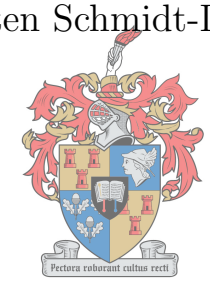
Reinforcement learning for the control of traffic flow on highways

Thorsten Schmidt-Dumont



UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Engineering at Stellenbosch University

Promoter: Prof JH van Vuuren
Co-promoter: Mrs MM Bruwer

December 2018

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2018

Abstract

Traffic congestion has become a significant problem around the world, not only in first-world countries, but also in third-world countries such as South Africa. Due to spatial limitations, especially in well-developed metropolitan areas, which typically experience the worst congestion problems, capacity expansion is not always feasible for relieving the pressure on the transportation network. Furthermore, the theory of induced traffic demand suggests that increasing highway capacity is not a long-term solution to traffic congestion due to additional traffic demand on new or updated routes, induced by commuters' perception that new or upgraded routes should be congestion free. As a result, various approaches toward improving highway traffic flow without increasing infrastructure capacity have been proposed in the literature.

Ramp metering and variable speed limits are the best-known control measures for effective traffic flow on highways. In most approaches towards solving the control problems presented by these control measures, optimal control techniques or online feedback control have been employed. Feedback control does not, however, guarantee optimality with respect to the on-ramp metering rate or the speed limit chosen, while optimal control techniques are limited to small networks due to their large computational burden.

Reinforcement learning is a promising alternative, providing the means and framework required to achieve near-optimal control policies at a fraction of the computational burden associated with optimal control algorithms. In this dissertation, a decentralised reinforcement learning approach is adopted towards simultaneously solving both the ramp metering and variable speed limit control problems.

The dawn of the autononomous vehicle promises further improvements in traffic flow which may be achieved over and above those of the aforementioned established highway traffic control measures, if their capabilities are harnessed effectively. A novel method of ramp metering by autonomous vehicles is introduced in this dissertation, based on the premise that specific instructions may be provided to autonomus vehicles travelling along an on-ramp. The control problem presented by this method of ramp metering *via* autonomous vehicles is also solved using a reinforcement learning approach.

The above solution approaches are implemented as a concept demonstrator within a simple, benchmark microscopic highway traffic simulation model. The effectiveness of the decentralised reinforcement learning approach is evaluated by means of statistical comparisons within the context of this simple benchmark simulation model. These approaches are finally applied within the context of a real-world case study simulation model of a section of the N1 highway outbound out of Cape Town, South Africa in order to demonstrate the effectiveness of the approaches within the context of a realistic scenario based on a real highway network and real traffic flow data.

Uittreksel

Verkeersopeenhoping het 'n ernstige probleem regoor die wêreld geword, nie net in eerste-wêreld lande nie, maar ook in derde-wêreld lande soos Suid-Afrika. As gevolg van ruimte-beperkings, veral in ontwikkelde, stedelike gebiede wat tipies die ernstigste verkeersopeenhoping ervaar, is die uitbreiding van infrastruktuur nie altyd 'n moontlike oplossing vir druk wat op vervoernetwerke ervaar word nie. Verder volg dit uit die teorie van geïnduseerde verkeersdruk dat die verhoging van snelwegkapasiteit nie 'n langtermynoplossing vir verkeersopeenhoping is nie vanweë die addisionele verkeer op nuwe of opgegradeerde roetes wat spruit uit die pendelaarspersepsie dat sulke roetes opeenhoping-vry behoort te wees. Gevolglik is 'n aantal benaderings in die literatuur voorgestel waarvolgens snelwegverkeersvloei verbeter kan word sonder om infrastruktuurkapasiteit te verhoog.

Opritmeting en veranderlike spoedbeperkings is die mees bekende beheermaatreëls vir doeltreffende verkeersvloei op snelweë. In die meeste benaderings tot die oplossing van die beheerprobleme wat met hierdie maatreëls gepaard gaan, word optimale beheertegnieke of intydse terugvoerbeheer toegepas. Daar is egter geen waarborg dat terugvoerbeheertegnieke optimale oplossings in terme van opritmetingstempo's of geselekteerde spoedgrense sal lewer nie, terwyl die gebruik van optimale beheertegnieke beperk is tot klein vervoernetwerke vanweë die noemenswaardige berekeningsvereistes van hierdie tegnieke.

Versterkingsleer is 'n belowende alternatief wat die middele en raamwerk verskaf waarvolgens byna-optimale beheerbeleide teen 'n fraksie van die berekeningsvereistes van konvensionele optimale beheeralgoritmes geformuleer kan word. 'n Gedesentraliseerde benadering tot versterkingsleer word in hierdie proefskrif gevolg om die verkeersvloei-beheerprobleme wat met opritmeting en veranderlike spoedbeperkings gepaard gaan, gelyktydig op te los.

Die koms van die outonome voertuig beloof verdere verbeterings in verkeersvloei wat behaal kan word bo en behalwe dié van die bogenoemde gevestigde snelwegverkeersbeheermaatreëls, indien hul doeltreffend aangewend word. 'n Nuwe metode van opritmeting deur middel van outonome voertuie word in hierdie proefskrif voorgestel, gebaseer op die veronderstelling dat spesifieke instruksies aan outonome voertuie op 'n oprit voorsien kan word. Die beheerprobleem wat deur hierdie metode van opritmeting deur middel van outonome voertuie daargestel word, word ook met behulp van 'n versterkingsleerbenadering opgelos.

Die bogenoemde oplossingsbenaderings word as 'n konsepdemonstrator in die konteks van 'n eenvoudige mikroskopiese snelweg-toetssimulasiemodel geïmplementeer. Die doeltreffendheid van die gedesentraliseerde versterkingsleerbenadering word deur middel van statistiese vergelykings in die konteks van die bogenoemde model evalueer. Die leerbenadering word laastens ook op 'n simulasiemodel van 'n realistiese gevallestudie oor die N1 snelweg uit Kaapstad, Suid-Afrika toegepas om die doeltreffendheid daarvan in terme van 'n werklike scenario en werklike verkeersvloeidata te demonstreer.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- My promoter, Prof JH van Vuuren, for sharing his broad knowledge, for his patience, guidance, relentless effort and his belief in me, for his critique, the challenging questions and the continual support, all of which have contributed to the quality of the work in this dissertation, which hopefully does not disappoint, knowing that you are "easily pleased with the very best."
- My co-promoter, Mrs Megan Bruwer from the Stellenbosch Smart Mobility Laboratory within the Department of Civil Engineering, for her assistance with regard to the traffic-specific questions, the brainstorming for ideas with respect to the novel highway traffic control measure, and her help in obtaining the traffic data required for the case study.
- SUnORE, the Harry Crossley foundation and the Department of Industrial Engineering, for their generous financial support over the past three years.
- SUnORE and the Department of Industrial Engineering, for the privilege to use the office space and computational facilities, and for affording me the opportunity to be a part of such an inspirational research environment.
- My colleagues at SUnORE, for their friendship, banter and inspiration over the past three years, making it a truly unforgettable time.
- My parents Hans and Henricke, and my brothers Georg and André, for their unconditional support whenever I needed it, throughout the course of my studies from BEng through to PhD. No matter what I had planned, you believed in me and encouraged me at every step along the way.
- Last but by no means least, my girlfriend Hester, thank you all for the unwavering support, the never-ending encouragement, for being my soundboard for ideas and allowing me to vent my frustration whenever necessary. Thank you for joining me for late nights and weekends at the office, for unexpectedly bringing me dinner when I was working late, and for always being there for me when I needed your support.

Table of Contents

List of Reserved Symbols

Variables

Symbol	Meaning
a_t	The action chosen at time step t
α	The step-size parameter (learning rate)
ϵ	The probability of choosing a random action during action selection
γ	The discount rate for future rewards
$P^a_{ss'}$	The probability of transition from state s to state s' under action a
q	The flow of vehicles along a section of highway
$Q(s,a)$	The value of taking action a in state s
$\mathcal{Q}(x,a)$	The value associated with centre-action pair (x,a) in kNN-TD learning
r_t	The reward obtained at time step t
R_t	The return following time step t
$R^a_{ss'}$	The expected immediate reward on transition from s to s' under action a
ρ	The density on a stretch of highway
$\hat{\rho}$	The target density that a ramp metering agent aims to achieve directly downstream of an on-ramp
s_t	The state of the environment at time step t
$V^\pi(s)$	The value of state s under policy π
w	The length of the queue building up at an on-ramp

Sets

Symbol	Meaning
$\mathcal{A}(s)$	The set of all possible actions in state s
$\mathcal{S}$	The set of all nonterminal states
$\mathcal{S}^+$	The set of all terminal states
$\mathcal{R}$	The set of all rewards

List of Acronyms

AHS: Automated highway system

ALINEA: Asservissement Linéaire d'Entreé Autoroutière

ANN: Artificial neural network

ANOVA: Analysis of variance

AV: Autonomous vehicle

AI: Artificial intelligence

CCTV: Closed circuit television

CI: Confidence interval

CRM: Conventional ramp metering

CRM-QL: Conventional ramp metering with queue limits

CSV: Comma separated value

CTM: Cellular transition model

GIS: Geographic information system

GPS: Global positioning system

GUI: Graphical user interface

HMI: Human machine interface

IRC: Iterative run controller

LA: Lane assignment

LSD: Least significant difference

kNN-TD: k nearest neighbour temporal difference reinforcement learning algorithm

MARL: Multi-agent reinforcement learning

MARLIN-ATCS: Multi-agent reinforcement learning for an integrated network of adaptive traffic signal controllers

MDP: Markov decision process

MLP: Multi-layer perceptron

MPC: Model predictive control

OSM: Open street map

PMI: Performance measure indicator

RL: Reinforcement learning

RM: Ramp metering

RMART: R-Markov average reward technique

SANRAL: South African National Roads Agency Limited

SARSA: State-action-reward-state-action

SATURN: Simulation and Assignment of Traffic in Urban Road Networks

SUMO: Simulation of Urban Mobility

TIS: Time spent in the system by individual vehicles

TMC: Traffic management centre

TTS: Total time spent in the system

VMS: Variable message sign

VSL: Variable speed limit

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Dissertation Background and Origin

Highways were originally built to provide virtually unlimited mobility to road users. The ongoing dramatic expansion of car ownership and travel demand have, however, led to the situation where, today, traffic congestion is a significant problem in major metropolitan areas all over the world. The reason for the severe traffic congestion experienced around the world is over-utilisation of the existing road networks which potentially leads to dense, stop-and-go traffic, as may be seen in Figure 1.1. In the United States, for example, travel delays increased by a factor of five from a cumulative 1.1 billion hours in 1982 to 5.5 billion hours in 2011 [144]. According to a report compiled by the Texas A&M Transportation Institute and the traffic information company Inrix [30], it is estimated that the average American citizen spends 42 hours per year stuck in traffic. This number rises to 82 hours in urban centres, which naturally are the more congested areas.

Perhaps the worst traffic jam ever recorded occurred in August 2010 on the National Highway 110 in China, and lasted longer than ten days [12]. The traffic jam was reported to be approximately 100 kilometres in length, with several motorists stuck in traffic for up to five days. Apart from the sheer inconvenience and frustration caused on the part of road users by typical rush hour congestion, it also has significant economic implications. Congestion in the United States resulted in a waste of more than three billion gallons of fuel and an accumulated seven billion hours spent by people stuck in traffic during 2015 at an annual nationwide cost of $160 billion, or $960 per commuter [30].

Traffic congestion is not only a major problem in first-world countries such as the United States, China or Germany, but also in South Africa. According to the TomTom Traffic Index [160], a congestion ranking based on GPS data collected from individual vehicles, Cape Town is the

(a) (b)

(c) (d)

FIGURE 1.1: *Severe traffic congestion on (a) National Highway 110, China, (b) Interstate 45, Texas, (c) Bundesautobahn 4, Germany, and (d) N2, South Africa [36].*

48$^{\text{th}}$ most congested city in the world, and the most congested city in Africa. In order to place these statistics into perspective, Cape Town has the same congestion ranking as New York City according to the TomTom Traffic Index published at the end of 2016, while the morning and afternoon peak congestion in Cape Town exceeds that experienced by commuters in New York City.

As may be seen in Figure 1.2, the traffic congestion levels in Cape Town have increased steadily since 2011, with a significant increase in congestion levels from 30% in 2015 to 35% in 2016. These percentages imply that a journey would take, on average, 35% longer in 2016 due to congestion than it would if free-flowing traffic conditions were to prevail. For the morning and afternoon peaks, the level of traffic congestion is naturally significantly larger than these average values suggest. During the morning peak, travellers experience a 75% increase in travel time, while during the afternoon peak commuters experience a 67% increase in travel time. The result of this level of traffic congestion is that the average Capetonian will spend an additional 42 minutes stuck in traffic per day, which accumulates to approximately 163 hours stuck in traffic congestion per year [160].

Although traffic congestion in Johannesburg is not quite as severe as it is in Cape Town, as travellers in Johannesburg experience average, morning peak and afternoon peak congestion levels of 30%, 62% and 60%, respectively, motorists in Johannesburg still spend 37 minutes per day, or a cumulative 141 hours stuck in congested traffic per year. As may be seen in Figure 1.2, congestion levels in Johannesburg temporarily decreased from 2009 until 2012. This decrease may be attributed to the Gauteng Freeway Improvement Project [160]. The aim of this project was significant highway capacity expansion through which the highways along major routes

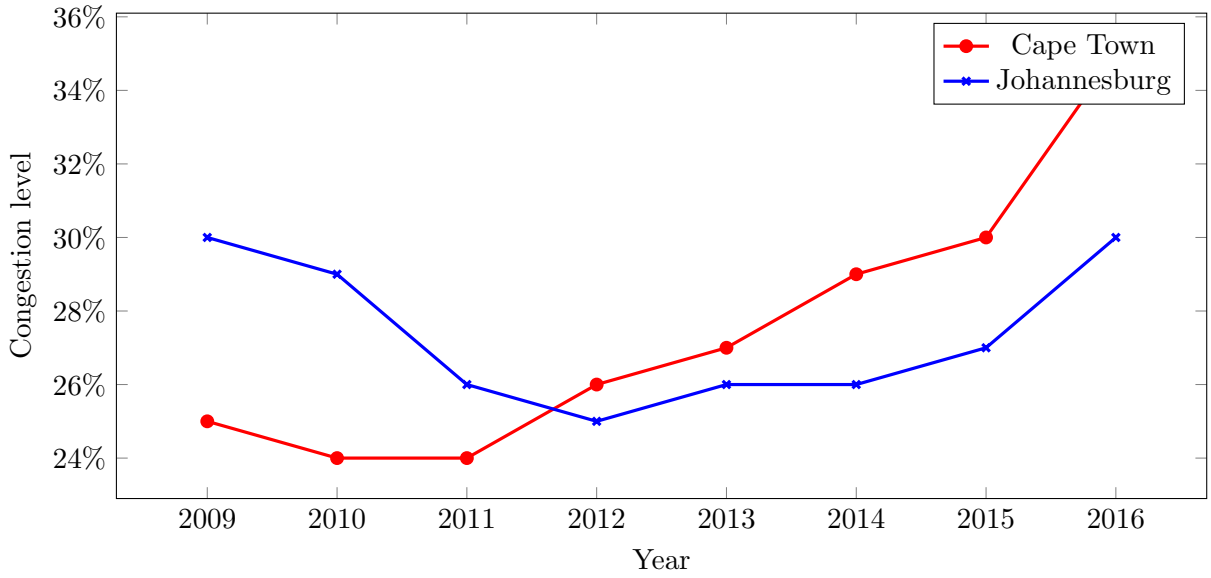


FIGURE 1.2: *Variation in traffic congestion levels in two major South African metropolitan areas, namely Cape Town and Johannesburg, during the period 2009–2016 [160].*

within the Johannesburg, Ekurhuleni and Tshwane metropolitan boundaries were expanded to at least four lanes in each direction, while along certain sections these highways were expanded to have six lanes in each direction [138]. The subsequent rise in congestion levels from 2012–2016, visible in the figure, may be attributed to the so-called *theory of induced travel demand*, in which it is suggested that increases in highway capacity will induce additional travel demand, thus not permanently alleviating congestion as envisioned [107]. The alternative to capacity expansion in order to improve traffic flow on highways is more effective control of the existing infrastructure. This may include dynamic traffic control measures such as ramp metering, variable speed limits, dynamic lane assignment, or the use of variable message signs to convey information on the current traffic situation to motorists.

Autonomous driving has often been hailed the future of human transportation with the promise of a congestion-free future due to perfect traffic flow coordination. Recent advances in the field of autonomous driving have led to the situation where in 2018 one is already able to purchase a vehicle that is essentially able to drive entirely by itself, although humans are required to be in the driver's seat, able to take over whenever required. Examples of such vehicles are the 2017 Mercedes-Benz E-Class [99] as well as the Tesla Model S [158]. These vehicles use a combination of cameras, ultrasonic sensors and radar to steer themselves on highways, change lanes and adjust their speeds according to traffic conditions [158].

Fully autonomous systems eliminate the driver from the control loop and may take complete control of the vehicle. Examples of commercially available vehicles capable of autonomous driving are the Tesla Model S and the Mercedes-Benz E-Class mentioned above. A possible configuration of the sensors employed in semi-autonomous and autonomous vehicles is shown in Figure 1.3.

Autonomous vehicles present a compelling case for their adoption, considering that already they are superior drivers to their human counterparts. This is due to the fact that a computer is simply better at parsing all the weather, GPS and traffic data that have to be taken into account when driving than an easily distracted human driver will ever be. A computer, for example, does not fall asleep behind the wheel, or remove its focus from driving to reply to

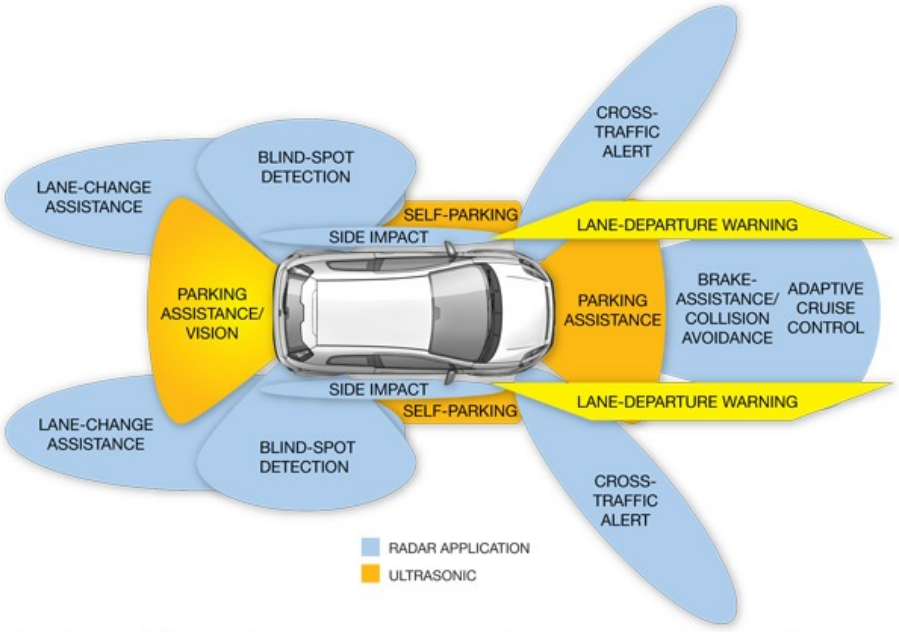


FIGURE 1.3: *The configuration and detection zones of the sensors of a semi-autonomous or autonomous vehicle [28].*

an urgent text message or answer a phone call [165]. Research reports have shown that human error is the main cause of motor vehicle accidents. In the United States alone, approximately six million vehicle accidents are reported annually to law enforcement [165]. According to the World Health Organisation [176], there are about 1.25 million traffic fatalities each year. Some 94% of these traffic accidents may be attributed to driver error. Furthermore, road traffic accidents are the leading cause of death among people aged 15–29 years — a statistic that is not too surprising when taking into account that 61% of drivers with smartphones admit to texting while driving [153]. An estimate of the annual costs associated with traffic accidents in the United States of America alone amounts to a staggering $836 billion [165]. Looking ahead at the transition phase from human-driven vehicles to autonomous vehicles, a study conducted by the Eno Center for Transportation suggests that a conversion of only 10% of the current vehicles on roads in the United States of America is expected to reduce the number of accidents each year by 211 000, saving approximately 1 100 lives. Cost savings from this modest change in traffic flow composition have also been estimated at $25.5 billion. If this number were to be increased to 90% over the course of time, the number of avoided traffic accidents may rise to 4.2 million annually, saving 21 700 lives per annum [165].

Various estimates have been made as to when the driverless vehicle transition will start in earnest. Elon Musk, the founder and CEO of Tesla Motors, has predicted the revolution to start around 2023, while industry analysts expect it to be between 2035 and 2050 [165]. From a purely technological point of view these numbers may be realistic. What is certain is that this revolution, once it starts, will bring about a self-compounding effect. As the number of autonomous vehicles on the road increases, new road designs will inevitably become more and more machine-centric. This will, in turn, make it harder for humans to drive their conventional vehicles on these roads, leading to more and more people trading in their keys [165]. This compounding effect will be further strengthened by the fact that, due to fewer accidents and smoother vehicle operation, insurance and running costs are expected to be considerably cheaper for autonomous vehicles, thus providing a further incentive to make the transition to driverless vehicles. Litman [89] has made predictions of autonomous vehicle adoption rates based on previ-

ous vehicle technology deployment, considering various technologies such as air bags, automatic transmissions, navigation systems, GPS services and hybrid technologies. The expected implementation rates are presented in Table 1.1. As may be seen in the table, it is assumed that fully autonomous vehicles will become available during the 2020s, but as is the case whenever a new technology is introduced, it will inevitably be flawed and overpriced initially, resulting in low adoption rates, with these adoption rates increasing once the autonomous vehicle can compete with human-driven alternatives on cost. The process to complete adoption (*i.e.* until such time that all vehicles on the roads are fully autonomous) is expected to take approximately five decades. The expected slow initial adoption rate, which should increase with time as the technology matures, is also illustrated graphically in Figure 1.4.

TABLE 1.1: *Autonomous vehicle implementation prediciton rates [89].*

Stage	Decade	Vehicle Sales	Vehicle Fleet	Vehicle Travel
Available with large premium	2020s	2–5%	1–2%	1–4%
Available with moderate premium	2030s	20–40%	10–20%	10–30%
Available with minimal premium	2040s	40–60%	20–40%	30–50%
Standard feature on most vehicles	2050s	80–100%	40–60%	50–80%
Saturation	2060s	?	?	?
Mandatory on all vehicles	?	100%	100%	100%

From the predictions in Table 1.1 it is clear that there will be a significant period of time during which mixed traffic flow of autonomous and human driven vehicles on the roads will prevail. The duration of the transition period seems long compared to the turnaround times of new innovations in the mobile telephone or personal computer technologies. One reason for this phenomenon is that motor vehicles typically cost fifty times as much and last ten times longer than mobile telephones or personal computers [89]. Since this transition phase is expected to take such a long time, it is important to implement traffic control measures which are not only able to take into account the mixed traffic flow of both human-driven vehicles and autonomous vehicles, but already start to exploit the expected benefits achievable through the efficient external control of autonomous vehicles, integrating these with human-driven vehicles in such a manner that every user of the system is able to experience the benefits.

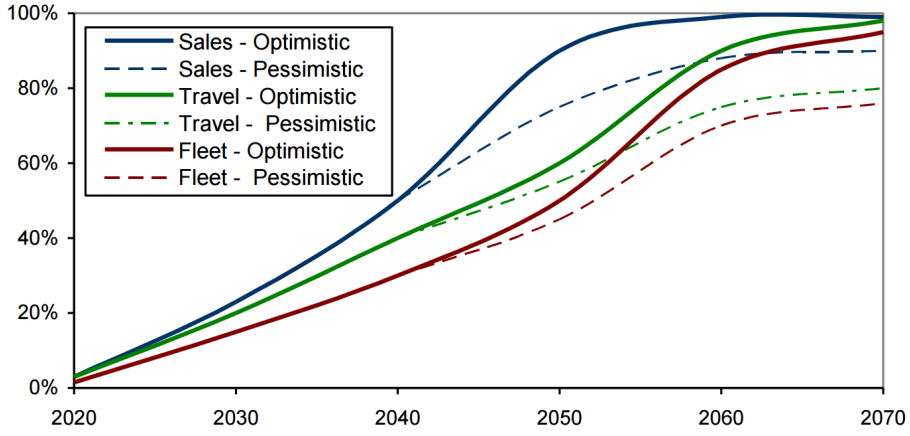


FIGURE 1.4: *Expected autonomous vehicle sales, fleet composition and travel distance projections, given as percentages of total vehicle compositions, for the years 2020–2070 [89].*

Recent advances in the field of *Artificial Intelligence* (AI) have shown great promise in terms of effective pattern recognition and successful strategy identification, even in situations where the range of alternatives is very large. Board games have proven to be a major testing ground for AI, by setting benchmarks for assessing the progress of AI, since an intelligent playing strategy is typically required in order to win these games. The game of Go has long held the reputation as the most challenging of classic games for AI due to its enormous search space and the difficulty of evaluating board positions and moves [149]. The Google-owned company DeepMind, however, mastered the formidable challenge posed by Go in March 2016, when its program, AlphaGo, beat the best Go player in the world, Lee Sedol, 4–1 in a five-match series [44]. A combination of AI techniques are employed in the program, so as to learn effective strategies for playing the game, without evaluating the entire range of possible moves at each stage of the game [149]. This remarkable feat has demonstrated the ability of AI algorithms to learn new strategies successfully within a complex, dynamic, uncertain environment.

It is, however, not only within the paradigm of board games that intricate AI systems have been applied with great success. Another remarkable application of AI is the so-called MogIA system. This system, developed by the Indian start-up company Genic.ai, took 20 million data points from public platforms such as Google, Facebook and Twitter, and, based on these data, correctly predicted Donald Trump as the winner of the 2016 United States presidential election, a result which was generally unexpected [19]. Furthermore, AI techniques have been applied to a wide variety of medical problems with great success. Esteva *et al.* [31] report on a deep convolutional neural network, which has been trained to identify melanoma (skin cancer) based on image classification. After training the neural network on 127 463 images, it was able to correctly classify the skin condition displayed in an image as benign or malignant in nature at a $72.1 \pm 0.9\%$ overall accuracy. Two dermatologists, on the other hand, achieved accuracies of 65.56% and 66.0%, respectively, when they were presented a subset of the validation set presented to the neural network.

The success of AI in respect of this wide variety of problems raises the question whether it would be possible to implement suitable AI algorithms to find effective highway traffic control measures in an online manner, allowing a computer to learn which control strategies work well in a dynamic traffic control environment.

1.2 Problem Statement

The problem considered in this dissertation is to investigate to what extent suitable reinforcement learning algorithms are able to identify high-quality traffic control policies for a portion of highway, taking into account known and novel control measures for various scenarios of traffic flow. As a concept demonstrator testbed, the reinforcement learning algorithms are to be implemented in a detailed agent-based microscopic simulation model of a traffic environment under investigation.

1.3 Dissertation Objectives

The following twelve objectives are pursued in this dissertation:

I To *conduct* a thorough survey of the literature related to:

(a) machine learning in general,

(b) reinforcement learning algorithms in particular,

(c) existing models and control measures for highway traffic control,

(d) the implementation of autonomous vehicles for controlling highway traffic flow,

(e) the application of machine learning to highway traffic control problems, and

(f) simulation principles and guidelines, with a specific focus on microscopic traffic simulation modelling.

II To *create* a suitable microscopic agent-based simulation model for use as a benchmark for evaluating the effectiveness of highway traffic control measures within the context of a simple highway network. This model should be able to facilitate the implementation of the highway traffic control measures researched in pursuit of Objectives I(c), I(d) and I(e) and should be informed by the research conducted in pursuit of Objective I(f).

III To *identify* a number of reinforcement learning algorithms capable of successfully altering traffic control policies by changing the variables of various highway traffic control measures.

IV To *implement* the reinforcement learning algorithms of Objective III in the context of the simulation model of Objective II with a view to identify high-quality highway traffic control policies, taking into account the subsequent improvements in the traffic flow along the highway made possible by changing the control policies associated with various existing highway traffic control measures.

V To *develop* and *implement* a novel highway traffic control measure in the simulation model of Objective II, based on the assumption that instructions may be given by reinforcement learning agents to varying percentages of autonomous vehicles with a view to improving the traffic flow along a stretch of highway.

VI To *verify* and *validate* the model and algorithmic implementations of Objectives II–V according to generally accepted modelling guidelines.

VII To *compare* statistically the relative effectiveness of various reinforcement learning algorithms with that of existing highway traffic control strategies in the context of the benchmark model of Objective II, taking variations in traffic demand along the stretch of highway into account.

VIII To *compare* statistically the relative effectiveness of the novel highway traffic control measure of Objective V with that of the best-performing existing highway traffic control measures identified in Objective VII, in the context of the benchmark model of Objective II, taking variations into account in the traffic demand along a stretch of highway.

IX To *apply* the concept demonstrator implementations of Objective IV and V to a special case study involving realistic traffic data for a specified stretch of a real highway.

X To *evaluate* the effectiveness of the associated reinforcement learning algorithms of Objective III in terms of their capability of identifying high-quality highway traffic control policies in the context of the case study of Objective IX.

XI To *compare* statistically the relative effectiveness of the novel highway traffic control measure of Objective V with that of the best-performing existing highway traffic control measures identified in Objective X, in the context of the case study.

XII To *recommend* sensible follow-up work related to the work in this dissertation which may be pursued in future.

1.4 Dissertation Scope

Due to the complexities involved in the highway traffic control problem, the scope in this dissertation is limited to the following control methods:

Ramp metering is the concept of controlling highway utilisation by effectively limiting the inflows of traffic onto the highway. This is achieved by changing traffic light phases at on-ramps, thereby controlling when vehicles are allowed to enter certain sections of the highway, ensuring that the highway capacity is fully utilised, preventing highway over-utilisation, and thus reducing congestion due to over-utilisation [115].

Variable speed limits are another method of controlling the flow of traffic on a certain section of highway [53]. By reducing the speed limit for a certain section of highway, the flow characteristics of that section are altered. As a result, the outflow out of that section may be reduced, thereby allowing a congested section upstream more time to resolve the congestion before further vehicles arrive. Furthermore, variable speed limits may lead to homogenisation of traffic flow, as the differences between the speeds of vehicles are reduced. This may result in a more stable traffic flow which may, in turn, lead to higher throughput and subsequently to a reduction in travel times [53].

The following traffic control measures are acknowledged, but are not implemented in this dissertation:

Dynamic lane assignments. In many cases, different lanes of a stretch of highway are not used effectively, resulting in over-utilisation of certain lanes, while other lanes may remain relatively under-utilised. One method of resolving this imbalance is to assign vehicles to specific lanes, thereby increasing the overall lane utilisation and hence increasing throughput on the highway. Dynamic lane assignment seems especially useful in a traffic paradigm where autonomous vehicles are present in the traffic flow, since direct and very detailed kinematic instructions may be given to such vehicles [137]. Due to the fact, however, that the focus in this dissertation is specifically on the period of mixed traffic flow with limited numbers of autonomous vehicles it is expected that dynamic lane assignments will not be effective due to the limited numbers of vehicles to which lane-changing instructions may be given. Dynamic lane assignments are therefore considered beyond the scope of this dissertation.

Variable message signs provide a manner of conveying information about upcoming traffic conditions to drivers through roadside infrastructure [164]. Due to the difficulty of measuring the effectiveness of these messages and their influence on driver behaviour, however, variable message signs are excluded as a control measure from the scope of this dissertation.

Platooning is the result of cooperative driving in the form of automated vehicles manoeuvring to achieve short inter-vehicle distances. Platooning may be facilitated by means of inter-vehicular communication, allowing vehicles to perform safe and efficient passing, lane changing and merging at close range. Platooning movement patterns have typically been modelled on the movement of wild geese and dolphins [68]. The benefits of platooning are, however, only expected to be fully exploitable once the traffic composition consists mainly of autonomous vehicles, and since the focus in this dissertation is on the transitional period during which limited numbers of autonomous vehicles are present in the traffic flow, platooning is beyond the scope of this dissertation.

1.5 Research Methodology

The work in this dissertation is executed in four stages. The first stage consists of a thorough literature review, focussing on the literature mentioned in Objective I of §1.3. The literature pertaining to machine learning in general is studied so as to present the reader with an overview of the prevailing techniques in fulfilment of Objective I(a). Thereafter, reinforcement learning algorithms, in particular, are studied in fulfilment of Objective I(b). This approach is followed in order to understand different machine learning techniques deemed suitable for solving the online traffic control problem described in §1.2 with a specific focus on reinforcement learning. In pursuit of Objective I(c), the study includes a review of multiple existing techniques for highway traffic control, with a specific focus on existing models and strategies for implementing the well-known control measures of ramp metering and variable speed limits. The aim here is to identify suitable techniques that have been implemented successfully within these contexts and may be adapted for implementation in this dissertation. Thereafter, in pursuit of Objective I(d), the focus shifts to previous attempts at controlling the traffic flow along a highway by providing autonomous vehicles with specific instructions. Finally, the literature study concludes with a review of previous attempts at implementing machine learning in a highway traffic control context, as well as a review of microscopic traffic simulation modelling and model validation guidelines, in fulfilment of Objectives I(e) and I(f).

The second stage of the study is the development stage. During this stage, Objectives II, III and IV of §1.3 are pursued. Initially the simple benchmark microscopic agent-based traffic simulation model of Objective II is established within a suitable software environment. The highway traffic control measures identified in the literature pertaining to Objective I(c) are incorporated into this simulation model in order to be able to effectively assess the ability of reinforcement learning algorithms to identify high-quality traffic control policies. This stage also includes the formulation of the highway traffic control problem as reinforcement learning problems in order to facilitate the implementation of the various reinforcement learning algorithms deemed suitable, in fulfilment of Objective III. Finally, this stage culminates in the development of a novel highway traffic control measure in pursuit of Objective V, informed by the literature reviewed in fulfilment of Objective I(d) on the application of autonomous vehicles for controlling highway traffic flow.

The next stage is the implementation stage. Objectives IV and V of §1.3 are pursued during this stage. This entails the implementation of the reinforcement learning algorithms for the existing and the novel highway traffic control measures within the context of the benchmark simulation model of Objective II. This implementation serves the purpose of a testbed for evaluating the traffic control protocols identified by reinforcement learning algorithms according to the improvements achievable in respect of the traffic flow along the highway.

The fourth and final stage of this study is the verification and evaluation stage. During this stage, Objectives VI to XI are pursued. The first step is to research appropriate, generally accepted modelling guidelines according to which a meaningful validation and verification of not only the simulation model implementation, but also the implementation of the reinforcement learning algorithms and highway traffic control measures within the simulation model may be carried out, in fulfilment of Objective VI. This is followed by a thorough statistical comparison of the relative performances of the various algorithms implemented within the context of the benchmark simulation model of Objective II for each of the highway traffic control measures, in fulfilment of Objectives VII and VIII. Thereafter, a case study of a specific instance of the highway traffic control problem is conducted in fulfilment of Objective IX. In this case study, the aforementioned reinforcement learning implementations for the existing and novel highway

traffic control measures are put to the test in the context of a realistic traffic data set, for a specified stretch of highway. This is again followed by a thorough statistical comparison of the relative performances of the various algorithmic implementations in fulfilment of Objectives X and XI. Finally, after having conducted a critical evaluation of the relative performances of the reinforcement learning algorithms in the context of the existing and novel highway traffic control measures, a summary is presented of what has been achieved in the dissertation, and suitable follow-up work and possible improvements are suggested for pursuit in the future, in fulfilment of Objective XII.

1.6 Dissertation Organisation

Apart from this introductory chapter, this dissertation consists of a further thirteen chapters, partitioned into four distinct parts. The first part, comprising Chapters 2–4, contains a literature review of material relevant to the work in this dissertation. More specifically, Chapter 2 is devoted to a literature review of machine learning, with a particular focus on reinforcement learning and a variety of solution techniques for the reinforcement learning problem. In Chapter 3, the focus shifts to the existing literature on highway traffic control measures, such as ramp metering and variable speed limits, and how machine learning has been implemented in these contexts. Furthermore, the existing literature on the implementation of autonomous vehicles for controlling highway traffic flow is reviewed. Part 1 is concluded in Chapter 4 with a comprehensive review of the literature pertaining to the principles of computer simulation with a particular focus on traffic simulation modelling.

The second part of the dissertation, comprising Chapters 5–10, is concerned with the development and implementation of the benchmark and case study microscopic highway traffic simulation models, as well as the implementation of the various reinforcement learning algorithms for the existing highway traffic control measures within the context of the benchmark and case study simulation models. In Chapter 5, a detailed description is provided of the simulation environment which acts as a testbed for the evaluation of the effectiveness of the machine learning algorithms. The implementation of the reinforcement learning algorithms is documented in Chapter 6 within the context of ramp metering, while a similar description of the implementation of reinforcement learning for solving the variable speed limit problem is provided in Chapter 7. Thereafter, a multi-agent approach to solving the ramp metering and variable speed limit problems simultaneously is presented in Chapter 8. Chapter 9 contains a description of the microscopic agent-based traffic simulation model developed for the purpose of the real-world case study. The ability of the reinforcement learning algorithms to identify high-quality highway traffic control policies in a real-world scenario is evaluated in Chapter 10, where a statistical evaluation of the relative algorithmic performances is performed.

The third part of the dissertation, comprising Chapters 11 and 12, is devoted to the development, implementation and evaluation of a novel highway traffic control measure. The focus thus shifts from existing technologies to future technologies involving fully autonomous vehicles. The concepts on which the novel highway traffic control measure is based, its formulation as a reinforcement learning problem and the solution by reinforcement learning algorithms within the context of the benchmark simulation model are detailed in Chapter 11. A statistical performance comparison of the novel highway traffic control measure with the best-performing existing highway traffic control measures is furthermore conducted. The ability of the reinforcement learning algorithms to identify high-quality highway traffic control policies within the context of the novel highway traffic control measure in a real-world scenario is evaluated in Chapter 12, where a sta-

tistical evaluation of the novel and the best-performing existing highway traffic control measures is performed.

Part four of the dissertation is the evaluation and assessment part, consisting of Chapters 13 and 14. A summary and critical appraisal of the contributions of the dissertation are provided in Chapter 13, and recommendations for related follow-up work which may be pursued in future follow in Chapter 14.

Part I

Literature Review

CHAPTER 2

Machine Learning

Contents

This chapter serves as an introduction to the field of machine learning, with a specific focus on reinforcement learning. In §2.1, the notion of machine learning is described in general and the different machine learning paradigms are discussed. Thereafter, the focus shifts in §2.2 to reinforcement learning in particular, introducing the key concepts of the reinforcement learning problem, together with a number of common solution approaches for this problem. In §2.3, two function approximation methodologies are reviewed which may be employed in order to extend the applicability of reinforcement learning to problems with continuous state and action spaces. The chapter finally closes in §2.4 with a brief summary of the material included.

2.1 Machine Learning in General

A scientific field is often best defined by the central question studied. Mitchell [103] states the central question of machine learning as follows:

> "How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?"

This central question covers a broad range of learning tasks, such as how to mine medical data records in order to determine which patients are likely to respond best to which treatments, how to design autonomous robots that are capable of navigating based on their own past experience, and how to build search engines that automatically take a user's needs into account and then customise themselves accordingly. More specifically, Mitchell [102] states that a machine is said

to *learn* with respect to a particular class of tasks T and a performance measure P, if it reliably improves its performance P at tasks in T following a gain in experience. Naturally then, three features have to be defined in order to have a well-defined learning problem — the class of tasks, the measure of performance which is to be improved upon, and the source of experience.

Marsland [93] classifies machine learning algorithms into four categories, according to the manner in which these algorithms find answers:

Supervised learning. In supervised learning, the algorithm is provided with a training set of examples for which the correct responses (targets) are known. Then, based on this training set, the algorithm aims to generalise in order to respond correctly to all possible inputs. This is also sometimes called *learning from exemplars*.

Unsupervised learning. In unsupervised learning, the correct responses are not known beforehand, but the algorithm aims to identify similarities between various inputs, such that those inputs which have something in common can be categorised together.

Reinforcement learning. Reinforcement learning falls somewhere between supervised and unsupervised learning, since the algorithm receives a signal if the answer is incorrect, but does not receive an indication as to how to correct it. The algorithm therefore learns by trial-and-error until the best answer is found. Reinforcement learning is sometimes referred to as *learning with a critic* because of this monitor which associates a score with each answer, but provides no suggestions as to how to improve it.

Evolutionary learning. Biological evolution may be interpreted as a learning process: biological organisms adapt in order to improve their own survival rate and the chance to produce offspring in their environment. This process is replicated in evolutionary learning, where each answer (or set of answers) is associated with a level of fitness, which provides an indication as to how good the current solution is.

Given the online nature of the highway traffic control problem, the potentially large number of variables that need to be taken into account, and the fact that until now, no perfect control method, or combination of methods has been found (which significantly complicates the learning process within the paradigm of supervised learning), it is reinforcement learning that has drawn the attention of the author for further investigation and implementation in this dissertation. This is due to the expectation that the performance measures of §1.3 are easily defined and measured within a simulation environment, which allows them to be translated into effective reward functions in order to provide high-quality feedback to a learning agent, thus allowing the performance of different control policies found during a trial-and-error search to be evaluated accurately in search of near-optimal policies.

2.2 Reinforcement Learning

When thinking about the nature of human learning, the first idea that comes to mind is that children learn by interacting with their surrounding environment. Whether a person is in the process of learning to drive or to hold a conversation, he or she is acutely aware of how the immediate environment reacts, and his or her actions are chosen in such a way as to influence what happens in that environment, typically in order to achieve a certain goal. Sutton and Barto [154], who are widely considered the pioneers of reinforcement learning [155], state that reinforcement learning is a computational learning approach focused on goal-directed learning

from interaction. This implies that the learning agent is not told which actions to take, but is rather instructed to attempt different actions, the results of which are then evaluated in the hope of finding the action reaping the most reward. This approach to machine learning results in two important characteristics, namely *trial-and-error search* and *delayed reward* — the actions taken now may not only affect the immediate reward, but also the following situation and, as a result, all subsequent rewards. These are the two most important distinguishing features of reinforcement learning when compared to other machine learning paradigms [154]. One challenge that arises due to these characteristics is that the right balance has to be found between exploration and exploitation in the sense that the learning agent needs to exploit what it already knows in respect of choosing actions which yield high rewards, but it also has to explore by choosing different actions so as to possibly uncover new, better actions.

Apart from the learning agent and the surrounding environment, there are four other main subelements of a reinforcement learning system: a *policy*, a *reward function*, a *value function* and a *model* of the environment [154].

The *policy* represents a mapping from perceived states of the environment to actions to be taken in the given state [154]. As a result, the policy defines the agent behaviour at a given time.

In a reinforcement learning problem, the *reward function* defines the goal of the problem. In other words, it maps each perceived state to a corresponding reward. This corresponding reward is typically a single number used to indicate the intrinsic desirability of that state [154]. It is then the goal of any agent to maximise the total reward received in the long run. As a result, the reward function is typically unalterable by the learning agent. The policy may, however, be altered in order to increase the reward obtained by the agent.

Whereas the reward function specifies what is desirable in the short term, the *value function* specifies what is good in the long run. The *value* of a state may be interpreted as the total reward an agent can possibly accumulate in future, starting from that state onwards. As such, the value function takes into account not only the reward gained from the current state, but also the rewards from the states which are likely to follow [154]. As a result, the objective in a reinforcement learning problem is typically to maximise the value obtained by the chosen actions in the long run. This may result in an action with a low initial reward being selected over one with a high initial reward, since the following state may be followed by states yielding even higher rewards. The converse may, however, also be true in certain cases. For this reason an efficient value function is often deemed the most important element of the formulation of a reinforcement learning problem [154].

The final component of a reinforcement learning system is a *model* of the environment. A model is, by definition, something that mimics the behaviour of the environment. The model is used for planning as, given the current state and action, it can predict the resulting next state as well as the associated reward.

2.2.1 Evaluative Feedback

According to Sutton and Barto [154], the most important feature distinguishing reinforcement learning from other machine learning paradigms is the fact that training information is employed to *evaluate* actions rather than giving correct actions as *instructions*. This distinguishing feature requires active exploration of the space of actions in the form of a comprehensive trial-and-error search. Purely evaluative feedback, however, only indicates how good the action taken is, without indicating whether it is the best or worst possible action.

Action-Value Methods

In reinforcement learning, an action is evaluated according to the cumulative reward, or value which results from it. One way to determine the value of an action is simply to take the average of the rewards received whenever that action has been selected. Thus, if at the t-th play of an iterative game, action a has been chosen $k_a > 0$ times previously, yielding rewards $r_1, r_2, \ldots, r_{k_a}$, its estimated value, denoted by $Q_t(a)$, is given by

$$Q_t(a) = \frac{r_1 + r_2 + \ldots + r_{k_a}}{k_a}. \tag{2.1}$$

If, however, $k_a = 0$, then $Q_t(a)$ is typically assigned a default value, such as $Q_0(a) = 0$ [154]. This method of value estimation is called the *sample-average* method, since it is simply the average of the sample of relevant rewards. The simplest action selection technique is then to choose the action yielding the highest value up to time t. This is called the *greedy method*. One shortcoming of this method, however, is that it does not allow for exploration of different actions, which may, in the long term yield a greater value. One solution to this problem is to behave in a greedy manner most of the time, but with a small positive probability ϵ of choosing an action which has, to that point, yielded a lower value [159]. This method is called *ϵ-greedy method*.

Softmax Action Selection

One criticism of the ϵ-greedy method is that, although exploration is encouraged, the exploration is random, choosing equally among actions. One solution to this problem is to vary the probability of an action being chosen as a graded function of the estimated value. In this case, the action with the highest estimated value is afforded the highest probability for selection, with all other actions ranked and weighted according to their value estimates. Such a method is called the *softmax* action selection rule [159]. In most cases the softmax method employs a Boltzmann or Gibbs distribution. An action a at the t-th play is then chosen with probability

$$\frac{e^{Q_t(a)/\tau}}{\sum_{a \in \mathcal{A}} e^{Q_t(a)/\tau}},$$

where τ is a positive parameter called the *temperature*. High temperatures result in nearly equiprobable (random) action selection, whereas low temperatures amplify the difference in selection probability based on differences in the value estimations [159].

Both the aforementioned methods require a record of all the rewards received up to time step t, which will result in infinitely increasing memory and computational requirements. As a result, Sutton and Barto [154] introduced an incremental update formula. Let Q_k denote the average of the first k rewards for some action a. Given this average and a new reward r_{k+1}, the average of all rewards may be computed as

$$Q_{k+1} = Q_k + \frac{1}{k+1}[r_{k+1} - Q_k]. \tag{2.2}$$

Tracking a Nonstationary Problem

The methods discussed above are applicable to a stationary environment — one which does not change over time. This is, however, often not the case in practice. In cases where the environment is nonstationary, it may be beneficial to weigh the more recent rewards more heavily

than rewards obtained far back in the past. One method of implementing this is to introduce a constant step-size parameter α. The incremental update rule in (2.2) is then modified such that

$$\begin{aligned} Q_k &= Q_{k-1} + \alpha[r_k - Q_{k-1}] \\ &= (1-\alpha)^k Q_0 + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} r_i, \end{aligned} \tag{2.3}$$

where $0 < \alpha \leq 1$ is constant. This is called a weighted average since the weights satisfy the condition $(1-\alpha)^k + \sum_{i=1}^{k} \alpha(1-\alpha)^{k-i} = 1$. Due to the fact that the weight decays exponentially, this method is called an *exponential, recency-weighted average* [154].

Several other methods for evaluative feedback have been formulated in the literature. Sutton and Barto [154] discuss a number of these methods.

2.2.2 The Reinforcement Learning Problem

This section is devoted to a generic formulation of the reinforcement learning problem in general. Key elements of the mathematical structure of the reinforcement learning problem are also introduced.

The Agent-Environment Interface

As stated in §2.2, the reinforcement learning problem is a framing of the problem of learning from interaction in order to achieve a goal. The learner and decision maker is called the *agent*, while the externalities it interacts with are called the *environment*. The actions chosen by the agent result in changes in *states* of the system and resulting *rewards*. The agent-environment interaction is illustrated graphically in Figure 2.1.

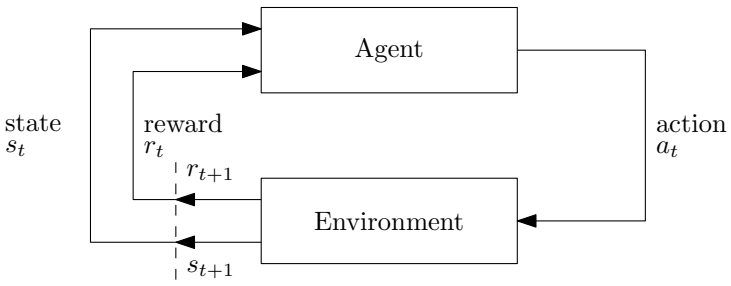


FIGURE 2.1: *The agent-environment interaction in reinforcement learning, adapted from [154].*

As may be seen in the figure, the agent and the environment interact at a sequence of discrete time steps $t = 0, 1, 2, \ldots$ At each time step t, the agent receives a representation of the environment's state, $s_t \in \mathcal{S}$, where $\mathcal{S}$ represents the set of all possible states. Based on the current state, the agent then chooses an action $a_t \in \mathcal{A}(s_t)$, where $\mathcal{A}(s_t)$ represents the set of all possible actions available to the agent when the environment is in state s_t. One time step later, the agent receives a numerical reward $r_{t+1} \in \mathcal{R}$, where $\mathcal{R}$ represents the set of all possible rewards, after which the environment finds itself in a new state, s_{t+1}. At each time step, the agent implements a mapping from the set of environment states to the unit interval $[0, 1]$ of real numbers representing probabilities of selecting each possible action. This mapping is called the agent's policy and is denoted by $\pi(s_t, a_t)$. Reinforcement learning methods specify how the agent may change its policy as a result of learning experience. The agent's goal is to maximise the total reward gained in the long run.

Backup diagrams, as depicted in Figure 2.2, are often used to illustrate the relationships which form the basis of the update operations that are at the heart of reinforcement learning methods. In these diagrams, each open circle represents a state, and each solid circle represents a state-action pair. In Figure 2.2 (a), for example, an agent finds itself in state $s \in \mathcal{S}$ and can take one of three possible actions $a \in \mathcal{A}(s)$, which may then lead to one of several next states $s' \in \mathcal{S}$, along with a corresponding reward $r \in \mathcal{R}$. The state nodes in backup diagrams do not necessarily all represent distinct states, as a state may be its own successor.

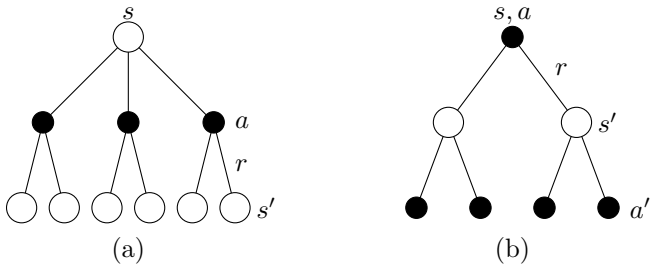


FIGURE 2.2: *Backup diagrams for a specific state s in (a) and a specific state-action pair (s, a) in (b), adapted from [154].*

Goals, Rewards and Returns

In reinforcement learning, the purpose or goal of an agent is formalised in terms of a special reward signal passed from the environment to the agent. Typically, this reward $r_t \in \mathcal{R}$ is simply a real number. The reward, formalised as the notion of a goal, is one of the key features of reinforcement learning. The agent always attempts to maximise its reward, and as a result, the reward should be a way of communicating to the agent *what* has to be achieved, instead of *how* to achieve it [22]. Take a robot playing chess as an example. A reward should only be obtained by actually winning a game, not for gaining control of an area of the board, for example, or taking its opponent's pieces, as these may not necessarily lead to a win. Furthermore, it is important that the reward should be calculated in the environment, and not by the agent, so as to ensure that the agent only has imperfect control in order to achieve this goal.

If the sequence of rewards received after some time step t is denoted by $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$, then generally the aim is to maximise the *expected return*, denoted by R_t and defined by some function of the reward sequence [22]. In the simplest case, the return may simply be the sum of the rewards,

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \ldots + r_T, \tag{2.4}$$

where T represents the final time step. This approach makes sense as long as the agent-environment interaction can naturally be partitioned into subsequences, called *episodes*, such as plays of a game. Critically, each episode must end in a *terminal state*, which may be followed by a reset to some standard starting state, drawn from a standard distribution of starting states. Tasks that may be partitioned into such episodes are called *episodic tasks* [154]. In episodic tasks, it should be possible to distinguish between the set of all non-terminal states, and the set of all terminal states, denoted by $\mathcal{S}^+$.

In many cases, however, tasks cannot be partitioned into identifiable episodes, but evolve continually. Such tasks are called *continuing tasks* [154]. For these tasks, the return formula (2.4) is problematic since neither the terminal time nor the accumulated return may be bounded. As a result, Sutton and Barto [154] suggested the concept of *discounting*. When adopting this

approach, the goal of the agent is to maximise the discounted return

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{2.5}$$

where γ is a scalable parameter in the unit interval $[0, 1]$, called the *discount rate*. The discount rate determines the value of future rewards: a reward received k time steps in the future is only worth γ^{k-1} times the value it would be worth if it were to be received immediately. As long as $\gamma < 1$ the reward sequence $\{r\}_{k=1,2,3,\ldots}$ is bounded, and the sum in (2.5) has a finite value. If $\gamma = 0$, the agent is said to be *myopic* in the sense of being concerned only with maximising the immediate rewards achieved. As γ approaches 1, however, future rewards gain more and more importance, and as a result, the agent becomes more *far-sighted*.

The quantification approaches in (2.4) and (2.5) may be combined into one formula which may be used in both episodic or continuing cases. The return may in this case be written as

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}, \tag{2.6}$$

which includes the possibilities that $T = \infty$ or $\gamma = 1$, but not both.

The Markov Property

As mentioned, the agent's decisions are made as a function of a signal received from the environment, known as a *state*. This state is usually determined by some preprocessing system, which forms part of the environment. Ideally, this state signal should summarise past sensations compactly, yet retain all the relevant information [154]. A state signal that succeeds in retaining all the relevant information is said to possess the *Markov property*. Take a game of chess as an example again: the current configuration of all the pieces on the board may be considered as a Markov state, since it summarises everything about the complete sequence of positions that lead to it. Much of the information about the exact sequence of moves is lost, but everything important going forward is retained. In the same way, the current position and velocity of a cannonball may be considered a Markov state, since this contains all the information necessary to trace the future trajectory of the object. For the purpose of tracing out the future trajectory, it is, however, not necessary to know how the cannonball achieved its current position and velocity.

Under the assumption that only a finite number of states and reward values exist, the Markov property of the reinforcement learning problem may be formalised as follows. Consider the response of a general environment at time $t + 1$ corresponding to an action taken at time t. In the most general case, this response may depend on everything that has happened, leading up to the current situation. In this case, the dynamics may be defined only by specifying the complete probability distribution

$$P_r(s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, \ldots, s_1, a_1, r_1, s_0, a_0), \tag{2.7}$$

for all $s' \in \mathcal{S}$, $r \in \mathcal{R}$, $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}(s_t)$, and all possible values of the past events: $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}(s_t)$, $r_t \in \mathcal{R}$, $\ldots$, $s_1 \in \mathcal{S}$, $a_1 \in \mathcal{A}(s_1)$, $r_1 \in \mathcal{R}$, $s_0 \in \mathcal{S}$, $a_0 \in \mathcal{A}(s_0)$. If, however, the state signal exhibits the Markov property, then the environment's response at time $t+1$ only depends on the state and action representations at time t, in which case (2.7) reduces to

$$P_r(s_{t+1} = s', r_{t+1} = r \mid s_t, a_t), \tag{2.8}$$

for all $s' \in \mathcal{S},\ r \in \mathcal{R},\ s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}(s_t)$. In other words, a state signal exhibits the Markov property if and only if (2.8) is equal to (2.7) for all $s' \in \mathcal{S},\ r \in \mathcal{R}$, and all histories, $s_t \in \mathcal{S},\ a_t \in \mathcal{A}(s_t),\ r_t \in \mathcal{R},\ \ldots,\ s_1 \in \mathcal{S},\ a_1 \in \mathcal{A}(s_1),\ r_1 \in \mathcal{R},\ s_0 \in \mathcal{S},\ a_0 \in \mathcal{A}(s_0)$.

As a result, if an environment exhibits the Markov property, then the one-step dynamics given in (2.8) allow for the prediction of the next state and associated reward, given only the current state and action. It follows that by iterating the expression in (2.8) one may predict all future states and rewards just as well as would be possible if the entire history up to the current time were known. This implies that the Markov states provide the best basis for choosing actions, which allows the action policy to be formulated as a function of the Markov states.

Markov Decision Processes

A reinforcement learning problem that satisfies the Markov property is called a *Markov decision process* (MDP) [155]. In the case where the state and action spaces are finite, the process is called a *finite* MDP. Any particular finite MDP is defined by its state and action sets, and by the one-step dynamics of the environment. Given any state $s \in \mathcal{S}$ and action $a \in \mathcal{A}(s)$, the probability of each possible next state s' is given by

$$P^a_{ss'} = P_r(s_{t+1} = s' \mid s_t = s, a_t = a). \tag{2.9}$$

These quantities are called *transition probabilities*. Similarly, given any current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}(s)$, together with any next state $s' \in \mathcal{S}$, the expected value of the next reward is given by

$$R^a_{ss'} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\}. \tag{2.10}$$

The quantities $P^a_{ss'}$ and $R^a_{ss'}$ in (2.9)–(2.10) completely specify the most important aspects of the dynamics of a finite MDP (only information about the distribution of rewards around the expected value is lost).

Value Functions

Almost all reinforcement learning algorithms are based on estimating *value functions* — functions of states (or state-action pairs) that provide an estimate as to how good it is for an agent to be in a certain state (or how good it is to perform a specific action in a given state) [155]. The notion of "how good" is typically defined in terms of the expected future rewards (*i.e.* in terms of the expected return). Naturally, the future rewards depend on the actions taken by the agent. Accordingly, the value functions are defined with respect to particular policies. The value of a state s under some policy π is the expected return when starting in state $s \in \mathcal{S}$ and following π thereafter. In MDPs, the *state-value function for policy* π, denoted by $V^\pi(s)$, is defined as

$$V^\pi(s) = E_\pi\{R_t \mid s_t = s\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\}, \tag{2.11}$$

where $E_\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\}$ denotes the expected value given that the agent follows policy π. Similarly, the value of taking an action $a \in \mathcal{A}(s)$ in state $s \in \mathcal{S}$ under policy π, denoted by $Q^\pi(s, a)$, is defined as the expected return, starting from state s, of taking action a, and thereafter following policy π. The function Q^π, called the *action-value function for policy* π, is given by

$$Q^\pi(s, a) = E_\pi\{R_t \mid s_t = s, a_t = a\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\}. \tag{2.12}$$

The value functions V^π and Q^π may be estimated from experience. For example, if an agent follows policy π and maintains an average, for each state encountered, of the actual returns that have followed the state, then the average will converge to the state's value, $V^\pi(s)$, as the number of times the state is encountered approaches infinity. If separate averages are kept for each action taken in a state, then these averages will similarly converge to the action values, $Q^\pi(s,a)$. Estimation methods of this kind are called *Monte Carlo methods* [154] due to the fact that they involve taking the average of actual returns from random samples.

A fundamental property of value functions used in reinforcement learning is that they satisfy certain recursive relationships. For any policy π and any state $s \in \mathcal{S}$, the consistency condition

$$\begin{aligned} V^\pi(s) &= E_\pi\{R_t \mid s_t = s\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\} \\ &= E_\pi\left\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s\right\} \\ &= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}\left[R^a_{ss'} + \gamma E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s'\right\}\right] \\ &= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V^\pi(s')] \end{aligned} \tag{2.13}$$

holds between the value of s and the value of its possible successor states, where it is implicit that the actions are taken from the set $\mathcal{A}(s)$, and the next states are taken from the set $\mathcal{S}$. The expression in (2.13) is known as the *Bellman equation for* V^π [22]. It expresses a relationship between the value of a state and the values of its successor states.

The Bellman equation (2.13) represents the average over all possibilities, taking the weight of the probabilities into account. It states that the value of the start state must equal the (discounted) value of the expected next state, together with the expected reward. The value function V^π is the unique solution to its Bellman equation. As a result, the Bellman equation forms the basis of a number of ways of computing, approximating and learning V^π.

For finite MDPs, an optimal policy may be defined in the following way. A policy π is said to *be better than or equal to* another policy π', denoted by $\pi \succeq \pi'$, if its expected return is greater than or equal to that of π' for all states. In other words, $\pi \succeq \pi'$ if and only if $V^\pi(s) \geq V^{\pi'}(s)$ for all $s \in \mathcal{S}$. If one policy exists that is better than or equal to all other policies, it is called an *optimal policy* [154], denoted by π^*. There may be more than one optimal policy. Each optimal policy π^* corresponds to an optimal state-value function value

$$V^*(s) = \max_\pi V^\pi(s) \tag{2.14}$$

for all $s \in \mathcal{S}$. Similarly, each optimal policy also has a corresponding optimal action-value function value

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \tag{2.15}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$. For each state-action pair, this function value represents the expected return associated with taking some action a in state s and thereafter following an optimal policy. As a result, one may write

$$Q^*(s,a) = E\{r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a\}. \tag{2.16}$$

Optimality and Approximation

Cases where an agent learns an optimal policy are very rare for real-life problem instances [154]. This is due to the fact that, because of time constraints, current processing technology still cannot compute an optimal policy for such a problem by solving the Bellman equation within a reasonable time available per stage. Furthermore, memory requirements also present a challenge. In tasks with small, finite sets of states, it is often possible to form approximations using tables or arrays containing an entry for each state-action pair. For large problems, which may have infinitely many states, this is, however, not possible. In such cases, the functions must be approximated at the cost of optimality, using some sort of more compact parameterised function representation. This does, however, present unique opportunities for achieving useful approximations. There may, for example, be many states which are reached with such a low probability that computing optimal behaviour for those states will have only a minimal impact on the amount of reward received by the agent. The online nature of reinforcement learning makes it possible to approximate optimal policies in such a way that more attention is afforded to frequently occurring states, resulting in good decisions being made when those states occur, at the expense of less effort being made in learning good policies for less frequently encountered states. This is one key property which distinguishes reinforcement learning from other approximate solution approaches to MDPs.

2.2.3 Reinforcement Learning Solution Approaches

The key idea of reinforcement learning may be summarised as the use of value functions in order to structure and organise the search for high-quality policies [154]. This section is devoted to a review of a variety of basic reinforcement learning algorithms which may be implemented in order to find optimal policies.

Policy iteration

In the case where the environment's dynamics are known, the Bellman equation in (2.13) results in a system of $|\mathcal{S}|$ linear equations in $|\mathcal{S}|$ unknowns, the $V^\pi(s)$-values for all $s \in \mathcal{S}$. Computing $V^\pi(s)$ directly from this system of equations is, however, often impractical, especially for problems which have large state spaces. As a result, Sutton and Barto [154] proposed estimating value functions by means of iterative methods. The value $V^\pi_{k+1}(s)$, which represents the estimation of $V^\pi(s)$ at the $(k+1)^{th}$ iteration, is given by

$$\begin{aligned} V^\pi_{k+1}(s) &= E_\pi\{r_{t+1} + \gamma V^\pi_k(s_{t+1}) \mid s_t = s\} \\ &= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V^\pi_k(s')], \end{aligned} \tag{2.17}$$

where the initial estimate V^π_0 is chosen arbitrarily. It has been shown that V^π_k converges to V^π as $k \to \infty$ under the condition that either $\gamma < 1$ or the events are episodic [22]. This method of estimating value functions through the repeated application of (2.17) until convergence is achieved, is called *policy evaluation*.

If both the state-value function $V^\pi(s)$ and the action value-function $Q^\pi(s,a)$ are known for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, one may easily determine the optimal policy by simply choosing at each state the action which appears to be best according to $Q^\pi(s,a)$. The new, *greedy* policy π' is given

by

$$\begin{aligned} \pi'(s) &= \max_a Q^\pi(s,a) \\ &= \max_a E\{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V^\pi(s')]. \end{aligned} \tag{2.18}$$

This process of greedily creating a policy that improves the existing policy with respect to the value function is called *policy improvement* [154]. Note that the policy $\pi(s)$ denotes the mapping from state $s \in \mathcal{S}$ to the action $a \in \mathcal{A}(s)$ the agent chooses according to the current policy. This convention is employed throughout the remainder of this dissertation.

As a result, once a policy π has been improved based on the value of V^π in order to find a better policy π', $V^{\pi'}$ may be computed, and again improved to find an even better policy π''. As a result, a sequence of monotonically improving policies and value functions

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$$

may be found, where $\xrightarrow{E}$ and $\xrightarrow{I}$ denote policy evaluation and policy improvement, respectively. A pseudo-code description of this algorithm, called *policy iteration*, is given in Algorithm 2.1.

Algorithm 2.1: **The policy iteration algorithm [154].**

Input : An arbitrary initial value $V(s) \in \Re$ and policy $\pi(s) \in \mathcal{A}(s)$ for all $s \in \mathcal{S}$.
Output: An optimal policy $\pi^*(s)$.

1 Policy evaluation;
2 $\triangle \leftarrow 0$;
3 **while** $\triangle > \delta$ (a small positive number) **do**
4 $\quad \triangle \leftarrow 0$;
5 $\quad$ **for** *each* $s \in \mathcal{S}$ **do**
6 $\quad\quad v \leftarrow V(s)$;
7 $\quad\quad V(s) \leftarrow \sum_{s'} P^{\pi(s)}_{ss'}[R^{\pi(s)}_{ss'} + \gamma V(s')]$;
8 $\quad\quad \triangle \leftarrow \max(\triangle, |v - V(s)|)$;
9 Policy improvement;
10 *policy_stable* $\leftarrow$ TRUE;
11 **for** *each* $s \in \mathcal{S}$ **do**
12 $\quad b \leftarrow \pi(s)$;
13 $\quad \pi(s) \leftarrow \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$;
14 $\quad$ **if** $b \neq \pi(s)$ **then**
15 $\quad\quad$ *policy_stable* $\leftarrow$ FALSE;
16 **if** *policy_stable* = FALSE **then**
17 $\quad$ go to line 1;
18 **else**
19 $\quad$ **return** $[\pi(s)]$;

Value iteration

One drawback of policy iteration, pointed out by Sutton and Barto [154], is that each iteration requires policy evaluation, which may itself be a protracted iterative computation, often

requiring multiple sweeps through the state set. The policy evaluation step may, however, be truncated without the loss of convergence guarantee of policy evaluation. In *value iteration*, policy evaluation does not continue until convergence, but is terminated after each state has been evaluated once, and thereafter policy improvement is completed immediately [154]. Thus, value iteration combines the policy improvement and truncated policy evaluation steps such that the estimated value is given by

$$\begin{aligned} V_{k+1}(s) &= \max_a E\{r_{t+1} + \gamma V_(s_{t+1}) \mid s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V_k(s')]. \end{aligned}$$

A pseudo-code desciption of the value iteration algorithm is given in Algorithm 2.2.

Algorithm 2.2: **The value iteration algorithm [154].**

Input : An arbitrary initial value $V(s) \in \Re$ for all $s \in \mathcal{S}$.
Output: An optimal policy $\pi^*(s)$.

1 $\triangle \leftarrow 0$;
2 **while** $\triangle > \delta$ (a small positive number) **do**
3 $\quad \triangle \leftarrow 0$;
4 $\quad$ **for** *each* $s \in \mathcal{S}$ **do**
5 $\quad\quad v \leftarrow V(s)$;
6 $\quad\quad V(s) \leftarrow \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$;
7 $\quad\quad \triangle \leftarrow \max(\triangle, |v - V(s)|)$;
8 **return** $[\pi(s) \leftarrow \arg\max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]]$;

Q-learning

Q-learning is another value iteration-based reinforcement learning algorithm first proposed by Watkins [170]. Unlike in value iteration, however, the goal in Q-learning is to attempt to directly compute the *optimal* action value function, $Q(s, a)$. This is achieved through the comparison of the current action-value estimation $Q(s_t, a_t)$ with a new estimate calculated using the reward r_t received as well as the maximum value of the future state, $\max_a Q(s_{t+1}, a)$. The update rule for the action values is given by

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \alpha \left[r_t + \gamma \max_a Q_k(s_{t+1}, a) - Q_k(s_t, a_t) \right], \tag{2.19}$$

where γ represents the discount factor as defined above, and α represents the learning rate, which is a small positive real number influencing the extent of the effect that the new estimation of the value has. For example, if the learning rate is 1, the old value will be replaced by the new estimation. Due to the stochastic nature of the MPDs, however, it is necessary to determine the average value obtained over multiple time steps. As a result, the learning rate is employed only to partially update the old values [130]. The final policy may then be extracted greedily from the final approximation of the state-action values once the algorithm has terminated. A pseudo-code description of the Q-learning algorithm is given in Algorithm 2.3.

Watkins and Dayan [169] have shown that Q-learning converges to the optimal action-value function $Q^*(s, a)$ as long as all state-action pairs are visited and updated infinitely many times, regardless of the policy being followed in line 4 of Algorithm 2.3. In order to promote efficient

learning, however, this policy should find a good balance between encouraging exploration (*i.e.* exploring the available action space for all states), as well as exploitation (*i.e.* encouraging the choice of good actions for each state). One popular method used for this purpose is the ϵ-greedy method, where the best action according to the current approximation of $Q(s,a)$ is chosen with a probability of $1-\epsilon$, and another action is selected randomly with probability ϵ.

Algorithm 2.3: **The Q-learning algorithm [170].**

Input : An arbitrary initial value $Q(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$.
Output: A near-optimal policy $\pi^*(s)$.

for all episodes **do**
  Initialise s;
  repeat for each step of each episode
    Choose a_t from s_t using some predefined policy derived from Q;
    Take action a_t, observe the reward r_t, and the next state s_{t+1};
    Update $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_t + \gamma \max_a Q_k(s_{t+1},a) - Q_k(s_t,a_t)\right]$;
    $s_t \leftarrow s_{t+1}$;
  until s is terminal;
return $[\pi(s) = \max_a Q(s,a)]$;

SARSA

The *state-action-reward-state-action* (SARSA) reinforcement learning algorithm is another notable algorithm derived directly from the Bellman equation (2.13) [154]. The algorithm's name is derived form the sequence of events that take place during the Q-value updating process. The SARSA algorithm functions similarly to the Q-learning algorithm. Unlike Q-learning, however, SARSA is a so-called on-policy algorithm. The effect of this is that when updating $Q(s_t,a_t)$, the next action a_{t+1} is chosen according to the current policy instead of taking the maximum Q-value over all actions [130]. The update rule for SARSA is thus given by

$$Q_{k+1}(s_t,a_t) = Q_k(s_t,a_t) + \alpha\left[r_t + \gamma Q_k(s_{t+1},a_{t+1}) - Q_k(s_t,a_t)\right]. \tag{2.20}$$

The result is that, as is typical in on-policy methods, a continual estimation of Q^π is provided for the current policy π, while simultaneously attempting to adapt the policy π over time to find the optimal policy π^* [154]. A pseudo-code description of the SARSA algorithm is provided in Algorithm 2.4.

R-Markov Average Reward Technique

The *R-Markov Average Reward Technique* (RMART) is, like Q-learning, an off-policy learning algorithm. The focus of the RMART algorithm, however, is that the value function is not defined with respect to the discounted accumulated reward, but rather with respect to the average expected reward per time step as

$$\varrho^\pi = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} E(r_t), \tag{2.21}$$

Algorithm 2.4: **The SARSA reinforcement learning algorithm [154].**

Input : An arbitrary initial value $Q(s, a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$.
Output: A near-optimal policy $\pi^*(s)$.

for all episodes **do**
  Initialise s;
  repeat for each step of each episode
    Choose a_t from s_t using some predefined policy derived from Q;
    Take action a_t, observe the reward r_t, and the next state s_{t+1};
    Update $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_t + \gamma Q_k(s_{t+1}, a_{t+1}) - Q_k(s_t, a_t)\right]$;
    $s_t \leftarrow s_{t+1}$;
  until s is terminal;
return $[\pi(s) = \max_a Q(s, a)]$;

where the process is assumed to be ergodic[1], and as a result, r^π does not depend on a specific starting state [154]. From any state, the long-term average reward is the same, but there is a transient reward, implying that from some states, better than average rewards may be received for a while, while other states may yield lower than average rewards. It is this transient reward which defines the value of a state as

$$\bar{V}^\pi(s) = \sum_{k=1}^{\infty} E_\pi\{r_{t+k} - \varrho^\pi \mid s_t = s\}. \tag{2.22}$$

Similarly, the action value of a state-action pair may then be defined as

$$\bar{Q}^\pi(s, a) = \sum_{k=1}^{\infty} E_\pi\{r_{t+k} - \varrho^\pi \mid s_t = s, a_t = a\}. \tag{2.23}$$

These are called *relative values*, since they are computed relative to the average reward achievable under the current policy [179]. Unlike in Q-learning, however, two policies are maintained in the RMART algorithm, a so-called behaviour policy and an estimation policy, based on the action-value function and an estimated average reward, respectively. A pseudo-code description of the RMART algorithm is given in Algorithm 2.5.

2.3 Reinforcement Learning with Function Approximation

When employing the above-mentioned solution approaches in their conventional form, the approximated values of the function $Q(s, a)$ are stored in a lookup table. This does, however, limit the practicality of using these algorithms in complex problems which often have a continuous state space [132]. The cause of this deficiency is the so-called curse of dimensionality, increasing the learning time, due to the continuous state space being discretised into an increasing number of states in order to achieve improved accuracy. Alternatively, a direct representation of a continuous state space may be achieved through the use of a general function approximator [132]. Function approximators have not only extended the applicability of the reinforcement learning solution approaches but, if implemented effectively, they have also been shown to use the feedback information more effectively, resulting in faster learning rates [130]. Two of the

[1] All states of an ergodic process communicate (*i.e.* all states can be reached from each state) and are thus recurrent, and all states are aperiodic (*i.e.* paths leading back to a state have lengths which are not all multiples of an integer $k > 1$) [175].

Algorithm 2.5: **The RMART algorithm [179].**

Input : Arbitrary initial values $\bar{Q}(s,a)$ and $\varrho(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$.
Output: A near-optimal approximation of the $\bar{Q}(s,a)$ and $\varrho(s,a)$ values.

repeat for each step of each episode
  Update learning rates for $\bar{Q}$ and ϱ;
  $\alpha^{(t)} \leftarrow 10^{\frac{\log(t+2)}{t+2}}$;
  $\beta^{(t)} \leftarrow \frac{A}{B+t}$, where A and B are scalars;
  Choose a_t from s_t using some predefined policy derived from $\bar{Q}$;
  Take action a_t, observe the reward r_t, and the next state s_{t+1};
  Update $\bar{Q}(s_t,a_t) \leftarrow \bar{Q}(s_t,a_t) + \alpha^{(t)}\left[r_t - \varrho(s_t,a_t) + \max_a \bar{Q}_k(s_{t+1},a) - \bar{Q}_k(s_t,a_t)\right]$;
  if $\bar{Q}(s_t,a_t) = \max_a \bar{Q}(s_t,a_t)$ **then**
    Update $\varrho(s_t,a_t) \leftarrow \varrho(s_t,a_t) + \beta^{(t)}\left[r_t - \varrho(s_t,a_t) + \max \bar{Q}(s_{t+1},a) - \max_a \bar{Q}(s_t,a_t)\right]$;
  $s_t \leftarrow s_{t+1}$;
until s is terminal;

most notable function approximation approaches are the k-nearest neighbour weighted average, and the multi-layer perceptron neural network, which are discussed in this section.

2.3.1 k-Nearest Neighbours Weighted Average

Martin *et al.* [94] introduced variations on the well-known temporal difference learning algorithms using weighted k-nearest neighbours for function approximation in a continuous state space.

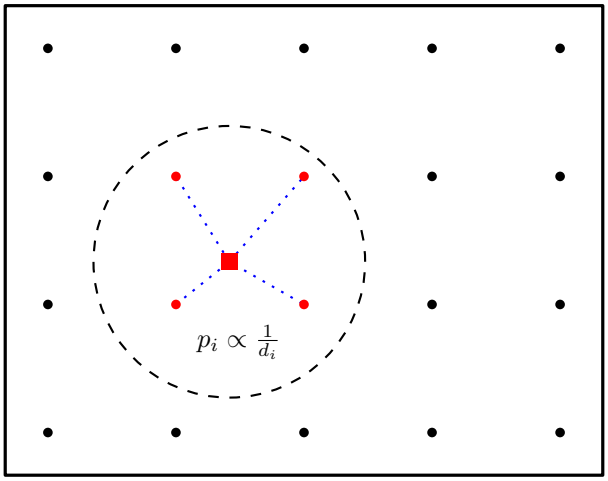


FIGURE 2.3: *Illustration of the k-nearest neighbour algorithm for estimating the value of a new point s based on Euclidean distance in two dimensions with* $k = 4$.

A pseudocode description of the kNN-TD reinforcement learning algorithm, introduced by Martin *et al.* [94], is given in Algorithm 2.6. The first task in this algorithm is to identify the k nearest neighbours of the current state s. In order to find these k nearest neighbours, a set $\mathcal{X}$ of centres which have been assigned explicit Q-values, is often defined and generated in the state space. Each member of the set of k nearest neighbours provides information on the current state s. This information is typically the associated Q-value in kNN-TD reinforcement learning. The k nearest neighbours of the current observation s may then be identified, based on the Euclidean distance d_i between s and its nearest neighbour $i \in \{1,\ldots,k\}$, as shown in Figure 2.3. Based

on this distance, an activation coefficient

$$w_i = \frac{1}{1 + d_i^2}, \tag{2.24}$$

which is inversely proportional to the distance between s and the nearest neighbour i, is determined for each of the k nearest neighbours for all $i \in \{1, \ldots, k\}$. The second task is then to obtain a probability distribution $p(i)$ over the set of k nearest neighbours $i \in \{1, \ldots, k\}$. These probabilities are determined according to the current weight vector w_i and may be calculated as

$$p(i) = \frac{w_i}{\sum_{i=1}^{k} w_i} \tag{2.25}$$

for all $i \in \{1, \ldots, k\}$. These probabilities associated with each of the k nearest neighbours may then be used to determine the expected value

$$\mathcal{Q}(knn, a) = \sum_{i=1}^{k} Q(i, a)p(i) \tag{2.26}$$

of the learning target for a given action a. Here $p(i)$ acquires the meaning of the probability $P(\mathcal{Q}(knn, a) = Q(i, a) \mid s)$, while $\mathcal{Q}(knn, a)$ takes the value $Q(i, a)$ given that the current state is s. Once the value $\mathcal{Q}(knn, a)$ has been determined for each action $a \in \mathcal{A}(s)$, action selection may follow, once again based on some pre-defined policy.

For the online learning process then, two such state representations are required, one for the current state s, and one for the next observed state s_{t+1}. Using these representations, the TD-error δ may be determined and the action value may be updated, using the rule in lines 13–15 of Algorithm 2.6.

2.3.2 Multi-layer Perceptron Neural Networks

It is well known that the human brain consists of a large number of nerve cells, called *neurons*, which are interconnected, and as a result form so-called neural networks. These neurons are effectively information processing units which, upon receiving an electrochemical input signal, have to determine whether to "fire" their own signal or not. If a neuron does fire, its electrochemical pulse is again sent out to millions of other neurons through so-called *synapses*, which then have to make their decisions about firing or not. These electrochemical pulses sent by the neurons through the synapses represent the information processing process that takes place within the human brain. An *artificial neural network* (ANN) is a computer representation of this information processing process, employing *artificial neurons* in order to mimic the human brain's behaviour. Haykin [50] defined an ANN as "a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use." Haykin [50] continued to point out the two most significant resemblances between neural networks and the human brain are (1) that knowledge is acquired by the network through a learning process, and (2) that interneuron connection strengths, known as synaptic weights, are used to store the knowledge.

The Neuron

McCulloch and Pitts [97] attempted to model the function of neurons mathematically, which led to the development of neural networks. Their formulation of a neuron comprises three basic elements [50, 93]:

Algorithm 2.6: **The *k*NN-TD algorithm [94].**

Input : A set $\mathcal{X} \in \Re^n$ of centres and an arbitrary initial value $Q(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$.
Output: A near-optimal approximation of the $Q(s,a)$ values.

repeat for each episode
  Initialise s;
  $knn \leftarrow k$-nearest neighbours of s;
  $p(knn) \leftarrow$ probabilities of each $i \in knn$;
  $\mathcal{Q}(knn,a) \leftarrow Q(knn,a) \times p(knn)$;
  Choose a from s according to $\mathcal{Q}(knn,a)$;
  repeat for each step of each episode
    Take action a_t, observe r_t, s_{t+1};
    $knn' \leftarrow k$-nearest neighbours of s_{t+1};
    $p(knn') \leftarrow$ probabilities of each $i \in knn'$;
    Choose a_{t+1} from s_{t+1} according to $\mathcal{Q}(knn',a)$;
    $\mathcal{Q}(knn',a_{t+1}) \leftarrow Q(knn',a_{t+1}) \times p(knn')$;
    $\delta \leftarrow r_t + \gamma \max_{a_{t+1}} \mathcal{Q}(knn', a_{t+1}) - \mathcal{Q}(knn, a)$;
    for $i \in knn$ **do**
      $Q(i,a) \leftarrow Q(i,a) + \alpha\delta p(i)$;
    $a \leftarrow a_{t+1}, s \leftarrow s_{t+1}, knn \leftarrow knn'$;
  until s is terminal;
until learning ends;

1. **A set of weighted inputs** which correspond to the synapses,
2. **an adder** which sums the input signals, and
3. **an activation function** which is employed in order to determine whether the neuron fires for the current inputs.

This mathematical model of the neuron is illustrated graphically in Figure 2.4. As may be seen in the figure, the neuron receives m inputs $(x_1, \ldots, x_m)$, weighted by the corresponding m weights $(w_1, \ldots, w_m)$ and produces a single output value y. The adder is employed to determine the net input $z = \sum_{i=1}^{m} w_i x_i + x_0\theta$. In this formulation, θ may represent either a bias or a threshold value, in which case x_0 is given a set value of 1 or -1, respectively [50]. The net input z is then passed to the activation function, which is used to determine the output value $y = \varphi(z)$ [93].

A typical activation function is the so-called *threshold activation function*. The threshold activation function is a step function, which takes the value $y = 1$ if the net input s is greater than d, or $y = 0$ otherwise [50]. In mathematical notation, the threshold activation function is thus given by

$$\varphi(s) = \begin{cases} 1, & \text{if } z \geq d \\ 0, & \text{if } z < d. \end{cases} \tag{2.27}$$

Another popular activation function is the so-called *sigmoid activation function*, defined as a strictly increasing function that exhibits suitable smoothness and asymptotic properties. A common example of the sigmoid function is the *logistic function*

$$\varphi(s) = \frac{1}{1 + e^{-dz}}, \tag{2.28}$$

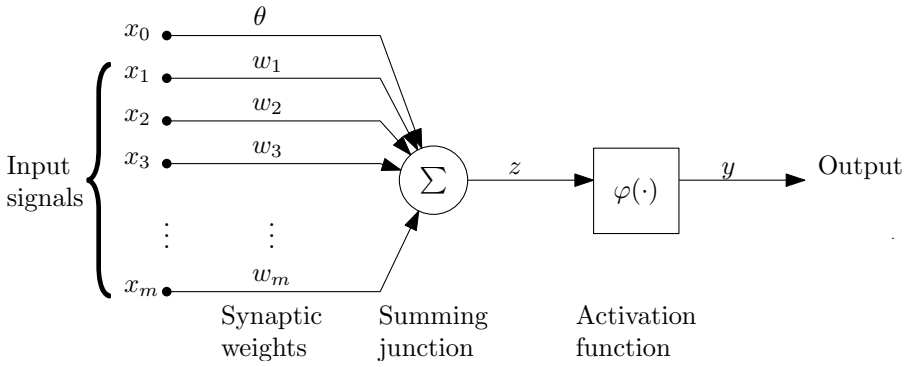


FIGURE 2.4: *A nonlinear model of a neuron. Adapted from Haykin [50].*

where d is the slope parameter of the sigmoid function, which influences how quickly the function transitions from low to high values. The larger the chosen value for the slope parameter, the more the sigmoid function resembles the threshold function. The sigmoid function is shown for the values of $d = -3$ and $d = -10$ in Figure 2.5.

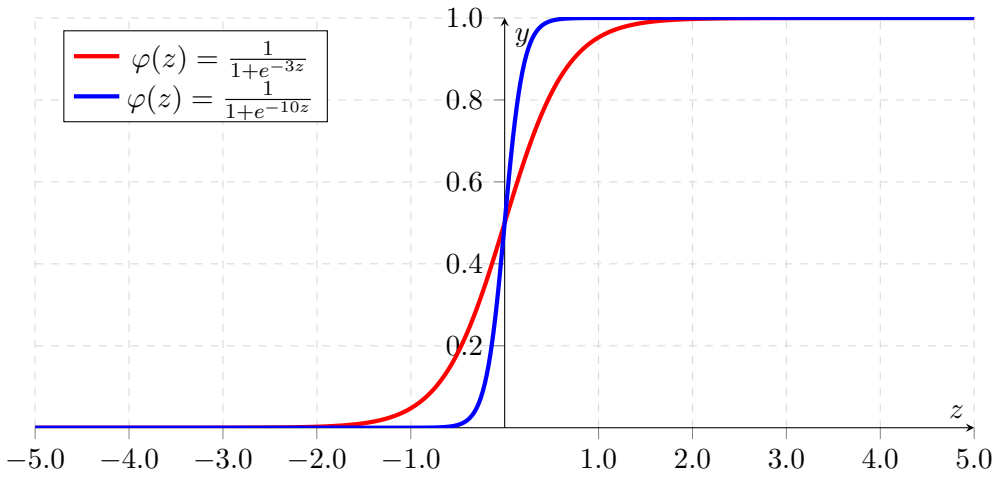


FIGURE 2.5: *The logistic sigmoid activation function (2.28) for values of $d = -3$ and $d = -10$.*

The Perceptron

Neurons, as described in the previous section, form the basis of ANNs. A single neuron, however, only provides limited information, and does not by itself answer the question of how an ANN has the capability to learn. In order to answer this question, the notion of a *perceptron* needs to be defined. Technically, a perceptron is simply a collection of McCulloch-Pitts neurons, together with a set of inputs and weights which connect the inputs to the neurons [93]. A graphical representation of a single layer perceptron is shown in Figure 2.6. The perceptron is considered to be a *feedforward* neural network, since signals are sent only in a forward direction through the network (from left to right in the figure). The perceptron shown in the figure has $m + 1$ inputs (including the bias or threshold), and n outputs. As in the neuron model, the inputs are connected to the outputs by weights, each input i being connected to each neuron j by a weight w_{ij}.

As mentioned in §2.1, in supervised learning, the algorithm is provided with a training set of examples for which the correct responses, known as targets, are known, based upon which a

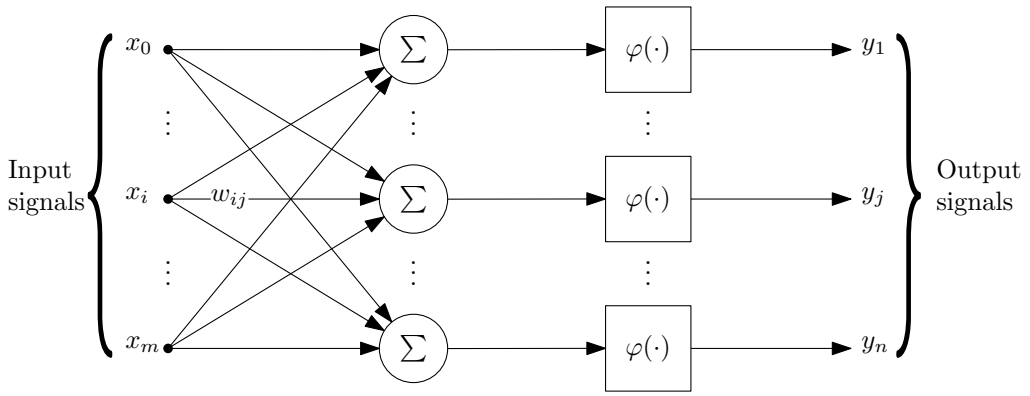


FIGURE 2.6: *The general architecture of a single layer perceptron. Adapted from Haykin [50].*

mapping from the inputs to the target outputs has to be learnt. In perceptron learning, this is achieved by updating the weights in the perceptron [93]. Let x_i denote the i^{th} input, let y_j represent the output of the j^{th} neuron, and let t_j denote the desired (target) output for neuron j. In the case where the output y_j differs from the target, the weight w_{ij}, connecting the i^{th} input to the j^{th} neuron is updated by

$$w_{ij} = w_{ij} + \eta(t_j - y_j)x_i, \tag{2.29}$$

where η represents a learning rate, controlling the magnitude of change affected to the weights. This learning rate is typically decreased over time in order to ensure convergence [3].

The single-layer perceptron is capable of solving linearly separable classification problems. Linear separability requires that the patterns to be classified are sufficiently separated from one another to ensure that the decision surface consists of hyperplanes, as illustrated in Figure 2.7(a) for a case of two dimensions. If, however, the classification problem is no longer linearly separable, as shown in Figure 2.7(b), the elementary single-layer perceptron may fail to classify them [50].

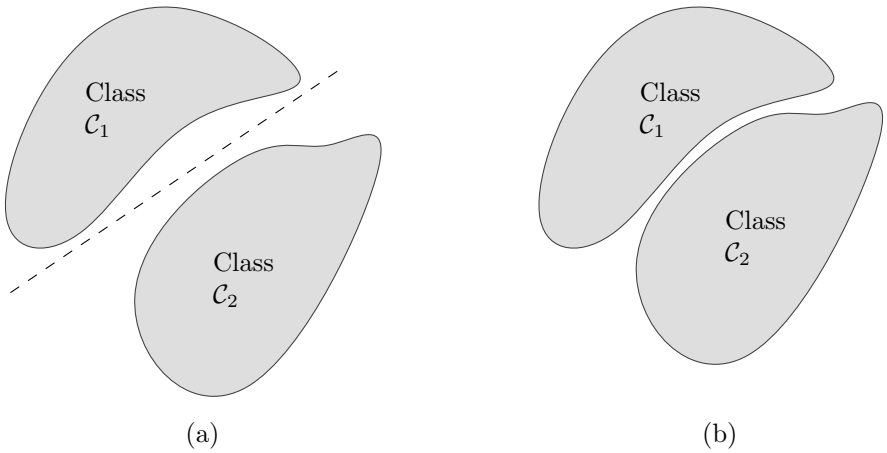


FIGURE 2.7: *A pair of linearly separable surfaces in (a), and a pair of nonlinearly separable surfaces in (b). Adapted from Haykin [50].*

The Multi-layer Perceptron

The multi-layer perceptron is another class of ANN, which, unlike the single-layer perceptron, which has only one layer of neurons, has multiple *hidden* layers of neurons between the input and

the output layer [93]. An illustration of the general architecture of an MLP with two hidden layers is provided in Figure 2.8. In the figure, each neuron is represented by a circle, which contains both the adder, as well as the activation function. The input signals are transferred to the first hidden layer, where the neurons within that layer produce their output signals based on their activation functions, which are then weighted and fed to the second hidden layer, where again, the neurons produce new outputs based on the specific activation functions. The outputs from the second layer are then fed to the neurons comprising the output layer. Finally, the outputs from the output layer are the outputs generated by the neural network [50].

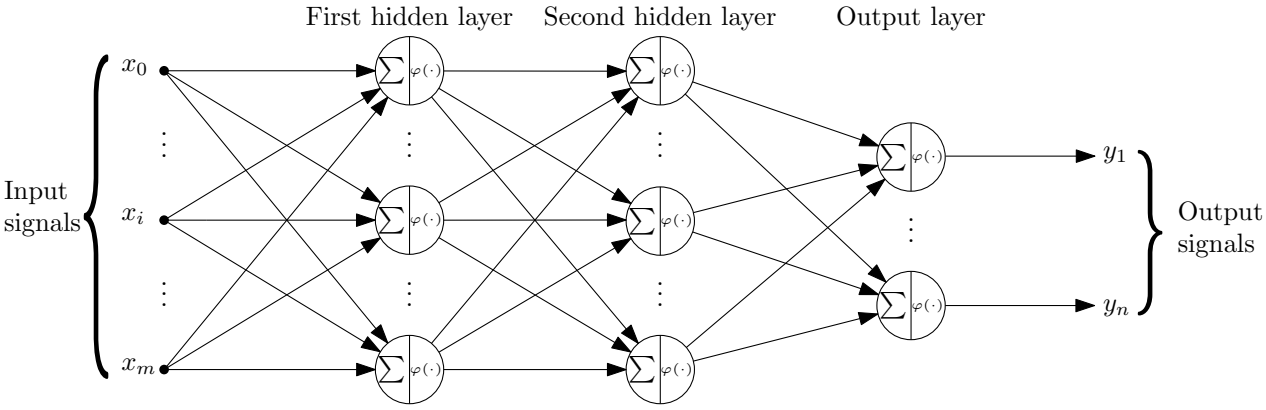


FIGURE 2.8: *The general architecture of the multi-layer perceptron with two hidden layers. Adapted from Haykin [50].*

It is important to note that, in order to extract the best performance from an MLP, the activation functions of each of the neurons within the networks should be of a *smooth* (everywhere differentiable), nonlinear nature (typically variations on the sigmoid function) so as to be able to generate accurate approximations of complex nonlinear functions [50]. In fact, Hornick *et al.* [58] have shown that MLPs are capable of approximating virtually any function of interest to an arbitrary degree of accuracy, provided that a sufficient number of hidden neurons are available. For the purpose of function approximation within reinforcement learning, an MLP would typically have only a single output signal, which would be the estimation of the state-value, or action-value function [130].

The Back Propagation Algorithm

Typically, the training paradigm adopted for MLPs when employed for function approximation is that of supervised learning. As with the perceptron, the training process for an MLP involves the adjustment of the network's synaptic weights. Due to the additional complexity contained within an MLP compared to that of a single-layer perceptron, this cannot be achieved by using a simple update rule such as the one in (2.29). The *back propagation training algorithm* is often employed for the training of MLPs, and it is essentially a gradient-based optimisation technique for minimising some appropriate error function [33]. The back propagation training algorithm comprises three phases, namely [33]:

1. The forward propagation of an input vector,
2. the calculation and back propagation of the associated error, and
3. the adjustment of the network weights.

During the forward propagation phase, an input vector is presented to and transmitted through the neural network. This entails the calculation of the activation of all hidden and output neurons, finally ending with the network's response to the input vector. Thereafter, during the second phase, the network output is compared to the target value of the given input vector, and based on this comparison, an error is calculated. These errors are then propagated backwards through the network with the goal of calculating the corresponding errors at each of the hidden neurons. Finally, in the third phase, all the network weights are updated simultaneously based on the error values for the hidden neurons [33]. There are two main approaches according to which this update process can take place. Either the weights are updated after each input vector has been presented to the network, which is referred to as *online learning*, or the weights are updated only once all the input vectors have passed through the network, which is referred to as *batch learning*.

As stated above, the aim of the back propagation algorithm is to minimise an appropriate error function while employing a gradient-based optimisation technique. Marsland [93] defined this error term as the sum of squared errors E, scaled by a factor of $\frac{1}{2}$ which, for a network with n output neurons, is given by

$$E = \frac{1}{2}\sum_{k=1}^{n}(t_k - y_k)^2, \tag{2.30}$$

where t_k denotes the target output of the k^{th} output neuron, and y_k is the actual network output of the k^{th} output neuron. Thereafter, the gradient of the error function is computed with respect to the weights, such that the weights may be adjusted in a manner so as to minimise the error. For a given weight w_{ij}, the update rule is then given by

$$w_{ij} = w_{ij} - \eta\frac{\partial E}{\partial w_{ij}}, \tag{2.31}$$

where η is again the learning factor. This update rule requires the activation function to be differentiable, which is the reason for the sigmoid function being a popular choice as activation function in MLPs. In order to calculate the gradient of the error function, which becomes difficult if one or more hidden layers are included in the network, the back propagation algorithm employs the chain rule of differentiation. The reader is referred to Haykin [50] (pp. 142–153) for a complete description of the derivation of the back propagation algorithm. A pseudo-code description of the back propagation algorithm is provided in Algorithm 2.7.

Algorithm 2.7 is specific to the online learning process, which means that the errors and weights are adjusted after each training sample has passed through the network. The functions presented in the algorithm are also specific to using the logistic sigmoid function as activation function. Furthermore, the algorithm is specific to networks with only one hidden layer, although it may be extended in order to include multiple hidden layers [93]. In the algorithm, the input vector is fed through the hidden layer in Steps 4–6, after which the output signal generated by the hidden layer is presented to and processed by the output layer in Steps 7–9. Thereafter, error calculation for the output layer is completed in Steps 11–12, and for the hidden layer in Steps 13–14. Finally weights of the output and hidden layers are updated in Steps 16 and 17, respectively. Upon completion of the training algorithm, the newly trained neural network may then be employed for function approximation within any of the reinforcement learning algorithms described in §2.2.3 [154].

Algorithm 2.7: **The back propagation algorithm for online learning [93].**

Initialise w_{ij} and w_{jk} with arbitrary small values;
repeat for each input vector
  Forward phase;
  for $j = 1, \ldots, \ell$ **do**
    $s_j \leftarrow \sum_{i=1}^{m} x_i w_{ij}$;
    $y_j \leftarrow \varphi(s_j) = \frac{1}{1+e^{-ds_j}}$;
  for $k = 1, \ldots, n$ **do**
    $s_k \leftarrow \sum_{j=1}^{\ell} y_j w_{jk}$;
    $y_k \leftarrow \varphi(s_k) = \frac{1}{1+e^{-ds_k}}$;
  Backward phase;
  for $k = 1, \ldots, n$ **do**
    $\delta_k \leftarrow (t_k - y_k) y_k (1 - y_k)$;
  for $j = 1, \ldots, \ell$ **do**
    $\delta_j \leftarrow y_j (1 - y_j) \sum_{k=1}^{n} w_{jk} \delta_k$;
  Update weights;
  $w_{jk} \leftarrow w_{jk} + \eta \delta_k y_j$;
  $w_{ij} \leftarrow w_{ij} + \eta \delta_j x_i$;
until learning ends;

2.4 Chapter Summary

In this chapter, basic concepts from the field of machine learning were reviewed, with a specific focus on reinforcement learning. In §2.1, the idea behind machine learning was reviewed in general, and the four different machine learning paradigms were described. Thereafter, there was a shift in focus in §2.1 to reinforcement learning in particular, with a brief introduction to the concept of evaluative feedback in §2.2.1, after which the reinforcement learning problem was outlined in §2.2.2. This was followed by an elucidation in §2.2.3 of some of the key solution approaches which may be employed when solving reinforcement learning problems. Finally, §2.3 served as an introduction to two important methodologies from the supervised learning paradigm which may be employed for value function approximation, namely the k nearest neighbours weighted average method and the multi-layer perceptron neural network, which allow the reinforcement learning methodology to be applied to problems which have large, continuous state and action spaces.

CHAPTER 3

Highway Traffic Control

Contents

This chapter is devoted to a review of certain well-known highway traffic control measures found in the literature. In §3.1, some of the fundamental theories of traffic flow, as well as the two major paradigms of traffic flow modelling are introduced. After these basic concepts have been explained, the focus shifts to the highway control problem, with a review of some of the best-known highway traffic control measures in the literature, as well as algorithms designed for their application in §3.2. This is followed in §3.3 by a brief description of several methods for improving traffic flow along a highway in the presence of autonomous vehicles. Thereafter, applications of machine learning are briefly highlighted in §3.4, with a focus on reinforcement learning as it is applicable to these traffic control measures. The chapter closes in §3.5 with a brief summary of the work reviewed.

3.1 Traffic Flow Fundamentals

Traffic flow theory may be traced back to the early 1950s [91], and its inception is largely attributed to Wardrop [168] who employed mathematical and statistical expressions in order to describe traffic flow. Over the following decade, the field continued to evolve with several important examples showing significant progress, such as the fluid-dynamic traffic flow models introduced by Lighthill and Whitham [86] and by Richards [133], which form the cornerstone of a number of macroscopic traffic models and theories developed since. Another notable example

of research conducted at the time is the car-following experiments and subsequent theories formulated by the General Motors research laboratory [26, 40, 41, 55]. Since then, traffic flow theory has diversified to incorporate a wide range of modelling influences, incorporating various fields of study, including environmental studies, economics, sociology and psychology, to name but a few [91].

Hoogendorn and Knoop [57] defined traffic flow theory as the description and analysis of the fundamental characteristics of traffic flow, such as flow and density relationships, road capacities and headway distributions. This theory may also be extended to include the effects of external factors such as driver behaviour, weather conditions and traffic control policies. Furthermore, traffic flow theory may be partitioned into two main fields, namely *microscopic traffic flow theory* and *macroscopic traffic flow theory*. In microscopic traffic flow theory, the behaviour of individual vehicles, related specifically to flow, speed and density, is studied, whereas the focus in macroscopic traffic flow theory shifts to a more aggregated view, considering the flow, speed and densities for numerous vehicles collectively, typically for a specified stretch of road.

3.1.1 Macroscopic Traffic Flow Theory

Traffic speed, density and flow are the underlying variables of traffic analysis [92]. Traffic *flow* is defined as the number of vehicles, n, that pass some designated point on a highway during a time interval of length t. According to this definition, the traffic flow is given by

$$q = \frac{n}{t}, \tag{3.1}$$

expressed in vehicles per time unit. It is, however, not only the number of vehicles that pass a point that are of interest, but also the amount of time that elapses between the arrival of successive vehicles at a specific point along a highway. This interarrival time of the vehicles is known as *time headway*, denoted by h_{t_i} and is measured from a common point on each vehicle (*e.g.* the front or rear bumper) as it passes a specific stationary point [161]. Headway may be related to flow by the relationship

$$q = \frac{n}{\sum_{i=1}^{n} h_{t_i}} = \frac{1}{\bar{h}_t}, \tag{3.2}$$

where

$$\bar{h}_t = \frac{1}{n} \sum_{i=1}^{n} h_{t_i}$$

represents the average time headway of n vehicles during a time interval of length t. Average speed may be defined in two ways, the first being the average speed at which vehicles travel when passing a specific stationary point, and the second based on the amount of time that vehicles require to traverse a set distance L. The first is known as the *time mean speed*, is denoted by $\bar{u}_t$ and is given by

$$\bar{u}_t = \frac{1}{n} \sum_{i=1}^{n} u_i, \tag{3.3}$$

where u_i represents the instantaneous speed of vehicle i when passing the designated point. The second measure of average speed, known as the *space mean speed*, is given by

$$u = \frac{1}{\frac{1}{n} \sum_{i=1}^{n} \frac{1}{L/t_i}}, \tag{3.4}$$

where t_i represents the amount of time required by vehicle i to traverse the set distance L. Finally, traffic *density* ρ is simply defined as the number of vehicles that occupy a stretch of road at any point in time, given by

$$\rho = \frac{n}{L} = \frac{n}{\sum_{i=1}^{n} h_{si}} = \frac{1}{\bar{h}_s}, \tag{3.5}$$

where

$$\bar{h}_s = \frac{1}{n} \sum_{i=1}^{n} h_{s_i}$$

is the average *space headway* between the n vehicles travelling along the specific stretch of road, again measured with respect to a common point on every vehicle [161]. Traffic density therefore provides an indication of how crowded the stretch of road under consideration is. It is, however, important to note that this definition of density does not take specific vehicle lengths, and as a result specific traffic composition, into account, as only the number of vehicles is considered.

Based on these definitions, a simple identity may be formulated to showcase the basic relationship between speed, density and flow, called the *fundamental relation of traffic flow theory* [168], or the *continuity equation* [57]. This identity is given by

$$q = u\rho, \tag{3.6}$$

with typical units of flow, speed and density being vehicles per hour (veh/h), kilometres per hour (km/h), and vehicles per kilometre (veh/km), respectively. The significance of (3.6) is that it allows an analyst to estimate any of the three macroscopic variables, given the other two. This is especially useful when estimating density, which is often difficult to measure [91].

The Fundamental Diagrams

Greenshields [46] defined three basic traffic stream models, namely the Speed-Density Model, the Flow-Density Model, and the Speed-Flow Model, based on the fundamental relationship (3.6). These models give rise to the so-called fundamental diagrams of traffic flow theory, which provide a graphical representation of the statistical relationships between the macroscopic traffic flow variables of speed, flow and density, based on the premise that drivers act in a similar manner when faced with similar traffic conditions [57]. As a result, Maerivoet and de Moor [91] distinguished between three categories of traffic flow conditions, namely *free-flow traffic*, *capacity-flow traffic* and *congested traffic*.

Free-flow traffic occurs when vehicles are able to travel at their desired speeds, untroubled by queues or other slower moving vehicles. As a result, free-flow traffic typically prevails under light traffic flow conditions [91]. The desired, or free-flow, speeds depend on the vehicle, as well as the driver and road section characteristics, and the current weather conditions and traffic rules (*e.g.* speed limits) [57]. This desired, free flow speed, denoted by u_f, is summarised by the average speed of the vehicles travelling along the section of road under consideration. Due to the low traffic densities observed when free-flow traffic prevails, the space headway between the vehicles is typically large, and minor disturbances due to overtaking manoeuvres or sudden braking do not have a significant effect on the aggregated traffic flow, which may, as a result be considered to be *stable* [91].

As traffic density increases, so does traffic flow, due to the smaller space headways between individual vehicles. This trend continues until the flow along a lane reaches its maximum, known as *capacity flow*, denoted by $q_{\max}$. This capacity flow depends not only on the current

traffic density, but also on the average speed along that specific lane. From (3.2) it is clear that capacity flow is reached at the point where the average time headway is at its minimum, which indicates tightly packed clusters of vehicles travelling at *capacity-flow speed*, which is typically lower than the free-flow speed [57]. These clusters of vehicles are, however, often unstable with the slightest braking action of one vehicle exhibiting a backward cascading effect, resulting in exaggerated braking by the following vehicles.

As the traffic density increases further, vehicles eventually start to slow down in order to avoid collisions caused by the decreased space and time headways. Due to these chain reactions of slowing vehicles, traffic flow starts to deteriorate, and the resulting, saturated traffic conditions are known as *congested traffic* [57]. Further increases in vehicle density will lead to so-called *stop-and-go traffic*, where vehicles often have to slow down significantly or even stop in order to avoid collisions. As traffic density further increases, the traffic becomes motionless, as the space headway between vehicles has reached a minimum bumper-to-bumper distance. In this state, the traffic conditions are referred to as *jammed traffic*. This maximum density, at which the traffic flow has deteriorated to such a point that the vehicles have become stationary, is known as the *jam density*, denoted by ρ_{jam}.

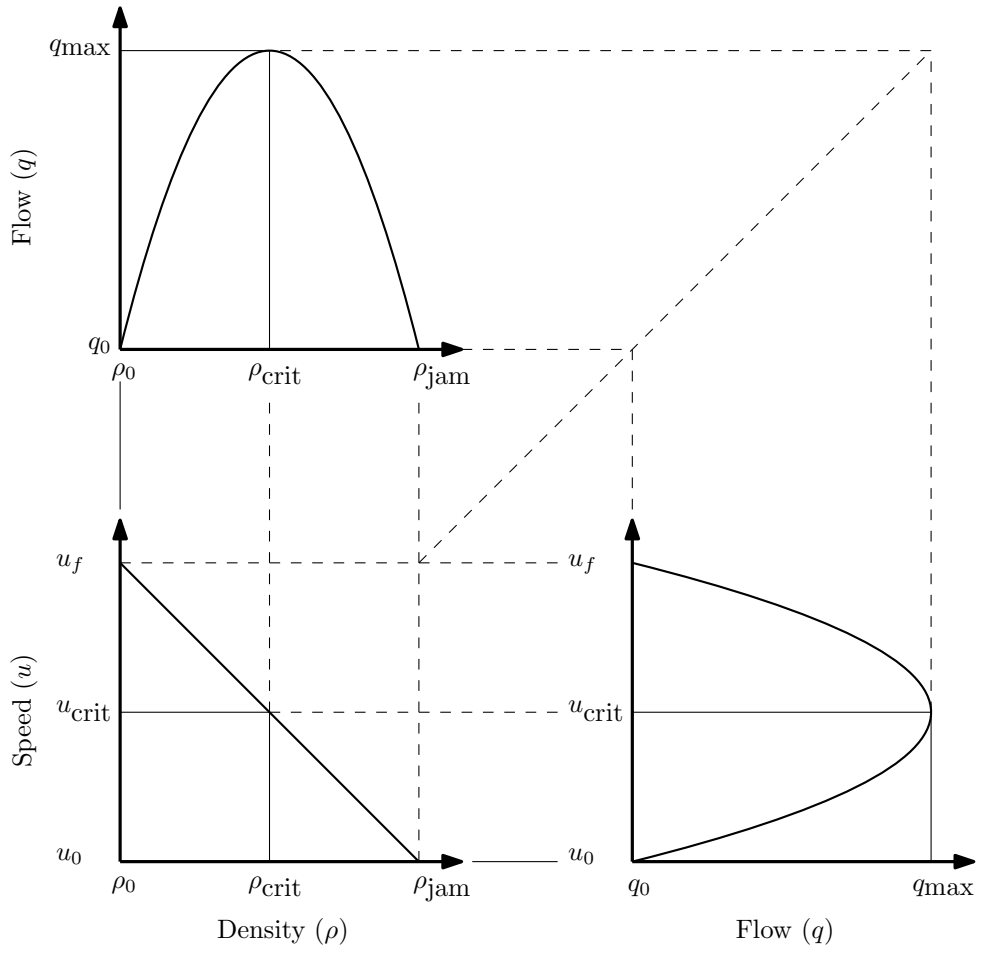


FIGURE 3.1: *The fundamental diagrams of macroscopic traffic flow theory, relating flow, speed and density (assuming a linear speed-density relationship). Adapted from Mannering and Kilareski [92].*

The fundamental diagrams, employed to illustrate the relationships and traffic behaviour described above are shown in Figure 3.1. In this instance, a linear speed-density relationship is assumed. As may be seen from the fundamental diagram corresponding to Mannering and Ki-

lareski's [92] Flow-Density Model, in the top left of Figure 3.1, the flow for a stretch of road increases until a critical density ρ_{crit} is reached at which the maximum flow occurs. This happens at a specific critical speed u_{crit}, as shown in the fundamental diagram corresponding to the Speed-Density Model, in the bottom left corner of Figure 3.1. As may be seen from the fundamental diagram corresponding to the Speed-Flow Model, shown in the bottom right of Figure 3.1, maximum flow occurs at this critical speed. This diagram is not as intuitive to interpret as the others due to the fact that each speed value corresponds to two distinct flow values. The diagram is separated into two regions of flow by the horizontal line corresponding to the critical speed u_{crit}. The region above this line corresponds to free-flowing traffic. Points along the curve in this region indicate that fewer vehicles pass a fixed point at higher speeds, with increased space headways separating them, while the point corresponding to the same flow on the curve below u_{crit} indicates that more vehicles travel past the fixed point, at a lower speed, with decreased space headways separating them. Various empirical models have also been formulated in order to determine the fundamental diagrams for specific sections of road, based on nonlinear speed-density relationships.

3.1.2 Microscopic Traffic Flow Theory

There are certain characteristics that are inherent to specific vehicles, as well as their drivers, which may have an influence on a traffic flow. In *microscopic* traffic flow models these individual characteristics are employed in order to describe the traffic flow in terms of the underlying interactions between drivers and their vehicles with one another [57]. Naturally, the behaviour of a vehicle in a given traffic environment is largely based on the behavioural aspects of its driver. For this reason, several models have been developed which are able to take varying driver behaviour into account in microscopic descriptions of traffic flow. The incorporation of human factors, however, greatly increases the model complexity [91] and, as a result, many microscopic traffic flow theories employ combined vehicle-driver combinations, modelling the vehicle and driver as a single entity in an attempt to reduce the model complexity.

Microscopic Traffic Flow Variables and Characteristics

When considering individual vehicles, several variables are typically associated with each vehicle in order to provide an accurate description of the traffic flow. These variables include the *length* of vehicle i, denoted by ℓ_i, the *longitudinal position* (typically taken to be the position of the rear bumper [91]) of vehicle i, denoted by x_i, the *speed* of the vehicle, given by

$$u_i = \frac{\mathrm{d}x_i}{\mathrm{d}t},$$

and its *acceleration*

$$a_i = \frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{\mathrm{d}^2x_i}{\mathrm{d}t^2}.$$

Microscopic speed characteristics describe the speed properties of an individual vehicle passing a fixed point or a short segment of road during a specified time period [96]. Roadway design features, interrupted flow situations (*e.g.* stop streets, signalised intersections) and other road users make up the immediate environment which, in turn, affects the speed at which each individual vehicle travels. It is typically only the accelerating capabilities of the vehicles that directly alter their speeds, not other factors such as road and wind friction [91].

As in the macroscopic modelling paradigm, two other important characteristics of the traffic flow are the space and time headways. Time headway is often considered to be one of the most

important microscopic traffic flow characteristics due to its direct influence on the capacity of a road section [57]. The time headway h_{t_i} of vehicle i is typically taken to be the difference in the passage time of the rear bumper of vehicle $i-1$ in front of it and its own bumper across a specific stationary point. The time headway therefore comprises a time gap t_{g_i}, which is defined as the time taken for the front bumper of vehicle i to reach the current position of vehicle $i-1$, and an occupancy time t_{o_i}, which is defined as the time required for vehicle i to traverse its own length, that is $h_{t_i} = t_{g_i} + t_{o_i}$ [91]. Hoogendoorn and Knoop [57] referred to the time gap as the *net time headway*, which is considered to be particularly important in respect of the analysis and modelling of the space requirements for overtaking manoeuvres. This type of analysis is also known as critical gap analysis, while the sum of the time gap and occupancy time is referred to as the *gross headway*.

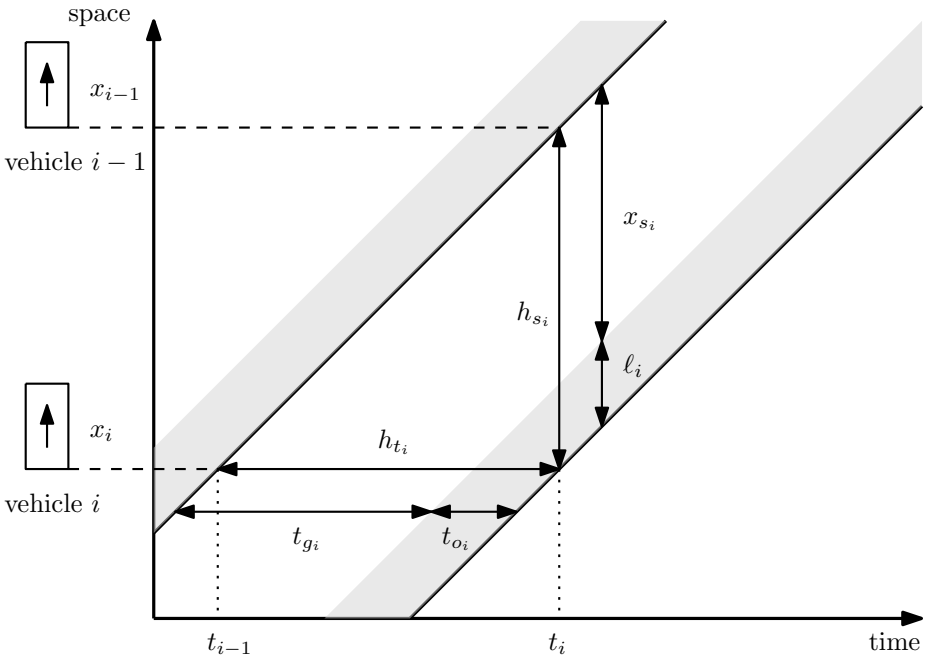


FIGURE 3.2: *A time-space diagram illustrating the trajectories of two vehicles ($i-1$ and i), as well as their time and space headways. Adapted from Logghe [90].*

Similarly, a space headway h_{s_i} is also associated with every vehicle i. The space headway is defined as the distance from the rear bumper of vehicle $i-1$, to the rear bumper of vehicle i [57]. As with the time headway, the space headway is also the sum of two components, namely the *space gap* x_{s_i} and the vehicle length ℓ_i. Again, the sum of these two components is typically referred to as the *gross space headway*, while the space gap alone is known as the *net space headway* [57]. It is important to note that time headways are local, microscopic characteristics as they relate to the behaviour of individual vehicles, and are typically measured from a fixed point along a roadway, whereas space headways are instantaneous measurements taken at a given point in time. As may be seen from the definition of the expressions for the space and time headways, these two characteristics are highly correlated. This correlation is illustrated by the relationship

$$\frac{h_{s_i}}{h_{t_i}} = \frac{x_{s_i}}{t_{g_i}} = \frac{\ell_i}{t_{o_i}} = u_i. \tag{3.7}$$

The relationship in (3.7) is illustrated graphically in a so-called *time-space diagram* in Figure 3.2. In the figure, the trajectories of two vehicles, $i-1$ and i, are traced out, showing their respective positions at every point in time. The speed of the vehicles is given by the gradient of the tangent to the line indicating each vehicle's trajectory. In the case shown in Figure 3.2, the two vehicles

are assumed to travel at the same constant speed for the sake of simplicty, resulting in the parallel trajectories.

Car-following Models

Car-following models have been established in the literature in order to capture the behaviour of one vehicle, following another along a specific road section, while incorporating the aforementioned microscopic traffic flow characteristics. Employing the notation presented above, a general car-following situation is depicted in Figure 3.3.

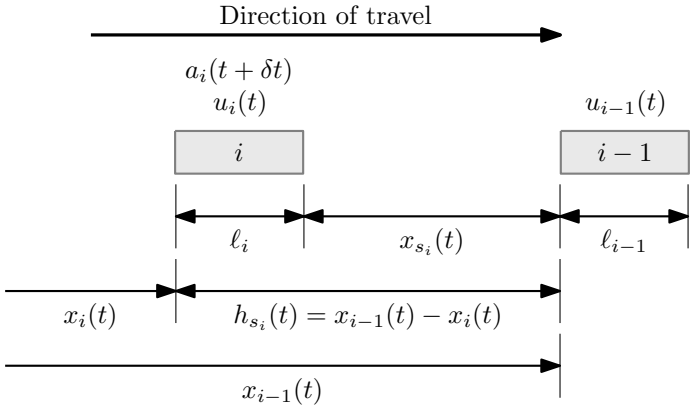


FIGURE 3.3: *Car-following theory notations and definitions. Adapted from May [96].*

In Figure 3.3, vehicle i is following vehicle $i-1$ in a left-to-right direction. It is important to note that the acceleration rate a_i of the following vehicle is specified as occurring at time $t+\delta t$, and not at t. The time duration δt represents a *reaction time*, required for the driver to react and subsequently apply the acceleration (or deceleration) rate [96]. The relative velocity of the lead vehicle and the following vehicle is denoted by $u_{i-1}(t) - u_t(t)$. Given a situation where this relative velocity is positive, the lead vehicle is travelling at a higher velocity, and as a result the magnitude of the distance headway between the vehicles is increasing. Conversely, if this relative velocity is negative, the following vehicle is travelling at a higher velocity, and the magnitude of the distance headway between the vehicles is decreasing. If the value of $a_i(t+\delta t)$ is positive, vehicle i will start accelerating at time $t+\delta t$, with a negative value of $a_i(t+\delta t)$ indicating that vehicle i will start decelerating at time $t+\delta t$. Finally, if the value of $a_i(t+\delta t)$ is equal to zero, vehicle i is travelling at a constant velocity [96].

Various rules and theories have been proposed in the literature for governing when and at what rate vehicles should accelerate (or decelerate), based upon the above car-following model. Pipes' [121] theory suggests that vehicles follow the guidelines set out in the California Motor Vehicle Code, stating that: "A good rule for following another vehicle at a safe distance is to allow yourself at least the length of a car between your vehicle and the vehicle ahead for every ten miles per hour of speed at which you are travelling." The resulting expression for distance headway is therefore

$$d_{\min} = \ell_i \left[\frac{u_i(t)}{(1.47)(16.0934)} \right] + \ell_i, \tag{3.8}$$

measured in metres. A comparison of the computed following distances with field data have shown that the computed values are sufficiently accurate for speeds ranging from 16–60 kilometres per hour, but significant differences were observed for speeds falling outside that range [96]. The approach adopted by Forbes and Simpson [37] considered the reaction time required

by the following vehicle to perceive the need for deceleration. As a result, the time gap between the lead and following vehicle should always be greater than, or at least equal to, this reaction time. Therefore, the minimum time headway should be equal to the reaction time added to the time it takes for the vehicle to traverse its own length. This relationship is given by

$$h_{t_i,\min} = \delta t + \frac{\ell_i}{u_i(t)}. \tag{3.9}$$

Again, as with Pipes' model, Forbes' model performs well in the range 16–60 kilometres per hour. Forbes' model outperforms Pipes' model at speeds higher than 60 kilometres per hour, but it still shows significant errors when compared to the in-field test data [96]. A third example of car-following theory comprises the suite of models proposed by the General Motors research laboratory [26, 40, 41, 55]. These models are significantly more extensive and particularly important due to the wide range of accompanying, comprehensive field experiments, as well as the discovery of the mathematical bridge between macroscopic and microscopic traffic flow theories. All the models proposed took the general form

$$\text{response} = f(\text{sensitivity, stimuli}),$$

where the response was always represented by the acceleration (or deceleration) to be performed by the following vehicle, and the stimulus was always represented as the relative velocity between the lead and following vehicle. Varying representations of the sensitivity, ranging from a constant sensitivity to empirically calibrated sensitivity functions based on vehicle speeds, differentiate between the five models formulated at the General Motors laboratories [96].

3.2 Highway Traffic Control Measures

Highways were originally built to provide virtually unlimited mobility to road users. As a result thereof, traffic control measures on highways were initially implemented mainly for safety reasons. The on-going drastic global expansion of car ownership and travel demand have, however, led to these measures being implemented in such manners as to maintain the efficiency of traffic flow on highways [115, 130]. Various control measures may be employed as a means of improving the efficiency of a highway network, including ramp metering, dynamic speed limits, and dynamic lane allocation [130].

3.2.1 Ramp Metering

Ramp metering (RM) has been claimed to be one of the most effective highway traffic control measures [115]. RM improves highway traffic flow by effectively regulating traffic flow onto the highway at an on-ramp, thus increasing mainline throughput and served traffic volume due to the avoidance of capacity loss and blockage of on-ramps due to congestion. RM strategies have been proven effective in both macroscopic as well as microscopic simulation environments, and have been implemented at various locations in the United States of America, France, Italy, Germany, New Zealand, the United Kingdom and the Netherlands [27, 75, 112].

Consider two scenarios of a highway on-ramp, as shown in Figure 3.4, one where RM is not employed (a), and one where RM is employed (b). Let q_{in} denote the upstream highway flow, d the on-ramp demand, q_{con} the mainstream outflow in the presence of congestion, and q_{cap} the highway capacity. It has been shown that the traffic outflow in the presence of congestion is between 5%–10% lower than the highway capacity [115]. It is assumed in Figure 3.4(b) that a

metering rate r is employed, allowing only limited numbers of vehicles onto the highway such that the highway capacity is never exceeded. Papageorgiou *et al.* [111] have shown that the amelioration $\triangle T$ (expressed as a percentage) of the total time spent is given by

$$\triangle T = \frac{q_{\text{cap}} - q_{\text{con}}}{q_{\text{in}} + d - q_{\text{con}}} \times 100. \tag{3.10}$$

For example, if the total demand exceeds the highway capacity by 20% (*i.e.* $q_{\text{in}} + d = 1.2q_{\text{cap}}$) and the capacity drop due to congestion is 5% (*i.e.* $q_{\text{con}} = 0.95q_{\text{cap}}$), then $\triangle T = 20\%$ results from (3.10), which demonstrates the power of effective RM strategies [115].

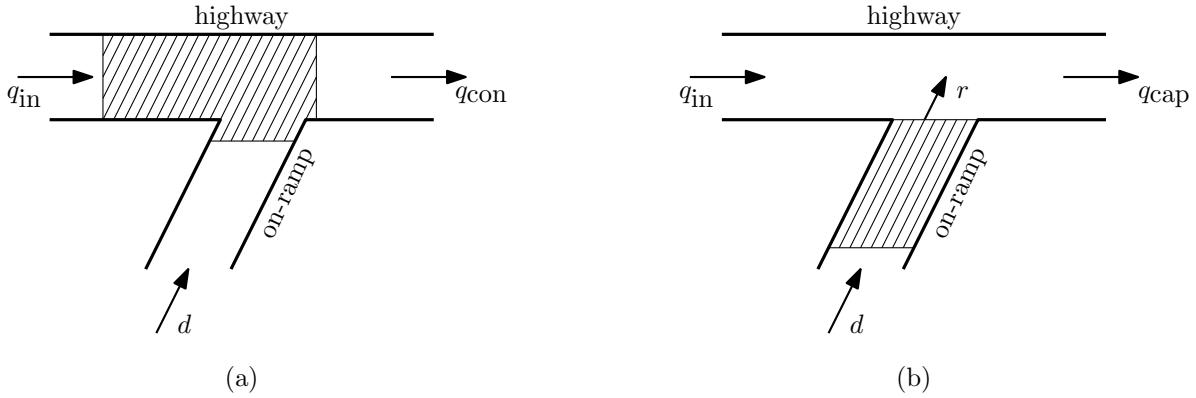


FIGURE 3.4: *Two cases of traffic flow onto a section of highway from an on-ramp, (a) without RM and (b) with RM, where shaded areas indicate congestion zones. Adapted from Papageorgiou and Kotsialos [115].*

Various RM strategies have been proposed in the literature. The most significant of these are briefly discussed in this section.

Fixed-time strategies

Fixed-time RM strategies are determined in an offline fashion for specific times of day, based on historical demands, without the use of real-time traffic information [115]. This approach was first introduced by Wattleworth [171]. According to this approach, a highway with several on- and off-ramps is partitioned into sections, each containing only one on-ramp. The flow q on a highway section j may then be defined as

$$q_j = \sum_{i=1}^{j} \alpha_{ij} r_i,$$

where r_i represents the on-ramp flow (in units of veh/h) for section i, and $\alpha_{ij} \in [0, 1]$ represents the proportion of vehicles that enter the highway in section i and do not leave the highway upstream of section j. In order to avoid congestion in the network, it is required that

$$q_j \leq q_{\text{cap},j},$$

where $q_{\text{cap},j}$ denotes the capacity of highway section j. Finally, the metering constraint

$$r_{j,\min} \leq r_j \leq \min\{r_{j,\max}, d_j\}$$

must hold, where d_j is the on-ramp demand and $r_{j,\max}$ is the on-ramp capacity at on-ramp j. As objective function then, one may wish to maximise the number of served vehicles at all

on-ramps, that is to

$$\text{maximise} \sum_j r_j,$$

or to minimise the on-ramp queues, *i.e.*

$$\text{minimising} \sum_j (d_j - r_j)^2,$$

while satisfying the highway and on-ramp capacity constraints. The most prominent fixed-time RM algorithm is AMOC, introduced by Kotsialos *et al.* [74], in which a second-order macroscopic traffic network model called METANET [100] is employed to solve a nonlinear optimisation problem with the goal of minimising the total travel time. The RM problem is formulated as a dynamic optimal control problem with constrained control variables in the AMOC control strategy. This dynamic control problem may then be solved numerically for given demands $d_i(t)$ and turning rates β_n^m over a specified time period. The general discrete-time optimal control problem formulation involves

$$\text{minimising } J = \vartheta[T] + \sum_{t=0}^{T-1} \varphi[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)] \tag{3.11}$$

subject to

$$\begin{aligned}
\boldsymbol{x}(t+1) &= \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)], && && (3.12)\\
\boldsymbol{x}(0) &= \boldsymbol{x}_0, && && (3.13)\\
u_{i,\min} &\leq u_i(t) && \text{for all } i \in \{1, \ldots, m\}, && (3.14)\\
u_i(t) &\leq u_{i,\max} && \text{for all } i \in \{1, \ldots, m\}, && (3.15)
\end{aligned}$$

where T is the time period under consideration, $\boldsymbol{x} \in \Re^n$ is the state vector, $\boldsymbol{u} \in \Re^m$ is the vector of control variables, $\boldsymbol{d}$ is the vector of disturbances acting on the traffic process, and ϑ and φ are arbitrary, twice-differentiable, nonlinear cost functions. Gomes and Horowitz [43] presented a similar nonlinear optimisation approach based on a first-order macroscopic traffic simulation model, called the *asymmetric cell transmission model* (ACTM). First-order models are significantly simpler than second-order models, and as a result, optimisation based on first-order approaches may be solved for much larger problem instances.

The main drawback of fixed-time strategies is that the optimal solutions found are specific to the historic period taken into account when solving the problem [115]. Thus real-time traffic fluctuations are not taken into account when RM strategies are defined. This may lead to congestion or underutilisation of the highway even though RM strategies have been determined and implemented. Traffic-responsive metering strategies are required to remedy this deficiency [115].

Traffic-responsive ramp metering strategies

Typically, the aim of traffic-responsive RM strategies is to keep the conditions on a highway close to a set of pre-specified values, based on real-time traffic information [115]. The simplest traffic-responsive RM controllers are local or independent controllers, which rely on measurements taken directly in the vicinity of the on-ramp in order to determine the metering rate [130]. One of the earliest local RM controllers is the demand-capacity algorithm introduced by Masher *et*

al. [95]. According to the demand-capacity algorithm, the metering rate at time t is determined for the next time by

$$r(t+1) = \begin{cases} \max\{q_{\text{cap}} - q_{\text{in}}(t), r_{\min}\} & \text{if } \rho_{\text{out}}(t) \leq \rho_{\text{crit}} \\ r_{\min} & \text{otherwise,} \end{cases} \tag{3.16}$$

where q_{cap} represents the highway capacity downstream of the on-ramp, q_{in} is the highway flow upstream of the on-ramp, ρ_{out} is the density of vehicles downstream of the on-ramp, ρ_{crit} is the critical highway density at which maximum flow occurs, and $r_{\min}$ is the minimum allowable metering rate. The demand-capacity algorithm is considered to be an open-loop or feed-forward control approach due to the output of the system not being directly employed in the determination of the control signal [115]. Due to the open-loop nature of the control algorithm, this control measure is prone to model deficiencies and its performance will degrade if the target value q_{cap} is not accurate [130].

As a closed-loop alternative, Papageorgiou *et al.* [112] introduced the well-known *Asservissement Linéaire d'Entreé Autoroutière* (ALINEA) RM strategy. The aim of the ALINEA algorithm is to regulate the downstream density so as to achieve maximum outflow. The metering rate according to ALINEA is given by

$$r(t+1) = r(t) + K_r[\hat{\rho} - \rho_{\text{out}}(t+1)], \tag{3.17}$$

where $K_r > 0$ is a control parameter, and $\hat{\rho}$ represents the desired downstream density (typically $\hat{\rho} = \rho_{\text{crit}}$), at which the highway outflow becomes close to q_{cap}. This control structure is known as integral-control, and is one of the simplest linear time-invariant controllers. A comparison of the functional structures of these algorithms is illustrated graphically in Figures 3.5 (a) and (b).

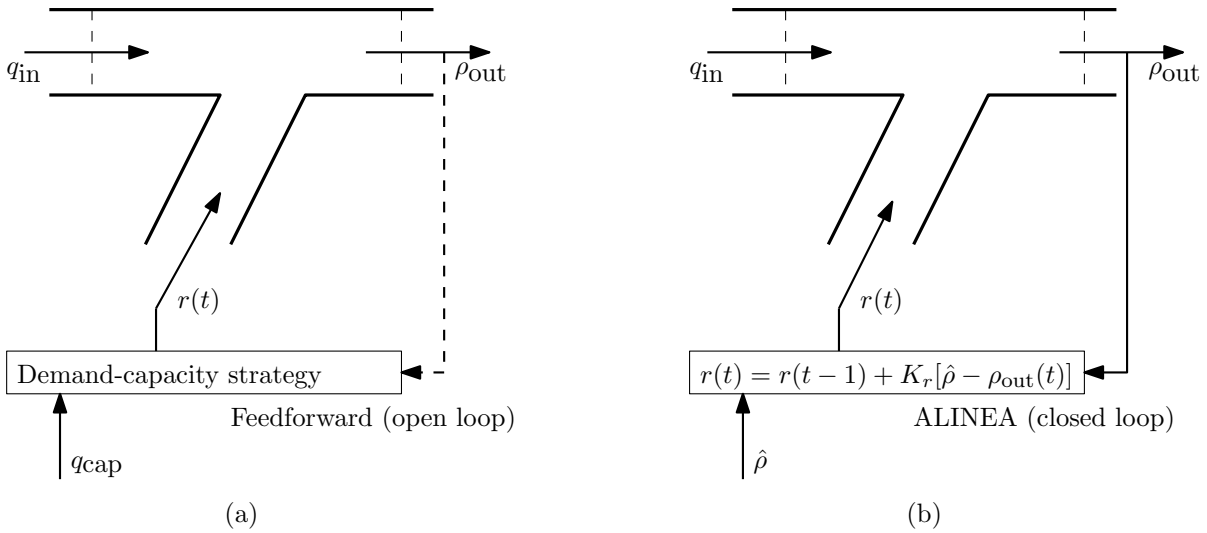


FIGURE 3.5: *Functional structure of (a) the demand-capacity algorithm and (b) the ALINEA algorithm. Adapted from Papageorgiou et al. [115].*

The simplicity and effectiveness of the ALINEA algorithm have made it the best-known RM controller, with validated real-world performance through field implementations [113]. Various alterations of the ALINEA algorithm have been proposed. Zhang and Ritchie [178], for example, proposed the use of an artificial neural network in the place of the control parameter K_r. The aim of such a neural network is to replace the constant parameter with one that varies according to the downstream density in order to provide improved traffic regulation based on the density.

Another well-known extension of the ALINEA RM control strategy is the so-called PI-ALINEA RM control strategy introduced by Wang *et al.* [167]. In this extension, a shortcoming of the ALINEA control strategy, namely that ALINEA is unable to take into account bottlenecks further downstream than the direct lane merge is addressed. This is achieved by adding an integral control loop to the feedback controller, which works in conjunction with the existing proportional control loop in the original ALINEA controller. The metering rate to be applied at the on-ramp is then determined according to

$$r(t) = r(t-1) - K_p[\rho_{\text{out}}(t) - \rho_{\text{out}}(t-1)] + K_r[\hat{\rho} - \rho_{\text{out}}], \tag{3.18}$$

where K_p and K_r denote the integral and proportional controller gain parameters, respectively. Following a theoretical analysis of the proposed controller, as well as extensive numerical experiments, Wang *et al.* [167] concluded that using extensive parameter tuning, PI-ALINEA would perform at least as well as ALINEA in situations with only the immediate downstream bottleneck, while PI-ALINEA was able to outperform ALINEA in situations where a distant downstream bottleneck had to be taken into account.

Although independent RM controllers are often effective and easily implemented, problems may arise when several on-ramps are located in close proximity, as then equity cannot be achieved in respect of all on-ramps. Furthermore, performance is often severely degraded when the on-ramp queue storage space is limited [130]. In order to deal with limited on-ramp storage space, a second algorithm is often implemented in order to determine a minimum metering rate so as to prevent the maximum permissible queue length being exceeded. This minimum metering rate may, in turn, lead to degraded RM performance. Coordinated RM approaches attempt to remedy this situation by simultaneously controlling the available queueing space, as well as the metering rate at multiple adjacent on-ramps [130].

Two early examples of coordinated RM algorithms are BOTTLENECK [65] and ZONE [80]. The BOTTLENECK algorithm has two major components, a local RM algorithm, determining metering rates at a local level based on occupancy, and a coordination algorithm for determining system-level metering rates, based on system capacity constraints. The local-level controller functions similarly to the demand-capacity algorithm. For the system level controller, the highway is partitioned into a number of sections. For each of these sections, the number of vehicles stored in that section are determined by monitoring vehicle inflows and outflows. If the number of vehicles stored in a section is positive, the metering rate for that section is reduced. Finally, the more restrictive of the local- and system-level metering rates is applied at each of the on-ramps. Similar to BOTTLENECK, the ZONE algorithm also consists of a local-level controller, and a system-level controller. RM at the local level is again determined based on occupancy, typically using a variation on the demand-capacity algorithm. At the system level, volume control is employed to ensure that the total traffic volume flowing through a pre-defined zone is not exceeded. Chu *et al.* [27] compared the performance of BOTTLENECK and ZONE with ALINEA, and showed that ALINEA performed better than both coordinated RM strategies. When the local-level controllers of BOTTLENECK and ZONE were replaced with the ALINEA controller, however, the coordinated strategies could be improved significantly, outperforming ALINEA in its original form.

The first attempt at a coordinated generalisation and extension of ALINEA was the METALINE multivariable regulator strategy proposed by Papageorgiou *et al.* [110]. RM volumes are calculated as

$$\boldsymbol{r}(t+1) = \boldsymbol{r}(t) - \boldsymbol{K}_1[\boldsymbol{\rho}(t+1) - \boldsymbol{\rho}(t)] + \boldsymbol{K}_2[\hat{\boldsymbol{\rho}} - \boldsymbol{\rho}(t)], \tag{3.19}$$

where $\boldsymbol{r} = [r_1, \ldots, r_m]^T$ is the vector of m controllable on-ramp metering rates, $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_m]^T$ is the vector of m measured densities along the highway stretch considered, and $\hat{\boldsymbol{\rho}} = [\hat{\rho}_1, \ldots, \hat{\rho}_m]^T$

is the vector of corresponding pre-specified desired density values. Finally, $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are the regulator's constant gain matrices. Field results and simulation comparisons of METALINE and ALINEA have shown that only in cases of non-recurrent congestion due to traffic incidents does METALINE outperform ALINEA [113]. The added design effort required for METALINE is therefore often not justified by the marginal improvements achieved compared with those of the simple ALINEA algorithm.

In an attempt to find a simpler coordinated RM strategy based on ALINEA, Papamichail and Papageorgiou [118] proposed a linked RM strategy with the aim of equalising the queue length of each on-ramp with the on-ramp downstream of its location. In this algorithm, three metering rates are calculated, the first being the local ramp flow $r(t)$, calculated from (3.17). The second ramp flow, called the *queue override ramp flow* $r^w(t)$, is given by

$$r^w(t) = -\frac{1}{T_c}[w_{\max} - w(t)] + d(t-1), \tag{3.20}$$

where T_c is the control cycle, $w_{\max}$ is the maximum allowable queue length, w is the current queue length, and d is the on-ramp demand. The aim of this control law is to maintain an on-ramp queue that does not exceed the maximum allowable queue length. The third control law links the coordinates of each on-ramp with the on-ramp downstream of its location in an attempt to ensure that they have queues of similar length. This linked control ramp flow r^{LC} is determined by

$$r^{LC}(t) = -K_w[w_{\min} - w(t)] + d(t-1), \tag{3.21}$$

where K_w is a control parameter for managing the smoothness of response, and $w_{\min}$ is the desired minimum queue length determined according to the queue present at the on-ramp situated downstream of the current on-ramp location. The value of $w_{\min}$ is initially set to zero, and is only changed once congestion forms at the downstream on-ramp, and the queue length at that on-ramp exceeds some threshold value. Then the minimum queue length $w_{\min}$ is enforced at the upstream on-ramp so as to provide more space on the highway for vehicles entering the highway at the downstream on-ramp. The minimum queue length is reset to zero once the queue length at the downstream on-ramp has fallen below the pre-specified threshold value. The final metering rate may then be calculated as

$$r(t) = \max\{\min[r(t), r^{LC}(t)], r^w(t)\}. \tag{3.22}$$

The linked RM algorithm has been evaluated within the context of a macroscopic traffic simulation model, and it has been shown that its performance is significantly better than that of the conventional ALINEA algorithm when ramp queue limits are imposed [118].

The heuristic adaptive RM control approaches presented above are generally easy to implement. They do, however, often require extensive tuning of parameters, which may result in subpar performance in a practical environment [130]. In another approach found in the literature, the aim is to overcome the problem of parameter tuning. This approach is called *model predictive control* (MPC). Following this approach, metering rates for multiple on-ramps are determined by solving nonlinear optimisation models with the goal of minimising the total travel time of vehicles. In order to provide traffic-responsive solutions, this optimisation problem is solved in a rolling horizon scheme. At each time instant t, a new optimisation is performed over the prediction horizon N_p, and only the first cycle of the solution is applied. This procedure is repeated at each time period. In order to reduce complexity, a control horizon $N_c \leq N_p$ is often introduced. After the control horizon has passed, the control signal is assumed to be constant. As a result, there are effectively two loops: The rolling horizon loop repeated at each time step t, and the optimisation loop inside the controller. The optimisation loop is repeated as often as

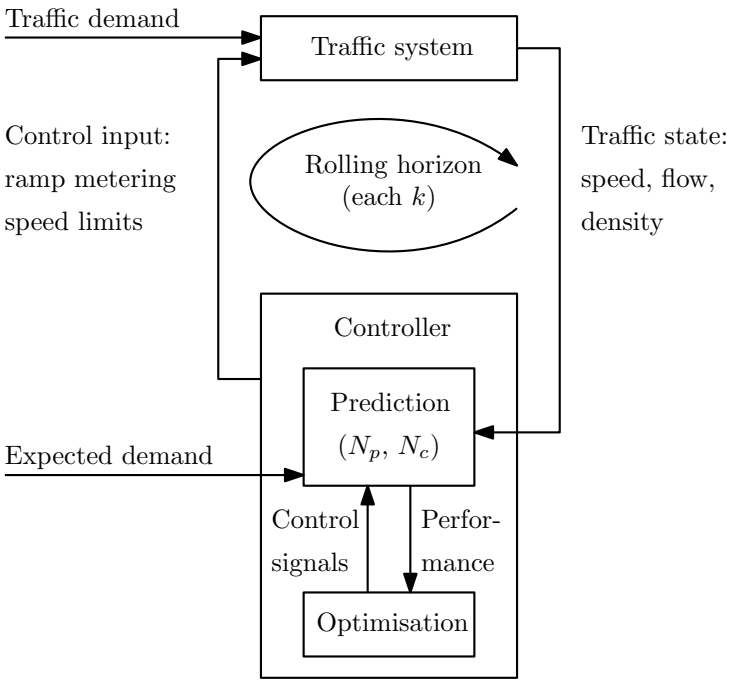


FIGURE 3.6: *Schematic illustration of the MPC structure for traffic control problems. Adapted from Hegyi et al. [53].*

required in order to find an optimal solution for the control signal at a time instant t, given the values of N_p and N_c, the current traffic state and the expected demand [53]. The structure of MPC is illustrated graphically in Figure 3.6.

Bellemans *et al.* [15] and Hegyi *et al.* [53] have successfully applied MPC for optimal, traffic-responsive RM. In both cases, the highway was modelled in the context of the second-order macroscopic traffic simulation model METANET. In both cases, only one on-ramp was considered due to the large computational overhead of solving the nonlinear optimisation problem. In order to overcome the problem of the computational overhead, Ghods *et al.* [42] proposed a decentralised solution approach based on the game theoretic concept of *fictitious play* for solving the nonlinear optimisation model. The decentralisation approach allows the computation to be handled by multiple nodes, thereby rendering the approach applicable to larger problem instances. Optimal fixed-time strategies provide optimal performance based on the premise that there are no disturbances, and that the forecast input data are accurate. MPC approaches may mitigate the performance drop caused by disturbances, but are limited to small networks due to the large computational overhead. As a trade-off, Papamichail *et al.* [117] proposed a hierarchical approach with the aim of providing semi-optimal control for large networks. The hierarchical control approach is illustrated in Figure 3.7. As may be seen in the figure, the hierarchical control approach consists of three layers. Real-time traffic measurements, together with historical data, are provided to the estimation/prediction layer, which are then used to provide an estimate of the state of the traffic, and provide predictions of future demands. The optimal control engine of the optimisation layer then solves the nonlinear optimisation problem in order to determine optimal RM values for each of the local regulators. This optimisation is carried out every ten minutes. Due to disturbances, the traffic flow does not remain stable during the ten-minute intervals and, as a result, the metering rates become suboptimal. Finally, instead of direct application of the optimal metering rates, ALINEA is employed at the direct control layer, in order to vary the metering rate around the point set by the optimisation layer [117].

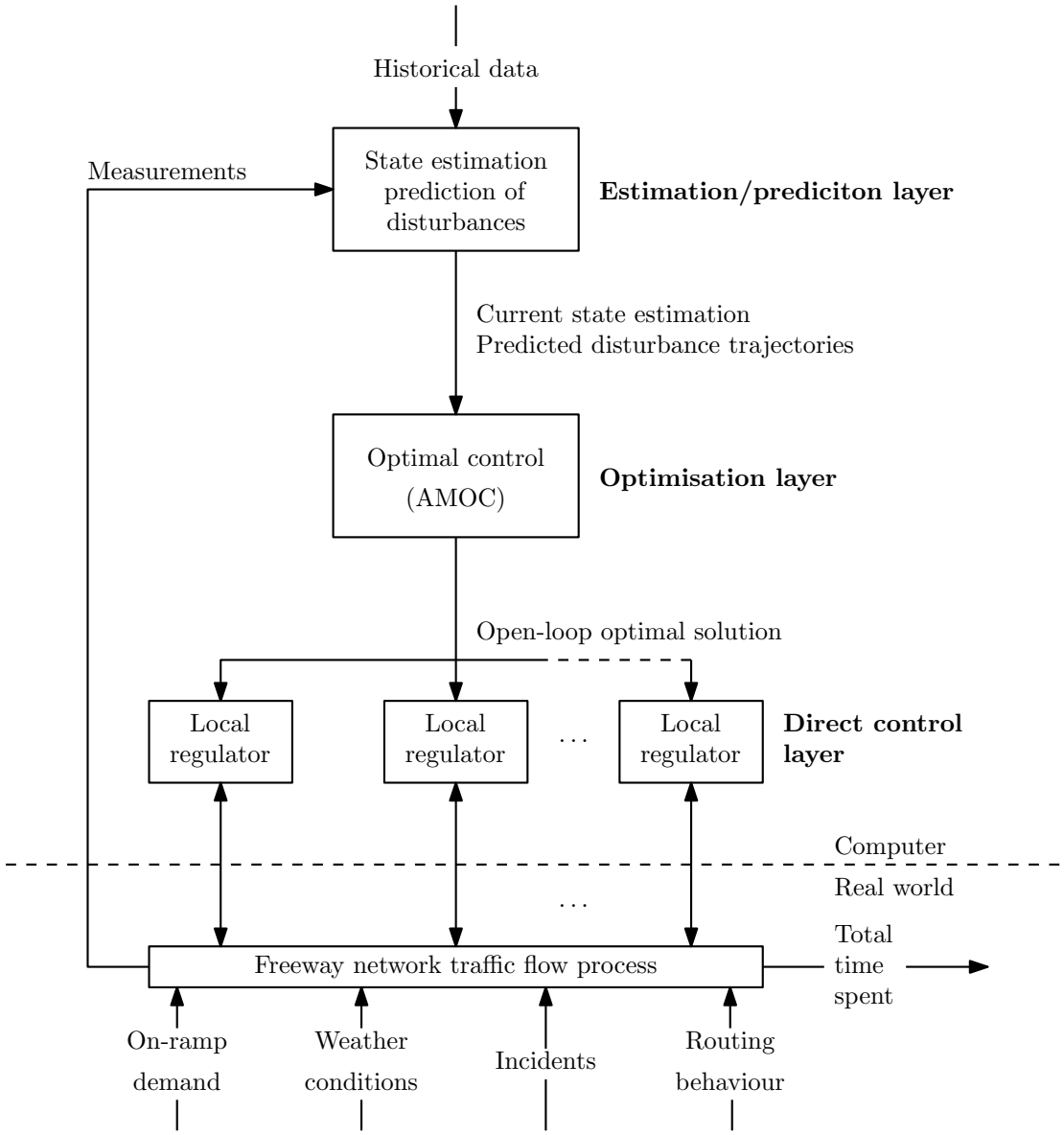


FIGURE 3.7: *A hierarchical MPC control structure with distributed controllers. Adapted from Papamichail et al. [117].*

3.2.2 Variable Speed Limits

Variable speed limits (VSLs) are another popular control measure implemented on highways in response to the prevailing weather conditions. VSL installations were first implemented in Germany during the 1980s. Today, numerous VSL installations are encountered throughout Europe and the United States of America [114]. Initially, the main goal of VSL was improved traffic safety achieved by lowering the speed limits upstream of congested areas. More recently, however, attempts have been made to increase traffic flow through the use of VSLs [52]. These are the two main approaches towards employing VSLs in the literature, the first emphasising the *homogenisation effect*, while the focus in the second approach is on *preventing traffic breakdown* by controlling the flow by means of VSLs [53].

The idea behind the homogenisation effect is that the reduced speeds due to the newly implemented speed limits result in a reduction of the differences in speed of vehicles travelling in the

same lane, as well as vehicles travelling in adjacent lanes [114]. Increased traffic flow homogenisation has a positive impact on traffic safety, and a correlation between VSLs and reduced accident probabilities has been demonstrated, with multi-year evaluations of the effect of VSLs on traffic safety showing reductions of up to 30% in accident numbers after VSL installation [23].

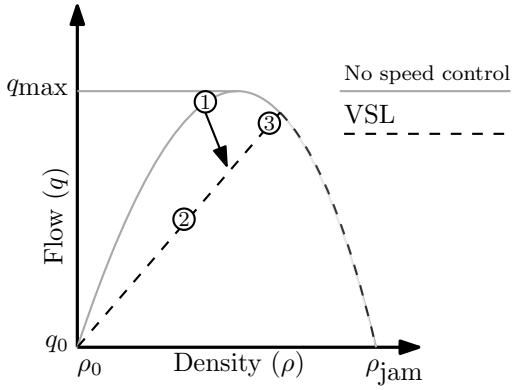


FIGURE 3.8: *The effect of VSLs on the fundamental diagram. Adapted from Hegyi et al. [53].*

The focus in the traffic breakdown prevention approach is on preventing overcritical densities. Typically, this is achieved by reducing the speed limit before a bottleneck area, or an on-ramp, thereby altering the fundamental diagram of traffic flow of that section, as shown in Figure 3.8. When traffic on the highway is in state 1, it is nearly unstable and even small disturbances or on-ramp flows may cause traffic breakdown. Adjusting the speed limit will change the state from 1 to somewhere between 2 and 3, changing the shape of the fundamental diagram from the grey line to the dashed black line. The resulting decrease in flow stabilises the traffic flow and allows more space for traffic entering the highway from the on-ramp [53]. The gradient of the straight line forming part of the altered fundamental diagram is directly proportional to the magnitude of the newly imposed speed limit. By resolving these high-density areas (bottlenecks), higher flow rates may be achieved due to the prevention of traffic breakdown [52].

One of the earliest examples of VSL control was introduced by Smulders [151]. The VSL control problem was formulated as an optimal control problem, based on a macroscopic simulation model, with the aim of finding the maximum expected time until congestion. This could be achieved by maximising the expected value of the time until congestion sets in, given by

$$V(\rho_0) = E\left[\int_0^{\tau} (\ell \rho_t u_t^i - \delta I_{i=1})\ \mathrm{d}t \mid \rho_0\right], \tag{3.23}$$

where ℓ represents the number of lanes of the highway section under consideration, ρ_t and u_t^i represent the density and velocity at time t, respectively, i is a binary variable which indicates whether VSLs are operational, δ is a control variable specific to the stretch of highway, and $I_{\{.\}}$ is an indicator function for determining whether congestion has formed on the highway stretch considered. Finally,

$$\tau = \inf\{t \geq 0 : \rho_t = \rho_{\text{jam}}\} \tag{3.24}$$

represents the time to congestion. Initially, a one-switch control policy was considered, taking the form

$$i(\rho) = I_{(\rho > \bar{\rho})},$$

which means that control is applied only for densities exceeding a pre-defined value $\bar{\rho}$. This did, however, lead to frequent switching on and off of the VSL control, and as a result, hysteresis control was introduced. In the resulting hysteresis control policy, a single variable speed sign is

employed, showing a reduced speed limit of 90 kilometres per hour, which is switched on and off at pre-defined values, as shown in Table 3.1.

TABLE 3.1: *Optimal hysteresis control policies [151].*

q_0 (veh/hour)	2 000	3 000	3 500	4 000	4 400	4 800
$\bar{\rho}_{\text{off}}$ (veh/km/lane)	56	26	2	5	8	12
$\bar{\rho}_{\text{on}}$ (veh/km/lane)	70	42	28	28	29	31

As may be seen in the table, the VSL control is switched on at specific densities, $\bar{\rho}_{\text{on}}$, based on the current traffic flow q_0, and then only switched off again once the density has reached a lower limit $\bar{\rho}_{\text{off}}$. This prevents frequent switching of the VSL control, and thus results in a more stable traffic control policy. It is also evident from the results of the table that for VSL control it is important to detect an increase in the traffic flow above 3 500 vehicles per hour, as the control policy hardly changes for these values, and the control is most effective at these high densities [151].

Another early example of VSL control is the sliding-mode approach proposed by Lenz [83], who defined a control law for adjusting the speed limit based on the current traffic density. According to this control law, the speed limit u_{limit} is adjusted according to

$$u_{\text{limit}} = \begin{cases} 120 & \text{if} \quad \rho \leq 14 \\ 100 & \text{if} \quad 14 \leq \rho \leq 17.5 \\ 80 & \text{if} \quad 17.5 \leq \rho \leq 23 \\ 60 & \text{if} \quad \rho \geq 23, \end{cases} \tag{3.25}$$

where all speeds are expressed in kilometres per hour, and all densities are expressed in vehicles per kilometre. It was found, however, that this control law led to so-called standing waves, or shock waves propagating downstream of the speed limit sign, and as a result, a predictive element was introduced by Lenz *et al.* [84], where the density measure ρ in (3.25) is replaced by $\bar{\rho} = \psi\rho_i + (1-\psi)\rho_{i+1}$ in order to take into account, as a predictive measure, the density of downstream traffic when adjusting the speed limit in order to prevent standing waves from forming.

Alessandri *et al.* [1, 2] introduced a nonlinear optimisation model based on a macroscopic traffic flow model. The optimal control problem formulated by Alessandri *et al.* [2] involves a stretch of highway partitioned into $K+1$ sections. For the macroscopic model, the state vector is defined as

$$\boldsymbol{x}_t = [\rho_0(t), \rho_1(t), \ldots, \rho_K(t), u_0(t), u_1(t), \ldots, u_K(t)], \tag{3.26}$$

where $\rho_i(t)$ and $u_i(t)$ represent the traffic density and average traffic velocity in section $i \in \{1, \ldots, K\}$ during time interval t, respectively. The performance measurement vector is given by

$$\boldsymbol{y}_t = [q_0(t), q_1(t), \ldots, q_K(t), w_0(t), w_1(t), \ldots, w_K(t)], \tag{3.27}$$

where $q_i(t)$ and $w_i(t)$ represent the exit flow from section i to section $i+1$, and the harmonic mean speed of vehicles coming from section i and moving into section $i+1$ during the time interval t, respectively. Furthermore, the vectors

$$\boldsymbol{r}_t = [r_0(t), r_1(t), \ldots, r_K(t)] \tag{3.28}$$

and

$$\boldsymbol{s}_t = [s_0(t), s_1(t), \ldots, s_K(t)] \tag{3.29}$$

represent the ramp in- and outflow values for section $i \in \{1, \ldots, K\}$ during time interval t. These vectors are updated at each model time interval t, based on the output of the underlying macroscopic traffic model. Finally, the control vector

$$\boldsymbol{c}_t = [b_0(t), b_1(t), \ldots, b_K(t)] \tag{3.30}$$

captures the speed limit control commands for time interval t, where a value of $0.5 \leq b_i < 1$ indicates various levels of speed restrictions being applied at section $i \in \{1, \ldots, K\}$. Finally, the objective function

$$J = \sum_{t=0}^{T} \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{c}_t) \tag{3.31}$$

is to be minimised, where $\boldsymbol{g}$ is a function of penalising arguments based on the state vector $\boldsymbol{x}_t$ and the control vector $\boldsymbol{c}_t$, respectively. This control vector also takes the form of hysteresis control, similar to that employed by Smulders [151]. This optimal control problem was solved using Powell's method for minimising an objective function approximately, and the results were implemented in the macroscopic simulation environment [2].

Kang *et al.* [67] employed a linearised traffic model, based on a linear speed-density relationship, such as the one shown in Figure 3.1, for example, in order to determine optimal VSLs for work zone operations on a highway adopting an MPC approach. This linear relationship is updated continually using real-time data from the microscopic simulation environment. The linearisation of the speed-density relationship allows the optimal control problem to be formulated as a *linear programming* (LP) problem in terms of macroscopic traffic flow variables. This LP problem is then reformulated at each MPC control time step using the latest traffic information from the simulation environment, and was solved using Lindo© [88], after which the new VSLs were applied in the simulation environment [67]. Another application of VSLs in an MPC context was demonstrated by Hegyi *et al.* [53], who extended their macroscopic traffic model to include control structures for both RM, as explained in §3.2.1, and VSLs in an integrated control approach. A second attempt at integrating the RM and VSL control approaches was demonstrated by Carlson *et al.* [23], who extended the AMOC strategy for finding optimal fixed-time RM strategies, introduced in §3.2.1, to include VSL control as well. Again the formulation entails minimising a nonlinear objective function, based on the underlying macroscopic traffic model, built in the METANET modelling environment.

In an attempt at simplifying the VSL control problem, Carlson *et al.* [25] proposed a feedback controller which takes as input real-time traffic flow and density measurements in order to calculate, in real time and within a closed loop, appropriate speed limits so as to maintain a stable traffic flow which is close to a pre-specified reference value, with the aim of achieving maximal throughput for any appearing demand. The control system developed takes the form of a cascade controller comprising two nested control loops, as may be seen from the controller structure shown in Figure 3.9.

The secondary controller is employed to adjust the VSL rate b which determines the outflow q_c, which is, in turn, compared to the reference value for the outflow $\hat{q}_c$. The outflow q_c is measured immediately downstream of the application area. The aim of the primary control loop is to control the measured density ρ_{out} with respect to the user specified reference density $\hat{\rho}_{\text{out}}$ which, should be set to the critical density for the highway stretch under consideration in order to maximise the throughput. As may be seen in Figure 3.9, the secondary controller is designed as an *integral* (I) controller, whose transfer function in the time domain is given by

$$b(t) = b(t-1) + K_I e_q(t),$$

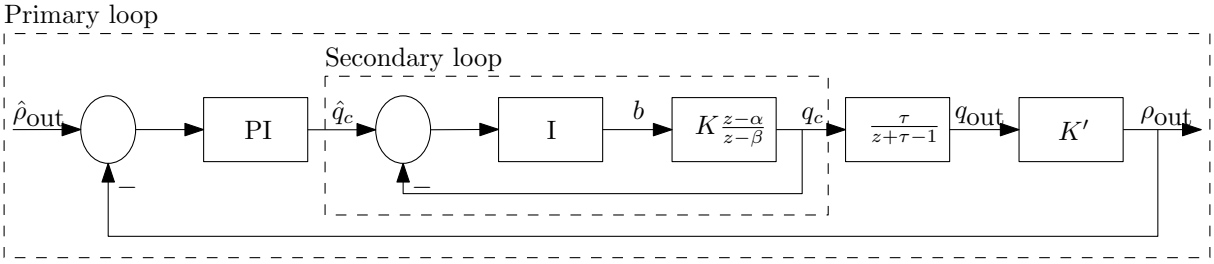


FIGURE 3.9: *An MTFC feedback cascade controller structure using VSLs as actuator. Adapted from Carlson et al. [25].*

where K_I is the integral gain of the controller and $e_q(t) = \hat{q}_c - q_c$ is the flow control error. In the primary loop, a *proportional-integral* (PI) controller is employed. The transfer function in the time domain for this PI controller is given by

$$\hat{q}_c(t) = \hat{q}_c(t-1) + (K'_P + K'_I)e_\rho(t) - K'_P e_\rho(t-1),$$

where K'_I and K'_P represent the integral and proportional gains of the controller, respectively, and $e_\rho(t) = \hat{\rho}_{\text{out}} - \rho_{\text{out}}(t)$ is the density control error [25]. The relationship between the difference in the applied speed limit and the resulting difference in flow is modelled as a linear discrete-time transfer function given in the frequency domain by

$$\frac{\triangle q_c(z)}{\triangle b(z)} = K\frac{z-\alpha}{z-\beta},$$

where α, β and K are model parameters which have to be tuned appropriately. In the time domain, this transform yields the difference equation

$$\triangle q_c(t+1) - \beta \triangleq_c (t) = K\triangle b(t+1) - \alpha\triangle b(t).$$

In the absence of congestion, the transform from the flow at the application area q_c to the flow at the bottleneck q_{out} is modelled as a first-order system with a time delay, given in the frequency domain by

$$\frac{\triangle q_{\text{out}}}{\triangle q_c} = \frac{\tau}{z+\tau-1},$$

where τ is again a model parameter. Finally, the transform of the bottleneck flow q_{out} to the bottleneck density ρ_{out} is enabled by a linearisation of the fundamental flow-density relationship, as shown in the fundamental diagram, around the critical density, and thus may be achieved simply through a proportional gain, given by K' [25]. This controller was implemented, and the parameters tuned within a METANET macroscopic simulation model.

Another, and arguably simpler feedback-based VSL controller was designed by Müller *et al.* [105]. The controller takes a form very similar to ALINEA, as it is also an integral controller. Therefore, there is only a single controller parameter which requires empirical parameter tuning. In this controller, the speed limit is adjusted according to a VSL metering rate $b \in [0.2, 0.8]$ which is calculated as

$$b(t) = b(t-1) + K_I[\hat{\rho} - \rho_{\text{out}}], \tag{3.32}$$

where K_I is the integral gain of the controller, $\hat{\rho}$ denotes the target density at the bottleneck that the controller aims to maintain, and ρ_{out} denotes the measured density at the bottleneck location during control interval t. The metering rate is then rounded to the nearest tenth and the VSL to be applied is determined by

$$\text{VSL} = 20 + 100b, \tag{3.33}$$

which resulted in the set of speed limits VSL $\in \{20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120\}$ in the original implementation. This controller was evaluated within the context of a simple highway network consisting of a dual carriageway and a single on-ramp joining the highway. The network and the VSL controller were implemented within the Aimsun microscopic traffic simulation environment.

3.2.3 Lane Assignment

Lane assignment (LA) is a control approach proposed for implementation on *automated highway systems* (AHSs). McMillin and Sanford [98] defined an AHS as a highway system in which some of the human functions in the driving process are supported and replaced by various technological systems. As defined by Ramaswamy *et al.* [126], LAs represent the scheduling of the path followed by a vehicle once it enters an automated multilane corridor. LA may be employed with two end goals in mind: Optimal routing of vehicles in order to reach their destination and traffic flow optimisation through increased lane utilisation, thereby increasing highway capacity and reducing the total travel time of vehicles [48].

One of the earliest formulations of the LA problem is due to Hall and Lotspeich [49] who employed an LP approach to LA with the goal of maximising highway flow. Their formulation is based on a network flow highway model, with segments being defined by type (on-ramp, off-ramp, or neither), length, capacity and number of lanes. Given the macroscopic nature of the underlying model, lane assignments are represented by lane flows, and as a result, the decision variables of the LP formulation are the flows entering a lane, the flows exiting a lane, the flows passing through a lane, and the flow of vehicles which remain within a lane in each segment [49]. The goal then is to find the flows which maximise the total flow

$$\sum_{i,j,k} F_{ijk}(t)$$

from each origin i to each destination j over each segment k for each time period t, subject to five constraint types, namely (1) flow conservation constraints, (2) lane and segment capacity constraints, (3) on-/off-ramp capacity constraints, (4) non-negativity constraints, and (5) a proportionally defined origin-destination constraint ensuring that the origin-destination flow demands are met [49]. In this formulation, lane and segment capacity is defined in terms of workload, where constant workload values are assigned to each of the various allowable vehicle manoeuvres (*i.e.* entering a lane, exiting a lane, passing through a lane, and remaining within a lane). These workload values are then multiplied by the corresponding flows to determine the total workload for each lane and segment, which may not exceed a predetermined maximum allowable workload. The resulting linear program was subsequently solved using the CPLEX LP Solver [62].

Ramaswamy *et al.* [126] employed a similar LP approach for solving the LA problem. In their formulation the aim was, however, not to maximise flow, but rather to minimise the total travel time of vehicles in the system, where the total travel time is the sum of the time spent travelling at the reference velocity T_{ss} and the time required for manoeuvring between lanes $T_{\text{manoeuvre}}$. The travel time at the reference velocity T_{ss} is given by

$$\sum_{m=1}^{n} \sum_{i=1}^{K_1} \sum_{j=i+1}^{K_2} \frac{\ell_{i,j}}{u_m} \rho_{i,j}^{m},$$

where n represents the number of lanes, K_1 and K_2 represent the number of highway entrance and exit ramps in the segment under consideration, respectively, $\ell_{i,j}$ denotes the length of the

segment connecting nodes i and j, u_m denotes the reference velocity of vehicles travelling between nodes i and j, and $\rho_{i,j}^m$ represents the number of vehicles travelling in lane m from node i to node j, while the time required to perform the necessary manoeuvres $T_{\text{manoeuvre}}$ is given by

$$\sum_{m=1}^{n} \sum_{i=1}^{K_1} \sum_{j=i+1}^{K_2} \gamma_{i,j}^m \rho_{i,j}^m,$$

where $\gamma_{i,j}^m$ is a manoeuvre time cost associated with the move to lane m while travelling from node i to node j. The resulting objective is then to

$$\text{minimise} \sum_{m=1}^{n} \sum_{i=1}^{K_1} \sum_{j=i+1}^{K_2} \left(\frac{\ell_{i,j}}{u_m} + \gamma_{i,j}^m \right) \rho_{i,j}^m$$

subject to flow conservation as well as capacity and non-negativity constraints [126]. This LP approach is, however, restricted to use in light traffic conditions only, where the manoeuvre time between lanes may be assumed to be constant when effective vehicle-to-vehicle communication is in place. In order to extend the approach to situations involving more severe traffic demand, resulting in increased vehicle densities, the manoeuvre time is taken to be a function of the respective densities of the two lanes affected by the manoeuvre. As a result, the objective function is no longer linear, and a quadratic programming solution approach is required in order to solve the LA problem [126].

Kim *et al.* [72] employed the same approach as Ramaswamy *et al.* [126] of minimising the total travel time as the sum of the manoeuvring time and the time spent travelling at the reference speed in their formulation of the LA problem in the context of an AHS. The definition of the time spent at the reference speed, as well as the flow conservation, capacity and non-negativity constraints, remains unchanged in their formulation, but the manoeuvring time is adapted to reflect the individual vehicle dynamics, taking into account the time loss due to deceleration and acceleration of both the vehicle changing lanes and the vehicle already in the target lane. The velocity profile of the vehicle already present in the target lane is shown in Figure 3.10.

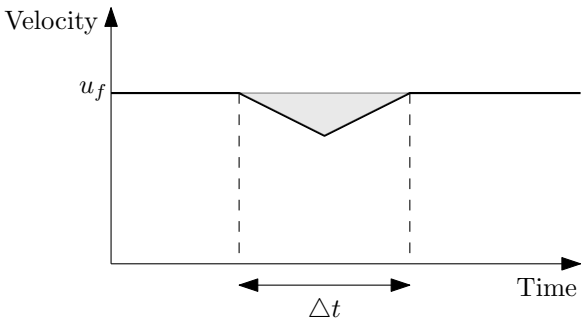


FIGURE 3.10: *The velocity profile when a vehicle makes space for lane changing. Adapted from Kim et al. [72].*

As may be seen in the figure, the vehicle initially decelerates in order to create space for the lane change, and then accelerates again to reach the free-flow traffic speed. The time loss of the manoeuvre is given by the shaded region in Figure 3.10. The time loss experienced by the vehicle changing lanes follows a similar velocity profile, initially decelerating, waiting for the space to open up, then performing the lane change, and finally accelerating to the free-flow speed in the target lane. Due to the increased complexity of incorporating these time costs in the resulting optimisation problem, Kim *et al.* [72] employed a genetic algorithm as an approximate solution approach.

3.3 Highway Control in the Presence of Autonomous Vehicles

As may have been expected, a substantial amount of research has been performed on the effect that autonomous vehicles are expected to have on traffic flow . Perhaps the earliest examples of such studies are due to Varaiya [164] and Rao and Varaiya [127], who investigated the impact that vehicles equipped with autonomous intelligent cruise control would have on traffic flow. In these early studies, mathematical models were formulated in order to assess the effect of vehicle platoons (*i.e.* vehicles following each other at relatively close headways) on traffic flow, enabled by automatic adaptive cruise control. In order to estimate highway traffic flow in the presence of vehicles with automatic adaptive cruise control, Rao and Varaiya [127] proposed the relationship

$$q = \frac{3600v\bar{N}}{\bar{N}(\ell + d) - d + \bar{\triangle}}, \tag{3.34}$$

where v denotes the vehicle speed in m/s, ℓ denotes the average individual vehicle length, d denotes the average headway between vehicles within a platoon, $\bar{\triangle}$ denotes the average distance between platoons, and $\bar{N}$ is the average size of a platoon. In this model formulation, the average platoon size was determined according to a pre-specified platoon size distribution whereby vehicles would join a platoon based on a certain probability when they would find themselves within a specific distance from the platoon.

In another paper investigating the effects of adaptive cruise control on traffic flow along highways, Van Arem *et al.* [163] extended the approach of Rao and Varaiya [127] by including vehicle-to-vehicle communication in their modelling approach. Their approach was evaluated within the so-called MIXIC link-level mesoscopic traffic simulation modelling environment. In their simulation model, the vehicle following rules were adapted so as to model the effect that vehicles equipped with connected adaptive cruise control have on the highway throughput. Due to the fact that vehicles were now able to communicate, the reaction times were reduced from 1.4 seconds for human drivers to 0.5 seconds for connected vehicles, resulting in a reduction in the minimum allowable headway between connected vehicles.

In another simulation study into the effects of adaptive cruise control on highway traffic flow, Kesting *et al.* [69] designed an adaptive cruise control controller that would alter the cruise control behaviour of a vehicle, based on the prevailing traffic conditions. A graphical representation of the controller is shown in Figure 3.11. The effectiveness of this controller in terms of altering the vehicle behaviour according to the prevailing traffic conditions was evaluated in the context of a microscopic traffic simulation model of a multi-lane highway exhibiting a single bottleneck at an on-ramp merge.

The influence of autonomous or connected vehicles on traffic flow stability and throughput was the focus of a study performed by Talebpour and Mahmassani [156]. They derived three different vehicle behaviour models in respect of acceleration and deceleration, according to different levels of vehicle connectivity capabilities. The first of these vehicle classes encompasses all vehicles that have no communication capabilities, as most of the vehicles currently on the roads. In the second class, vehicles that are communication-ready were modelled. This class encompasses all vehicles which are equipped with the necessary infrastructure for vehicle-to-vehicle and vehicle-to-infrastructure communication, although connectivity between vehicles and/or infrastructure cannot be guaranteed. Accordingly, four scenarios could be defined within this class: Active/inactive vehicle-to-vehicle communication and active/inactive vehicle-to-infrastructure communication. If both types of communication are inactive, the vehicle behaviour is the same as that of vehicles in class one, while if vehicle-to-vehicle communication is active, the car-following rules are updated, allowing for smaller headways between two successive vehicles. Vehicle-to-

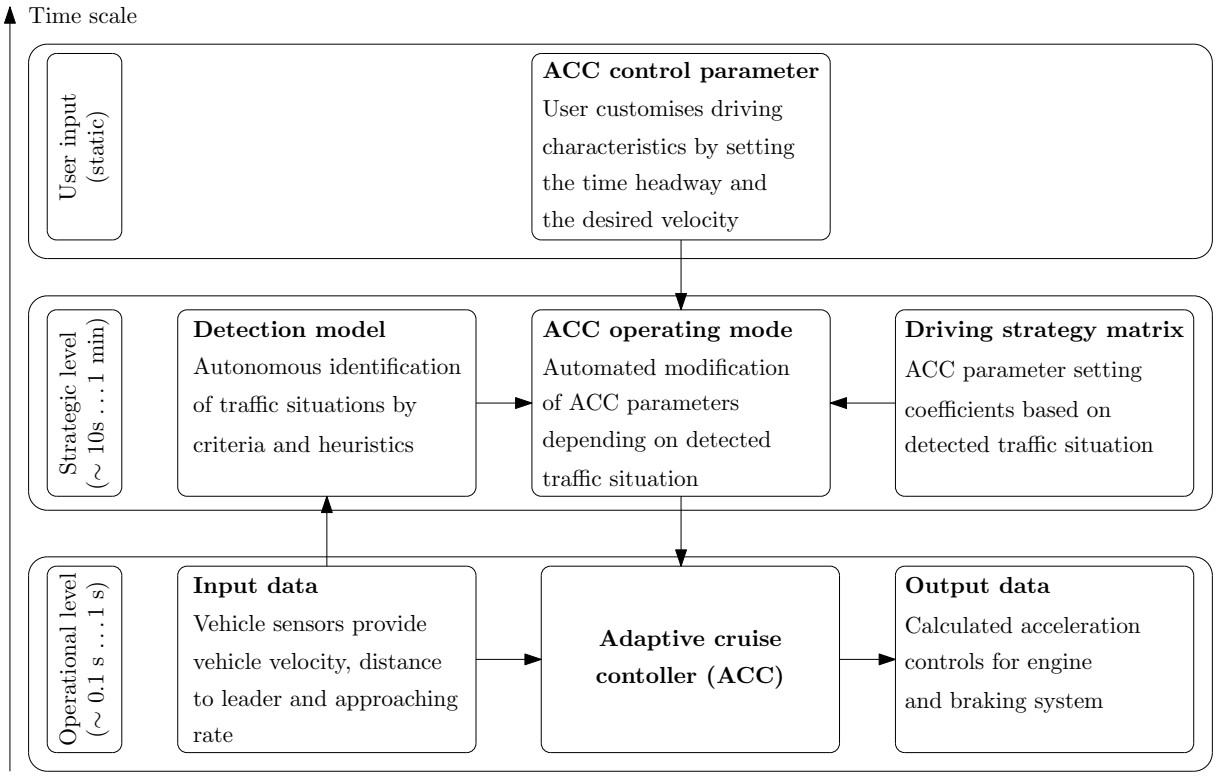


FIGURE 3.11: *An adaptive cruise control controller architecture. Adapted from Kesting et al. [69].*

infrastructure communication allows drivers to receive information from *traffic management centres* (TMCs), such as real-time information on VSLs, route guidance or congestion warnings. The vehicle behaviour is, however, not adjusted when vehicle-to-infrastructure communication is active [156]. Finally, the third class of vehicles are fully autonomous vehicles. Reaction times for autonomous vehicles were assumed to be instantaneous, and minimum following distances were subsequently determined as a function of the vehicle deceleration capabilities and speed at which the vehicle is travelling. The effect that various percentages of either connected or autonomous vehicles have on the resulting traffic flow was subsequently evaluated in the context of a microscopic traffic simulation model of a simple hypothetical highway network consisting of a 3.5 mile segment with two lanes in the forward direction, and a single lane on-ramp which merges with the highway at the 1.75 mile mark.

An early attempt at controlling highway traffic flow by means of providing direct instructions to autonomous vehicles is due to Baskar *et al.* [11]. In this implementation, the traffic flow was assumed to consist only of autonomous vehicles, which may receive direct instructions from the roadside TMC. Another assumption was that all vehicles travel in platoons, and that, as such, the dynamics of all the vehicles within a platoon could be described by the lead vehicle of that platoon. Any action carried out by the leader of the platoon would thus also be performed by all of the follower vehicles in the platoon. These actions involved the speed at which the platoon should travel, the lane in which the platoon should travel, and the time at which a platoon should enter the highway from an on-ramp. An MPC approach was adopted with the aim of determining these actions for each of the platoons present within the simulated environment, such that the total time spent in the system by all vehicles would be minimised. The objective

of the MPC approach was then to minimise

$$J_{\mathrm{TTS}} = \sum_{N_{\mathrm{sim}}}^{\ell=0} \left(n(\ell) + q_m(\ell) + q_0(\ell)\right) T_{\mathrm{TTS}}, \tag{3.35}$$

where J_{TTS} denotes the total time spent by all vehicles in the system over the course of the entire simulation period, $n(\ell)$ denotes the number of vehicles present in the simulation model at the start of control interval ℓ, $q_m(\ell)$ denotes the number of vehicles entering the simulated area along the mainline during the control interval ℓ and $q_o(\ell)$ denotes the number of vehicles entering the simulated area from an on-ramp during the control interval ℓ. Here $q_m(\ell)$ may be negative in the case where there are more vehicles leaving the simulated area than there are vehicles entering the simulated area from the mainline during control interval ℓ. The constraints incorporated in their formulation ensure that sufficient space is available for an entire platoon to change lanes, and that sufficient space is available for an entire platoon to enter the highway traffic flow from an on-ramp. This formulation was implemented in a MITSIM [177] inspired traffic simulation model within MATLAB. The MPC optimisation problem was solved using the pattern search method [7] available within the `patternsearch` command incorporated in the Genetic Algorithm and Direct Search Toolbox in MATLAB.

Although the presence of autonomous vehicles was not assumed, Schakel and Van Arem employed vehicle-to-infrastructure communication in order to develop an in-car advice algorithm, based on which drivers of "connected" vehicles would receive specific advice regarding lane choice, speed and following distance [141]. The architecture of the system is illustrated in Figure 3.12.

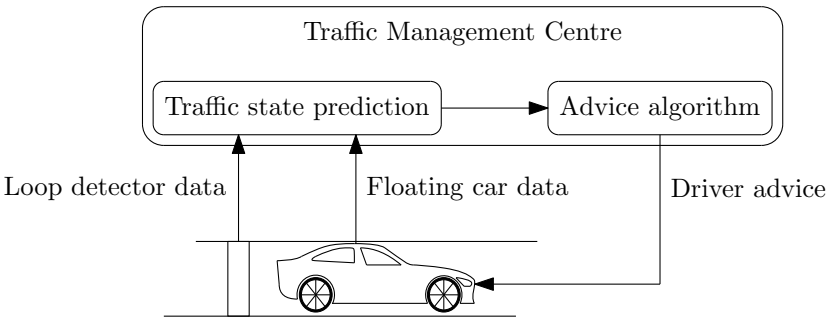


FIGURE 3.12: *Overview of an in-car advisory system. Adapted from Schakel and van Arem [141].*

As may be seen in the figure, a combination of loop detector and floating vehicle data are employed to generate an estimation of the current traffic flow, as well as a prediction of the traffic flow that a connected vehicle is likely to encounter downstream of its current location. This traffic state prediction is the input to the advice algorithm. The algorithm is based on two principles of traffic flow. The first is the so-called *acceleration advice principle*. According to this principle, drivers are encouraged to maintain short, but safe, headways at the end of a congested traffic zone. The basis for this advice is the principle that the capacity drop due to congestion is mainly due to the fact that drivers only accelerate once the actual headway is larger than the desired headway [141]. Thus, by maintaining a shorter headway, the effects of the capacity drop may be reduced. The second component of the advice algorithm is the so-called *distribution advice principle*. It has been shown that drivers do not fully utilise all highway lanes once traffic breakdown occurs [73]. The aim of the distribution advice principle then, was to redistribute traffic flow more evenly across the lanes, thereby utilising the available highway capacity more effectively [141]. Based on these two principles, drivers were given three distinct types of advice, namely *speed advice* (*i.e.* drivers are advised on the speed at which they should be travelling), *headway advice* (*i.e.* drivers are advised on the following distance they should maintain), and

lane advice (*i.e.* drivers are advised on the lane in which they should be travelling). In order to model varying levels of driver compliance, a variable $\omega \in [0, 1]$ was introduced, where a value of 0 denotes no compliance, while a value of 1 denotes full compliance. This system was implemented and evaluated within the context of a microscopic traffic simulation model.

In an attempt at coordinating the traffic control measures described in §3.2.1–§3.2.3 in the presence of autonomous and connected vehicles, Roncoli *et al.* [136] formulated an optimal control problem with the aim of providing optimal metering rates, VSLs and lane change advice. The basis of this optimal control approach was a macroscopic lane-based *cellular transition model* (CTM) [135]. In this formulation, RM is applied in the conventional way, by directly regulating the inflow of traffic onto a highway from an on-ramp by means of a traffic light. Thus, no autonomous capabilities, vehicle-to-infrastructure or vehicle-to-vehicle communication are required or assumed for the RM implementation. For VSL control, sufficient penetration of autonomous or connected vehicles is assumed, where *sufficient* is defined as the number of vehicles required to enforce a speed limit on all vehicles, even if this speed limit is only assigned to a limited number of vehicles [136]. It is therefore assumed that the new VSL is imposed on the entire traffic flow at a specific link during every control time step. For the LA component of the optimisation problem, an intermediate algorithm is employed. This algorithm receives as input the optimal lateral flows between lanes, together with the probability for random lane changes by human-driven vehicles and the probability that a vehicle will exit the highway system at an off-ramp. The intermediate algorithm then determines an appropriate number of autonomous vehicles which should receive a lane-change command such that the optimal lateral flows may be achieved [136]. The aim of the optimal control approach is to determine the optimal metering rate for every on-ramp within the study area, the optimal VSL to be applied at each highway segment and the optimal lateral traffic flows of every highway segment, all based on a piecewise-linear fundamental diagram of traffic flow. Furthermore, the optimal control interval lengths for each of these control measures were also determined by the optimisation model. The objective function to be minimised comprised seven terms, three of which were linear, while the four remaining terms were quadratic. The first, and most important, of these terms represented the total time spent in the system by all vehicles during the entire control period. The second and third terms were penalty terms, introduced to avoid the build-up of impractically long on-ramp queues, and impractically large numbers of lane change manoeuvres, respectively. Finally, the four quadratic terms were introduced to either penalise variation in control variables from one time step to the next, or from one segment to the adjacent downstream segment [136]. The underlying CTM and the optimal control approach were implemented in MATLAB, while the Gurobi optimisation solver was employed for solving the quadratic programming problem.

The above-mentioned optimal control approach by Roncoli *et al.* [135, 136] was refined and implemented by Roncoli *et al.*[137] in the context of a hierarchical MPC control structure with the aim of enabling the optimal control problem to be solved in an online manner. A graphical illustration of the structure of this hierarchical control structure may be seen in Figure 3.13. The purpose of the adaptation and prediction layer is to process the data obtained from roadside traffic sensors, as well as the data collected from autonomous vehicles, and subsequently generate a traffic demand forecast for the duration of the following control interval. This traffic state estimation, as well as the predicted traffic demand, is then provided to the optimisation layer. In the optimisation layer, the optimal control problem outlined above, as formalised by Roncoli *et al.* [136], is solved. Due to the fact that the optimal control problem is solved in the context of a link-based macroscopic traffic simulation model, while the hierarchical MPC approach was implemented within a microscopic traffic simulation model, a local control layer was introduced. The function of the local control layer is to translate the optimal macroscopic densities, as determined in the optimisation layer, to physical speed limit values and red phase times for the

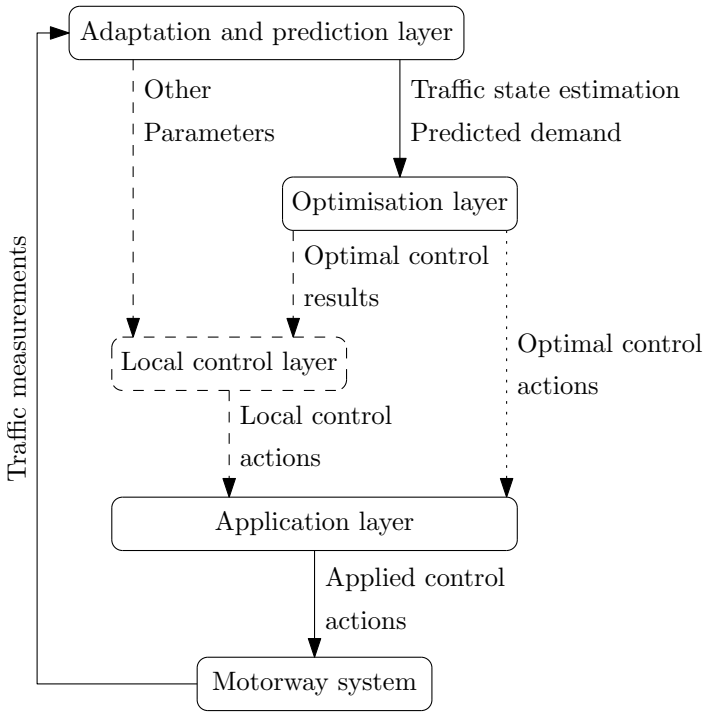


FIGURE 3.13: *A hierarchical MPC control structure with distributed controllers in the presence of autonomous vehicles. Adapted from Roncoli* et al. *[137].*

MTFC and RM components respectively. This is achieved by employing the target densities suggested by the optimal control layer as set points for several local feedback controllers, which then determine suitable red phase times and VSLs in order to achieve these set target densities [137]. I-type feedback controllers, such as those employed in ALINEA and the MTFC controller by Müller *et al.* [105], are employed for this purpose in the hierarchical MPC control structure. The VSLs to be applied are then determined as

$$v_{i,j}(t) = v_{i,j}(t-1) + K_v \left[\hat{\rho}_{i+1,j}(t) - \rho_{i+1,j}(t)\right], \tag{3.36}$$

where $v_{i,j}$ denotes the speed limit applied at lane j of segment i, $\hat{\rho}_{i,j}$ denotes the target density set point for lane j of segment i, $\rho_{i,j}$ denotes the measured density at lane j of segment i, and K_v denotes the integral gain of the controller. The same controller structure is employed for the RM component as the optimal control layer specifies target densities for the merge sections, which the controller aims to achieve. Finally, the application layer serves the purpose of applying the lane changing actions suggested by the optimal control layer, as well as the VSL and RM actions suggested by the feedback controllers within the simulation model. As stated above, this hierarchical MPC approach was implemented in the context of a simplified microscopic highway network comprising a 5 km stretch of a three-lane highway with a single on-ramp at 3.5 km, implemented within the AIMSUN microscopic traffic simulation software.

Perraki *et al.* [120] applied the hierarchical MPC framework of Roncoli *et al.* [137] in the context of a real-world case study of the A20 highway connecting Rotterdam and Gouda in the Netherlands. This case study model was also implemented within the AIMSUN microscopic traffic simulation software. While an improvement of 17.7% in respect of the total time spent in the system by all vehicles was recorded, several shortcomings were also pointed out. In this implementation, the control actions were carried out under the assumption that all vehicles are equipped with vehicle automation and communication systems, while different vehicle types (such as passenger vehicles, delivery vehicles or trucks) were not considered. It is expected that

modelling mixed traffic flows consisting of both autonomous and human-driven vehicles may have the largest impact on the results obtained, particularly on the manner in which the VSLs are applied.

Apart from the heuristic, optimal and feedback control approaches outlined above in the context of autonomous or connected vehicles, traffic state estimation, as employed in the forecasting component of the hierarchical MPC framework of Roncoli *et al.* [137], is another area of research within the broader field of autonomous driving which has received a significant amount of attention [34]. This is demonstrated in the work of Rempe *et al.* [129], Bekiaris-Liberis *et al.* [14], Fountoulakis *et al.* [38] and Roncoli *et al.* [134]. The aim in all of these studies was to generalise information obtained from individual vehicles (regarding their immediate traffic surroundings) in order to form a reliable picture of the state of traffic flow in general. Typically, the aim then was to employ this new-found, real-time traffic information in order to be able to better control traffic flow and even prevent congestion, as illustrated by the fact that most of the control approaches presented above make use of traffic flow predictions in one way or another.

3.4 Machine Learning in Highway Traffic Control

Machine learning has been applied to various traffic control problems in the literature. Neural networks have, for example, been employed by Spall and Chin [152] in order to determine optimal traffic signal timings for networks with fixed demand profiles. Kwon and Stephanedes [79] also employed neural networks in order to predict the exit demand profiles for highways in order to improve highway control. Reinforcement learning has furthermore been employed multiple times in order to solve the control problem of finding optimal signal timings for urban traffic networks, as demonstrated by Kuyer *et al.* [78], as well as by Khamis and Gomaa [71, 70].

3.4.1 Reinforcement Learning for Ramp Metering

One of the first applications of reinforcement learning to the RM problem was introduced by Wen *et al.* [173], who implemented Q-learning (described in §2.2.3) for RM control in a simple METANET macroscopic simulation model. In their implementation, the state space consists of three variables, namely the average speed of vehicles directly downstream of the diversion point, the density directly downstream of the diversion point, and the current metering rate. The action space comprises five different adjustments that may be performed on the current metering rate, which are chosen using an alteration of the softmax action selection procedure (described in §2.2.1). Finally, the reward function is based on the total time spent in the system by vehicles [173].

Another early reinforcement learning approach to RM is due to Davarynejad *et al.* [29], who implemented Q-learning within a discretised state space in the context of a METANET macroscopic simulation environment in order to perform RM with a queueing consideration. The simulation model comprised a six-kilometre stretch of highway with a single on-ramp located at the four-kilometre mark. Five possible system states were incorporated. The first state is the density directly downstream of the diversion point ρ, which is normalised with respect to the traffic jam density ρ_{jam} and discretised into n_ρ equi-spaced grid points. The second state is the on-ramp queue length, which is normalised with respect to a maximum allowable queue length $w_{\max}$ and discretised into n_w elements. The third state is the on-ramp demand d, which is normalised with respect to the on-ramp capacity D and then discretised into n_d elements. The fourth state is a one-time-step prediction of the on-ramp demand d_{+1} which may take one

of three values, based on the current on-ramp demand. Its value is either equal to the current on-ramp demand, or is one step up or down from the current demand. The final state is the metering rate r, which is discretised into n_r equal parts, ranging from a lower bound r_ℓ to an upper bound r_u. An incremental action space is employed, resulting in three possible actions, $\triangle r \in \{r_-, r_0, r_+\}$, where r_- represents a single-step decrease in the metering rate, r_0 means that the metering rate remains unchanged, and r_+ implies a one-step increase of the metering rate [29]. Two learning agents were employed, one to control the metering rate and the other to control the on-ramp queue length. The reward function of the agent controlling the metering rate is a direct function of the traffic outflow after the on-ramp, whereas the reward for the agent controlling the on-ramp queue length is given by

$$R = \begin{cases} \frac{1}{\ln||1-w||} & \text{if } 0.01 \leq w \leq 1.99 \\ -100 & \text{otherwise,} \end{cases}$$

with the reward set to 1 when $0.975 \leq w \leq 1.025$ in order to smooth the response of the learning agent. The maximum metering rate, as determined by the two agents, is applied at the on-ramp [29].

Fares and Gomaa [32] presented an alternative Q-learning formulation to that of Davarynejad *et al.* [29], based on the same METANET simulation model. In their formulation, the state space is described by three variables, namely the number of vehicles in the mainstream N, the number of vehicles that entered the mainstream from the on-ramp during the previous time interval $\triangle N$, and the on-ramp traffic signal during the previous time step. In order to control the density on the highway, the phase of the traffic signal at the on-ramp is adjusted. Thus, the action space simply consists of two actions only — a red and a green phase. In order to employ density control, reward is defined in terms of a deviation from the critical density ρ_{crit} at which flow is maximised as

$$R = \frac{1}{|\ \rho - \rho_{\text{crit}}\ |},$$

where ρ represents the average density downstream of the off-ramp during the current time period. This accumulated reward is to be minimised by the Q-learning agent.

Another application of reinforcement learning to RM was proposed by Rezaee *et al.* [130, 131, 132] who applied the kNN-TD reinforcement learning algorithm (described in §2.2.3) to a stretch of highway in Toronto, Canada. In order to reproduce real-life complexities of traffic flow, a microscopic simulation model was developed using Paramics© [124]. The state space was again defined using three variables, namely the downstream density ρ_{ds}, with centres placed at $\{12, 16, 19, 22, 25, 28, 33, 40, 50, 60\}$, the upstream density ρ_{us}, with centres placed at $\{12, 16, 20, 24, 28, 40\}$, and the on-ramp density ρ_{or}, with centres at $\{3, 5, 7, 9, 11, 20, 40, 60\}$. The aim of the learning agent was to minimise the total travel time

$$TTT = T_c \sum_{i=o}^{\infty} N(i),$$

where $N(i)$ represents the average number of vehicles present in the control area, confined by upstream, downstream and on-ramp detectors. In order to minimise TTT, the number of vehicles present in the control area has to be minimised. As a result, the reward function is defined as

$$R(i) = -N(i) = T_c \sum_{t=0}^{i-1} [d(t) - s(t)],$$

where $d(t)$ and $s(t)$ represent the entrance and exit rates to and from the control area during the time interval t, respectively. In order to increase the rate of convergence, Rezaee *et al.* [132]

employed a variable learning rate, based on the number of visits to each centre-action pair. The learning rate is then calculated as

$$\alpha_{n^i(x,a)} = \left[\frac{1}{1 + C(x,a)(1-\gamma)}\right]^{0.7} \quad \text{for all } i = C(x,a),$$

where $n^i(x,a)$ is the index of the i-th time that action a is attempted in centre x, and γ is the discount rate, as defined in §2.2.2. In order to find the right balance between exploration and exploitation, the ϵ-greedy method (described in §2.2.1) with an adaptive ϵ-value was employed. As with the learning rate, the value of ϵ depends on the number of visits to the centre-action pairs. The estimated number of visits to a certain state is determined in a fashion similar to the calculation of the $\mathcal{Q}$-values in the kNN-TD algorithm. The estimated number of visits to a state is thus given by

$$C^{kNN}(s,a) = \sum_{i \in kNN} p_i C(x,a),$$

where p_i is the weighted probability in (2.25) and $C(x,a)$ represents the number of visits to the centre-action pair (x,a). The state-dependent value of ϵ may then be calculated as

$$\epsilon(s) = \max\left\{0.1, \left(\frac{1}{1 + \frac{1}{11} \cdot \frac{1}{N_a(s)} \cdot \sum_a C^{kNN}(s,a)}\right)\right\},$$

where $N_a(s)$ is the number of possible actions available when the system is in state s. Thus, ϵ initially has a value of 1, and this parameter decreases over time to a minimum of 0.1 as the agent begins to exploit the knowledge gained [132].

3.4.2 Reinforcement Learning for Variable Speed Limits

One of the first demonstrations of reinforcement learning to the VSL problem is due to Zhu and Ukkusuri [179]. In their formulation of the VSL problem as an MDP, the state space is characterised and discretised according to various levels of congestion. Four such levels of congestion are defined as follows

$$u_{\text{limit}} = \begin{cases} 1 & \text{if} \quad 0 < \rho_i(t) \leq 0.25\rho_{\text{jam}} \\ 2 & \text{if} \quad 0.25\rho_{\text{jam}} < \rho_i(t) \leq 0.5\rho_{\text{jam}} \\ 3 & \text{if} \quad 0.5\rho_{\text{jam}} < \rho_i(t) \leq 0.75\rho_{\text{jam}} \\ 4 & \text{if} \quad 0.75\rho_{\text{jam}} < \rho_i(t) \leq \rho_{\text{jam}}, \end{cases} \tag{3.37}$$

where level 1 is characteristic of a free-flow state, level 2 is characteristic of a state of slight congestion, level 3 is characteristic of a state of moderate congestion, and finally, level 4 is characteristic of a state of heavy congestion. The action space comprises a discretised function of speed limits which may be employed by the learning agent on each controlled link i, given by

$$V_i(t) = V_0 + a_i(t)I, \tag{3.38}$$

where $a_i(t) \in \{1, 2, \ldots, A\}$ denotes the space of actions available to the learning agent on controlled link i, V_0 denotes the minimum allowable speed limit, and $V_0 + AI$ denotes the maximum allowable speed limit. The objective to be minimised in this study is the total travel time spent by vehicles in the system. As in the formulation of Rezaee *et al.* [132], the reward function employed in order to achieve this goal of minimising the total travel time is defined as

$$R(i) = -N(i), \tag{3.39}$$

where $N(i)$ represents the average number of vehicles present on the controlled link i during the current control period. This formulation of the reinforcement learning problem was employed in a link-based dynamic network loading model, which is a second-order macroscopic traffic simulation model. The RMART reinforcement learning algorithm (Algorithm 2.5) was employed as solution technique to the reinforcement learning problem.

Walraven *et al.* [166] demonstrated another application of reinforcement learning to the VSL problem, once again using METANET as the underlying macroscopic traffic modelling tool. The state space is again defined in such a manner as to provide a representation of the current traffic flow conditions on the highway. The state of the highway is given by

$$s_t = \left(\frac{a_{t-1}}{u_f}, s_{t-1}(0), \frac{u_1(t)}{u_f}, \ldots, \frac{u_N(t)}{u_f}, \frac{\rho_1(t)}{\rho_{\text{jam}}}, \ldots, \frac{\rho_N(t)}{\rho_{\text{jam}}} \right), \tag{3.40}$$

where the first and second state variables represent the current and previous speed limits assigned to the highway [166]. The remaining state variables represent the current speeds and velocities for the N highway sections during time period t. The current speed u_n and density ρ_n for each section $n \in \{1, \ldots, N\}$ are normalised with respect to the free flow speed u_f and jam density ρ_{jam}, respectively. The speed and density information for all highway sections is included in order to allow the learning agent to detect an oncoming traffic jam in one of the sections under consideration. The action space $\mathcal{A} = \{60, 80, 100, 120\}$ contains a number of discrete speed limits which may be applied by the learning agent. In order to smooth the increase and decrease of the speed limits, this state space may also be defined in a state-specific manner, thus allowing only certain speeds to be selected in relation to the current applied speed limit. For example, if a current speed limit of 120 km/h is enforced, the action space may be reduced to $\mathcal{A}(s_t) = \{80, 100\}$, thus allowing the agent to choose only between a new speed limit of 80 km/h or 100 km/h, excluding the speed limit of 60 km/h from the available action space [166]. Finally, the reward function employed is

$$r_t = \begin{cases} 0 & \text{if } \min\{u_i(t+1) \mid i = 1, \ldots, N\} > u \\ -h(t, t+1) & \text{otherwise,} \end{cases} \tag{3.41}$$

where u is a pre-specified threshold speed, and $h(t, t+1)$ is a function denoting the number of vehicle hours accumulated during the time interval from time t to time $t+1$. As may be seen from the definition of the reward function, the objective to be achieved by the learning agent is once again to minimise the total time spent in the system by the vehicles. In order to solve the reinforcement learning problem, Walraven *et al.* [166] employed Q-learning, in conjunction with a neural network using the back propagation algorithm as a function approximator.

Another implementation of Q-learning for solving the VSL control problem is due to Li *et al.* [85]. In this implementation, the same basic highway network, consisting of a dual carriageway with a single on-ramp, as previously employed by Hegyi *et al.* [53] and Müller *et al.* [105] was implemented within a macroscopic cell transmission simulation model. In this implementation, the state space comprised three variables, namely the density upstream of the bottleneck location at the lane merge, the density directly downstream of the lane merge and the traffic density at the on-ramp. This state space was discretised in intervals of magnitude 5, between 5 and 80 vehicles/mile/lane on the mainline, while intervals of magnitude 3, between 3 and 30 vehicles/mile/lane were employed for the on-ramp density. This discretisation was employed because a table-based implementation of Q-learning was adopted [85]. The action space consisted of three actions: To either reduce the current speed limit by 5 miles per hour, to maintain the current speed limit or to increase the speed limit by 5 miles per hour. As a result, the speed limit was adjusted incrementally in order to avoid introducing major disturbances in the traffic

flow. Finally, the speed limits that could be applied were bounded between 20 and 65 miles per hour, resulting in an action space $\mathcal{A} = \{20, 25, 30, 35, 40, 45, 50, 55, 60, 65\}$. The reward was given in terms of the Poisson mass function

$$R(s) = \mu \frac{\lambda^s e^{-\lambda}}{s!}, \tag{3.42}$$

where $R(s)$ denotes the reward achieved when in state s, μ denotes the parameter used to scale the magnitude of the reward, and λ is the Poisson parameter. The value of μ was taken as 1×10^4 while the parameter λ was set to the critical density at the bottleneck. In order to increase the convergence speed of the Q-learning algorithm, an additional incentive of 200 was added to the reward function when the agent found itself in the two states closest to the critical density, while a penalty of 400 was subtracted for severely congested states (*i.e.* those states with a bottleneck density above 40 veh/mile/ln). This implementation was finally evaluated in the context of a real-world case study involving a section of the Interstate 880 highway in California.

3.5 Chapter Summary

This chapter contained reviews of traffic flow theory and specific highway traffic control measures. In §3.1, the two basic traffic flow modelling paradigms, namely macroscopic and microscopic traffic flow theory, as well as some of the basic notions within each of these paradigms, were introduced. Thereafter, the focus shifted in §3.2 to the control of traffic on a highway, with a review of RM as a means of controlling the number of vehicles allowed onto the highway in §3.2.1. Dynamic speed limits, which may be employed so as to control the flow of traffic already on the highway, were next reviewed in §3.2.2. In §3.2.3, the notion of LA was reviewed, which may be employed to improve the utilisation of the available space on the highway in the most efficient manner. This was followed in §3.3 by a review of various techniques which have been employed in order to improve the traffic flow along a highway in the presence of varying percentages of autonomous vehicles. Finally, applications of machine learning, and more specifically, reinforcement learning to these highway traffic control methodologies, were reviewed in §3.4.

CHAPTER 4

Computer Simulation Modelling

Contents

This chapter serves as a brief introduction to the extensive field of computer simulation modelling. In §4.1 simulation modelling itself, as well as a few key concepts pertaining to simulation modelling, are defined. This is followed in §4.2 by an introduction to the four major simulation modelling paradigms found in the literature. Thereafter, twelve generic steps that are typically followed during the completion of a simulation study are briefly discussed in §4.3. In §4.4, more detail is provided on some of the various methods suggested in the literature for verification and validation of a simulation model. Before the three currently prevailing traffic simulation modelling paradigms are introduced in §4.6, some of the advantages and disadvantages of simulation modelling are mentioned in §4.5. The chapter finally closes in §4.7 with a brief summary of the material included.

4.1 Simulation Modelling Concepts

Several interpretations and definitions of the notion of simulation have been proposed in the literature. Perhaps most famously, Banks *et al.* [9] defined simulation as "the imitation of the

operation of a real-world process over time." Law and Kelton [81] defined simulation as "a broad collection of methods and applications to mimic the behaviour of real systems, usually on a computer with appropriate software." Simulation may, as a result, be seen as a process of experimentation, using a model of a real-world system, with the aim of studying the behaviour of the underlying system, given certain starting conditions. In order to achieve this, the behaviour of the model has to be a sufficient predictor of the behaviour of the real-world system, so that specific "what-if" questions may be answered using the simulation model [119].

While several different modelling paradigms exist within the broader concept of simulation, there are a number of key concepts that are common to all of these paradigms, as they form the basis of most simulation models. The *system, model, events, entities, attributes, activities, resources* and *system state variables* are these common concepts on which the notion of a simulation model is built. This section serves as a brief introduction to these concepts.

A *system* may be defined as a set of interrelated objects, or *entities*, which cooperate in order to achieve a common goal [175]. A *model* was defined by Shannon [147] as the representation of an entity/object in a form other than itself. This representation usually comes paired with a number of assumptions, and is used in order to predict the behaviour of the real-world system under various conditions.

System state variables are the collection of all the information required to sufficiently describe the current system status at any given point in time [10]. This collection of variables used to provide a snapshot description of the system is known as the system *state* [175]. In the case of a traffic highway simulation, for example, the state of the system may be defined according to the various traffic densities, speeds, and flows on the particular stretch of highway under consideration. *Events* are specific occurrences which have the potential to change the system state variables, as well as the resulting state that the system finds itself in.

Entities are objects, such as persons or vehicles, which possess the ability to cause changes in the system state variables [63]. They may either be dynamic (*i.e.* possess the ability to move through a system), or they may be static (*i.e.* remain stationary and serve other entities in the system) [9, 10]. All entities possess a number of unique characteristics, called *attributes*, which are used to describe the performance, as well as the functions of these entities [10, 63]. Events are created by the interaction of entities with *activities* [9, 10, 63]. *Activities* are the processes and the logic which govern the execution of the simulation. Within the context of a simulation, there are three major types of activities, namely delays, queues and logic [63]. A *resource* is a special type of entity, which is typically of a static and capacity-restricted nature, and provides a service to other dynamic entities [10, 63]. Examples of resources are bank tellers, order windows or packaging machines. Resources may find themselves in one of a number of given states, such as idle, busy, blocked of failed.

Banks *et al.* [9] stated that simulation models themselves may be classified as being either *static* or *dynamic*, *deterministic* or *stochastic*, and *discrete* or *continuous*. A *static* simulation model, commonly referred to as a *Monte Carlo* simulation model, is a model that is independent of time, and thus only describes a system at a specific instant in time [9, 175]. A simulation of the process of rolling a die is a good example of a static simulation model. *Dynamic* simulation models, on the other hand, attempt to capture the behaviour of real-world systems as they evolve over time [175]. The simulation of a bank teller from the time when the bank opens at 09:00 until the bank closes at 15:30 is an example of a dynamic simulation model [9].

A simulation model that is devoid of randomness is called a *deterministic* simulation model. As a result, such models assume certainty with respect to every aspect of the model. An example of a deterministic model would be that of a dentist's office if all patients were to arrive at their

precisely scheduled times, and all procedures would take exactly the planned amount of time [9, 175]. A *stochastic* model, on the other hand, is one that contains at least one random variable. Therefore, the model output may be expected to be different for every simulation run performed. Typically, probability distribution functions are employed within stochastic simulation models in order to specify the starting times or durations of specific events.

In *discrete* simulation modelling, all system state variables are updated at discrete or countable points in time. In contrast, the system state variables are updated continuously as time progresses in a *continuous* simulation model [175]. An example of a discrete simulation model is that describing the events in a banking hall, where customers arrive at specific points in time, whereas an example of a continuous simulation model is that of the temperature distribution within an engine block as the engine runs for an extended period of time.

4.2 Prevailing Simulation Modelling Paradigms

Depending on the nature of the problem and the resulting level of abstraction required, there are four distinguishable simulation modelling paradigms currently available to the analyst. These paradigms are *agent-based modelling, discrete-event modelling, system dynamics modelling,* and *dynamic systems modelling* [17].

4.2.1 Agent-based Modelling

In *agent-based modelling*, a system is modelled as the collection of a number of autonomous decision making entities, called *agents*. Each of these agents responds to its current situation based on a set of predefined rules. This results in certain collective behaviours which are then exhibited by the system [16]. People, vehicles, products or companies are examples of possible agents whose behaviour is defined at the microscopic level in an agent-based model, while reactions, population dynamics and traffic flows are examples of possible emerging behaviours which result from the collective behaviour of the individual agents.

4.2.2 Discrete-event Modelling

As stated above, a discrete system is one in which the state variables are updated at specific, discrete points in time [175]. In the same manner, *discrete-event modelling* is a modelling paradigm in which events that change the system state occur only at discrete, but possibly random, time points [145]. In discrete-event modelling, the dynamics of the system are captured as the simulation model time advances, but the system state remains constant between consecutive events. As a result, flow charts, or the so-called "transaction-flow world view," is often employed to describe the movement of entities within the simulation model [145]. A simple example of a discrete-event simulation model is that of a drive-through window at a fast-food vendor, where customers arrive and queue at various stages until orders have been processed and the customers leave the system.

4.2.3 System Dynamics Modelling

System dynamics modelling was defined by Borshchev and Filippov [17] as "the study of information-feedback characteristics of industrial activity to show how organisational structure, amplification, and time delays interact to influence the success of the enterprise." In system dynamics

modelling, processes are represented in terms of stocks (*e.g.* material, knowledge, money or people), the flows between these stocks, and the information according to which the values of these stocks are determined. In system dynamics, however, the focus is not on individual agents or entities as in agent-based modelling, but rather on capturing and experimenting with the policies that govern these flows of stocks [17].

4.2.4 Dynamic Systems Modelling

Dynamic systems modelling is often considered to be the ancestor of system dynamics modelling. It has been employed extensively in mechanical, electrical, chemical and other technical engineering disciplines as part of the design process. In dynamic systems modelling, the underlying system is represented in the form of various state variables and algebraic differential equations. Unlike in system dynamics modelling, however, these equations and variables have a direct physical meaning, such as velocity, current, or pressure, and are not used as aggregate quantities of entities [17].

4.3 Typical Steps in a Simulation Study

As a guideline to the completion of an effective and successful simulation study, Banks *et al.* [9, 10] suggested a twelve-step procedure to be followed. These steps are summarised in flowchart-form in Figure 4.1 and are discussed briefly in this section, using the same numbering as in the figure.

1. *Problem identification and formulation.* The first step of any simulation study is the formulation of the problem at hand by means of a formal problem statement [9, 82]. It is imperative that the problem is stated unambiguously and that the simulation analyst understands the problem [10]. This step may, however, have to be revisited at a later stage due to unexpected findings or due to a better understanding of the underlying system being modelled.

2. *Setting of objectives and overall project plan.* The project plan defined during this stage serves to highlight the scope of the project, as well as the objectives which are to be achieved (*i.e.* the questions that are to be answered) [10, 82]. The project plan should also include a statement specifying the various scenarios that are to be investigated, as well as an indication of the time frame and budget available for the completion of the simulation study. The staff and equipment requirements, as well as a guideline for expected outcomes at various stages of the project may also be included in this project plan [81].

3. *Model conceptualisation.* During this step, an abstraction of the real-world system is established, based on a series of mathematical and logical relationships between the various components and structure of the system [10]. It is often best to start with a simple model, to which complexity can be added as the modelling process continues and the model evolves. The model complexity should, however, not exceed that which is required in order to accomplish the purpose of the model [9], as an unduly complex model will typically only increase both the computational and monetary expense of the study without a justification of these increases in terms of a higher-quality output [10].

4. *Data collection.* As indicated in Figure 4.1, this step occurs concurrently with model conceptualisation. This is due to the fact that there is a constant interplay between

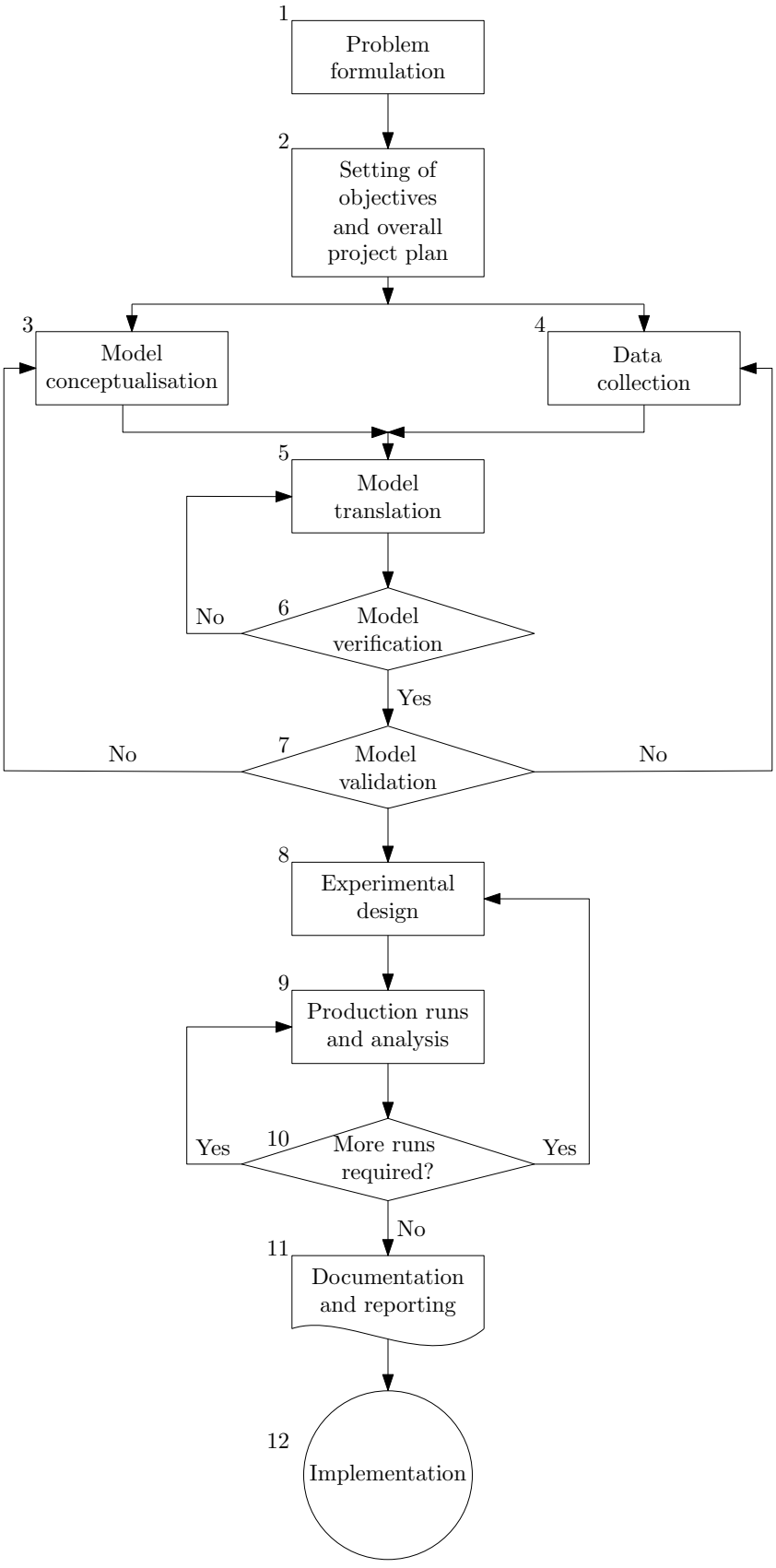


FIGURE 4.1: *The twelve steps in a typical simulation study. Adapted from Banks et al. [9].*

model construction and the corresponding data requirements, with the objectives largely dictating the data requirements [9]. As the model complexity changes during Step 3, the data requirements may change as well. It is thus important to keep data availability from the real-world system in mind when performing the model conceptualisation [10].

5. *Model translation.* This step involves translating the conceptual model, with its underlying logic and arithmetic developed during Step 3, into an appropriate computer recognisable simulation modelling language [9, 10, 82].

6. *Model verification.* Once the model has been programmed into a suitable computer language, it is necessary to verify whether or not the actual model represents the originally envisaged model that was to be formulated and built (*i.e.* whether the model and the underlying logic execute correctly) [9, 10, 148]. This process is often referred to as debugging, and it is advisable that it is performed continuously throughout the model building process [82].

7. *Model validation.* Validation is the process of determining whether the simulation model which has been built provides an accurate representation of the underlying real-world system. Ideally, model validation is performed by means of a comparison of the model output with output data taken from the real-world system [9, 10, 82].

8. *Experimental design.* During this step, various different system designs which are to be investigated are decided upon [82, 148]. For each of these scenarios, decisions as to the number of simulation runs required, an adequate length of each such run, and the manner of initialisation which will yield the desired results need to be specified and determined [9].

9. *Production runs and analysis.* The results for the various runs and scenarios, as determined in the previous step, are recorded and subsequently analysed statistically in order to compare the model output performance for the various scenarios [10, 82, 148]. Common statistical measures employed include sensitivity analyses as well as the determination of confidence intervals for various performance measures.

10. *Additional runs.* Based on the analysis completed in the previous step, it is decided whether additional simulation runs or different experiments are required for further or more accurate performance assessment [9].

11. *Documentation and reporting.* Documentation refers to both the simulation model and its implementation in a software suite, as well as reporting on the progress of the study itself. Program documentation is especially important if the final model implementation is to be used by a client who was not involved in the original model building process. Furthermore, good documentation may facilitate model modifications should they be required in future [9, 148]. The final report should be written in a clear and concise manner, stating all assumptions made during the modelling process, and report on all analyses and findings as well as recommendations for implementation [10].

12. *Implementation.* In the final step of any simulation study, the simulation analyst acts primarily as a reporter, providing suggestions for possible improvements that may be pursued, but the final decision on implementation of such recommendations depends on the decision maker. The likelihood of the recommendations being implemented often depends on the rigour with which the previous steps have been performed, as well as the outcomes of these steps [10].

4.4 Verification and Validation of a Simulation Model

Verification and validation of a simulation model are critical to ensure the success of a simulation study, as they ensure the validity of the model output, and the recommendations based on these outcomes [81]. As such, the verification and validation processes are both aimed at producing a credible and suitable model. Verification is primarily concerned with the correct building and implementation of the conceptual model developed, whereas validation is concerned with ensuring that an appropriate model, giving an accurate representation of the real-world system, is built [9, 81]. A graphical representation of the verification and validation processes is given in Figure 4.2.

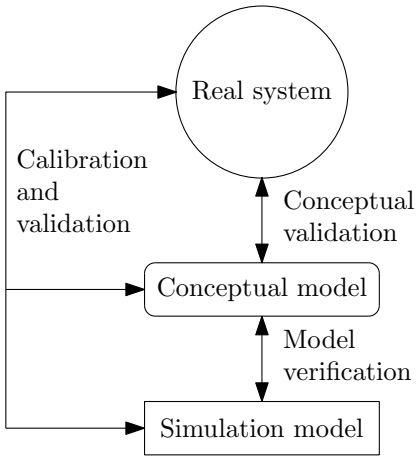


FIGURE 4.2: *The role of verification and validation in the simulation modelling process. Adapted from Banks et al. [9].*

4.4.1 Verification of a Simulation Model

As stated above, verification of a simulation model involves determining whether the simulation model has been built correctly within the chosen simulation environment [82, 139] (*i.e.* whether the model acts as expected and in accordance with the underlying model logic). Balci [8] described model verification as "substantiating that the model is transformed from one form into another, as intended, with sufficient accuracy." Typically, debugging of a model is the main component of model verification, ensuring that the computer code faithfully captures the designed model [125].

The primary technique employed for debugging logic errors in a simulation model is that of performing structured walk-throughs and tracing model output at various stages [139]. Conducting the walk-through requires that the analyst manually emulates execution of the model, often by following a single entity along its path throughout the model. Animation is usually an effective tool simplifying the model walk-through, providing a visual medium for following the path of an entity through the model. This is especially effective when combined with the tracing of variable values throughout the model execution [9]. Another popular technique for model verification is that of examining the model output for a large variety of input parameters and determining whether or not the model output makes logical sense. This may be facilitated by the use of an interactive run controller or debugger [9].

The above-mentioned techniques are generic, and applicable to any type of simulation model. Rakha *et al.* [125], however, suggested a five-step verification procedure to be followed for the verification of traffic simulation models specifically.

The first step involves the selection of model input parameters, such as expected traffic demands, route choices and preferred speed values. These values should be selected in such a manner so as to represent the expected domain of application of the simulation model.

Step two involves an independent test to ascertain whether each of the selected values from step one agrees with real-world data. Suppose that for a highway model, the required model parameters are the free-flow speed and critical speed values. Suppose further that field data collected from a stretch of highway considered indicates that these values typically fall within the range 80km/h to 120km/h and 60km/h to 90km/h, respectively. Then the independent test should ensure that the initial selected values do, in fact, fall within these ranges.

Following the independent test of step two, an additional test is performed in step three, in order to ascertain whether or not the combination of selected input parameter values is consistent with the field data. For the previous example, a combination of a free-flow speed of 80km/h and a capacity speed of 90km/h would be separately consistent with the input data from the real-world data, but the combination is infeasible, since one cannot have a capacity speed which is higher than the free-flow speed. Thus the assessment of the combination of input parameters ensures that the combination of chosen values is also consistent with typical measured values.

Step four involves the generation of results, not only by the simulation model, but also through the direct application of the model logic without the use of the computer code.

During the fifth and final step, the two sets of results emanating from step four are compared. Should these results conform with a specific level of required accuracy, the verification process may be considered successful, while inconsistencies greater than the required accuracy will require revision of the simulation model after which the verification process needs to be repeated.

4.4.2 Validation of a Simulation Model

Simulation model validation involves determining whether an appropriate model, which is able to represent the real-world system with acceptable accuracy, has been built, taking into account the particular objectives of the simulation study [8, 82, 139]. Law and Kelton [81] stated that when determining whether a model is, in fact, a valid model, three types of validity have to be considered, namely *conceptual validity, operational validity* and *credibility.*

Conceptual validity answers the question whether the model is, in fact, a valid representation of the real world system [81]. Often the type of validation procedure employed to confirm conceptual validity is *face validation*, which involves asking individuals who are knowledgeable about the real-world system whether the model and its behaviour are reasonable [139]. *Turing tests* are sometimes employed for this purpose, where outputs from the real system, as well as model outputs, are given to a subject matter expert, who is knowledgeable about the real-world system, and the subject matter expert is asked to distinguish between the real and model outputs [82, 139].

Operational validity provides an answer to the question whether the model's output is in line with the real-world system's behavioural data [81]. This is typically achieved by *results validation*, and is only possible if there are real-world data available for comparison. The comparison typically consists of a wide range of statistical analyses so as to assess whether the model output is (statistically) significantly different from the real-world data [82, 139]. Additionally,

an operationally valid model should exhibit reasonableness, in the sense that the model should exhibit *continuity, consistency* and *degeneracy*. *Continuity* implies that if small changes are made to the model's input parameters, these should be reflected in the model's outputs and variables by similarly small changes [81]. *Consistency* implies that the model output should be similar for separate simulation runs with the same input parameters (*i.e.* the model output should not change significantly due to a change in the random number generator seed). Finally, *degeneracy* implies that the model should reflect the removal of one or more objects. For example, if a banking hall has two tellers, and one of these tellers is removed, the effect should be reflected in the model output [81, 139]. Another test for degeneracy is known as an *extreme condition test*, where inappropriate input parameters are specifically chosen so as to ascertain whether an appropriate effect is displayed by the model. If, for example the inventory of raw material is set to zero in a simulation of a production plant, the resulting production rate should also be equal to zero [139].

Credibility is determined by the end-user and decision-maker who employs the model in order to answer the questions set out in the project objectives. The decision-maker will trust a credible model, whereas if the decision-maker feels that the model is not credible, the results and recommendations emanating from the simulation study will typically not be trusted and implemented [81].

4.5 Some Advantages and Drawbacks of Simulation Modelling

As technological breakthroughs in the computer industry lead to ever-faster and more powerful computational hardware, more powerful, more accurate and more user-friendly simulation software suites are developed. The combination of these two factors has led to a rapid expansion of the number of companies employing simulation as a tool in their daily operations [10], as ever more complex systems may thus be studied.

Probably the greatest advantage of a simulation study is that it allows for the investigation of various scenarios, additions or modifications to a real-world system, without actually disrupting the real-world system during the time of the investigation [9, 81]. Experimentation with alternatives can take place in the simulated environment, rather than with the actual system, thus reducing the risk of unexpected problems or unforeseen side-effects. Since the modelling process requires considerable insight into the operation of the real-world system in order to build an accurate representation thereof in the simulation environment, simulation may help to identify microscopic problems or critical parameters, which may subsequently be studied [9].

Another advantage of simulation modelling is that it allows for the evaluation of alternatives, generally at a fraction of the cost that would be required to experiment with the real-world system (typically around 1% of the real-world implementation cost [10]). This allows for greater experimentation possibilities involving a wider range of alternatives, with the possibility of finding optimal, or near-optimal, solutions to a given problem [10]. Furthermore, due to the notion of virtual time in a simulation model, simulation allows for the compression as well as expansion of time. Compression of time enables the study of long processes within relatively short time periods, while expansion of time may allow a user to study certain near-simultaneous events separately in order to gain more insight into process intricacies [10]. As a result, simulation can also act as a tool for problem area or constraint identification.

In cases where the system under consideration is too complex to be studied analytically, simulation often offers a practical alternative [81]. Finally, simulation has the potential to be used as a tool for building consensus among decision-makers as potential gains are backed up numerically

with simulation outputs [10]. Thereafter, the simulation model may be employed as a tool in the process of preparing for the change that is to be implemented, answering "what-if" questions which may hamper the change implementation. With the aid of animation, a simulation model may also be used as a training tool for the end users of certain equipment, helping them to understand the underlying system implications of various actions [9].

Simulation is, however, not without drawbacks. The most prominent of these being that, since building an accurate, credible simulation model is as much an art as a science, the simulation analyst requires extensive training and experience, not only within the realm of simulation itself, but also in respect of the use of specific simulation software [10, 148]. Furthermore, for the effective and accurate interpretation of simulation outputs and results, the analyst also needs to have a sound statistical background. The continual improvement of new simulation software and packages aims to minimise the effect of these skills requirements by including many built-in analysis tools, although the analyst still has to have a sound understanding of the underlying techniques in order to be able to interpret the results correctly. Another drawback of these improved software packages is that they tend to be expensive [9]. Simulation modelling, if executed properly, can be time consuming and expensive. This may be attributed to large computational overheads required by some complex simulation models [10].

The creation of a simulation model relies on the making of valid and accurate assumptions. If an assumption made is incorrect, or not properly documented, certain conclusions drawn from the study may not be valid. In many cases, the results from the simulation study may also be non-trivial and difficult to interpret [10]. That being said, in some cases, simulation may be employed for solving problems where an analytical solution would have been possible. In such cases, the analytical solution is preferable due to reduced randomness which may influence the results, and hence, in such a case, the problem should be solved analytically rather than through the use of simulation [10]. Finally, since simulations are based on a certain degree of randomness, in some cases it may be difficult to distinguish between the occurrence of specific results due to underlying relations in the system or due to randomness [9].

4.6 Traffic Simulation Modelling Paradigms

Simulation in the context of traffic flow and control has been defined by May [96] as "a numerical technique for conducting experiments on a digital computer, which may include stochastic characteristics, and involve mathematical models that describe the behaviour of a transportation system over extended periods of time." Three different paradigms of traffic simulation modelling exist, based on the level of abstraction of the underlying simulation model, namely *macroscopic, mesoscopic* and *microscopic* simulation modelling. For each of these three paradigms, several examples of simulation models exist. Boxill and Yu [18], for example, reviewed and evaluated sixteen macroscopic traffic simulation models, three mesoscopic simulation models, and sixty five microscopic simulation models. This section is devoted to a review of the three traffic simulation modelling paradigms, as well as the provision of examples of commercially available traffic simulation software within each of these specific paradigms. A graphical illustration of the difference between the two extreme modelling paradigms — macroscopic and microscopic traffic simulation — is shown in Figure 4.3.

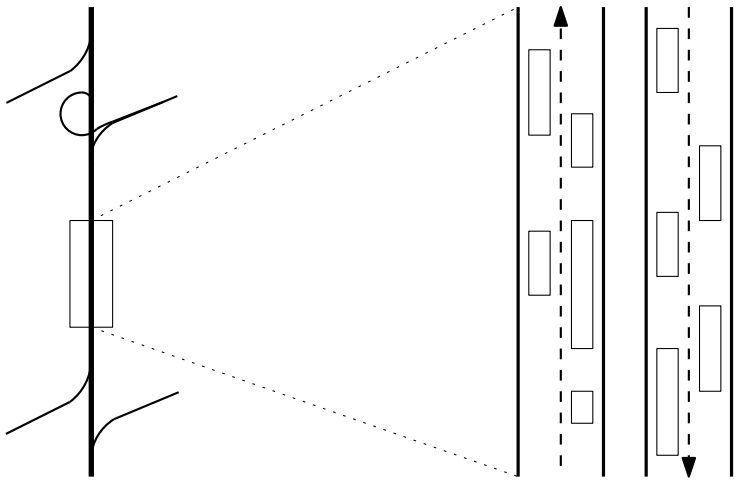


FIGURE 4.3: *A comparison of the level of detail in a macroscopic traffic model in (a) and a microscopic traffic simulation model in (b).*

4.6.1 Macroscopic Traffic Simulation

In *macroscopic* traffic simulation modelling, the fundamental theories of traffic flow, introduced in §3.1.1, are translated into simulation models. Macroscopic traffic simulation therefore involves the analysis of aggregated vehicle data [18]. Using average vehicle flows, densities and speeds, vehicle movement may thus be modelled as a compressible fluid [61]. Due to the conservation of the number of vehicles, the underlying flow equation of macroscopic traffic modelling is governed by the flow equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho V)}{\partial x} = V(x,t),$$

where $\rho(x,t)$ represents the density of vehicles at position x along a roadway at time t, and $V(x,t)\,\mathrm{d}x$ denotes the rate of vehicles that are either joining or leaving the road network of length $\mathrm{d}x$ [54]. The most significant advantage that macroscopic simulation holds over the other traffic simulation modelling paradigms is that due to the relatively low model complexity, it has a relatively low computational overhead [61].

The most popular macroscopic modelling software for the simulation of highway networks is METANET, developed by Messner and Papageorgiou [100] in 1990. In METANET, the underlying highway network is represented by a directed graph, where each edge represents a stretch of highway with uniform characteristics, while the nodes represent bifurcations, junctions, on-ramps and off-ramps [116].

TRANSYT/10 (Traffic Network Study Tool) is an off-line macroscopic simulation program used for determining and studying optimal fixed-time, coordinated traffic signal timings within a transportation network for which the average flows are known, in an attempt to minimise delays [128]. TRANSYT/10 has been widely accepted as the standard for setting fixed-time signal timings [18].

Another example of macroscopic traffic simulation software is KRONOS, which was developed during the early 1980s. Like METANET, KRONOS is used mainly for the simulation of highway networks, and is based on the *Lighthill-Whitham-Richards* (LWR) theory of traffic flow [122].

4.6.2 Microscopic Traffic Simulation

In a *microscopic* traffic simulation model, each vehicle is modelled individually. This involves the unique assignment of movements (speed, acceleration and deceleration) as well as characteristics (vehicle length and position) to each individual vehicle [18, 66, 77]. The behaviour and resulting movement of each individual vehicle through the road network is based on a set of rules in terms of a car-following protocol, lane changing rules and gap acceptance algorithms [66]. The most significant advantage of microscopic traffic simulation is the high level of detail that may be incorporated into a simulation study, allowing the traffic system to be studied on an operational level [56].

Simulation of Urban Mobility (SUMO) [64] is an open-source microscopic traffic simulation software suite designed for the analysis of large road networks, allowing up to 10 000 roads to be incorporated in a single simulation model [76]. Due to the relatively coarse nature of SUMO, it is not often employed for the detailed evaluation of existing intersections. It has, however, been found useful for the evaluation of traffic signal control algorithms due to its fast execution time [13]. Kotusevski and Hawick [76] found that SUMO is less user-friendly, and more writing intensive than other software due to the fact that the user is required to write multiple XML files. Another drawback of SUMO is that only left-hand driving is supported in the software.

Vissim [123] is another example of a readily available microscopic traffic simulation software suite. Vissim is a behaviour-based software suite suitable for modelling both urban and highway traffic networks, typically employed in order to study and optimise flow of traffic within the network. Vissim provides the user with an intuitive *graphical user interface* (GUI), allowing the user to easily construct the road network and populate it with vehicles and traffic signals, while relaying relevant information such as vehicular movements as well as travel time summaries of vehicles in an animated format [18].

Quadstone Paramics [124] is another commercially available microscopic traffic and pedestrian simulation software suite which allows the user to build large traffic networks, including highways and urban networks, as well as combinations thereof. The Paramics software includes a large variety of built-in features for traffic analysis, allowing the user to incorporate measuring devices, such as loop detectors, or other features, such as *variable message signs* (VMSs), in the simulation environment [124].

4.6.3 Mesoscopic Traffic Simulation

A *mesoscopic* traffic simulation model comprises a combination of microscopic and macroscopic elements. In a mesoscopic traffic simulation model, the individual vehicle characteristics of a microscopic traffic simulation model are retained, but the traffic flow dynamics are aggregated as in a macroscopic model [20]. In contrast with a microscopic simulation model, the performance measures and characteristics of groups of vehicles are recorded. Unlike macroscopic traffic simulation, however, mesoscopic traffic simulation models can simultaneously accommodate any number of such groups of vehicles and record the actions between these groups [140]. Effectively, single vehicle entrance and exit flows from a section are recorded, while the flow within a section is modelled using flow equations and macroscopic variables. An advantage of adopting a mesoscopic traffic modelling approach is that it typically allows for a high degree of flexibility when it comes to input data specifications and that complex algorithms for traffic control are typically easily implemented in the mesoscopic context [140].

The Simulation and Assignment of Traffic in Urban Road Networks (SATURN) [47] model is an example of a ready-made mesoscopic traffic simulation model. In the SATURN model, two

distinct phases are employed, the first being a detailed simulation of the delays occurring at the various intersections, while the second phase entails route selection for vehicles according to their origin-destination demands [47]. During execution, the simulation model iterates between these two phases, using the respective output of each phase as the input for the subsequent iteration of the other phase [109].

INTEGRATION is another routing-oriented mesoscopic traffic simulation implementation. In INTEGRATION, each vehicle is modelled individually, but vehicle movement is determined according to macroscopic traffic flow theory, thus incorporating both microscopic and macroscopic traffic flow principles [109]. This facilitates the modelling of lane changing as well as the incorporation of toll plazas and vehicle emissions [162]. Boxill and Yu [18] found INTEGRATION to be the leading software for modelling and evaluating intelligent transport systems along corridors when real-time demand is incorporated.

4.7 Chapter Summary

Some of the most pertinent general principles and concepts of computer simulation modelling were introduced in §4.1. This discussion included the description of a number of concepts that are common to simulation models from all four prevailing modelling paradigms, namely agent-based modelling, discrete-event modelling, system dynamics modelling and dynamic systems modelling, as described in §4.2. This was followed in §4.3 by the discussion of a twelve-step process for successfully completing a typical simulation study. Following a description of some of the methods available for the verification and validation of a simulation model in §4.4, a number of advantages and disadvantages of employing simulation modelling as an analytical problem solving tool were mentioned in §4.5. Finally, the three traffic simulation modelling paradigms, classified according to their respective levels of abstraction, were reviewed in §4.6. This was accompanied by an overview of some of the commercially available software packages within each of these modelling paradigms.

Part II

Current Technologies

CHAPTER 5

A Microscopic Highway Simulation Model

Contents

This chapter is devoted to a detailed description of the microscopic (agent-based) traffic simulation model designed and implemented as a test-bed environment for the experiments conducted in this dissertation. The simulation model was implemented in the AnyLogic 7.3.5 [5] software suite, making specific use of its built-in Road Traffic Library. The chapter opens in §5.1 with a detailed description of the modelling framework, with a specific focus on the process of building the road network, implementing highway intersections, and generating individual vehicles as well as recording model output data. This is followed by a description of the verification and validation techniques employed throughout the model building process in §5.2. Thereafter, the experimental design employed in later chapters of the dissertation for the purpose of comparing highway traffic control policies is described in §5.3. Finally, the chapter closes in §5.4 with a brief summary of the work included in the chapter.

5.1 Model Framework

An agent-based, microscopic highway traffic simulation model was designed and implemented as a test-bed environment for the evaluation of the effectiveness of various highway traffic control measures under the guidance of policies provided by reinforcement learning algorithms.

This model has been built in such a manner that it is able to represent a section of highway, together with intersections, consisting of both on- and off-ramps, with sufficient accuracy so as to be able to conduct a thorough evaluation of the effectiveness of highway control policies proposed by reinforcement learning agents. The simulation model developed is stochastic in nature, since Monte Carlo methods are utilised, and Poisson, exponential and uniform distributions are employed when attributes are assigned to the various model entities. Furthermore, the model is continuous as well as dynamic, as its state variables are updated continually throughout model execution.

The static entities of the simulation model comprise road mark-up elements. These entities are roads, intersections, traffic signals, and stop lines. The only dynamic entities in the simulation model are vehicles, as they are the only entities that physically move within the simulated environment during execution of the simulation model. The traffic signals implemented in order to enforce ramp metering at on-ramps are a special type of entity, since they may also be classified as a resource, allocating green time to vehicles and thereby controlling the vehicle flow.

Each of the aforementioned entities possesses a number of unique attributes. For vehicles, these attributes include speed, acceleration, deceleration, colour, length, arrival rate, arrival location, destination, position, as well as travel time and distance travelled. Some of these attributes are assigned random values through the use of built-in probability distributions. The attributes unique to road segments include length and the number of lanes in each direction, while intersection attributes include the roads connected by the intersection, as well as the manoeuvres that the vehicles are allowed to perform as they pass through the intersection. The current phase, the elapsed time during the current phase, the time remaining for the current phase, as well as phase durations and sequences are the attributes specific to each traffic signal. Finally, the attributes associated with a stop line include its position along a specific road segment, as well as the type of traffic sign associated with the stop line. In the case of a speed limit sign, the value of the speed limit is also an attribute of the stop line.

The events occuring during the execution of the simulation model may either be endogenous (internal) or exogenous (external). Endogenous events include vehicle manoeuvres, the changing of traffic signal phases, and changes in vehicle speeds, while exogenous events include vehicle arrivals into the system and vehicle exits from the system.

5.1.1 Constructing the Road Network

One of the most important aspects to consider when constructing any simulation model is the requirement that the model has to be an accurate representation of the real-world system. In the case of a traffic simulation model, the road network implemented within the simulation model should accurately represent the corresponding real-world network. In order to facilitate the accurate construction of road networks in terms of scale and shape, AnyLogic [5] offers a built-in *geographic information system* (GIS) function which allows access to the *open street map* (OSM) [108] server. The OSM server provides a readily available global map of road networks. Within the so-called `gisMap` in AnyLogic, specific `gisPoints` may then be specified, and routes between these points may be generated, based on existing roads between the two points. An example of this is shown in Figure 5.1. These GIS routes may then be converted to road mark-up elements such as roads and intersections, which form part of AnyLogic's built-in Road Traffic Library. The advantage of employing this approach is that the scale and underlying shape of the automatically generated road sections are an accurate representation of the real-world equivalent. Alternatively, the user may manually trace the road network over an image of a map. When this approach is adopted, however, the choice of the appropriate scale is of primary importance,

so as to ensure that road segments are of a realistic length. Road networks may be constructed by dragging and connecting the various space mark-up elements (roads, intersections, bus stops, parking lots and stop lines) using the AnyLogic Road Traffic Library [5]. Once the size and layout of the road network have been created, the user may assign specific attributes to the individual components which form part of the space mark-up of the road network.

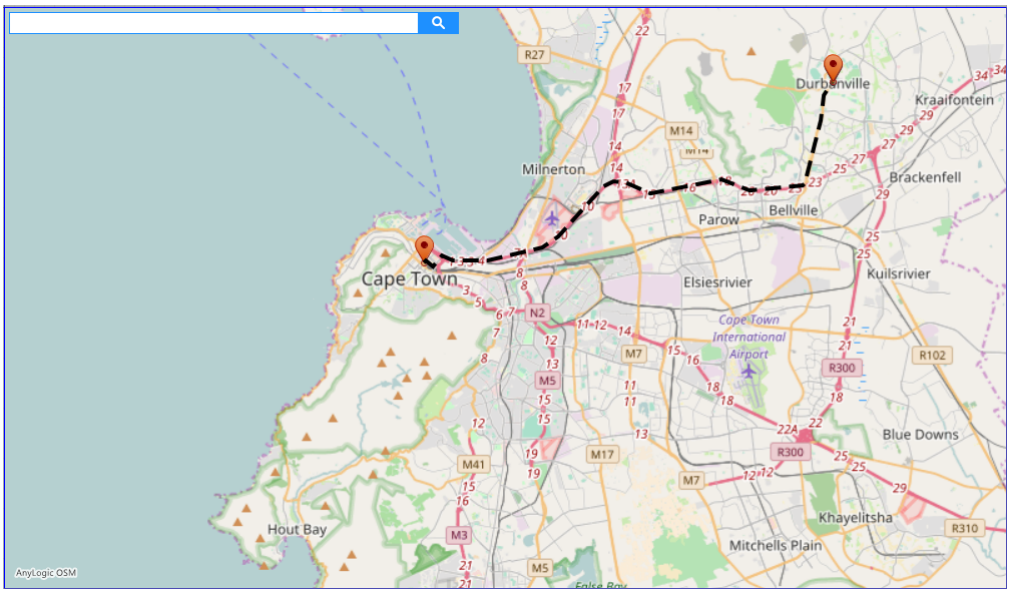


FIGURE 5.1: *A screenshot illustrating the GIS routing capabilities within AnyLogic, with an automatically generated route (indicated by the black, dashed line) between Cape Town and Durbanville in the Western Cape province of South Africa.*

Roads are arguably the most important components of the space mark-up of a road network. Within the AnyLogic Road Traffic Library, such roads may comprise straight or curved segments and possess a number of properties as specified by the user, including whether the road is a one-way or two-way road, the number of lanes in the "forward" direction as well as the number of lanes in the "backward" direction. Lanes in opposite directions are separated using a so-called lane divider of a user-specified width. Roads also allow the user to access the number of vehicles, as well as a list of the individual vehicles travelling on a specific road section at any point in time during execution of the simulation model. Through the use of this list, attributes specific to the vehicles may then be accessed and varied. Certain properties are applicable not only to individual roads, but also to the entire road network. These properties are the traffic flow direction, lane width and the road appearance in the visualisation animation of the simulation model.

Intersections are employed to connect various sections of road to one another. This may include intersections that control traffic flows from multiple directions, the gradual increase from an n-lane road into a m-lane road where $m > n$, or the gradual decrease from an n-lane road to an m-lane road where $m < n$. The movement of vehicles through an intersection is governed by so-called lane connectors which specify paths which may be followed by the vehicles as they travel through the intersection.

Stop lines are another method of controlling traffic flow within the simulation model. Stop lines may be placed at any location along a section of road. These entities may be employed in order to introduce road signs, thereby enforcing traffic rules such as the indication of a stop street, a speed limit, the end of the scope of a speed limit, or a yield sign. Finally, stop lines may be used

for the facilitation of the execution of a specific portion of code which is to be executed every time a vehicle passes over the stop line.

5.1.2 The Benchmark Model

In order to demonstrate the working of the process of reinforcement learning as applied to the highway traffic control problem, and to evaluate the performance of the policies proposed by a reinforcement learning agent, a simple benchmark simulation network is introduced in this section. This benchmark network consists of a hypothetical highway section following the general layout shown in Figure 5.2.

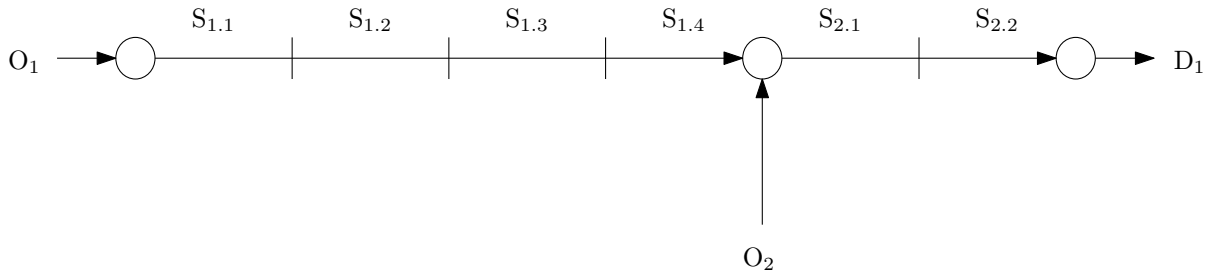


FIGURE 5.2: *The benchmark highway network considered in this study.*

As may be seen in the figure, the network has two demand nodes, denoted by O_1 and O_2, which occur in the mainline and at a single on-ramp, respectively. The stretch of highway before the on-ramp consists of four sections, denoted by $S_{1.1}$–$S_{1.4}$ which are all 1 km in length. After the on-ramp there are two further 1 km sections of highway, denoted by $S_{2.1}$ and $S_{2.2}$, which lead to a single destination node, denoted by D_1. All highway sections have two lanes in the forward direction, while the on-ramp has only a single lane joining into the highway stream.

A more detailed representation of the on-ramp implementation in the benchmark network is given in Figure 5.3. As may be seen in the figure, the vehicles entering the main stream from the on-ramp are given a lateral lane space of 110 metres in order to join the traffic flow on the highway. `StopLine1` and `StopLine3`, positioned as indicated in the figure, are used to display speed limits, while `StopLine2` is used for the placement of a traffic signal in the case where ramp metering is applied. `StopLine4` is employed so as to display a warning sign regarding the lane merge ahead.

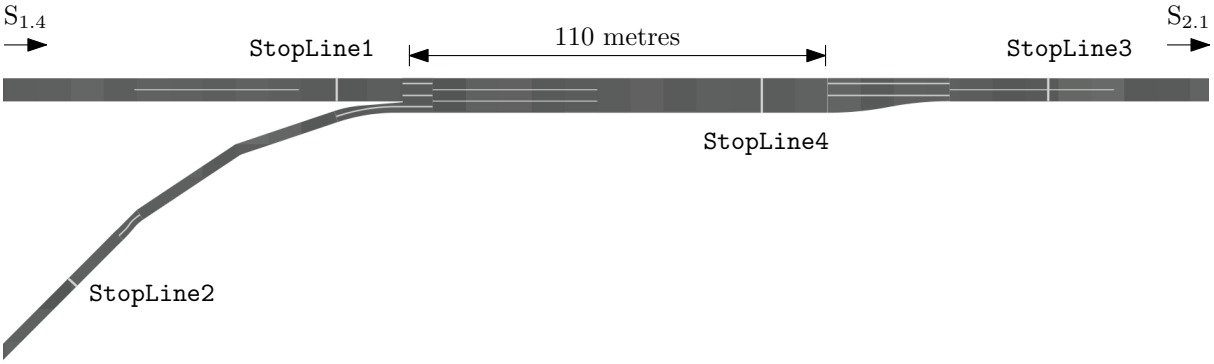


FIGURE 5.3: *A screenshot from the simulation environment showing the highway intersection as the on-ramp joins the highway in the benchmark network. The direction of travel is from left to right.*

The lane connectors, indicated by the solid white lines in the middle of the lanes within an intersection indicate the path that a vehicle will take when travelling through the intersection. As may be seen in the figure, in the intersection representing the lane merge from the three-lane section to the two-lane section, there is no lane connector from the on-ramp lane to the highway lanes. This forces vehicles to choose a suitable position along the 110 metre stretch at which to join the highway traffic flow, rather than specifying the exact point at which the vehicles should enter the highway traffic flow.

5.1.3 The Generation of Vehicles

Vehicles are generated and removed from a simulation run by means of a number of state chart blocks included in the Process Modelling and Road Traffic Libraries. These blocks include a `source` block, which is used to generate vehicles, a `queue` block which acts as a buffer in the case where vehicles that have been generated have to wait before entering the simulated road network (such as when congestion spill back reaches past the boundaries of the simulated environment), a `carEnterRoadNetwork` block, where vehicle attributes are specified, a `carMoveTo` block which is used to define the destination of vehicles, and finally a `carDispose` block which removes vehicles from the simulation once they have reached their destination. An example of such a configuration, specifically for the benchmark network described in §5.1.2, is given in Figure 5.4.

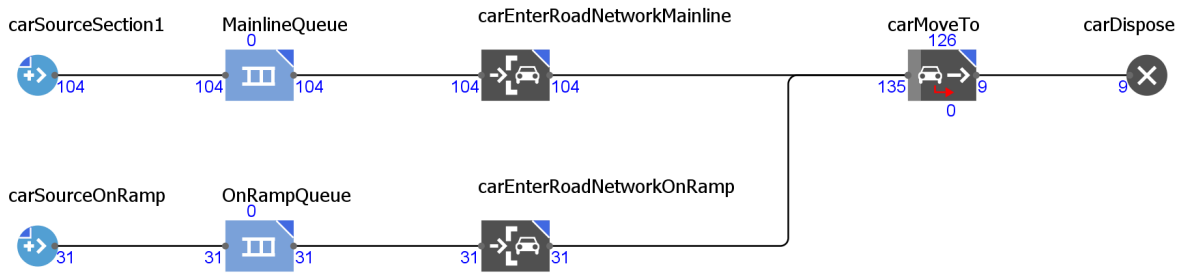


FIGURE 5.4: *A number of connected blocks in the simulation model for the benchmark network, indicating that 104 vehicles have been generated at O_1, none of which are waiting in a queue to enter the simulated road network. Similarly, 31 vehicles have been generated at O_2 which have, again, all entered the road network. A total of 126 vehicles remain within the simulation, while 9 vehicles have reached their destination and have thus been removed from the network.*

In the situation where the entry point to the road network has multiple lanes in the forward direction (such as at the demand point O_1), the vehicles generated may enter the network in either a user-defined or randomly allocated lane. If, however, the entry road consists of a single lane, the vehicle will appear in the single lane once it enters the road network. Vehicle generation may be performed in a number of different ways. Vehicles may be generated according to an arrival rate following a Poisson distribution with an input mean corresponding to the desired traffic volume. Alternatively, the desired vehicle interarrival times may be specified either explicitly or through the use of a suitable probability distribution (*e.g.* an exponential distribution, a normal distribution or a uniform distribution). Finally, vehicle arrivals may take place according to a deterministic arrival schedule in which case the vehicles are generated at exact times following a user-specified schedule, or vehicles may be generated by calling a so-called vehicle inject function.

Once a vehicle has been generated, and there is sufficient space available on the road network at the arrival location to accommodate the vehicle, a number of attributes are simultaneously assigned to it. Among these attributes are its length, initial speed, preferred speed, maximum

acceleration and deceleration, as well as its entry point in the simulation model. These attributes are assigned to the vehicle when it passes through the `carEnterRoadNetwork` block. If, however, sufficient space to enter the road network is not available, the vehicle waits in the `queue` block until space becomes available at which point the vehicle may enter the road network. The destination of the vehicle is only assigned to it once it reaches the `carMoveTo` block.

In the simulation environment, all vehicles obey all traffic laws. As a result, the model is unable to account for vehicles that perform illegal manoeuvres such as running red signals or exceeding the imposed speed limit. Furthermore, vehicles maintain a suitable following distance which is stochastically calculated based on the vehicles' deceleration abilities. The vehicle following distance is, however, always of such a magnitude that if both vehicles were to decelerate at the maximum deceleration rate, a collision of the vehicles would be avoided. The gaps between stationary vehicles are uniformly distributed distances ranging from 1 to 3 metres in length.

5.1.4 Model Output Data

Performance data recorded throughout the execution of each simulation run are saved and written to an excel file at the end of each simulation run. These data may be partitioned into three major classes of *performance measure indicators* (PMIs), based on which the relative performance of the different control policies, as determined by the various reinforcement learning algorithms, may be evaluated.

The first of these PMIs is the *total time spent in the system by the vehicles* (TTS), which is simply the sum total of the times spent in the system by all vehicles. This PMI is then broken down into two further PMIs, namely the *total time spent in the system by vehicles travelling along the highway* (TTSHW) only, and the *total time spent in the system by vehicles that join the network from the on-ramp* (TTSOR). The reason for this breakdown is that it is expected that there may be an increase in the total time spent in the system by vehicles that join the network from the on-ramp due to ramp metering, which may not be reflected sufficiently in the single total time in the system measure.

The second of the PMI classes is the mean vehicle travel time. This is again broken down into the *mean travel time of vehicles travelling along the highway* (TISHW) only, and the *mean travel time of vehicles joining the highway from the on-ramp* (TISOR). During the data collection process, the maximum travel time achieved by a vehicle travelling along the highway only, as well as the maximum travel time of a vehicle joining the highway from the on-ramp, is also recorded. This is due to the fact that road users may not only be interested in how long it would take them to travel the same distance on average, but also what their travel time would be in a worst-case scenario. These values constitute the third PMI class.

For all of the aforementioned output data generated by the simulation model, further information is also recorded in addition to the explicit values taken as PMIs. These include the corresponding minimum values, maximum values, standard deviations and confidence intervals, as well as the number of sample points included in these calculations.

5.2 Model Verification and Validation

This section is devoted to the description of the application of some of the verification and validation techniques reviewed in §4.4 which were applied to the simulation model for the benchmark

network described in §5.1.2. These techniques were not applied only once, but throughout the entire model building process.

5.2.1 Verification of the Traffic Simulation Model

Simulation model verification is essential in ensuring that the simulation model performs as expected. This includes ensuring that the model is free of programmed and logical errors. While a wide variety of verification techniques exist, only the major methods employed in this study are described in this section.

In order to facilitate efficient debugging of programming and logical errors, AnyLogic [5] has a built-in *interactive run controller* (IRC) and a debugger. Upon model compilation, the debugger searches through the models' source code and reports any errors detected. If an error is found, the model may not be executed, and the user is given a description of the error. Furthermore, the location of the error within the source code is specified and possible explanations as to the cause of the error are provided. In the case where the debugger does not find any errors, the model source code is compiled successfully, and the simulation model may be executed.

During the execution of the simulation model, two types of runtime errors may occur: Java exceptions or simulation errors. Java exceptions are caused by computational errors within the Java code (such as division by zero or attempting to access a null pointer), while simulation errors result from erroneous programmed logic. In the case of a Java exception, the simulation run is terminated, and the user is pointed to the portion of source code which caused the error. An example of a simulation error is when a vehicle is generated at a specific location, and there exists no route to its specified destination. An example of a simulation error caused by the unavailability of a route to the user-specified destination, as explained above, is shown in Figure 5.5.

For the verification of the algorithmic implementations of the reinforcement learning algorithms reviewed in Chapter 2, variable tracing and print statements were employed. Variables may be set to be "visible" during the execution of the simulation model, allowing their values to be constantly monitored visually throughout a simulation run. For the ramp metering implementation this was useful so as to monitor whether the changes in red phase duration were correctly applied. Similarly, print statements were central to ensuring the correct performance of the reinforcement learning algorithms, specifically when determining the k nearest neighbours and their corresponding weights in the kNN-TD algorithm. In this case, the centre values of the neighbours, the Euclidean distances from these centres to the current point, as well as the corresponding weights were printed out so as to facilitate manual verification. Print statements were also central in ensuring that the action-value update rule was followed correctly, as outlined in the respective algorithms.

5.2.2 Validation of the Traffic Simulation Model

Successful model validation implies that the simulation model is an accurate representation of the underlying real-world system modelled. Two of the techniques described in §4.4.2 were employed in this study to ensure that the simulation model design and implementation are valid.

The first of these methods involved performing a sensitivity analysis in which input parameters which are central to the model performance were altered, after which it was ensured that the expected variations are represented in the model output. An example of this was varying the arrival rate and observing the effect that this variation has on vehicle occupation of the road

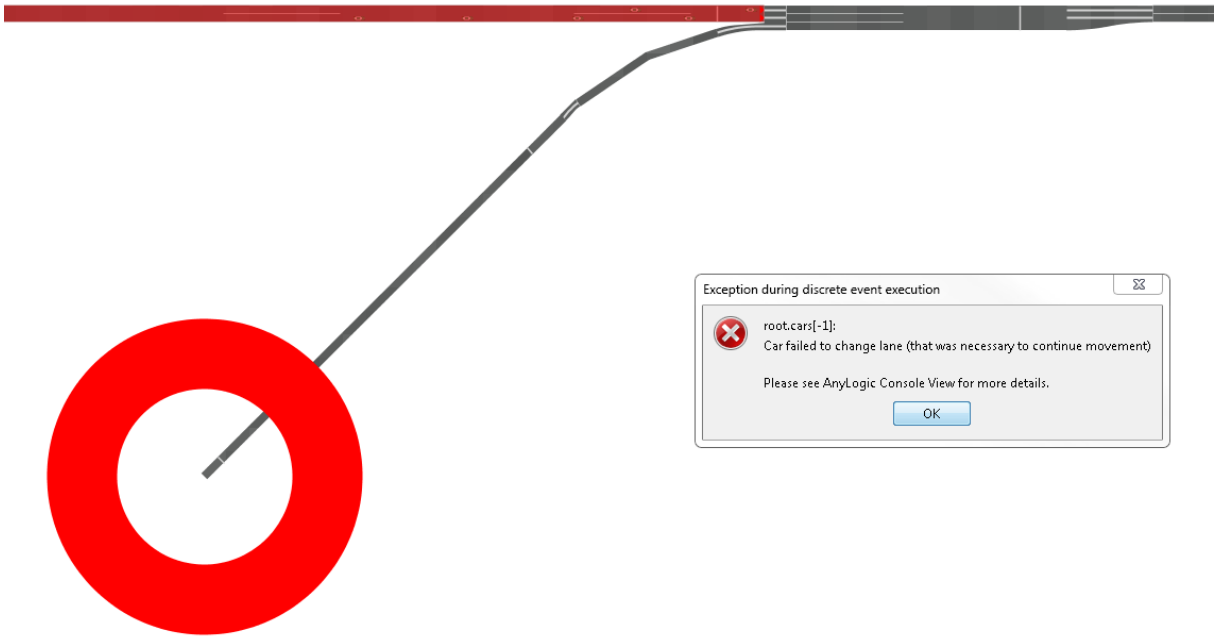


FIGURE 5.5: *A vehicle is generated and expected to enter the road network at the on-ramp, as indicated by the red circle. The exit road is, however, specified to be $S_{1.4}$, as indicated by the red road segment. Naturally, no path exists for the vehicle to reach this section of road, resulting in the error message. This example displays a user input error, as the exit should have been specified as $S_{2.2}$.*

network. In the scenario where the arrival rate of vehicles entering the highway at demand node O_1 is increased, and all other parameters are kept constant, it is expected that the level of congestion should increase, and that the resulting total, as well as the individual, travel times of the vehicles travelling along the highway only should increase. If, on the other hand, the arrival rate of vehicles entering the highway at demand node O_1 is lowered, the converse is expected to occur. Similarly, if the arrival rate of vehicles entering the highway from the on-ramp is increased, congestion on the highway is expected to increase as more vehicles enter the highway stream from the on-ramp, obstructing the traffic flow along the highway. This should again result in increased total and individual travel times, especially for vehicles travelling along the highway only. A number of simulation runs were performed in order to verify these expectations.

The second validation method involved an analysis and interpretation of the simulation model results. Due to the stochastic nature of the simulation model, the output data are bound to contain a certain degree of randomness influenced by the stochasticity of the arrival rates, as well as the vehicle speeds and randomness inherent within the reinforcement learning algorithms. Furthermore, significant variation in the output data may be the result of specific vehicle actions resulting in shockwave formation and propagation along the highway. As a result, one may expect differences in model output data from run to run. The variance contained within these data should, however, not be of an undue extent. The resulting variance was thus analysed in order to ensure that the discrepancies between the output data from the various simulation runs were of acceptable magnitude. Finally, the results were also analysed to ensure that the values made physical and logical sense. It was, for example, ensured that the vehicle travel times and delays were not unlikely values (*i.e.* negative or unacceptably large values).

5.3 Experimental Design

This section is devoted to a discussion on various aspects pertaining to the experimental design according to which the algorithmic comparison of some of the algorithms discussed in Chapters 2 and 3 are performed later in this dissertation. This includes the determination of a suitable simulation warm-up period, as well as some of the general specifications pertaining to the road network simulated, such as vehicle and road attributes. Finally, the types of statistical analysis to be performed in respect of the simulation output data collected from the various simulation runs are described.

5.3.1 The Simulation Warm-up Period

At commencement of the simulation model execution, there are initially no vehicles present in the road network. As vehicles are generated at the source nodes, and vehicles begin to travel through the road network, the number of vehicles present in the road network gradually increases until a steady state is reached. The recording of vehicle travel times and delays during this initial period may potentially yield misleading results, due to the lower traffic demand implied by the comparatively small number of vehicles present in the network. For this reason it is necessary to determine a simulation warm-up period of a suitable length, which is long enough to ensure consistency in the recorded results, yet short enough in order to avoid wasted computation time during model execution.

In order to determine a suitable length of this warm-up period, the method described by Law and Kelton [81] is employed in this dissertation.

Let $Y_1, Y_2, Y_3, \ldots$ denote observations of the number of vehicles present in the network at discrete points $1, 2, 3, \ldots$ in time, respectively. The steady state mean $\overline{m}$ of the number of vehicles Y in the network may then be determined as

$$\overline{m} = \lim_{i \to \infty} E(Y_i). \tag{5.1}$$

As stated by Law and Kelton [81], the relationship in (5.1) does not, however, hold during the finite initial warm-up period x (*i.e.* $E[\overline{Y}(x)] \neq \overline{m}$ during this period). It is therefore suggested that a warm-up period $[1, x^*]$ is introduced, and that all observations made during this period are to be disregarded. A better estimation of $\overline{m}$ is thus given by

$$\overline{Y}(x, x^*) = \frac{\sum_{i=x^*+1}^{x} Y_i}{x - x^*} \tag{5.2}$$

as opposed to $\overline{Y}(x) = \frac{\sum_{i=1}^{x} Y_i}{x}$. In order to determine a suitable length for this warm-up period $[1, x^*]$, Law and Kelton [81] suggest the following four step procedure:

1. Run the model ω times, each for a length of x time units. The resulting output Y_{ij} represents the i^{th} observation from the j^{th} model run, for $i = 1, 2, \ldots, x$ and $j = 1, 2, \ldots, \omega$.

2. Calculate the average of the observations by dividing their sum by the number of simulation runs, *i.e.* $\overline{Y}_i = \sum_{j=1}^{\omega} Y_{ij}/\omega$ for $i = 1, 2, \ldots, x$. The average value $\overline{Y}_i$ has mean $E(\overline{Y}_i) = E(Y_i)$ and variance $\mathrm{Var}(\overline{Y}_i) = \mathrm{Var}(Y_i)/\omega$ for all $i = 1, 2, \ldots, x$.

3. Calculate a moving average using a window over the averaged processes $\overline{Y}_1, \overline{Y}_2, \ldots, \overline{Y}_x$ in order to remove high frequency oscillations. The moving average is determined according

to

$$\overline{Y}_i(y) = \begin{cases} \frac{\sum_{s=-y}^{y} \overline{Y}_{i+s}}{2y+1}, & \text{if } i = y+1, \ldots, x-y \\ \frac{\sum_{s=(i-1)}^{i-1} \overline{Y}_{i+s}}{2i-1} & \text{if } i = 1, \ldots, y, \end{cases} \tag{5.3}$$

where y is the size of the moving average window and is selected such that $0 < y \leq x/4$.

4. The warm-up period x^* is then chosen as the value of i for which mean values $\overline{Y}_i(y)$, $\overline{Y}_{i+1}(y), \ldots, \overline{Y}_{x-y}(y)$ have converged to a constant value.

For the determination of the length of the warm-up period of the simulation model described in §5.1, the value of ω was chosen to be 30 replications as it is expected that this value will give an accurate indication of the steady state of the system. Each iteration was run for 1 800 seconds, and observations regarding the number of vehicles present in the system were made every second, resulting in 1 800 observations for each simulation run. It was found that for the initial traffic flows of 2 000 vehicles per hour at O_1 and 250 vehicles per hour at O_2, a warm-up period of 200 seconds is sufficient. A graph depicting the convergence to the steady traffic state at these initial traffic conditions is shown in Figure 5.6.

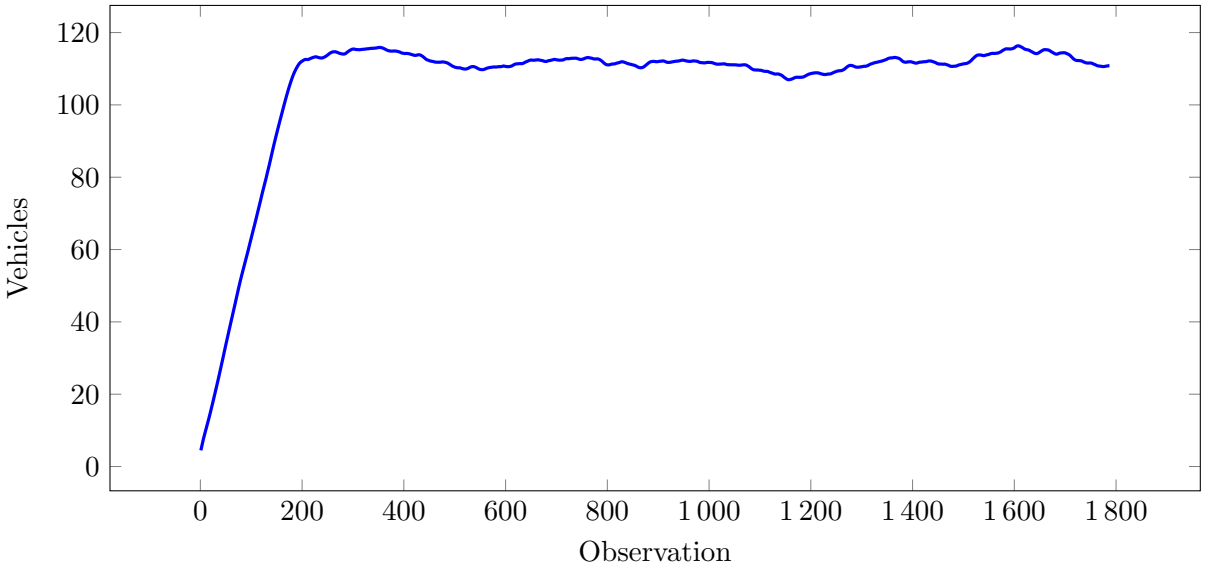


FIGURE 5.6: *An indication of the warm-up time under the initial free-flow traffic conditions. The warm-up time is approximately 200 seconds, while at the steady state there are approximately 110 vehicles present in the network.*

5.3.2 General Specifications of the Simulation Framework

In the simulation model of §5.1, vehicle arrivals are determined according to exponentially distributed interarrival times with rate parameter $\lambda = 1/\mu$, where μ denotes the mean. This mean, usually given in veh/s, may be calculated by simply dividing the desired hourly traffic volume by 3 600. The reinforcement learning algorithms are to be evaluated in four different scenarios of varying traffic demand, as shown in Figure 5.7. A rush hour is imitated in each of the scenarios, initially accommodating free-flowing traffic due to low demand. This is followed by a 30-minute period of steady increase in demand, until the demand reaches a peak, after which it remains constant for an hour. Thereafter, the demand decreases steadily back to the free-flow demand over a 30-minute period. Finally, in order to account for congestion which

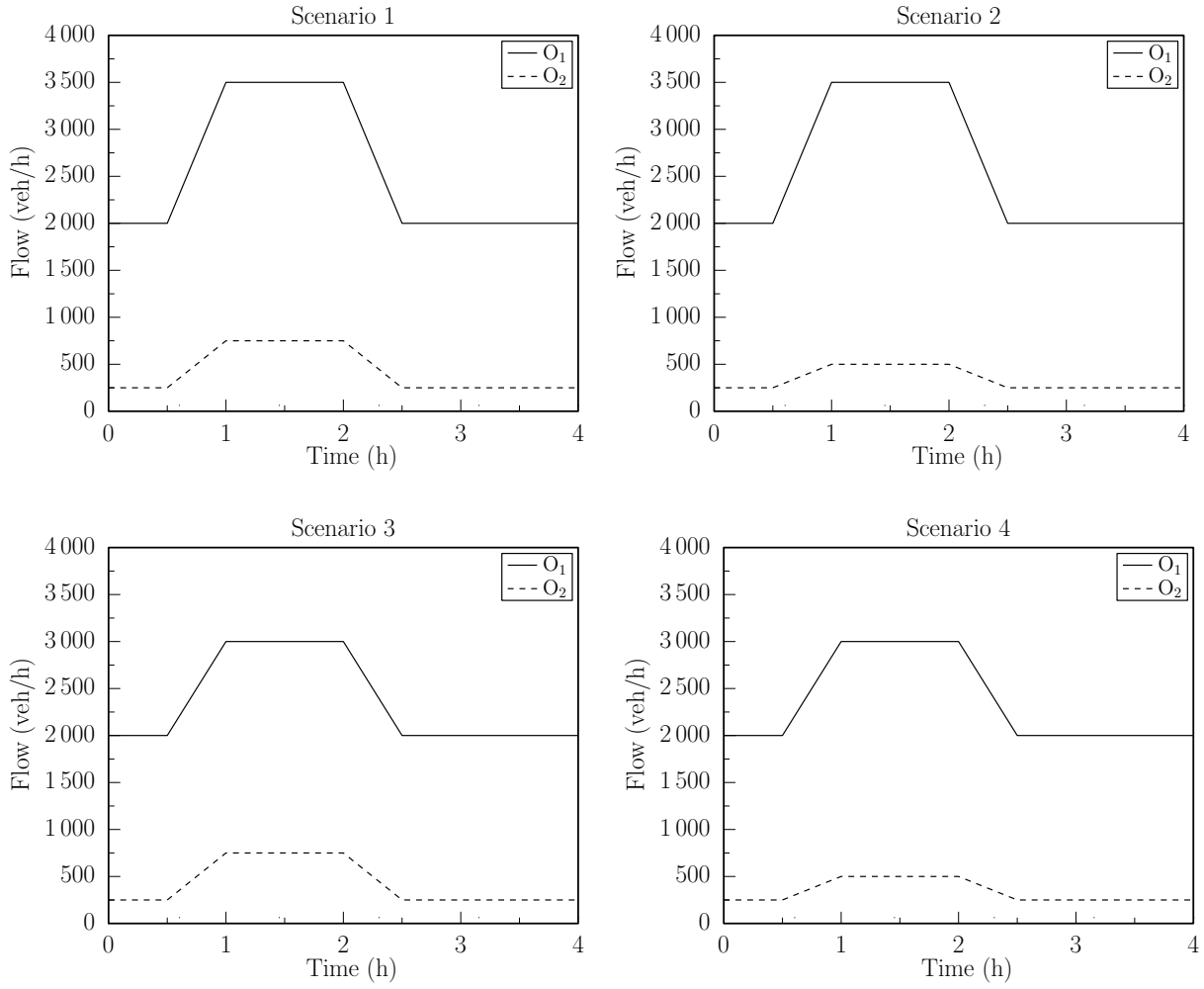


FIGURE 5.7: *The four scenarios of varying traffic demand at the origins O_1 and O_2 of the benchmark network considered in this dissertation.*

has built up over the peak demand period, a period of 90 minutes is allowed at the end of each experiment run for the system to once again reach the initial free-flow traffic conditions. As may be seen in Figure 5.7, the heaviest traffic demand is generated in Scenario 1, with the demand at O_1 peaking at 3 500 vehicles per hour, and the demand at O_2 peaking at 750 vehicles per hour. In Scenario 2, the demand at O_1 remains unchanged, while the peak demand at O_2 is reduced to 500 vehicles per hour. Scenario 3 has a reduced peak demand of 3 000 vehicles per hour at O_1, while the on-ramp demand at O_2 remains as in Scenario 1. Finally, demands at both O_1 and O_2 are lowered in Scenario 4, with peaks of 3 000 vehicles per hour and 500 vehicles per hour, respectively.

For the purposes of this study, default values for the vehicle properties, as suggested in the AnyLogic Road Traffic Library [5], are employed in order to demonstrate the working of the algorithms in the aforementioned hypothetical scenarios. All vehicle lengths are thus fixed at 5 metres, while the initial speeds of the vehicles entering the highway and the on-ramp are set to 120 km/h and 60 km/h, respectively. In order to account for variation in driver aggressiveness, the preferred speed of the vehicles is uniformly distributed between 110 km/h and 130 km/h, while the maximum acceleration and deceleration values are taken as 1.8 m/s^2 and -4.2 m/s^2, respectively.

5.3.3 Types of Statistical Analysis to be Performed on Model Output Data

For each scenario in which the reinforcement learning algorithms described in Chapter 2 are compared later in this dissertation, differences in the output data which are reported to be significantly different are so at a 5% level of significance. In order to determine whether there are, in fact, statistically significant differences in the algorithmic means of the PMIs described in §5.1.4, an *analysis of variance* (ANOVA) [104] is carried out. The ANOVA, however, only indicates whether there is at least one significant difference between two means. As a result, *post hoc* tests are required in order to determine *where* this difference actually occurs. Unfortunately, most *post hoc* tests assume homogeneity of sample variances, which is not necessarily the case in the PMI data of §5.1.4.

The two *post hoc* tests employed in this dissertation for determining where the differences in the algorithmic means of the PMIs lie are *Fisher's Least Significant Difference* (LSD) test [174] and the *Games-Howell* test [39]. After an ANOVA has been performed and a significant difference has been identified between the means of two samples, a *Levene* test [146] is carried out in order to determine whether or not the corresponding variances differ significantly from one another. In the case where the variances are found not to differ statistically from one another at a 5% level of significance, the LSD test (which requires homogeneity of sample variances) is employed in order to identify the location of the differences in the PMIs. If, however, the Levene test reveals that the variances are, in fact, statistically different at a 5% level of significance, the Games-Howell test (which does not require homogeneity of sample variances) is employed in order to determine where these differences lie.

The working of the ANOVA, Levene, LSD and Games-Howell statistical tests are reviewed briefly in this section. In all the tests described below, n samples are to be compared, each containing m observations. Denote the i-th sample by $x_1^{(i)}, \ldots, x_m^{(i)}$ for all $i \in \{1, \ldots, n\}$. Furthermore, let the mean of the i-th sample be denoted by $\bar{x}^{(i)}$ and let s_i denote the sample's standard deviation. Finally, let $\bar{x}$ denote the mean of all sample means $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$.

In all of the above-mentioned statistical tests it is assumed that the data are normally distributed. According to the central limit theorem, even if samples are taken from an unknown distribution, the distribution of the sample mean will still be approximately normally distributed, if the sample size m is sufficiently large [104]. This implies that although the underlying probability distributions of the PMIs of §5.1.4 may be unknown, the sample means may be considered to be approximately normally distributed and, as a result, the requirement of the statistical tests that the data are normally distributed, is not violated, if m is large.

In order to determine a suitable sample size m, the technique outlined by Lindley [87] is employed. Initially, a sample of m PMI values is generated from simulation runs. An estimate of the standard deviation of the sample PMI values may be calculated as

$$S_x = \sqrt{\frac{\sum_{i=1}^{m}(x - \bar{x}^{(i)})^2}{m-1}}. \tag{5.4}$$

Thereafter, a *confidence interval* (CI) around the true mean may be determined. In order to determine an accurate CI, the studentised range distribution[1] [104, Appendix A] is employed. The CI half-width h, at a $(1-\alpha)$-level of confidence, is given by

$$h_{1-\alpha} = t_{(1-\alpha/2),(m-1)} \frac{S_x}{\sqrt{m}}, \tag{5.5}$$

[1] A distribution which may be used for the estimation of the range of a normally distributed population in the case where the standard deviation of the population is unknown and the population may be considered to be small.

where $t_{(1-\alpha/2),(m-1)}$ is the critical value for a two-sided error summing to α with $m-1$ degrees of freedom. If the half-width h is judged to be too wide, the required number of replications in order to achieve a desired half-with h^* is given by

$$m^* = m\left(\frac{h}{h^*}\right)^2. \tag{5.6}$$

For the purposes of this study, a CI half-width h not exceeding a value greater than 5% of the sample mean, as suggested by Lindley [87], is deemed sufficiently accurate. This procedure is repeated for each PMI, after which the largest m^*-value is chosen.

The ANOVA test

The null-hypothesis H_0 to be tested when performing an ANOVA may be formulated as *there are no statistically significant differences between the means of any of the samples*. It follows that the alternative hypothesis H_1 is that *there are significant differences between at least two of the sample means*. In the ANOVA test, both the sum of squares of observations *within* samples and the sum of squares *between* sets of the sample data are used in order to test the null-hypothesis. The sum of squares of observations within samples is calculated as

$$S_w = \frac{1}{mn-n}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_j^{(i)} - \bar{x}^{(i)})^2. \tag{5.7}$$

Similarly, the sum of squares between sets of data is given by

$$S_b = \frac{m}{n-1}\sum_{i=1}^{n}(\bar{x}^{(i)} - \bar{x})^2. \tag{5.8}$$

The test statistic is given by the ratio S_b/S_w. This test statistic is compared with the critical value $F(d_1, d_2, \alpha)$ of the F-distribution, where $d_1 = n-1$ denotes the number of degrees of freedom of the numerator, $d_2 = mn-n$ denotes the number of degrees of freedom of the denominator, and α denotes the level of statistical significance. The value of $F(n-1, mn-n, \alpha)$ may be found in [104, Appendix A]. In the case where

$$S_b/S_w > F(n-1, mn-n, \alpha), \tag{5.9}$$

the null-hypothesis H_0 is rejected at an α-level of significance. This implies that there are, in fact, significant differences between the means of at least two samples at a $(1-\alpha)$-level of confidence. Alternatively, if the inequality in (5.9) does not hold, it may be concluded that no statistically significant differences exist between the sample means at a $(1-\alpha)$-level of confidence.

The Levene test

The Levene test is used in order to assess whether the variances of two or more data sets are statistically different at an α-level of significance. This is encapsulated in the null-hypothesis H_0 that *there are no statistically significant differences between the variances of any of the original samples*, while the alternative hypothesis H_1 becomes *there are statistically significant differences between at least two of the original sample variances*. In order to perform the test, two variables have to be determined. The first of these values, the test statistic F_L is calculated as

$$F_L = \frac{(mn-n)\sum_{i=1}^{n} m(\bar{x}^{(i)} - \bar{x})^2}{(n-1)\sum_{i=1}^{n}\sum_{j=1}^{m}(|x_j^{(i)} - \bar{x}^{(i)}| - \bar{x}^{(i)})^2}. \tag{5.10}$$

The critical value $F(n-1, mn-n, \alpha)$ is again obtained from the F-distribution table. If

$$F_L > F(n-1, mn-n, \alpha), \tag{5.11}$$

then the null-hypothesis is rejected which implies that variances between at least two of the data sets are statistically different at a $(1-\alpha)$-level of confidence and the Games-Howell test is subsequently performed in respect of each pair of samples. If, however, the inequality in (5.11) does not hold, it may be concluded there are no statistical differences between the variances at a $(1-\alpha)$-level of confidence, and the LSD test is performed.

The Fisher LSD *post hoc* test

The Fisher LSD *post hoc* test has proven to be a powerful parametric statistical test. Criticism has, however, been offered due to the belief that its protection against inflated Type I error[2] rates is insufficient, although this has only been proven to be the case when more than three data sets are being compared [51].

The null-hypothesis H_0 for Fisher's LSD test is that *there is no statistically significant difference between the means* $\bar{x}^{(k)}$ *and* $\bar{x}^{(\ell)}$ *of two samples.* The test statistic of the LSD test is given by $|\bar{x}^{(k)} - \bar{x}^{(\ell)}|$, while the critical value at a level of significance α is

$$L_\alpha = t_{\alpha/2,d_2}\sqrt{2S_w/m}, \tag{5.12}$$

where $t_{\alpha/2,d_2}$ denotes the entry in the table corresponding to the two-sided t-distribution [104, Appendix A] at a significance level of α with $d_2 = mn - n$ degrees of freedom and where S_w is the value of the sum of squares within samples, as determined in (5.7).

If $|\bar{x}^{(k)} - \bar{x}^{(\ell)}| > L_\alpha$, the null-hypothesis is rejected at a level of confidence $1-\alpha$ (*i.e.* there is a statistical difference between the means $\bar{x}^{(k)}$ and $\bar{x}^{(\ell)}$ at an α-level of significance). Otherwise, the means may not be considered different at an α-level of significance. This procedure has to be repeated for all $\binom{n}{2}$ pairs of samples. When performing the Fisher LSD *post hoc* test it is important to keep the *practical significance*[3] as well as the statistical significance of the results in mind.

The Games-Howell test

The Games-Howell *post hoc* test [59, 60] is a non-parametric test recommended for use in cases with unequal sample sizes or if the assumption of homogeneity of variances required for Fisher's LSD test is violated [35]. According to Armstrong and Hilton [6], the Games-Howell *post hoc* test is one of the most robust modern methods of *post hoc* testing. Furthermore, it is said to be a more conservative test than the majority of other *post hoc* tests [6]. The critical value required for the test employs Welch's degrees of freedom (from Welch's t-test[4]) and the studentised range

[2] A Type I error is the error of rejecting a null-hypothesis when it is actually true.

[3] Practical significance refers to the evaluation of whether statistically significant differences are large enough to be relevant in a practical sense. As an example, consider the mean travel times of vehicles returned after the implementation of the policies as suggested by two reinforcement learning agents. Now assume that after a number of simulation runs, these means have been found to be statistically significantly different although they only differ by 0.5 seconds. While it may have been proven that these means are different from a statistical perspective, it is clear that this difference is negligible in a practical sense.

[4] Welch's t-test is a two-sample location test used for determining whether the means of two different populations are equal. In this test, homogeneity of variance is not assumed, but normality of data is assumed.

distribution [104, Appendix A], and is denoted by $q_{\sigma(k,\ell),d(k,\ell),\alpha}$, where

$$\sigma(k,\ell) = \sqrt{\frac{s_k^2 + s_\ell^2}{2m}} \tag{5.13}$$

is the standard error, $d(k,\ell)$ denotes the degrees of freedom, calculated here as

$$d(k,\ell) = \frac{m-1}{\left(s_k^2/m\right)^2 + \left(s_\ell^2/m\right)^2} \left(\frac{s_k^2 + s_\ell^2}{m}\right)^2 \tag{5.14}$$

and α is again the level of statistical significance. If $|\bar{x}^{(k)} - \bar{x}^{(\ell)}| > q_{\sigma(k,\ell),d(k,\ell),\alpha}$, then there is a statistical difference between the means of the two samples at an α-level of significance and the null-hypothesis is rejected. If, on the other hand, this inequality does not hold, then the means of the two samples do not differ at an α-level of significance and the null-hypothesis may not be rejected at a $(1-\alpha)$-level of confidence.

P-values in Hypothesis Tests

One method of reporting the results of an hypothesis test involves stating whether or not a null-hypothesis should be rejected at a specified level of significance α, and is called *fixed significance level* testing [104]. A so-called *p-value* is employed in fixed significance level testing and denotes the probability that the test statistic will take on a value that is at least as extreme as the observed value in the case that the null-hypothesis is true. In other words, the p-value is the smallest level of significance which would lead to rejection of the null-hypothesis H_0 based on the given data. Consider, for example, the two-sided hypothesis test employed in the Fisher LSD test, where

$$H_0 : |\bar{x}^{(k)} - \bar{x}^{(\ell)}| = 0 \text{ and } H_1 : |\bar{x}^{(k)} - \bar{x}^{(\ell)}| \neq 0 \tag{5.15}$$

are the null and alternative hypotheses, respectively. Then the p-value is given by

$$1 - P\left(-\frac{|\bar{x}^{(k)} - \bar{x}^{(\ell)}|}{\sqrt{2S_w/m}} < t_{\alpha/2,d_2} < \frac{|\bar{x}^{(k)} - \bar{x}^{(\ell)}|}{\sqrt{2S_w/m}}\right). \tag{5.16}$$

Operationally, once the p-value has been computed it is compared with a predefined level of significance α, in order to make a decision. It is then standard practice to report the observed p-value, along with the decision made in respect of rejection of the null-hypothesis. Apart from stating this decision on the null-hypothesis, the p-value provides a measure of credibility of the null-hypothesis. More specifically, the p-value provides a measure of risk that an incorrect decision regarding the null-hypothesis has been made, as the p-value denotes the probability that the null-hypothesis is wrongly rejected [104] (in other words, it is the probability of making a Type I error). The p-values for the ANOVA, Levene and Games-Howell tests may be computed similarly, but using the appropriate probability distributions in each case, as mentioned above.

5.4 Chapter Summary

This chapter opened in §5.1 with a description of the various entities involved in the simulation model building process, culminating in a detailed description of the simple, hypothetical, benchmark highway network which will be used as a test-bed and concept demonstrator for the working of the reinforcement learning algorithms implemented in the following chapters. This

was followed in §5.2 by a description of the verification and validation techniques employed so as to ensure a valid simulation. Finally, an experimental design was described in §5.3, with a specific focus on the simulation warm-up period, as well as some general parameter specifications and the statistical analysis which is to be performed in respect of the simulation model output data.

CHAPTER 6

Reinforcement Learning for Ramp Metering

Contents

The purpose of this chapter is to provide a detailed description of the implementation of RL in the context of RM. The chapter opens in §6.1 with a description of the implementations of the well-known ALINEA and PI-ALINEA RM control strategies within the microscopic traffic benchmark model of §5.1.2. Thereafter, the RM problem is formulated in §6.2 in the context of RL, which then serves as the blueprint for the algorithmic implementations of Q-Learning and the kNN-TD RL algorithms. These algorithmic implementations are presented in §6.3 and §6.4, respectively. This is followed by an algorithmic parameter evaluation in §6.5.1, after which the relative algorithmic performances of the RM techniques are compared in §6.5.2, using suitable algorithmic parameter values. Thereafter, queueing considerations are introduced within each of the RM implementations in §6.6, so as to prevent the formation of excessively long on-ramp queues. A thorough algorithmic performance comparison of the RM implementations incorporating these queueing considerations follows in §6.6.3. The chapter finally closes in §6.7 with a brief summary of the work included in the chapter.

6.1 ALINEA and PI-ALINEA in a Microscopic Context

The ALINEA RM control strategy, widely regarded as the benchmark RM control strategy [130], has been designed for application in a macroscopic traffic modelling environment. As a result, a number of minor adjustments have to be made to the control strategy in order to facilitate its successful application within a microscopic traffic simulation model. According to the ALINEA strategy, the metering rate is adjusted based on the traffic density along the highway directly downstream of the on-ramp. In the macroscopic case, this is achieved simply by adjusting the maximum allowable flow entering the highway from the on-ramp.

As in several real-world applications of RM [52, 130], a one-vehicle-per-green-phase approach is adopted for its microscopic implementation in this dissertation. The flow of vehicles onto the highway from the on-ramp may then be controlled by adjusting the red phase duration of the traffic signal enforcing RM at the on-ramp. Due to the fact that the control law only returns a metering rate, this metering rate is converted to practically implementable red phase time

$$R(t) = \max\left[0, \left(\frac{3600}{r(t)}\right) - G(t)\right], \tag{6.1}$$

where $r(t)$ denotes the metering rate (in veh/h) determined according to (3.17), and $G(t)$ denotes the fixed green phase duration. It is evident that a larger metering rate (*i.e.* allowing more vehicles to enter the highway traffic stream) results in shorter red phase times, while a smaller metering rate restricts the traffic flow allowed to enter the highway by enforcing longer red phase durations.

Due to the fact that the ALINEA control law dates back to 1997, it may be considered outdated, especially considering the large volume of work performed since. Therefore, PI-ALINEA, a more recent extension of the ALINEA control law, first published in 2014, is also implemented as a second benchmark control strategy against which the performance of the RL implementations may be measured. In PI-ALINEA the metering rate is determined according to (3.18) and similarly to ALINEA based on the traffic density directly downstream of the on-ramp. This metering rate is then again converted to red phase times which may be applied in the microscopic traffic simulation model in the same manner as for ALINEA, by means of (6.1).

6.2 Formulation as a Reinforcement Learning Problem

Wen *et al.* [173] have shown that the RM control problem may be formulated as an MDP and, as a result, may be solved using RL algorithms. This section is devoted to a description of the formulation of the RM problem as an RL problem, which serves as the blueprint for the various highway control algorithmic implementations later in this dissertation. The modelling approach adopted here was inspired by the work of Davarynejad *et al.* [29] and Rezaee [130], and is applied to the benchmark model described in Chapter 5. RM is enforced by a single traffic signal placed at an on-ramp, as shown in Figure 6.1.

6.2.1 The State Space

The three principal components that make up the state space are described in this section. These components are illustrated graphically in Figure 6.2. The first state is the density ρ_{ds} directly downstream of the on-ramp. This state has been selected as it provides the agent with direct feedback in respect of the quality of the previous action, because this is the bottleneck

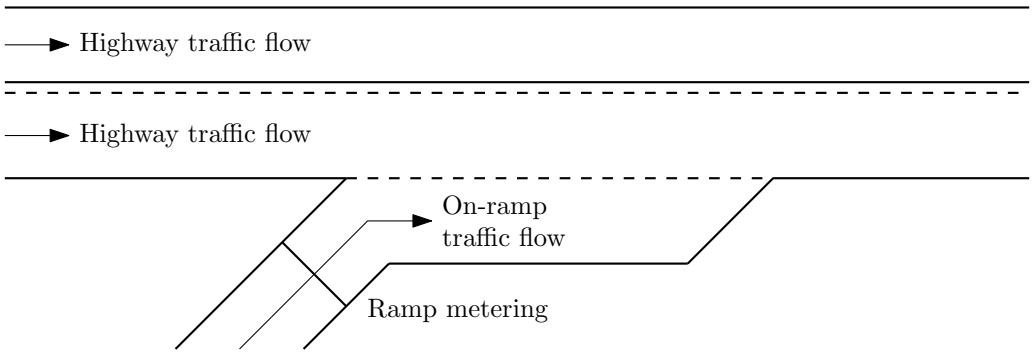


FIGURE 6.1: *The RM implementation adopted within the benchmark model of §5.1.2.*

location and thus the source of congestion. As a result it is expected that the earliest indicator of impending congestion may be the downstream density.

The second state is the density ρ_{us} upstream of the on-ramp. This state has been selected because it provides an indication as to how far the congestion, if any, has propagated backwards along the highway.

Finally, the third state is the on-ramp queue length w. This state is included so as to provide the agent with information on the prevailing traffic conditions along the on-ramp, as well as providing information about the on-ramp demand.

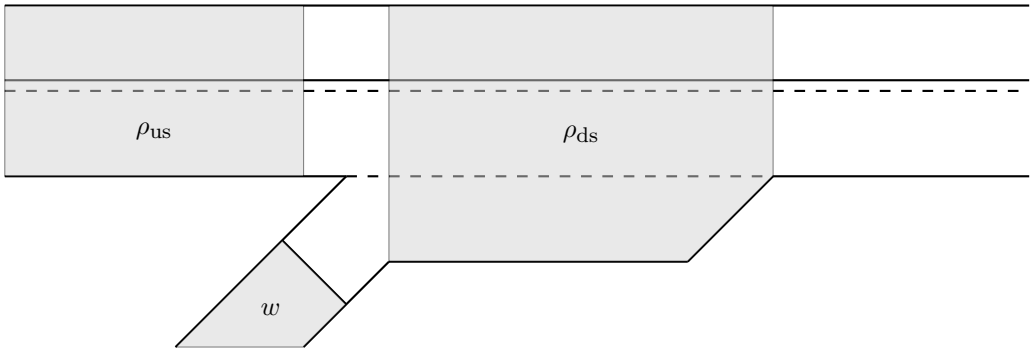


FIGURE 6.2: *A representation of the state space for the RM problem in the context of the benchmark model of §5.1.2.*

6.2.2 The Action Space

In order to improve the state of traffic flow, the learning agent may select a suitable action based on the prevailing traffic conditions. Rezaee [130] showed that the use of a direct action selection policy (*i.e.* selecting a red phase duration directly from a set of pre-specified red times) instead of an incremental action selection policy (*i.e.* adjusting the red phase duration incrementally) yields better results when applied to the RM problem. As a result, a direct action selection policy is adopted for the work presented in this dissertation.

As stated above, red phase times are varied in order to control the flow of vehicles that enter the highway from the on-ramp. Direct action selection then implies that the agent chooses pre-specified red phase durations from the set of actions $\mathcal{A}$. In this case, the actions available to the agent are $a \in \{0, 2, 3, 4, 6, 8, 10, 13\}$, where each action represents a corresponding red phase duration measured in seconds. These red phase durations correspond to the respective on-ramp

flows $q_{\text{OR}} \in \{1\,600, 720, 600, 514, 400, 327, 277, 225\}$ vehicles per hour, assuming a green phase duration of three seconds in each case.

6.2.3 The Reward Function

Typically, the objective when designing a traffic control system is to minimise the combined total travel time spent in the system by all transportation network users. From the fundamental traffic flow diagram (see Figure 3.1) it follows that the maximum throughput, which corresponds to maximum flow, occurs at the critical density [115]. Density is usually the variable that the RM agent aims to control. This is the case in ALINEA, the most celebrated RM technique. As a result of the successful implementation of ALINEA in several studies and real-world applications [113], the reward function adopted in order to provide feedback to the RM agent has been inspired by the ALINEA control law. According to the ALINEA control law, given in (3.17), the metering rate is adjusted based on the difference between the measured density downstream of the on-ramp and a desired downstream density. The reward awarded to the RM agent is calculated as

$$r(t) = -(\hat{\rho} - \rho_{\text{ds}})^2, \tag{6.2}$$

where $\hat{\rho}$ denotes the desired density the RL agent aims to achieve directly downstream of the on-ramp, and ρ_{ds} denotes the measured density downstream of the on-ramp during time interval t, as indicated in Figure 6.2. The difference between the desired and measured densities is squared in order to amplify the effect of large deviations from the desired density, thereby providing amplified negative feedback for actions which result in such large deviations. A portion of this reward function is shown in Figure 6.3.

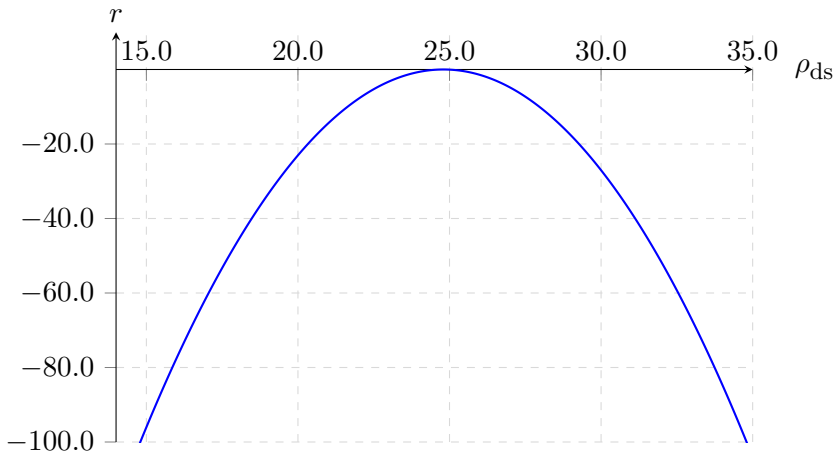


FIGURE 6.3: *The reward function employed for the RM agent in the context of the benchmark model of §5.1.2 with a desired density $\hat{\rho} = 24.8$ veh/km.*

6.2.4 Learning Rate and Action Selection

Watkins and Dayan [169] have shown that Q-Learning suppresses uncertainties and converges to the optimal Q-values if a decreasing learning rate is employed, as long as the sum

$$\sum_{i=1}^{\infty} \alpha_{n^i(s,a)} \tag{6.3}$$

diverges (whether or not the sum

$$\sum_{i=1}^{\infty} \left[\alpha_{n^i(s,a)}\right]^2 \tag{6.4}$$

diverges) for all state-action pairs, where $n^i(s,a)$ denotes the index of the i-th time that the state-action pair (s,a) has been visited. As a result, the learning rate

$$\alpha_{n^i(s,a)} = \left[\frac{1}{1+i(1-\gamma)}\right]^{0.85}, \tag{6.5}$$

which decreases as a function of the number of visits to state-action pairs, is employed in this dissertation, where i denotes the index of the i-th visit to the state-action pair (s,a), and γ denotes the discount factor, as defined in §2.2.2. The discount factor is set to $\gamma = 0.94$, which was found to be near-optimal for traffic applications by Rezaee [130].

As stated in §2.2.2, a trade-off between exploration and exploitation of the state-action space is of primary importance when solving RL problems. In order to achieve a balance between exploration and exploitation, an adaptive ϵ-greedy policy is employed in this dissertation. As with the learning rate above, the adaptive ϵ-value is determined as a function of the number of visits to a state s. This state-dependent ϵ-value is given by

$$\epsilon(s) = \max\left\{0.05, \left[\frac{1}{1+\frac{1}{5}\frac{1}{N_a(s)}\sum_{i=1}^{a} i(s)}\right]\right\}, \tag{6.6}$$

where $N_a(s)$ denotes the number of available actions a when the system is in state s and $i(s)$ denotes the number of visits to state s. Employing such a state-dependent ϵ-value encourages exploration in the case where a state has not yet been visited, but encourages exploitation as the number of visits to the state increases, as the ϵ-value steadily decreases to a minimum value of 0.05. The methods of determining the adaptive learning rate $\alpha_{n^i(s,a)}$ and the state-dependent ϵ-value are based on the work of Rezaee [130], and have been fine-tuned empirically so as to yield the most effective results.

6.3 Q-Learning for Ramp Metering

Due to the guarantee of optimality as a result of the conditions outlined above, as well as its ease of implementation, Q-Learning is selected as one of the RL techniques implemented and evaluated in this dissertation.

Since Q-Learning is a lookup table-based method of RL, the state space has to be discretised so as to facilitate a tabular representation of the state space and the resulting action-value function $Q(s,a)$. The downstream density is therefore discretised into $n_{\rho_{\text{ds}}} = 10$ equi-spaced intervals. In the same way, the upstream density is discretised into $n_{\rho_{\text{us}}} = 10$ equi-spaced intervals. In order to be able to facilitate the introduction of a penalty term for long queue formation on the on-ramp, the on-ramp queue length is discretised into nine intervals, according to

$$n_w = \begin{cases} 2.50 & \text{if } \frac{w}{100} > 2.00, \\ 2.00 & \text{if } \frac{w}{100} > 1.75, \\ 1.75 & \text{if } \frac{w}{100} > 1.50, \\ 1.50 & \text{if } \frac{w}{100} > 1.25, \\ 1.25 & \text{if } \frac{w}{100} > 1.00, \\ 1.00 & \text{if } \frac{w}{100} > 0.75, \\ 0.75 & \text{if } \frac{w}{100} > 0.50, \\ 0.50 & \text{if } \frac{w}{100} > 0.25, \text{ and} \\ 0.25 & \text{if } \frac{w}{100} \geq 0. \end{cases} \tag{6.7}$$

This discretisation results in a state space consisting of $|n_{\rho_{\text{ds}}}| \times |n_{\rho_{\text{us}}}| \times |n_w| = 900$ states. AnyLogic 7.3.5 [5] has a built-in database function which allows a simulation model implemented in this environment to read in and update the database values during a simulation run through the execution of Microsoft SQL Server [101] code which links the simulation model to the built-in database. The lookup table used in Q-Learning in order to approximate the state action value $Q(s, a)$ is implemented within this built-in database, and may thus be updated throughout the simulation model execution using the real-time information on state and action values, as well as the immediate reward. Adopting the state space discretisation and action space described above, Q-Learning (as outlined in Algorithm 2.3) may be implemented within the context of the benchmark model of §5.1.2.

6.4 kNN-TD Learning for Ramp Metering

As stated in §2.3, function approximators extend the applicability and, if implemented correctly, improve the accuracy and learning speeds which may be achieved by RL agents. As a result, kNN-TD learning (as described in Algorithm 2.6) is also implemented in this dissertation.

Due to the fact that maximum vehicle throughput is achieved at the critical traffic density, the centres chosen for both the downstream density and the upstream density should be clustered around the critical density value so as to be able to provide more accurate approximations of the action value when the measured density is close to the critical density. The critical density of highway segments is typically around 28 vehicles/km [130]. As a result, the downstream centres are chosen as $\{15, 22, 25, 27, 29, 33, 38, 45, 55, 70\}$, while the centres for the upstream density are placed at $\{12, 20, 25, 30, 70, 75, 80\}$. Finally, the centres for the on-ramp queue length are chosen as $\{3, 5, 7, 9, 11, 20, 40, 60, 80\}$. The lookup table used for storing and updating the centre-action values is, as in the case of Q-Learning, created using AnyLogic's built-in database functionality.

The learning rate α is determined as in (6.5), except that it is now determined for centre-action pairs rather than for state-action pairs. The calculation of the state-dependent ϵ-value, however, changes slightly to

$$\epsilon(s) = \max\left\{0.05, \left[\frac{1}{1 + \frac{1}{11}\frac{1}{N_a(s)}\sum_{i=1}^{a} C^{k\text{NN}}(s)}\right]\right\}, \tag{6.8}$$

where $C^{k\text{NN}}(s)$ is the estimated number of visits to state s, given by

$$C^{k\text{NN}}(s) = \sum_{i=1}^{k} p(i)C(x, a),$$

where $p(i)$ is the weighted probability linked to each of the k nearest neighbours, as determined in (2.25), and $C(x, a)$ denotes the number of visits to the state-action pair (x, a). The results achieved by ALINEA, Q-Learning and kNN-TD learning are described in the following section.

6.5 Computational Results

In this section, the performance of the ALINEA, PI-ALINEA, Q-Learning and kNN-TD learning are compared in each of the four scenarios of traffic demand of §5.3.2, implemented within the benchmark simulation model described in §5.1.2. Initially, the performance of the various algorithms is fine-tuned by means of algorithmic parameter evaluations in §6.5.1, after which

the algorithmic comparison is performed in §6.5.2 adopting parameter values found in §6.5.1 to perform most satisfactorily.

6.5.1 Parameter Evaluation

This section is devoted to determining good parameter values for the ALINEA and PI-ALINEA control strategies, as well as the Q-Learning and kNN-TD RM implementations (described in §6.1, §6.3 and §6.4, respectively) within the context of the benchmark model of Chapter 5.

ALINEA parameter evaluation

For the ALINEA control strategy, a good combination of two parameter values has to be determined. These combinations consist of a value for K_R, the nonnegative control parameter, as well as a value for the target density $\hat{\rho}$. For the control parameter K_R, three values, which were judged to be of a small, medium and large magnitude, are evaluated. Papageorgiou *et al.* [112] suggested a value $K_R = 70$ veh/h in the macroscopic case. Wang *et al.* [167], however, determined the best-performing K_R-value as 40 veh/h. Furthermore, Papageorigou *et al.* [112] found that the ALINEA control strategy is fairly insensitive to changes in a wide range of K_R-values within their macroscopic implementation. It was, however, found that increasing the K_R-value leads to faster response of the regulator, while extremely large K_R-values lead to unstable, oscillatory behaviour of the controller. Therefore, the small, medium and large K_R-values considered in this dissertation were chosen as 10 veh/h, 40 veh/h and 70 veh/h, respectively.

Due to the fact that the main aim in this dissertation is to reduce the *total time spent in the system* (TTS), described in §5.1.4, by as much as possible, this is the performance measure selected for the purpose of comparison in the parameter evaluation. In order to determine the TTS-values presented in Table 6.1, thirty simulation runs using different seeds were performed for the comparison of each of the parameter combinations. The same thirty seed values were, however, used for the comparison of each of the 39 combinations of parameters. It is expected that effective RM in Scenario 2 will lead to large savings in travel time on the highway, while the lower on-ramp demand should result in acceptable increases in travel time for vehicles joining the highway from the on-ramp. As a result, it is expected that RM should be most effective in Scenario 2. Therefore, Scenario 2 was chosen as the scenario in which to perform the parameter evaluation.

The target density was initially examined in unit intervals ranging from 24 veh/km to 34 veh/km. The results obtained from this initial parameter variation suggested that the best total time in the system value is achieved at 26 veh/km, while setting the controller parameter $K_R = 40$ consistently achieved the best performance. Subsequently, the surrounding unit interval was examined more closely in steps of 0.1 veh/km as may be seen in Table 6.1.

As may be seen from the results presented in Table 6.1, the microscopic implementation is not as insensitive to changes in the K_R-value, as claimed by Papageorgiou *et al.* [112], and no clear trend in terms of the performance of the control strategy emerged. It is, however, evident that similarly to the implementation by Wang *et al.* [167], setting the K_R-value to 40 consistently returned good performance. The smallest TTS-value was achieved when taking the value of K_R as 40 with a target density of 26 veh/km. As a result, the parameter combination $(K_R, \hat{\rho}) = (40, 26)$ is employed for all further comparisons carried out in this chapter.

A similar approach was adopted for the parameter evaluation in respect of the PI-ALINEA control law. In the PI-ALINEA control law in (3.18), there are two controller parameters, as

TABLE 6.1: *Parameter evaluation results for the ALINEA RM control policy, measured in terms of the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
K_R	25.0	25.5	25.6	25.7	25.8	25.9	26.0
10	955.33	—	—	—	—	—	919.16
40	919.38	897.65	897.76	900.77	891.10	886.42	873.49
70	917.96	—	—	—	—	—	890.89
	Target density $\hat{\rho}$						
K_R	26.1	26.2	26.3	26.4	26.5	27.0	
10	—	—	—	—	—	934.21	
40	890.16	884.01	909.71	905.56	876.10	906.14	
70	—	—	—	—	—	908.27	

well as the target density which require fine tuning. Given the vast combination of possible controller configurations, and the fact, that for the ALINEA control law the best-performing target controller parameter was found to be $K_R = 40$, as corroborated by Wang *et al.* [167] in the paper in which PI-ALINEA was first published, the values of K_P and K_R were taken as 60 and 40, respectively, as suggested in the original publication. Different target density values were then considered as for ALINEA. The intial rough parameter evaluation for target densities between 24 veh/km and 34 veh/km indicated that the best-performing density is 24 veh/km. The densities around 24 veh/km were subsequently investigated. These results are summarised in Table 6.2. As may be seen in the table, the smallest TTS-value is achieved when setting the target density equal to 24.2 veh/km. Therefore, this is the target density value employed for all further comparisons performed in this chapter.

TABLE 6.2: *Parameter evaluation results for the PI-ALINEA RM control policy, measured in terms of the TTS by the vehicles (in veh·h).*

Target density $\hat{\rho}$						
23.0	23.5	23.6	23.7	23.8	23.9	24.0
904.88	906.03	923.79	934.15	915.58	893.00	894.46
Target density $\hat{\rho}$						
24.1	24.2	24.3	24.4	24.5	25.0	
936.92	877.90	921.46	915.49	886.92	916.26	

Q-Learning parameter evaluation

Similarly to the parameter evaluation performed for the ALINEA and PI-ALINEA control strategies, the effectiveness of the Q-Learning algorithm was investigated for target density values ranging from 24 veh/km to 34 veh/km. This initial investigation revealed that the best TTS is achieved at a target density of 24 veh/km. This was again followed by a closer examination of the surrounding unit interval in subintervals of 0.1 veh/km. These results are presented in Table 6.3. As may be seen in the table, the target density at which the smallest TTS was achieved is 23.8 veh/km. Therefore, 23.8 veh/km is the target density employed for all further comparisons conducted in respect of Q-Learning within the context of RM in this dissertation.

TABLE 6.3: *Parameter evaluation results for the Q-Learning RM implementation, measured in terms of the TTS by the vehicles (in veh·h).*

Target density $\hat{\rho}$						
23.0	23.5	23.6	23.7	23.8	23.9	24.0
905.99	880.60	912.77	890.91	869.87	913.51	891.92
Target density $\hat{\rho}$						
24.1	24.2	24.3	24.4	24.5	25.0	
895.99	929.04	953.67	935.76	1 075.46	961.33	

kNN-TD learning parameter evaluation

As with all prior RM implementations, the effectiveness of the kNN-TD algorithm, when applied to the RM problem, was investigated in unit intervals for target densities ranging from 24 veh/km to 34 veh/km. This initial investigation indicated that the smallest TTS could be achieved at a target density of 25 veh/km. Therefore, the unit interval around 25 veh/km was once again investigated more closely in steps of 0.1 veh/km. The results of this investigation are presented in Table 6.4. As may be seen in the table, the target density corresponding to the smallest TTS has a value of 24.8 veh/km.

TABLE 6.4: *Parameter evaluation results for the kNN-TD RM implementation, measured in terms of the TTS by the vehicles (in veh·h).*

Target density $\hat{\rho}$						
24.0	24.5	24.6	24.7	24.8	24.9	25.0
888.46	888.03	878.98	875.93	860.61	903.39	870.99
Target density $\hat{\rho}$						
25.1	25.2	25.3	25.4	25.5	26.0	
901.05	871.63	888.99	900.03	893.04	900.38	

The parameter evaluation aimed at determining the best target density value was performed using $k = 4$ nearest neighbours. The value $k = 4$ was chosen based on the findings of Rezaee *et al.* [130], who reported that $k = 4$ yielded the best results in their RM implementation. For the sake of completeness, however, values of $k = 2$ and $k = 8$ are also considered in this dissertation. Due to the fact that it is not expected that the ordering in terms of the best target density will be affected by the number of nearest neighbours, it was deemed sufficient only to evaluate various k-values in conjunction with a fixed target density of $\hat{\rho} = 24.8$ veh/km, rather than evaluating all possible combinations of k-values and $\hat{\rho}$-values. The results of this investigation are illustrated graphically in Figure 6.4. As may be seen in the figure, the quickest time to convergence is achieved with $k = 2$ nearest neighbours. As expected, the time until convergence increases as the number of nearest neighbours increases. It is, however, interesting to note that the algorithmic performance is not proportional to the number of nearest neighbours. As may be seen in the figure, the best performance is achieved with $k = 4$ nearest neighbours. This was further confirmed after 30 comparative simulation runs had been performed, with TTS-values of 873.79, 857.09 and 890.64 being recorded for the cases with $k = 2$, $k = 4$ and $k = 8$, respectively. As a result, $k = 4$ nearest neighbours are employed for all further comparisons in this chapter.

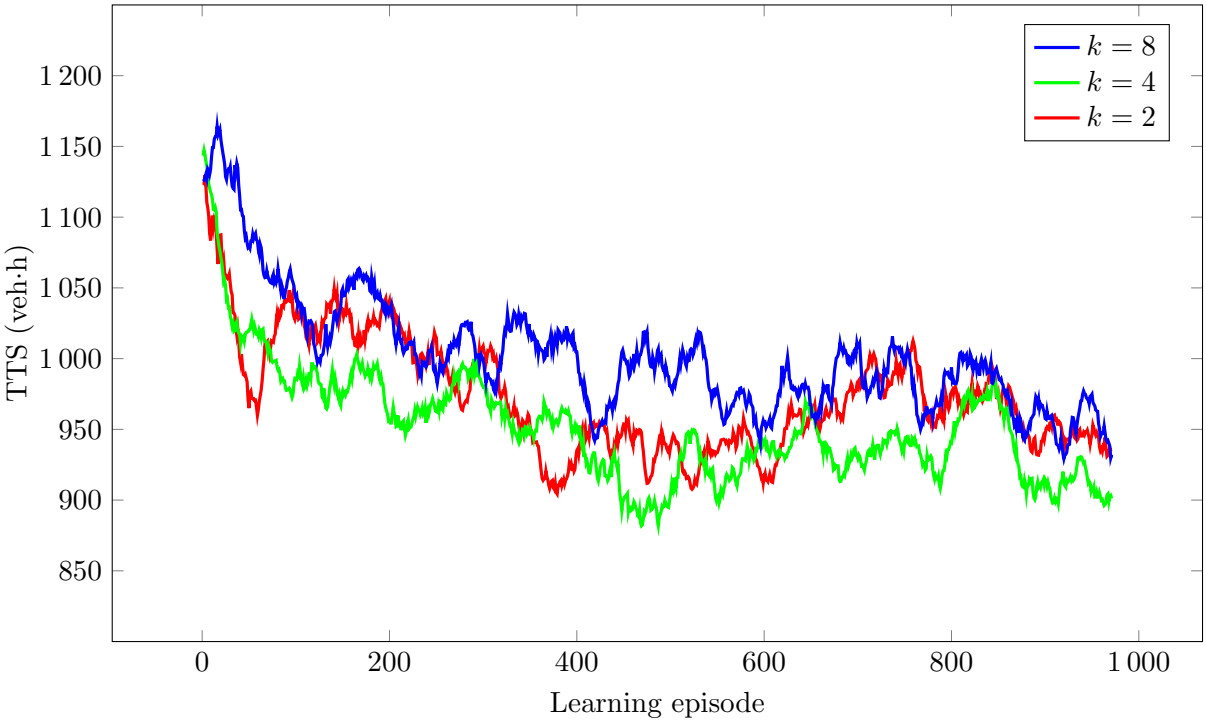


FIGURE 6.4: *The kNN-TD learning progression over the course of 1 000 training epochs for Scenario 2, shown for values of $k = 2$, $k = 4$ and $k = 8$. In order to filter out some simulation noise, a moving average over 30 epochs is shown.*

6.5.2 Algorithmic Comparison

In this section, the simulation results and the relative RM algorithmic performances are compared. This comparison is performed in each of the four different scenarios of traffic demand, as described in §5.3.2. The results are presented and interpreted through the use of box plots in which the means, medians and interquartile ranges of the PMIs are indicated, as well as tables indicating whether or not statistical differences exist between the PMI values for each pair of algorithms at a 5% level of significance.

Scenario 1

As may be seen from the p-values of the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms for Scenario 1, presented in Table 6.5, the ANOVA test revealed that there are, in fact, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all seven PMIs (not necessarily the same pairs in each case). Furthermore, Levene's test revealed that the variances of the PMIs returned by the algorithms were only statistically indistinguishable at a 5% level of significance in respect of the TTS and maximum TISHW PMIs, while the variances between at least some pair of algorithms were found to differ statistically for the other five PMIs at a 5% level of significance. Therefore, the Fisher LSD test was employed in order to determine between which pairs of algorithms' PMI values the differences occur in respect of the TTS and maximum TISHW, while the Games-Howell test was employed for this purpose in respect of all other PMIs.

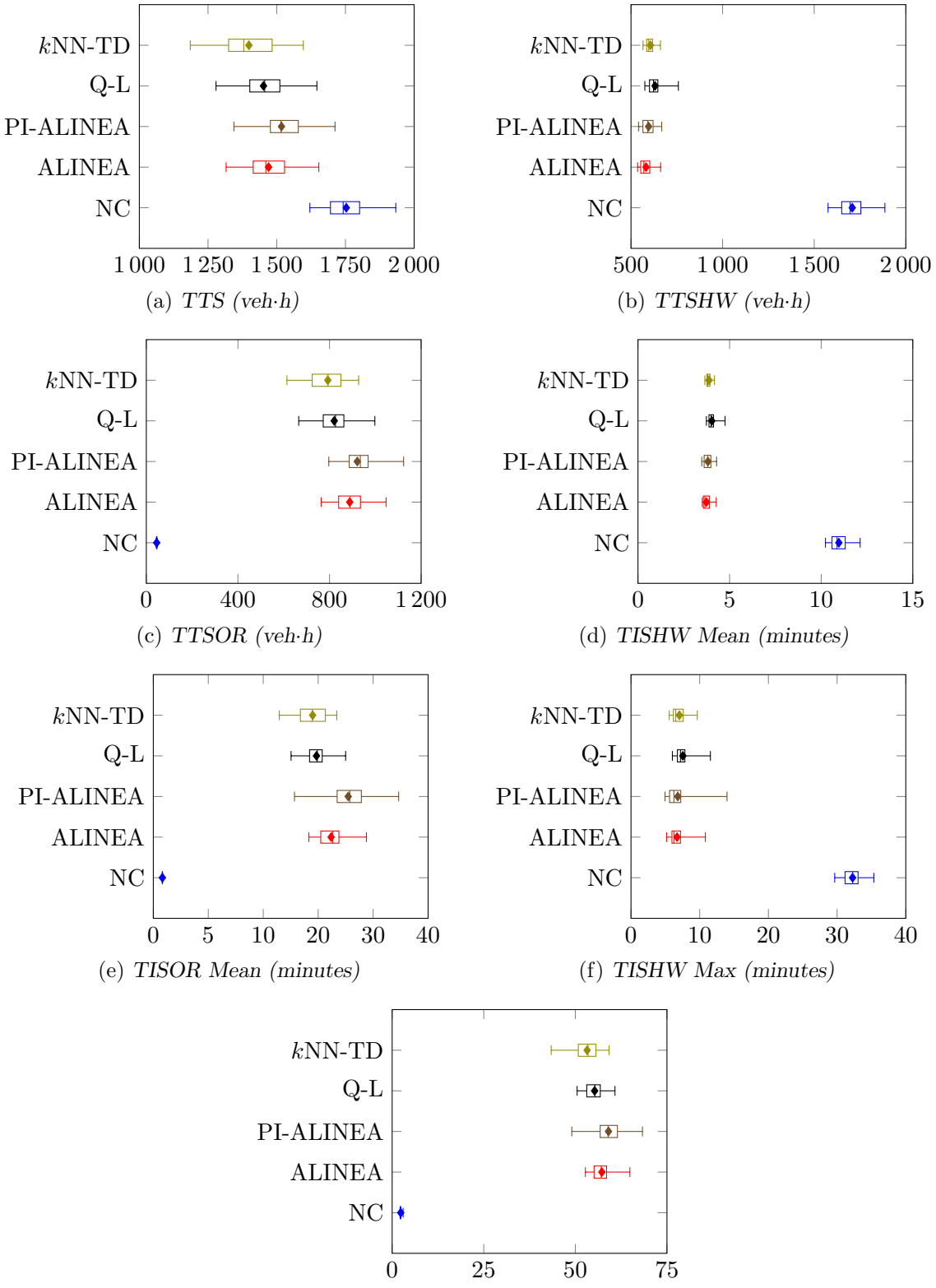
TABLE 6.5: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 1. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

PMI	Mean value					*p*-value	
	No Control	ALINEA	PI-ALINEA	Q-Learning	*k*NN-TD	ANOVA	Levene's Test
TTS	1 753.01	1 470.25	1 517.30	1 451.80	1 398.80	$< 1 \times 10^{-17}$	6.7483×10^{-1}
TTSHW	1 707.70	582.04	596.02	631.19	606.16	$< 1 \times 10^{-17}$	1.4413×10^{-7}
TTSOR	45.31	888.21	921.28	820.61	792.64	$< 1 \times 10^{-17}$	2.9143×10^{-8}
TISHW Mean	10.96	3.73	3.82	4.03	3.88	$< 1 \times 10^{-17}$	1.1087×10^{-7}
TISOR Mean	1.66	32.35	35.47	29.69	28.99	$< 1 \times 10^{-17}$	1.7548×10^{-9}
TISHW Max	32.25	6.70	6.81	7.55	7.04	$< 1 \times 10^{-17}$	5.2954×10^{-1}
TISOR Max	2.34	57.23	59.05	55.27	53.21	$< 1 \times 10^{-17}$	2.0492×10^{-8}

As may be seen in the box plot in Figure 6.5(a), all four RM implementations were able to achieve statistically significant improvements over the no-control case in terms of the TTS. This is also clear from the *p*-values in Table 6.6. The *k*NN-TD RM implementation outperformed ALINEA, PI-ALINEA and Q-Learning as it returned the best performance in terms of the TTS, achieving a value of 1 398.80 veh·h, which is a 20.21% improvement over the no-control case. The performance of ALINEA and Q-Learning are not statistically different at a 5% level of significance, as may be seen in Table 6.6. These two algorithms achieved improvements of 16.13% and 17.18%, respectively over the no-control case. Furthermore, Q-Learning was able to outperform PI-ALINEA, which achieved a reduction in the TTS of 13.45%, while the performances of ALINEA and PI-ALINEA were found the be statistically indistinguishable at 5% level of significance.

As may be expected for an RM implementation, the savings in terms of travel times are achieved on the highway, which is protected by reducing the on-ramp flows in order to avoid congestion. This trend is clear in all RM implementations, as all algorithms achieved significant improvements in the TTSHW, as may be seen in Figure 6.5(b). This is confirmed by the p-values in Table 6.7, where it is shown that ALINEA, PI-ALINEA and *k*NN-TD RM achieved the best performance in respect of the TTSHW (but did not perform statistically different from one another at a 5% level of significance), achieving improvements of 65.92%, 65.10% and 64.50%, respectively, over the no-control case. Although Q-Learning was outperformed by both the ALINEA and PI-ALINEA implementations, it still managed to achieve a 63.04% improvement over the no-control case in respect of the TTSHW, as its performance was found not to differ statistically from that of *k*NN-TD RM at a 5% level of significance.

These significant improvements in respect of the TTSHW do, however, lead to significant increases in respect of the TTSOR, as RM potentially leads to the build-up of long on-ramp queues. As may be seen in Figure 6.5(c), this is clearly the case for all RM implementations. Taking into account that an increase in the TTSOR is to be expected, and that, as a result, the no-control scenario should, by default, achieve the smallest TTSOR-value, it is *k*NN-TD RM which, in fact, achieved the smallest TTSOR-value of 792.64 veh·h, outperforming both the ALINEA and PI-ALINEA implementations, which achieved values of 888.21 veh·h and 921.28 veh·h, respectively. Although the *k*NN-TD algorithm is able to achieve a smaller value for the TTSOR than Q-Learning, which returned a TTSOR-value of 820.61 veh·h, the two algorithms do not perform statistically differently from one another at a 5% level of significance, as may be seen in Table 6.8. Similarly, ALINEA returned a smaller TTSOR-value than PI-ALINEA, yet these two control strategies were also found to perform statistically similarly at a 5% level of significance.



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.5: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 1.*

The trends in respect of the mean and maximum travel times along the highway only are very similar to those for both the TTS and TTSHW, as may be seen in Figures 6.5(d) and 6.5(f), respectively. The ALINEA and PI-ALINEA implementations achieved the best performance in respect of the mean TISHW, outperforming the Q-Learning implementation, while they were both found the perform statistically similarly to the kNN-TD RM implementation. Although kNN-TD RM returned a smaller mean TISHW-value than Q-Learning, the performances of these two implementations could not be classified as statistically different at a 5% level of significance. The ALINEA control strategy achieved a reduction in the mean TISHW from 10.96 minutes in the no-control case to 3.73 minutes, while PI-ALINEA, kNN-TD RM and Q-Learning achieved mean travel times along the highway only of 3.82 minutes, 3.88 minutes and 4.03 minutes, respectively, as may be seen in Table 6.9. Similarly, the ALINEA implementation achieved a reduction of 79.22% over the no-control case, in respect of the maximum time spent travelling along the highway only, again outperforming the Q-Learning algorithm which was able to achieve a reduction of 76.59%, while its performance was found to be statistically indistinguishable from that of both PI-ALINEA and kNN-TD RM, at a 5% level of significance. Furthermore, PI-ALINEA, Q-Learning and kNN-TD RM were found to perform statistically similarly in respect of the maximum TISHW, as these control strategies achieved reductions of 78.88%, 76.59% and 78.18% respectively, over the no-control case. The corresponding p-values are summarised in Table 6.11.

As in the case of the TTSOR, increases were again to be expected in both the mean and maximum travel times for vehicles joining the highway from the on-ramp. This trend is clearly visible in the box plots in Figures 6.5(e) and 6.5(g). As was the case with the TTSOR, the kNN-TD RM implementation was able to achieve the smallest mean and maximum TISOR-values, outperforming both the ALINEA and PI-ALINEA implementations in respect of both these performance measures, as may be seen in Tables 6.10 and 6.12. The kNN-TD RM implementation is followed in the order of relative algorithmic performances by Q-Learning, which was able to outperform ALINEA in respect of the mean TISOR, and PI-ALINEA in respect of both the mean and maximum TISOR, while the performances of ALINEA and PI-ALINEA were found to be statistically similar at a 5% level of significance in respect of both of these PMIs. In the kNN-TD RM implementation, the mean travel time for vehicles joining the highway from the on-ramp is 28.99 minutes, while vehicles require on average 29.69 minutes, 32.35 minutes and 35.47 minutes in the Q-Learning, ALINEA and PI-ALINEA implementations, respectively. The maximum travel time is limited to 53.21 minutes by kNN-TD RM, while this value increases to 55.27 minutes in the case of Q-Learning, 57.23 minutes when ALINEA is implemented and 59.05 minutes in the case of PI-ALINEA.

TABLE 6.6: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	2.2204×10^{-16}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	6.4782×10^{-2}	4.6674×10^{-1}	5.3821×10^{-3}
PI-ALINEA			—	1.0558×10^{-2}	6.3393×10^{-6}
Q-Learning				—	3.7813×10^{-2}
kNN-TD					—
Mean	1 753.01	1 470.25	1 517.30	1 451.80	1 398.80

TABLE 6.7: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.5158×10^{-13}	3.6782×10^{-13}	1.1183×10^{-12}	5.1992×10^{-14}
ALINEA		—	5.7573×10^{-1}	1.0051×10^{-4}	9.6577×10^{-2}
PI-ALINEA			—	9.43551×10^{-3}	8.2730×10^{-1}
Q-Learning				—	1.2374×10^{-1}
kNN-TD					—
Mean	1 707.70	582.04	596.02	631.19	606.16

TABLE 6.8: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	7.7720^{-15}	$< 1 \times 10^{-17}$	2.4420×10^{-15}	1.3320×10^{-15}
ALINEA		—	6.0405×10^{-1}	6.0878×10^{-3}	1.7841×10^{-4}
PI-ALINEA			—	2.9585×10^{-4}	1.0247×10^{-5}
Q-Learning				—	6.0924×10^{-1}
kNN-TD					—
Mean	45.31	888.21	921.28	820.61	792.64

TABLE 6.9: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.0150×10^{-14}	3.4694×10^{-13}	1.0022×10^{-12}	5.6266×10^{-13}
ALINEA		—	5.8152×10^{-1}	5.1698×10^{-5}	1.0193×10^{-1}
PI-ALINEA			—	6.6438×10^{-3}	8.3697×10^{-1}
Q-Learning				—	1.0065×10^{-1}
kNN-TD					—
Mean	10.96	3.73	3.82	4.03	3.88

TABLE 6.10: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.9950×10^{-15}	$< 1 \times 10^{-17}$	3.7750×10^{-15}	1.9980×10^{-15}
ALINEA		—	9.1746×10^{-3}	4.4174×10^{-3}	7.7021×10^{-5}
PI-ALINEA			—	3.0064×10^{-7}	4.5458×10^{-8}
Q-Learning				—	8.0721×10^{-1}
kNN-TD					—
Mean	1.66	32.35	35.47	29.69	28.99

TABLE 6.11: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	7.7103×10^{-1}	2.7038^{-2}	3.6991×10^{-1}
PI-ALINEA			—	5.4068×10^{-2}	5.4423×10^{-1}
Q-Learning				—	1.8423×10^{-1}
kNN-TD					—
Mean	32.25	6.70	6.81	7.55	7.04

TABLE 6.12: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	3.3309×10^{-15}	6.9399×10^{-15}	$< 1 \times 10^{-17}$	7.719×10^{-15}
ALINEA		—	3.5514×10^{-1}	6.7870×10^{-2}	1.7417×10^{-4}
PI-ALINEA			—	2.0004×10^{-3}	7.4744×10^{-6}
Q-Learning				—	1.0354×10^{-1}
kNN-TD					—
Mean	2.34	57.23	59.05	55.27	53.21

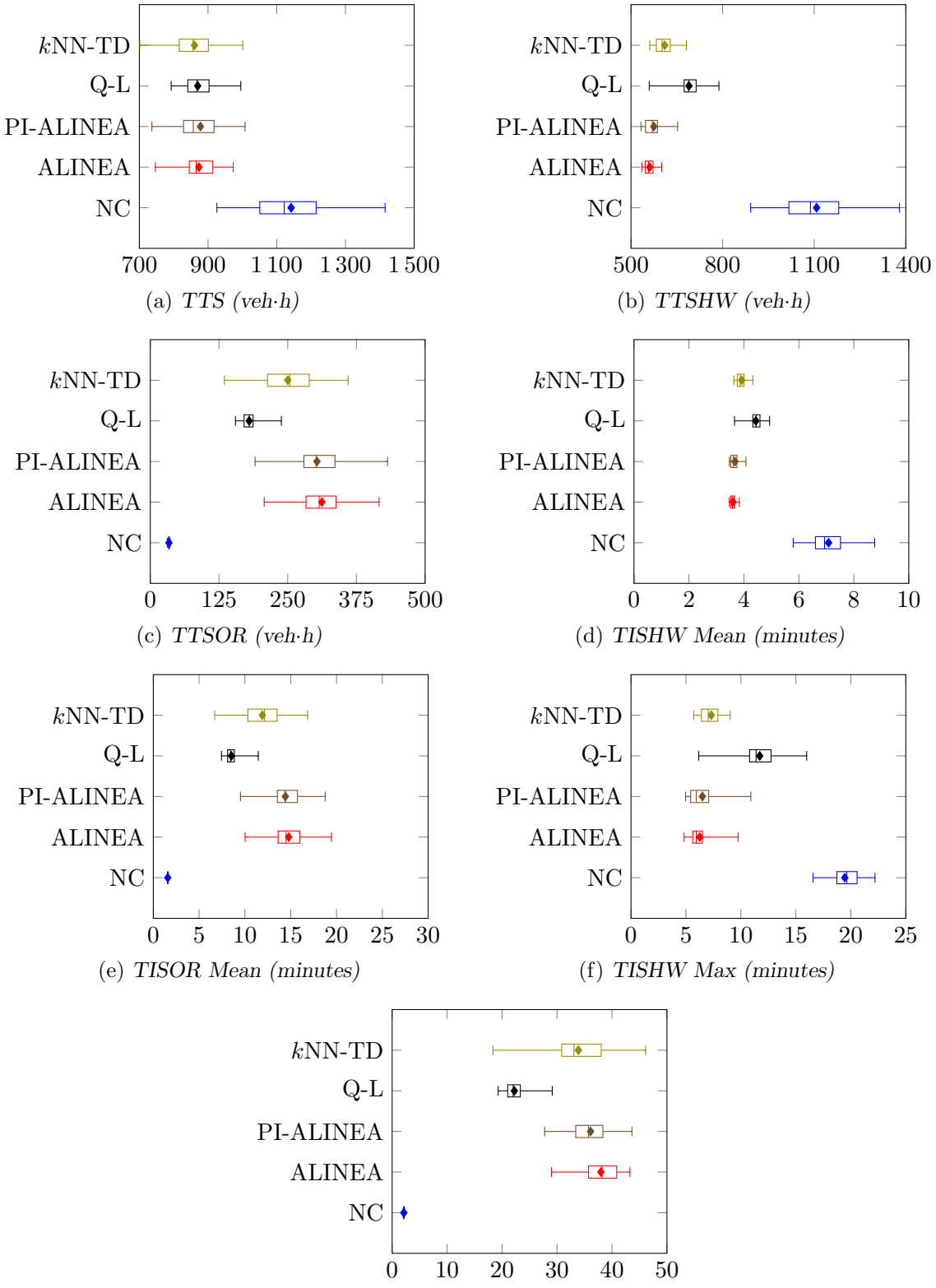
Scenario 2

As in Scenario 1, the p-values returned by the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms for Scenario 2, presented in Table 6.13, revealed that there are, once again, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all seven PMIs. Unlike in Scenario 1, however, Levene's test revealed that the variances returned by at least some pair of algorithms in Scenario 2 are also statistically different at a 5% level of significance in respect of all seven PMIs. Hence the Games-Howell *post hoc* test was performed so as to ascertain where the differences between the algorithms' output data occur in respect of all seven PMIs.

TABLE 6.13: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 2. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 141.80	873.49	877.90	869.87	860.61	$< 1 \times 10^{-17}$	1.6056×10^{-5}
TTSHW	1 107.88	561.21	574.68	689.93	610.40	$< 1 \times 10^{-17}$	1.6986×10^{-14}
TTSOR	33.92	312.28	303.22	179.94	250.21	$< 1 \times 10^{-17}$	4.4579×10^{-11}
TISHW Mean	7.08	3.60	3.67	4.44	3.92	$< 1 \times 10^{-17}$	5.5511×10^{-15}
TISOR Mean	1.58	14.78	14.41	8.53	11.91	$< 1 \times 10^{-17}$	8.9253×10^{-12}
TISHW Max	19.45	6.26	6.49	11.71	7.31	$< 1 \times 10^{-17}$	3.5866×10^{-2}
TISOR Max	2.13	37.97	36.10	22.23	33.89	$< 1 \times 10^{-17}$	6.8268×10^{-9}

All four RM implementations were again able to achieve significant improvements in respect of the TTS in Scenario 2, as may be seen in Figure 6.6(a). The kNN-TD RM implementation was able to achieve a 24.63% reduction in the TTS over the no-control case, while the Q-Learning,



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.6: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 2.*

ALINEA and PI-ALINEA implementations achieved 23.81%, 23.50% and 23.11% improvements, respectively. Interestingly none of the RM implementations was able to clearly outperform any of the others at a 95% level of confidence, as may be seen in Table 6.14.

ALINEA and PI-ALINEA were able to achieve the largest reduction in the TTSHW over the no-control case, with neither of these algorithms outperforming each other, while they were both able to outperform Q-Learning and kNN-TD RM in respect of the TTSHW, as may be seen from Table 6.15. The kNN-TD RM implementation was also able to outperform Q-Learning at a 5% level of significance. This trend is also evident in the box plots in Figure 6.6(b). ALINEA and PI-ALINEA were able to reduce the TTSHW by 49.34% and 48.13%, respectively, while kNN-TD RM and Q-Learning were able to achieve reductions of 44.90% and 37.73%, respectively.

Interestingly, the order of relative algorithmic performances of the four RM strategies in respect of the TTSOR was found to be exactly opposite to that in respect of the TTSHW in Scenario 2. Taking the natural increase in travel time by vehicles joining the highway from the on-ramp into account, the Q-Learning algorithm outperformed ALINEA, PI-ALINEA and kNN-TD, achieving a TTSOR-value of 179.94 veh·h. As may be seen in Table 6.16, kNN-TD was able to outperform both ALINEA and PI-ALINEA, with the three algorithms achieving TTSOR-values of 250.21 veh·h, 321.28 veh·h and 303.22 veh·h, respectively. Finally, the performances of ALINEA and PI-ALINEA were again found to be statistically indistinguishable at a 5% level of significance. These results are summarised in the box plots shown in Figure 6.6(c).

As may be seen in Figures 6.6(d) and 6.6(f), ALINEA and PI-ALINEA were again able to achieve the largest reductions in both the mean and the maximum time spent in the system by vehicles travelling along the highway only. This is confirmed by the p-values presented in Tables 6.17 and 6.19. As may be seen in the tables, ALINEA and PI-ALINEA outperformed both Q-Learning and kNN-TD RM in respect of the mean TISHW, while ALINEA and PI-ALINEA were both able to outperform Q-Learning in respect of the maximum TISHW, with the algorithms achieving reductions in the mean TISHW over the no-control case of 49.15%, 48.16%, 38.14% and 44.63%, respectively. Similarly, improvements of 67.81%, 66.63%, 39.79% and 62.42% were achieved over the no-control case by the four algorithms, respectively, in respect of the maximum TISHW. As is evident from the tables, the reductions achieved by kNN-TD RM in respect of both the mean and maximum TISHW-values were large enough to outperform Q-Learning at a 5% level of significance in respect of both these PMIs.

When considering the mean and maximum travel times for vehicles joining the highway from the on-ramp, Q-Learning again outperformed all three other RM implementations at a 5% level of significance, as may be seen in Tables 6.18 and 6.20. The kNN-TD RM implementation returned the next-best performance, outperforming both ALINEA and PI-ALINEA in respect of the mean TISOR, while kNN-TD RM was able to outperform only ALINEA in respect of the maximum TISOR at a 5% level of significance. The performances of ALINEA and PI-ALINEA were once again statistically indistinguishable at a 5% level of significance in respect of both these PMIs. This trend is also evident in the box plots of Figures 6.6(e) and 6.6(g). As may be seen in Table 6.18, Q-Learning was able to achieve a mean TISOR of 8.53 minutes compared with 11.91 minutes for kNN-TD RM, 12.41 minutes for PI-ALINEA and 14.78 minutes for ALINEA. Furthermore, Q-Learning was able to limit the maximum TISOR to 22.23 minutes, while this value increased to 33.89 minutes for kNN-TD RM, 36.10 minutes for ALINEA, and 37.97 minutes for PI-ALINEA, as may be seen in Table 6.20.

TABLE 6.14: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.2428×10^{-12}	$< 1 \times 10^{-17}$	2.6867×10^{-13}	$< 1 \times 10^{-13}$
ALINEA		—	9.9876×10^{-1}	9.9871×10^{-1}	9.3592×10^{-1}
PI-ALINEA			—	9.8465×10^{-1}	8.7929×10^{-1}
Q-Learning				—	9.7611×10^{-1}
kNN-TD					—
Mean	1 141.80	873.49	877.90	869.89	860.61

TABLE 6.15: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.6843×10^{-14}	4.8850×10^{-14}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	2.6590×10^{-1}	$< 1 \times 10^{-17}$	2.5781×10^{-6}
PI-ALINEA			—	$< 1 \times 10^{-17}$	2.8731×10^{-3}
Q-Learning				—	5.9342×10^{-9}
kNN-TD					—
Mean	1 107.88	561.21	574.68	689.93	610.40

TABLE 6.16: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance..*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.9900×10^{-15}	$< 1 \times 10^{-17}$	1.3320×10^{-14}	3.3309×10^{-14}
ALINEA		—	9.4583×10^{-1}	$< 1 \times 10^{-17}$	6.9172×10^{-5}
PI-ALINEA			—	$< 1 \times 10^{-17}$	1.3679×10^{-3}
Q-Learning				—	3.0816×10^{-7}
kNN-TD					—
Mean	33.92	321.28	303.22	179.94	250.21

TABLE 6.17: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.6399×10^{-14}	9.0483×10^{-14}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	3.3179×10^{-1}	$< 1 \times 10^{-17}$	3.6889×10^{-7}
PI-ALINEA			—	$< 1 \times 10^{-17}$	2.3740×10^{-4}
Q-Learning				—	2.5637×10^{-10}
kNN-TD					—
Mean	7.08	3.60	3.67	4.44	3.92

TABLE 6.18: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	9.9900×10^{-16}	$< 1 \times 10^{-17}$
ALINEA		—	9.5175×10^{-1}	$< 1 \times 10^{-17}$	3.4075×10^{-5}
PI-ALINEA			—	$< 1 \times 10^{-17}$	4.5281×10^{-4}
Q-Learning				—	1.06165×10^{-7}
kNN-TD					—
Mean	1.58	14.78	12.41	8.53	11.91

TABLE 6.19: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	6.7815×10^{-12}	1.4906×10^{-11}	$< 1 \times 10^{-17}$
ALINEA		—	9.5789×10^{-1}	$< 1 \times 10^{-17}$	7.4934×10^{-3}
PI-ALINEA			—	6.8673×10^{-12}	1.7764×10^{-1}
Q-Learning				—	$< 1 \times 10^{-17}$
kNN-TD					—
Mean	19.45	6.26	6.49	11.71	7.31

TABLE 6.20: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.9979×10^{-15}	1.1100×10^{-15}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	2.7085×10^{-1}	$< 1 \times 10^{-17}$	1.5095×10^{-2}
PI-ALINEA			—	6.2461×10^{-13}	4.0502×10^{-1}
Q-Learning				—	2.2509×10^{-11}
kNN-TD					—
Mean	2.13	37.97	36.10	22.23	33.89

Scenario 3

As in Scenarios 1 and 2, an ANOVA test revealed that there are, once again, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all seven PMIs, as may be seen from the p-values presented in Table 6.21. The Levene test revealed that the variances returned by at least some pair of algorithms were, in fact, found to differ statistically for each of the seven PMIs at a 5% level of significance. Therefore, the Games-Howell test was employed in order to determine between which pairs of algorithmic output values differences occur in respect of all seven PMIs.

In Scenario 3, for the first time, only the RL implementations were able to achieve statistically significant improvements over the no-control case in respect of the TTS, while no statistical differences were detectable between the performances of ALINEA, PI-ALINEA and the no-

control case at a 5% level of significance. Both Q-Learning and kNN-TD were able to outperform ALINEA and PI-ALINEA, achieving 14.72% and 11.09% improvements over the no-control case, respectively, but were found to be statistically indistinguishable from one another at a 5% level of significance, as may be seen in Table 6.22. This dominance of the RL algorithms in respect of the TTS is evident from the p-values presented in Table 6.22. Although the means of the TTS-values achieved by Q-Learning and kNN-TD do not differ statistically at a 5% level of significance, it is clear from Figure 6.7(a) that their variances do, in fact, differ, with kNN-TD exhibiting a more consistent performance (indicated by the smaller interquartile range). This statistical difference in the variances at a 5% level of significance was confirmed statistically by the Levene test, as may be seen in Table 6.21.

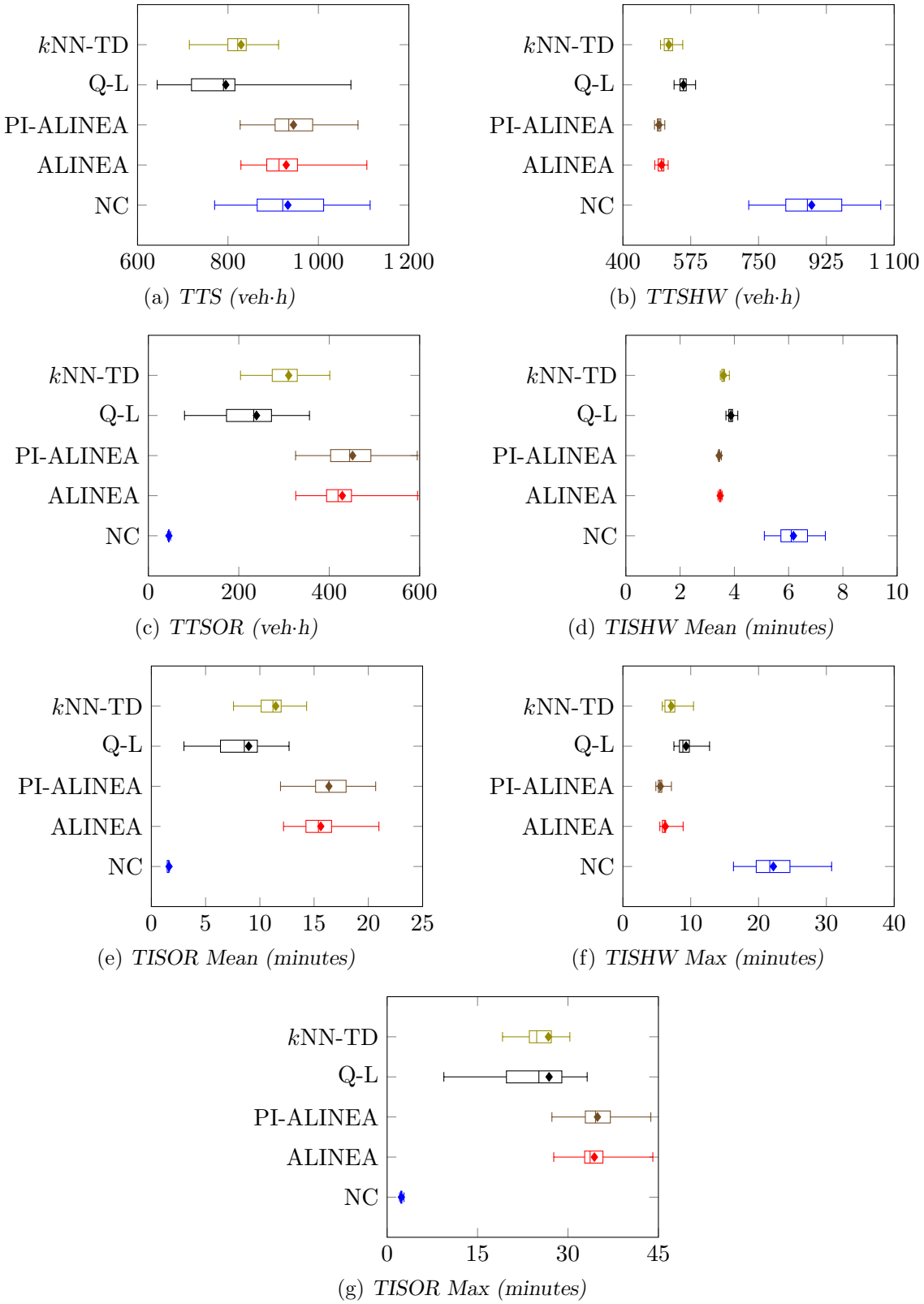
As may be seen in Table 6.23, statistical differences were detectable between all RM implementations in respect of their TTSHW-values at a 5% level of significance. Interestingly, although the performances of Q-Learning and kNN-TD are statistically indistinguishable in respect of the TTS, there is a statistical difference in the TTSHW-values achieved at a 5% level of significance. The same is true for ALINEA, PI-ALINEA and the no-control case. These differences are evident in the box plots of Figure 6.7(b). As may be seen in the figure, PI-ALINEA outperforms all other algorithms in respect of the TTSHW, followed by ALINEA, which outperforms Q-Learning, kNN-TD RM and the no-control case. Finally, kNN-TD only outperforms both Q-Learning and the no-control case in respect of the TTSHW, while Q-Learning was able to outperform only the no-control case.

In respect of the TTSOR, statistical differences were again found between all algorithms, except ALINEA and PI-ALINEA at a 5% level of significance, as may be seen in Table 6.24. It is interesting to note, however, that the ordering of the relative performances of the algorithms is exactly opposite to what it was for the TTSHW, with the no-control scenario expectedly achieving the smallest TTS-value of 45.40 veh·h. The no-control case is followed by Q-Learning, achieving a value of 239.16 veh·h. Q-Learning thus outperformed kNN-TD RM, ALINEA and PI-ALINEA, which returned values of 310.36 veh·h, 428.94 veh·h and 451.72 veh·h, respectively. Finally, kNN-TD outperformed ALINEA and PI-ALINEA. This ordering of the relative algorithmic performances is very clear in the box plots of Figure 6.7(c).

As for the TTSHW, the algorithms once again all performed statistically different in terms of both the mean and maximum time spent in the system by vehicles travelling on the highway only at a 5% level of significance, as may be seen in Tables 6.25 and 6.27. As shown in Figures 6.7(d) and 6.7(f), the pattern of relative performances of the algorithms for these two PMIs is the same as for the TTSHW. PI-ALINEA yielded the largest reduction, outperforming all other RM algorithms in terms of both the mean and maximum travel times with improvements of 44.34% and 75.03% over the no-control case, respectively. PI-ALINEA was followed by ALINEA which

TABLE 6.21: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 3. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

PMI	Mean value					p-value	
	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	932.46	928.75	944.88	795.18	829.02	4.6629×10^{-15}	4.0257×10^{-2}
TTSHW	887.07	499.81	493.16	556.02	518.66	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TTSOR	45.40	428.94	451.72	239.16	310.36	$< 1 \times 10^{-17}$	7.1521×10^{-8}
TISHW Mean	6.18	3.48	3.44	3.87	3.60	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISOR Mean	1.63	15.62	16.36	8.97	11.47	$< 1 \times 10^{-17}$	3.1641×10^{-6}
TISHW Max	22.19	6.26	5.54	9.32	7.14	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISOR Max	2.37	34.38	34.89	26.87	26.76	$< 1 \times 10^{-17}$	3.3493×10^{-5}



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.7: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 3.*

outperformed Q-Learning and kNN-TD RM, achieving improvements of 43.69% and 71.79% over the no-control case in respect of the mean and maximum travel times for vehicles travelling along the highway only. The kNN-TD RM implementation was able to achieve improvements of 41.75% and 67.82% in terms of the mean TISHW and maximum TISHW, respectively, when compared with the no-control case, thereby outperforming Q-Learning which returned improvements of 37.38% and 57.99%, in respect of the mean and maximum TISHW.

As may have been expected, the ordering of the relative algorithmic performances in respect of the mean and maximum time in the system spent by vehicles joining the highway from the on-ramp is again almost opposite to that of the mean and maximum travel times spent by vehicles travelling along the highway only. These orderings are clear in the box plots in Figures 6.7(e) and 6.7(g). The no-control case exhibits the smallest values for both the mean TISOR and maximum TISOR, achieving values of 1.63 minutes and 2.37 minutes, respectively, as may be seen in Tables 6.26 and 6.28. The Q-Learning and kNN-TD implementations were able to outperform both PI-ALINEA and ALINEA, achieving values of 8.97 minutes and 11.47 minutes, respectivley, for the mean TISOR, compared with values of 15.62 minutes and 16.36 minutes for ALINEA and PI-ALINEA. The performances of kNN-TD RM and Q-Learning, as well as those of PI-ALINEA and ALINEA, were found to be statistically indistinguishable at a 5% level of significance in respect of the mean TISOR. In respect of the maximum TISOR, kNN-TD RM achieved the smallest value of 26.76 minutes, compared with 26.87 minutes for Q-Learning. As may be expected, due to the closeness of these values, the performances of kNN-TD RM and Q-Learning were again found to be statistically indistinguishable at a 5% level of significance. Both kNN-TD RM and Q-Learning were, again, able to outperform PI-ALINEA and ALINEA, which returned values of 34.89 minutes and 34.38 minutes, respectively, in respect of the maximum TISOR, while the latter two implementations were found to perform statistically on par with one another at a 5% level of significance.

TABLE 6.22: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.9974×10^{-1}	9.7371×10^{-1}	6.9184×10^{-6}	3.1995×10^{-5}
ALINEA		—	8.7065×10^{-1}	1.1873×10^{-6}	4.1385×10^{-7}
PI-ALINEA			—	9.5709×10^{-8}	1.3786×10^{-8}
Q-Learning				—	5.0197×10^{-1}
kNN-TD					—
Mean	932.46	928.75	944.88	795.18	829.02

TABLE 6.23: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.8319×10^{-10}	1.0325×10^{-10}	5.4956×10^{-10}	6.9611×10^{-10}
ALINEA		—	2.6107×10^{-2}	$< 1 \times 10^{-17}$	3.9563×10^{-6}
PI-ALINEA			—	$< 1 \times 10^{-17}$	1.8811×10^{-9}
Q-Learning				—	1.3157×10^{-11}
kNN-TD					—
Mean	887.07	499.81	493.16	556.02	518.66

TABLE 6.24: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	4.7739×10^{-15}	$< 1 \times 10^{-17}$	5.2711×10^{-10}	$< 1 \times 10^{-17}$
ALINEA		—	6.2847×10^{-1}	4.1576×10^{-10}	1.4199×10^{-9}
PI-ALINEA			—	1.3145×10^{-11}	3.1801×10^{-11}
Q-Learning				—	1.7473×10^{-2}
kNN-TD					—
Mean	45.40	428.94	451.72	239.16	310.36

TABLE 6.25: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	7.9940×10^{-15}	6.6599×10^{-15}	6.5836×10^{-13}	4.3632×10^{-13}
ALINEA		—	3.1771×10^{-4}	$< 1 \times 10^{-17}$	1.2145×10^{-8}
PI-ALINEA			—	$< 1 \times 10^{-17}$	8.1881×10^{-11}
Q-Learning				—	2.6794×10^{-11}
kNN-TD					—
Mean	6.18	3.48	3.44	3.87	3.60

TABLE 6.26: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.8809×10^{-15}	2.1090×10^{-15}	9.5070×10^{-9}	$< 1 \times 10^{-17}$
ALINEA		—	6.2002×10^{-1}	6.6082×10^{-8}	5.6063×10^{-8}
PI-ALINEA			—	5.1779×10^{-9}	8.4921×10^{-10}
Q-Learning				—	8.614×10^{-2}
kNN-TD					—
Mean	1.63	15.62	16.36	8.97	11.47

TABLE 6.27: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	8.0935×10^{-13}	5.4734×10^{-14}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	2.5139×10^{-4}	4.5404×10^{-10}	5.8632×10^{-3}
PI-ALINEA			–	$< 1 \times 10^{-17}$	1.9554×10^{-7}
Q-Learning				—	2.9339×10^{-7}
kNN-TD					—
Mean	22.19	6.26	5.54	9.32	7.14

TABLE 6.28: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	1.2704×10^{-8}	$< 1 \times 10^{-17}$
ALINEA		—	9.8182×10^{-1}	9.3808×10^{-3}	3.0570×10^{-4}
PI-ALINEA			—	6.3021×10^{-3}	1.0409×10^{-4}
Q-Learning				—	9.9999×10^{-1}
kNN-TD					—
Mean	2.37	34.38	34.89	26.87	26.76

Scenario 4

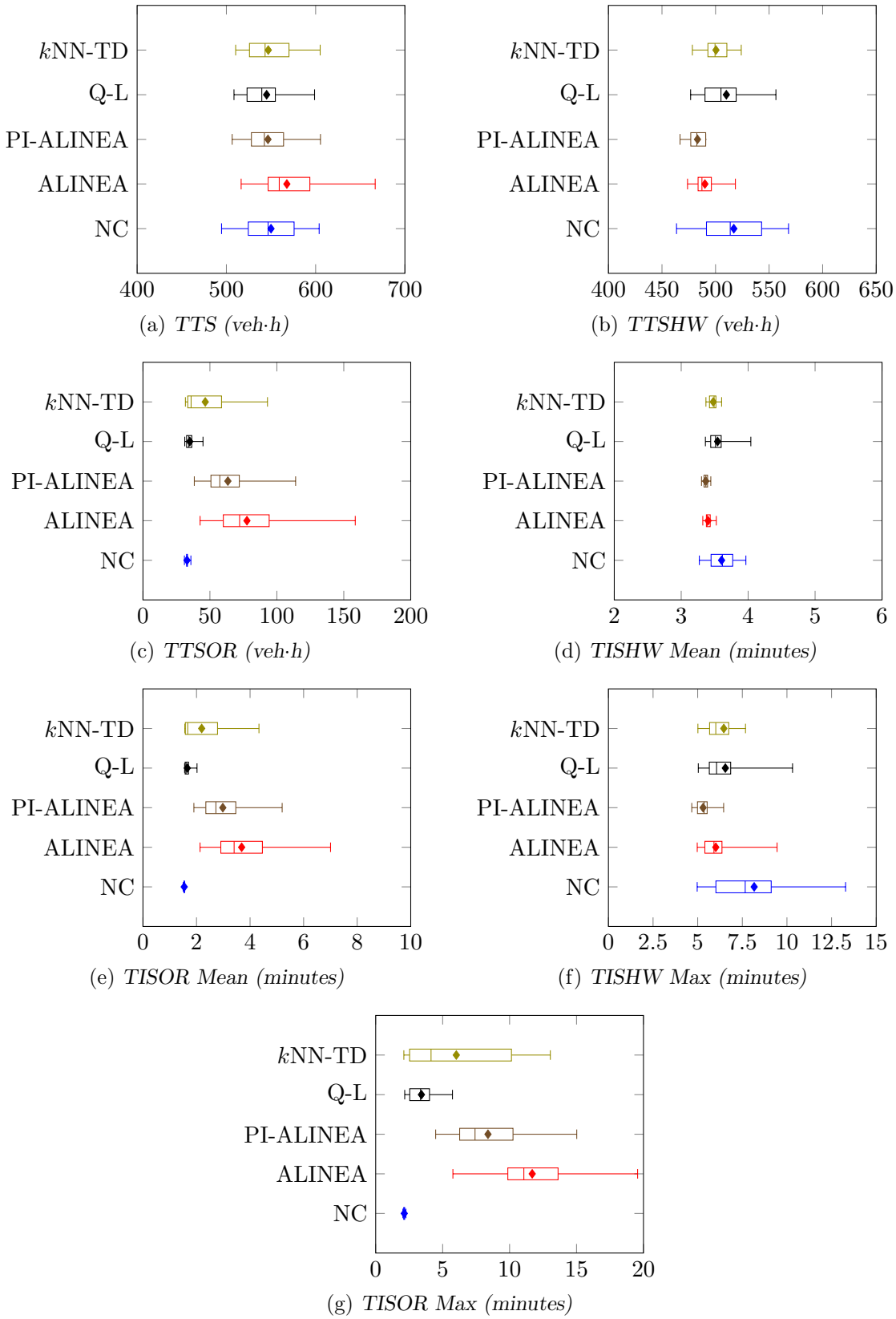
As in all three scenarios above, the ANOVA test performed on the PMI-values returned by the algorithms in the case of Scenario 4 revealed that there are, in fact, statistical differences at a 5% level of significance between the means returned by the algorithms in respect of all seven PMIs, as may be seen from the p-values presented in Table 6.29. The Levene test furthermore revealed that the variances returned by the algorithms were only found to be statistically indistinguishable at a 5% level of significance in respect of the TTS. As may be seen in the table, the variances between at least some pair of algorithms' output in respect of each of the other six PMIs were found to differ statistically at a 5% level of significance, and hence the Games-Howell test was subsequently performed in order to determine between which pairs of algorithmic output the differences occur in respect of all of these PMIs, while the Fisher LSD test was performed for this purpose in respect of the TTS.

TABLE 6.29: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 4. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	550.00	567.76	546.58	545.10	546.93	1.6027×10^{-2}	1.6449×10^{-1}
TTSHW	517.07	490.10	483.15	510.10	500.40	2.2642×10^{-3}	1.5810×10^{-13}
TTSOR	32.93	77.66	63.43	35.00	46.53	$< 1 \times 10^{-17}$	1.4433×10^{-15}
TISHW Mean	3.60	3.40	3.37	3.54	3.48	1.6209×10^{-13}	1.1102×10^{-16}
TISOR Mean	1.54	3.69	2.98	1.65	2.19	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Max	8.16	6.01	5.30	6.55	6.46	1.2329×10^{-9}	1.6780×10^{-9}
TISOR Max	2.13	11.69	8.39	3.42	6.02	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$

As outlined in §5.3.2, Scenario 4 represents the smallest overall traffic demand. As a result, it was expected that RM would be the least effective in this scenario. As may be seen in Figure 6.8(a), this expectation was confirmed, with the box plots of all four cases spanning roughly the same interval, the only exception being ALINEA for which an increase in the variance of the TTS was observed. As may be seen from the p-values in Table 6.30, ALINEA was outperformed by all other implementations at a 5% level of significance, while no statistical differences were observed in the performances of any of the other algorithms at a 5% level of significance. Q-Learning achieved the smallest TTS-value, followed by PI-ALINEA, kNN-TD and the no-control case, while, ALINEA returned the largest TTS-value.

Interestingly, ALINEA and PI-ALINEA were able to outperform both the Q-Learning and kNN-TD RM algorithms, as well as the no-control case in respect of the TTSHW, while the latter



(a) *TTS (veh·h)* (b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)* (d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)* (f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.8: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 4.*

two were found not to perform statistically different from the no-control case at a 5% level of significance. These results are summarised in Table 6.31. As may be seen in Figure 6.8(b), the dominance of ALINEA and PI-ALINEA is due to the combination of an absolute reduction in the TTSHW, and a smaller variance from the minimum value, which resulted in the improvement of the mean values.

As may have been expected, the improvements achieved by ALINEA and PI-ALINEA in respect of the TTSHW are offset by a deterioration in the TTSOR. This deterioration is clearly visible in the box plots of Figure 6.8(c). Interestingly, a similar increase in the TTSOR is observed for kNN-TD RM. As shown in Table 6.32, the no-control case was able to outperform all four of the RM implementations. The no-control case is followed by Q-Learning in the order of relative algorithmic performances, as Q-Learning outperformed kNN-TD RM, ALINEA and PI-ALINEA at a 5% level of significance. The kNN-TD implementation returned the next best performance, outperforming both ALINEA and PI-ALINEA, while the performances of the latter two were found to be statistically indistinguishable at a 5% level of significance.

When employing the PI-ALINEA control policy, a mean travel time of 3.37 minutes was achieved for vehicles travelling along the highway only. This value was small enough to be able to outperform all other algorithms, as well as the no-control case, as may be seen in Table 6.33. The same pattern emerges for the maximum TISHW, where PI-ALINEA again outperformed all other algorithms, limiting the maximum TISHW to a value of 5.30 minutes. For both of these PMIs, PI-ALINEA is followed by ALINEA in the order of relative algorithmic performances, as ALINEA outperformed both Q-Learning and kNN-TD RM in respect of the mean TISHW, while the performance of ALINEA was statistically on par with that of the two RL approaches in respect of the maximum TISHW. The performance of kNN-TD RM and Q-Learning was found to be statistically indistinguishable at a 5% level of significance in respect of both of these PMIs. In respect of the maximum TISHW, however, only kNN-TD RM was able to outperform the no-control case, while in respect of the mean TISHW the no-control case was outperformed by both the RL implementations. This hierarchy of algorithmic performances is also evident in the box plots of Figures 6.8(d) and 6.8(f). As may be seen from the figures, the improved performances by the feedback controllers are not due to an absolute reduction in respect of the travel times by vehicles along the highway, but rather due to reduced variances in these travel times, as may be seen from the smaller interquartile ranges of the corresponding box plots in the figures.

The differences in respect of the travel times of vehicles joining the highway from the on-ramp when comparing ALINEA and PI-ALINEA with the no-control case, as well as the RL RM implementations, are again evident in Figures 6.8(e) and 6.8(g), where a clear increase in both the mean TISOR and maximum TISOR may be seen for ALINEA, PI-ALINEA and kNN-TD RM. The no-control case was again able to outperform all of the RM implementations in respect of both the mean and maximum TISOR-values, as may be seen in Tables 6.34 and 6.36. Q-Learning returned the next-best performance, achieving increases in the mean and maximum TISOR-values of only 7.14% and 60.56% respectively, thereby outperforming all other RM implementations in respect of both these PMIs. Q-Learning was followed by kNN-TD RM, which returned increases of 42.21% and 182.63%, respectively, in respect of the mean and maximum TISOR, outperforming both ALINEA and PI-ALINEA at a 5% level of significance. The performances of ALINEA and PI-ALINEA were found not to differ statistically at a 5% level of significance in respect of the mean TISOR, as the control strategies resulted in increases of the mean travel time of 139.61% and 93.51%, respectively. Finally, PI-ALINEA outperformed ALINEA at a 5% level of significance in respect of the maximum TISOR, as motorists could expect to travel 393.90% or 548.83% longer than in the no-control case in the worst-case scenario.

TABLE 6.30: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.8789×10^{-2}	6.4781×10^{-1}	5.1275×10^{-1}	6.8189×10^{-1}
ALINEA		—	5.2504×10^{-3}	2.8735×10^{-3}	6.0308×10^{-3}
PI-ALINEA			—	8.4301×10^{-1}	9.6253×10^{-1}
Q-Learning				—	8.0644×10^{-1}
kNN-TD					—
Mean	550.00	567.76	546.58	545.10	546.93

TABLE 6.31: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.3739×10^{-4}	1.4707×10^{-5}	8.7566×10^{-1}	5.9648×10^{-2}
ALINEA		—	4.9426×10^{-2}	2.9734×10^{-3}	4.1369×10^{-3}
PI-ALINEA			—	4.1690×10^{-5}	1.2531×10^{-7}
Q-Learning				—	3.4116×10^{-1}
kNN-TD					—
Mean	517.07	490.10	483.15	510.10	500.40

TABLE 6.32: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.9216×10^{-9}	6.2060×10^{-9}	1.3446×10^{-2}	3.2264×10^{-3}
ALINEA		—	1.1607×10^{-1}	4.9473×10^{-9}	1.7059×10^{-5}
PI-ALINEA			—	2.4458×10^{-8}	7.0248×10^{-3}
Q-Learning				—	1.6495×10^{-2}
kNN-TD					—
Mean	32.93	77.66	63.43	35.00	46.53

TABLE 6.33: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	3.2276×10^{-5}	2.1358×10^{-6}	6.8898×10^{-1}	1.4588×10^{-2}
ALINEA		—	1.6432×10^{-2}	3.2466×10^{-4}	1.1718×10^{-5}
PI-ALINEA			—	1.0884×10^{-5}	6.1642×10^{-10}
Q-Learning				—	2.3630×10^{-1}
kNN-TD					—
Mean	3.60	3.40	3.37	3.54	3.48

TABLE 6.34: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	4.3571×10^{-10}	2.9771×10^{-9}	1.3720×10^{-4}	2.1368×10^{-3}
ALINEA		—	6.8520×10^{-2}	1.3548×10^{-9}	5.3421×10^{-6}
PI-ALINEA			—	1.5598×10^{-8}	5.8431×10^{-3}
Q-Learning				—	1.4647×10^{-2}
kNN-TD					—
Mean	1.54	3.69	2.98	1.65	2.19

TABLE 6.35: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.8596×10^{-3}	2.6307×10^{-5}	4.1509×10^{-2}	2.7965×10^{-2}
ALINEA		—	2.9785×10^{-3}	3.9119×10^{-1}	5.6738×10^{-1}
PI-ALINEA			—	2.5957×10^{-4}	6.7367×10^{-4}
Q-Learning				—	9.9914×10^{-1}
kNN-TD					—
Mean	8.16	6.01	5.30	6.55	6.46

TABLE 6.36: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	2.3547×10^{-12}	5.2558×10^{-7}	5.0068×10^{-5}
ALINEA		—	5.0282×10^{-4}	$< 1 \times 10^{-17}$	5.4022×10^{-7}
PI-ALINEA			—	2.2692×10^{-10}	5.7636×10^{-4}
Q-Learning				—	8.3283×10^{-3}
kNN-TD					—
Mean	2.13	11.69	8.39	3.42	6.02

Discussion

Although its performance was statistically indistinguishable from that of ALINEA, PI-ALINEA and Q-Learning in Scenario 2, Q-Learning in Scenario 3, and Q-Learning, PI-ALINEA and the no-control case in Scenario 4 at a 5% level of significance, the kNN-TD RM implementation was able to consistently achieve either the smallest or the second-smallest value in respect of the TTS of all algorithms over all four scenarios, never once being outperformed in respect of the TTS. The second-best performing algorithm in respect of the TTS in all four scenarios was Q-Learning which was only outperformed once by kNN-TD RM, namely in Scenario 1, while it was able to outperform PI-ALINEA in Scenario 1, ALINEA and PI-ALINEA in Scenario 3, and ALINEA in Scenario 4, and it was able to match the performance of ALINEA and PI-ALINEA in all other scenarios in respect of the TTS. ALINEA was, however, able to achieve significant

improvements over the no-control case in Scenarios 1 and 2, while it did not lead to a statistically significant increase in the TTS in Scenario 3 at a 5% level of significance. In Secnario 4, however, ALINEA resulted in a statistically significant increase in the TTS at a 5% level of significance. PI-ALINEA, on the other hand, was also able to improve on the no-control case in respect of the TTS in Scenarios 1 and 2, while it did not lead to statistically significant increases in respect of the TTS at a 5% level of significance in Scenarios 3 and 4.

ALINEA and PI-ALINEA were shown to be the most effective algorithms in terms of protecting the highway flow, never being outperformed in respect of the TTSHW, mean TISHW or maximum TISHW PMIs. Their strong performances in terms of the highway traffic flow were, however, offset by poor performances in terms of on-ramp traffic flow. This is evident from the fact that ALINEA and PI-ALINEA consistently achieved the largest values in respect of the TTSOR, mean TISOR and maximum TISOR. ALINEA managed to return the smallest TTSHW-values in all four scenarios, while PI-ALINEA achieved the second-smallest TTSHW-value in each of the four scenarios. In respect of the TTSOR, ALINEA returned the largest values in Scenarios 1 and 3, and the second largest TTSOR-values in Scenarios 2 and 4, while PI-ALINEA returned the largest TTSOR-values in Scenarios 2 and 4, and the second-largest TTSOR-values in Scenarios 1 and 3.

The kNN-TD RM implementation was able to match the performance of ALINEA and PI-ALINEA in terms of the TTSHW, mean TISHW and maximum TISHW in Scenario 1, while outperforming the two algorithms in terms of the TTSOR, mean TISOR and maximum TISOR in Scenario 1 at a 5% level of significance. In Scenario 2, the kNN-TD algorithm was, however, outperformed by ALINEA and PI-ALINEA in terms of the TTSHW, mean TISHW and maximum TISHW, while outperfroming the feedback controllers in respect of the TTSOR, mean and maximum TTSOR-values at a 5% level of significance. In Scenarios 3 and 4, the approach taken by the kNN-TD RM agent seemed to change, as marginally less emphasis was placed on protecting the highway flow in favour of finding a better balance in terms of the on-ramp queue. This is evident from the fact that in Scenario 3, kNN-TD RM outperformed both ALINEA and PI-ALINEA in respect of the TTSOR, mean TISOR and maximum TISOR-values, while it was outperformed by ALINEA and PI-ALINEA in terms of the TTSHW, mean TISHW and maximum TISHW. In Scenario 4, the kNN-TD RM agent was able to recognise that there is reduced traffic demand and was thus able to reduce the metering rate in such a manner that it did not perform worse than the no-control case in any of the PMI values corresponding to those vehicles travelling along the highway only, while the increases in respect of the travel times on the on-ramp were small enough not to compromise the gains in TTSHW and thereby result in increases in the TTS, while simultaneously not being able to improve on the no-control case.

Q-Learning typically found a middle ground between protecting the highway flow and balancing the on-ramp queue. This may be seen from the fact that it was outperformed by ALINEA and PI-ALINEA in all four scenarios in respect of the TTSHW, mean TISHW and maximum TISHW, but it was consistently able to outperform ALINEA and PI-ALINEA in respect of the TTSOR, mean TISOR and maximum TISOR.

In summary, both of the RL implementations demonstrated an ability to adapt well to changes in the demand profile, never resulting in an increase in the TTS over the no-control case. As expected, the more complex kNN-TD RM implementation generally performed marginally better than Q-Learning when considering purely the TTS-value. Therefore, kNN-TD is judged to be the best performing algorithm based on the results from all four scenarios analysed in this section.

6.6 Ramp Metering with a Queueing Consideration

From the results presented in the previous section it is evident that RM, in its original incarnation may often lead to long on-ramp queues which result in excessively long travel times by vehicles joining the highway from the on-ramp. The increase in travel time is not, however, the only problem associated with the build-up of these long on-ramp queues. Typically, the on-ramp originates in an urban traffic network, and the on-ramp queue may propagate back into this arterial network which may, in turn, cause major congestion problems in this network [150]. Therefore, queueing considerations also have to be implemented in the context of each of the four RM controllers.

6.6.1 ALINEA and PI-ALINEA with Queue Limits

The simplest, and perhaps most popular, countermeasure for regulating the on-ramp queue length is to place a detector at the ramp entrance and terminate ramp metering when the loop detector occupancy exceeds a certain threshold. This approach may, however, yield an oscillatory override behaviour as well as underutilisation of the available storage space on the on-ramp [45]. This may be avoided by tighter on-ramp queue control under the assumption that a good estimate of the current queue length is available. Under this assumption, Smaragdis and Papageorgiou [150] designed an on-ramp queue length controller which may be employed in conjunction with any controller yielding a metering rate $r(t)$ as output. This controller takes as input a maximum allowable queue length $\hat{w}$, the estimate of the queue length at time t, denoted by $w(t)$, an estimate of the on-ramp demand during the previous time period, denoted by $d(t-1)$, and the control interval length T. Given this information, the controller returns a metering rate

$$r'(t) = -\frac{1}{T}\left[\hat{w} - w(t)\right] + d(t-1) \tag{6.9}$$

with the aim of maintaining an on-ramp queue length as close to the maximum permissible queue length $\hat{w}$ as possible. Naturally, it would not make sense to regulate the on-ramp queue length in cases where the on-ramp demand is low, as this may induce unnecessary on ramp queue formation. The final metering rate to be applied is therefore given by

$$r''(t) = \max\left[r(t), r'(t)\right], \tag{6.10}$$

where $r(t)$ denotes the metering rate set by a ramp metering strategy such as ALINEA or PI-ALINEA. Employing this approach results in the ramp metering strategy being employed until the on-ramp queue reaches the maximum permissible value, at which point in time the metering rate determined in (6.9) is employed to maintain an acceptable on-ramp queue.

The control law in (6.9) was implemented, and its effectiveness evaluated for a maximum permissible queue length $\hat{w} = 100$ vehicles. The results of this investigation are presented in Table 6.37. As may be seen in the table, when employing this combination of controllers, the on-ramp queue formation may be limited to a value close to the maximum permissible queue length value. Note that the values presented are, again, the average maximum queue length values obtained over thirty simulation runs. The fact that the values returned are marginally larger than the maximum allowable value of 100, may be attributed to the fact that the queue limit controller is only activated once the maximum permissible queue limit has been reached and, as a result, the increased metering rate is only employed during the time period following the period in which the maximum permissible value was reached for the first time. Finally, as may be seen from the results in the table, the maximum allowable queue length was never reached in Scenario 4, and

so the queue limit controller is never triggered, resulting in the fact that the same maximum queue length values were returned by both controllers, with or without the queue limitation in place.

TABLE 6.37: *Queue limit effectiveness evaluation results for ALINEA and PI-ALINEA, measured in terms of the average maximum queue length reached (in number of vehicles).*

	Scenario 1		Scenario 2	
Queue Limit	ALINEA	PI-ALINEA	ALINEA	PI-ALINEA
∞	607	596	269	247
100	110	110	109	108
	Scenario 3		Scenario 4	
Queue Limit	ALINEA	PI-ALINEA	ALINEA	PI-ALINEA
∞	345	335	62	40
100	110	111	62	40

6.6.2 Q-Learning and kNN-TD with Queue Limits

A popular technique for preventing an RL agent from reaching certain states is to provide negative feedback through some sort of additional punishment embedded into the reward function for those specific states. This approach was, for example, employed by Li *et al.* [85] in their RL implementation for VSLs to punish the agent when severely congested states were reached. This approach towards punishing the learning agent is employed in this dissertation in order to enforce queue limitations. The reward function is thus adjusted such that the reward achieved by the agent is

$$r(t) = \begin{cases} -(\hat{\rho} - \rho_{\text{ds}}(t))^2 & \text{if } w(t) < \hat{w}, \\ -(\hat{\rho} - \rho_{\text{ds}}(t))^2 - \zeta & \text{otherwise,} \end{cases} \tag{6.11}$$

where $w(t)$ denotes the queue length measured during time interval t, $\hat{w}$ denotes the maximum permissible queue length and ζ is a scalar value employed as the punishment for the agent.

The adjusted reward function was implemented for both the Q-Learning and kNN-TD RM agents, with maximum permissible queue length $\hat{w} = 100$ and the punishment for excessively long on-ramp queue set to $\zeta = -100\,000$. The results from this evaluation are presented in Table 6.38. As may be seen in the table, employing the punishment in order to prevent the formation of excessively long on-ramp queues was effective for both Q-Learning and kNN-TD for RM.

As may have been expected, limiting the metering rate in order to prevent excessively long on-ramp queues from forming impairs the performance of the RM strategies. This is clearly visible from the summarised results presented in Table 6.39. Naturally, the decreases in the TTSHW achieved in conventional RM cannot be achieved when queue limits are applied. The implementation of queue limits does, however, result in smaller increases in the TTSOR when compared with the case where queue limits are not enforced.

6.6.3 Algorithmic Comparison

In this section, the simulation results and the relative algorithmic performances in the context of RM with the incorporation of on-ramp queue limitations are compared. This comparison

TABLE 6.38: *Queue limit effectiveness evaluation results for Q-Learning and kNN-TD learning, measured in terms of the average maximum queue length reached (in number of vehicles).*

	Scenario 1		Scenario 2	
Queue Limit	Q-Learning	kNN-TD	Q-Learning	kNN-TD
∞	543	581	136	229
100	103	41	80	59
	Scenario 3		Scenario 4	
Queue Limit	Q-Learning	kNN-TD	Q-Learning	kNN-TD
∞	163	257	0	21
100	107	54	0	28

TABLE 6.39: *The effect of employing queue limitations in the RM implementations overall performance.*

	Scenario 1							
	ALINEA		PI-ALINEA		Q-Learning		kNN-TD	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS	1 766.00	1 470.25	1 753.68	1 517.30	1 561.05	1 451.80	1 640.32	1 398.80
TTSHW	1 488.75	582.04	1 475.46	596.03	1 323.47	631.19	1 555.85	606.16
TTSOR	277.25	888.21	278.22	921.28	237.58	820.61	84.47	792.63
	Scenario 2							
	ALINEA		PI-ALINEA		Q-Learning		kNN-TD	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS	1 033.93	873.49	990.95	877.90	918.68	869.87	941.17	860.61
TTSHW	830.38	561.21	791.45	574.68	759.88	689.93	832.22	610.40
TTSOR	203.54	312.28	199.50	303.22	158.79	179.94	108.95	250.21
	Scenario 3							
	ALINEA		PI-ALINEA		Q-Learning		kNN-TD	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS	911.23	928.75	944.46	944.88	815.96	795.18	871.34	829.02
TTSHW	699.32	499.81	723.97	493.16	676.46	556.02	771.09	518.66
TTSOR	211.91	428.93	220.49	451.72	139.50	239.16	100.25	310.36
	Scenario 4							
	ALINEA		PI-ALINEA		Q-Learning		kNN-TD	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS	570.31	567.76	549.32	546.58	550.31	545.10	545.19	546.93
TTSHW	491.21	490.09	483.57	483.15	516.95	510.10	493.60	500.40
TTSOR	79.10	77.66	65.75	63.43	33.36	35.00	51.59	46.53

is again performed in the context of the four different scenarios of traffic demand described in §5.3.2. As in the previous section, the results are presented and interpreted through the use of box plots in which the means, medians and interquartile ranges of the PMI-values are indicated, as well as tables indicating whether or not statistical differences exist between the PMI values for each pair of algorithms at a 5% level of significance.

Scenario 1

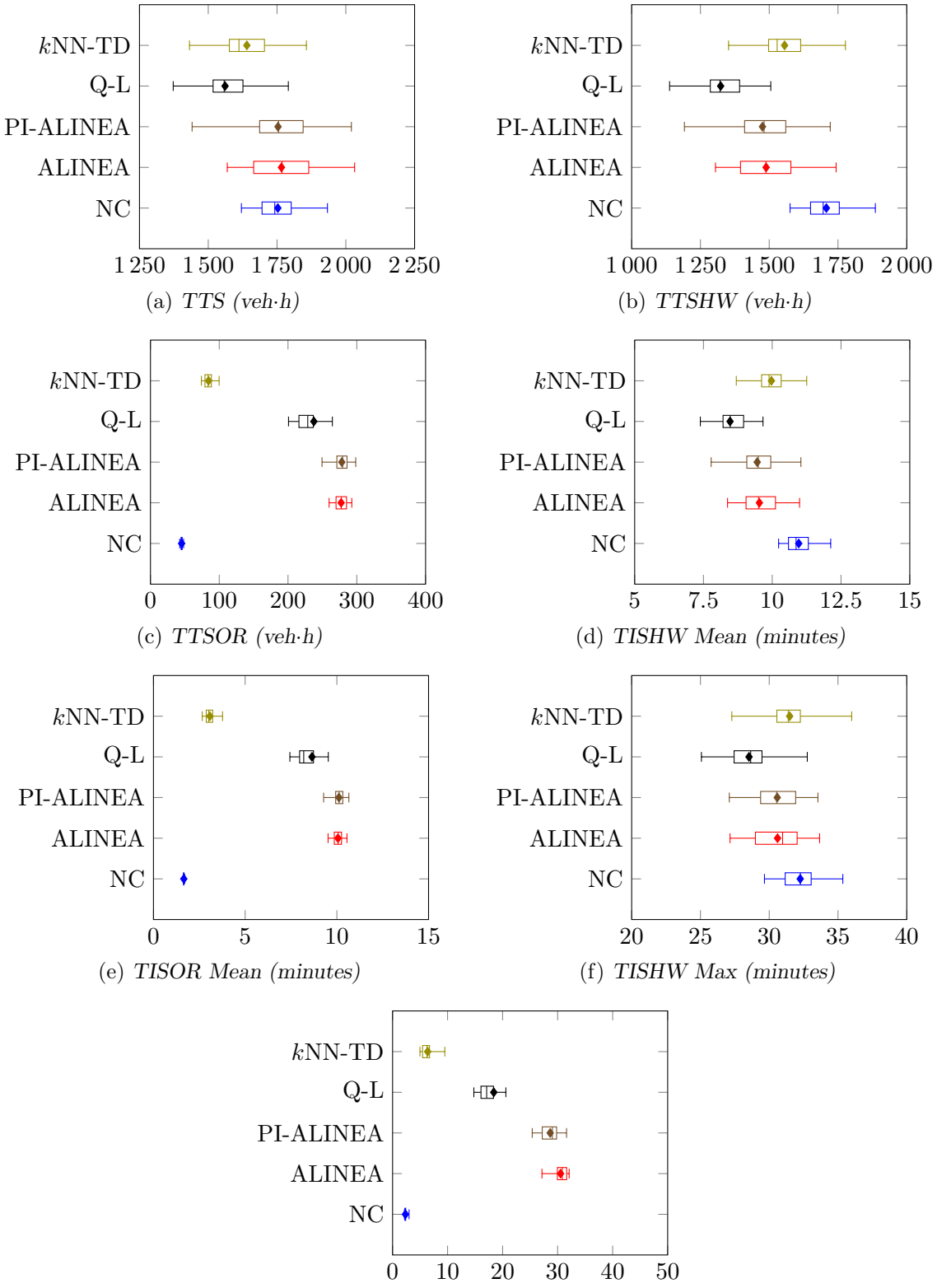
The p-values of the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms for Scenario 1, are presented in Table 6.40. As may be seen in the table, the ANOVA revealed that there are, in fact, statistical differences between the means returned by at least some pair of algorithms in respect of each of the seven PMIs. Furthermore, Levene's test revealed that the variances of the PMI-values returned by the algorithms were only statistically indistinguishable at a 5% level of significance in respect of the TTS, mean TISHW and maximum TISHW PMIs, while the variances between at least some pair of algorithmic output data were found to differ statistically for the other four PMIs at a 5% level of significance. Therefore, the Fisher LSD test was employed in order to ascertain between which pairs of algorithmic outputs the differences between the algorithmic performances occur in respect of the TTS, mean TISHW and maximum TISHW PMIs, while the Games-Howell test was employed for this purpose in respect of the other four PMIs.

TABLE 6.40: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in the case of RM with queue limits in Scenario 1. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

PMI	Mean value					p-value	
	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 753.01	1 766.00	1 753.68	1 561.05	1 640.32	1.3545×10^{-13}	2.0465×10^{-1}
TTSHW	1 707.70	1 488.75	1 475.46	1 323.47	1 555.85	$< 1 \times 10^{-17}$	2.1738×10^{-1}
TTSOR	45.31	277.25	278.22	237.58	84.47	$< 1 \times 10^{-17}$	6.7252×10^{-4}
TISHW Mean	10.96	9.53	9.45	8.47	9.97	$< 1 \times 10^{-17}$	2.5698×10^{-1}
TISOR Mean	1.66	10.06	10.11	8.64	3.07	$< 1 \times 10^{-17}$	4.3682×10^{-4}
TISHW Max	32.25	30.61	30.58	28.53	31.49	7.6883×10^{-13}	7.8705×10^{-1}
TISOR Max	2.34	30.53	28.65	18.40	6.38	$< 1 \times 10^{-17}$	4.5092×10^{-3}

As is evident from the box plots in Figure 6.9(a), only the RL implementations were able to achieve statistically significant improvements over the no-control case in respect of the TTS. This is corroborated by the p-values presented in Table 6.41. Q-Learning achieved the smallest TTS-value, outperforming kNN-TD, ALINEA, PI-ALINEA and the no-control case, as it still managed to achieve a 10.95% improvement over the no-control case. Q-Learning is followed by kNN-TD in the order of relative algorithmic performances, which was able to achieve a 6.43% improvement over the no-control case, thereby outperforming ALINEA, PI-ALINEA and the no-control case at a 5% level of significance. The performances of ALINEA, PI-ALINEA and the no-control case were found to be statistically indistinguishable in respect of the TTS, as ALINEA achieved an increase in the TTS of 0.74% over the no-control case, while PI-ALINEA returned an increase of 0.04% over the no-control case in respect of the TTS.

Although the savings in the travel times for vehicles travelling along the highway only were not as pronounced when queue limitations are implemented, the savings to be made in respect of the TTS were still achieved on the highway. This trend is clearly visible in the box plot in Figure 6.9(b), as all of the RM implementations were able to outperform the no-control case at a 5% level of significance. Q-Learning again yielded the best performance, outperforming all other RM implementations at a 5% level of significance, as it returned a TTSHW-value of 1 323.47 veh·h. Q-Learning was followed in the order of relative algorithmic performances by ALINEA and PI-ALINEA, which achieved TTSHW-values of 1 488.75 veh·h and 1 475.46 veh·h, respectively. As may be seen in Table 6.42, the performances of ALINEA and PI-ALINEA were found to be statistically indistinguishable at a 5% level of significance. Finally, kNN-TD which returned a TTSHW-value of 1 640.32 veh·h, concludes the order of relative algorithmic performances.



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.9: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation with queue limits in Scenario 1.*

Perhaps unexpectedly, the order of relative algorithmic performances in respect of the TTSOR is not the opposite of the order in respect of the TTSHW. Naturally, the no-control case achieved the smallest TTSOR-value, as no RM is employed in this case. The no-control case is followed by kNN-TD learning in the order of relative algorithmic performances, which returned a TTSOR-value of 84.47 veh·h — a value small enough to allow it to outperform the other three RM implementations at a 5% level of significance, as may be seen from the p-values in Table 6.43. The kNN-TD implementation is followed by Q-Learning in the order of algorithmic performances, which returned a TTSOR-value of 237.58 veh·h, outperforming both ALINEA and PI-ALINEA at a 5% level of significance. Finally, the performances of ALINEA and PI-ALINEA were again found to be statistically similar at a 5% level of significance, as they returned TTSOR-values of 277.25 veh·h and 278.22 veh·h, respectively. This similarity and the order of the relative algorithmic performances are very clear in the box plots in Figure 6.9(c).

As may have been expected, the relative algorithmic performances in respect of both the mean and maximum TISHW are the same as that for the TTSHW, as may be seen in Figures 6.9(d) and 6.9(f), respectively. Once again, Q-Learning achieved the best performance in respect of both of these PMIs, outperforming all other implementations. Q-Learning achieved reductions of 22.72% and 11.53% over the no-control case in respect of the mean and maximum TISHW PMIs, respectively. The performance of Q-Learning was again followed by ALINEA and PI-ALINEA, whose performances were found to be statistically indistinguishable at a 5% level of significance for both of these PMIs, while they were both able to outperform kNN-TD learning in respect of the mean TISHW, as may be deduced from the p-values in Table 6.44. In respect of the maximum TISHW, only PI-ALINEA was able to outperform kNN-TD learning, while the performances of ALINEA and kNN-TD learning were found to be statistically indistinguishable at a 5% level of significance, as may be seen from the p-values in Table 6.46. ALINEA and PI-ALINEA were able to outperform the no-control case at a 5% level of significance in respect of both of these PMIs as they achieved reductions of 13.05% and 13.78%, respectively, in respect of the mean TISHW, while reductions of 5.09% and 5.19%, were recorded over the no-control case in respect of the maximum TISHW. The 9.03% reduction recorded by the kNN-TD implementation was large enough for its performance to be classified as statistically different from the no-control case at a 5% level of significance in respect of the mean TISHW, while in respect of the maximum TISHW, kNN-TD and the no-control case were found to perform statistically similarly at a 5% level of significance, as may be seen in Tables 6.44 and 6.46.

Naturally, due to RM being employed, increases were again to be expected in respect of both the mean and maximum TISHW-values. This trend is clearly visible in the box plots in Figures 6.9(e) and 6.9(g). From these box plots it is evident that the order of relative algorithmic performances in respect of both of these PMIs is the same as that for the TTSOR. As was the case with the TTSOR, the kNN-TD implementation was able to outperform all three of the other RM implementations at a 5% level of significance in respect of both of these PMIs, as may be seen in Tables 6.45 and 6.47. The kNN-TD implementation was again followed by Q-Learning in the order of relative algorithmic performances in respect of both of these PMIs, as Q-Learning was able to outperform both ALINEA and PI-ALINEA at a 5% level of significance. Although the performances of ALINEA and PI-ALINEA were found to be statistically similar at a 5% level of significance in respect of the mean TISOR, PI-ALINEA was able to outperform ALINEA in respect of the maximum TISOR.

TABLE 6.41: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	6.4392×10^{-1}	9.8108×10^{-1}	1.9841×10^{-10}	9.3463×10^{-5}
ALINEA		—	6.6099×10^{-1}	1.6573×10^{-11}	1.4877×10^{-5}
PI-ALINEA			—	1.7501×10^{-10}	8.5348×10^{-5}
Q-Learning				—	5.3551×10^{-3}
kNN-TD					—
Mean	1 753.01	1 766.00	1 753.68	1 561.05	1 640.32

TABLE 6.42: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.3134×10^{-13}	7.6605×10^{-15}	$< 1 \times 10^{-17}$	7.3733×10^{-8}
ALINEA		—	6.2036×10^{-1}	6.3145×10^{-9}	1.3280×10^{-2}
PI-ALINEA			—	7.1796×10^{-8}	3.1460×10^{-3}
Q-Learning				—	7.4385×10^{-15}
kNN-TD					—
Mean	1 707.70	1 488.75	1 475.46	1 323.47	1 555.85

TABLE 6.43: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.0182^{-14}	1.9651×10^{-14}	$< 1 \times 10^{-17}$	8.1934×10^{-14}
ALINEA		—	9.9610×10^{-1}	1.9554×10^{-3}	1.2599×10^{-12}
PI-ALINEA			—	1.5511×10^{-3}	$< 1 \times 10^{-17}$
Q-Learning				—	5.0071×10^{-14}
kNN-TD					—
Mean	45.31	277.25	278.22	237.58	84.47

TABLE 6.44: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	3.2196×10^{-15}	2.2204×10^{-16}	$< 1 \times 10^{-17}$	8.2798×10^{-9}
ALINEA		—	6.2541×10^{-1}	8.0178×10^{-10}	7.7016×10^{-3}
PI-ALINEA			—	9.6133×10^{-9}	1.7336^{-3}
Q-Learning				—	2.2204×10^{-16}
kNN-TD					—
Mean	10.96	9.53	9.45	8.47	9.97

TABLE 6.45: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	6.2169×10^{-15}	1.6650×10^{-16}	2.1090×10^{-16}	$< 1 \times 10^{-17}$
ALINEA		—	9.5957×10^{-1}	1.5334×10^{-3}	1.4887×10^{-11}
PI-ALINEA			—	1.0261×10^{-3}	1.1159×10^{-11}
Q-Learning				—	4.9405×10^{-14}
kNN-TD					—
Mean	1.66	10.06	10.11	8.64	3.07

TABLE 6.46: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	3.1470×10^{-4}	2.6424×10^{-4}	5.2958×10^{-14}	8.7909×10^{-2}
ALINEA		—	9.6145×10^{-1}	7.4874^{-6}	5.0336×10^{-2}
PI-ALINEA			—	9.1764×10^{-6}	4.5021×10^{-2}
Q-Learning				—	6.4776×10^{-10}
kNN-TD					—
Mean	32.25	30.61	30.58	28.53	31.49

TABLE 6.47: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.7089×10^{-14}	1.3767×10^{-14}	8.6708×10^{-14}	5.4290×10^{-14}
ALINEA		—	1.2029×10^{-3}	4.6050×10^{-11}	9.0099×10^{-12}
PI-ALINEA			—	2.4014×10^{-9}	$< 1 \times 10^{-17}$
Q-Learning				—	6.8636×10^{-11}
kNN-TD					—
Mean	2.34	30.53	28.56	18.40	6.38

Scenario 2

As in Scenario 1, the p-values returned by the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms for Scenario 2, as presented in Table 6.48, revealed that there are statistical differences between the means of at least some pair of algorithms in respect of all seven PMIs. Furthermore, Levene's test revealed that the variances in algorithmic output are statistically indistinguishable at a 5% level of significance for the TTS and maximum TISHW PMIs, while there are statistical differences between the variances of at least some pair of algorithmic output data in respect of all other PMIs. Therefore, the Fisher LSD test was performed in order to ascertain between which pairs of algorithmic output the differences occur in respect of the TTS and maximum TIHSW, while the Games-Howell test was employed for this purpose in respect of the other five PMIs.

All four RM implementations were able to achieve significant improvement over the no-control case in Scenario 2, as may be seen in Figure 6.10(a). The ALINEA control strategy was able to achieve a 9.45% reduction in the TTS over the no control case, while the PI-ALINEA, Q-Learning and kNN-TD RM implementations achieved 13.21%, 19.54% and 17.57% improvements, respectively. As may be seen from the p-values in Table 6.49, Q-Learning and kNN-TD returned the best performance as they were found to perform statistically indistinguishably at a 5% level of significance, while outperforming ALINEA and the no-control case. Q-Learing was also able to outperform PI-ALINEA, while the performances of PI-ALINEA and kNN-TD RM were found to be statistically similar at a 5% level of significance. Finally, unlike in Scenario 1, ALINEA and PI-ALINEA were both able to outperform the no-control case at a 95% level of confidence.

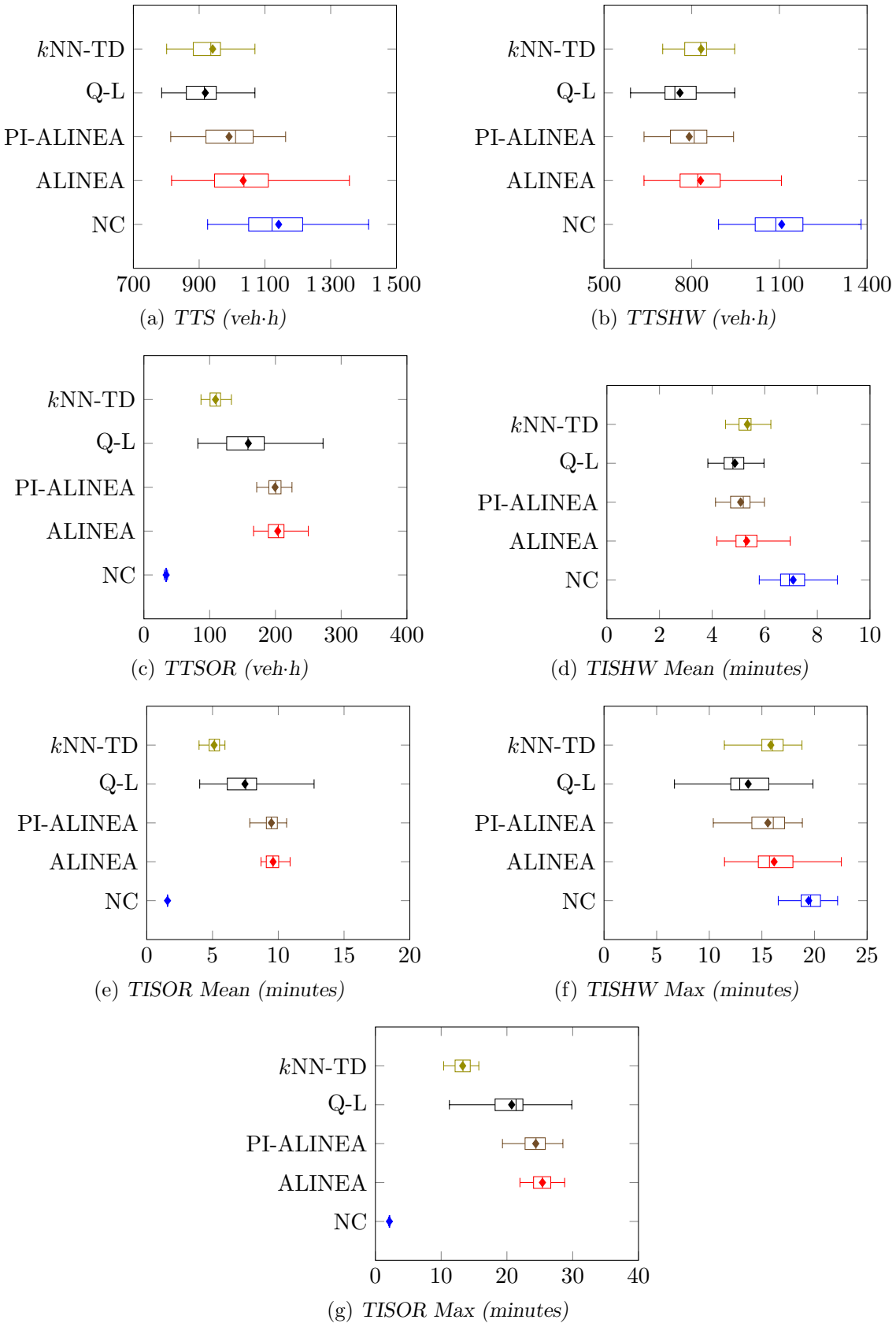
In a trend similar to that in Scenario 1, Q-Learning achieved the smallest TTSHW value of 759.45 veh·h, followed by PI-ALINEA and ALINEA with TTSHW-values of 791.45 veh·h and 830.38 veh·h, respectively, while kNN-TD RM returned a TTSHW-value of 832.22 veh·h. As is clearly visible in Figure 6.10(b), all of the RM implementations were able to outperform the no-control case, for which a TTSHW-value of 1 107.88 veh·h was recorded. This fact is corroborated by the p-values in Table 6.50. While Q-Learning was able to outperform both ALINEA and kNN-TD RM, its performance was found to be statistically indistinguishable from that of PI-ALINEA at a 5% level of significance. PI-ALINEA, on the other hand, was found to perform statistically similarly to both ALINEA and kNN-TD RM, while the latter two were also found to perform statistically indistinguishably from one another at a 5% level of significance.

Interestingly, in respect of the TTSOR, the performances of all the RM implementations, except ALINEA and PI-ALINEA, were found to differ statistically at a 5% level of significance in Scenario 2. Taking the natural increase in travel times for vehicles joining the highway from the on-ramp due to RM into account, kNN-TD outperformed all other RM implementations at a 5% level of significance as it returned a TTSOR-value of 108.95 veh·h. As may be seen from the p-values in Table 6.51, kNN-TD RM is followed by Q-Learning in the order of relative algorithmic performances, as Q-Learning was also able to outperform both ALINEA and PI-ALINEA at a 5% level of significance — these control measures returned TTSOR-values of 158.79 veh·h, 203.54 veh·h and 199.50 veh·h, respectively. The order of relative algorithmic performances, as well as the similarity in performance between ALINEA and PI-ALINEA, is also clearly visible in the box plots of Figure 6.10(c).

As may be seen in Figures 6.10(d) and 6.10(f), all four of the RM implementations were able to outperform the no-control in the mean and maximum TISHW PMIs. This fact is corroborated by the p-values in Tables 6.52 and 6.54. Q-Learning achieved the smallest mean and maximum TISHW-values, thereby outperforming ALINEA and kNN-TD RM at a 5% level of significance in respect of the mean TISHW, while Q-Learning outperformed all three of the other RM

TABLE 6.48: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 2. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

PMI	Mean value					p-value	
	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 141.80	1 033.93	990.95	918.68	941.17	6.6613×10^{-15}	5.6135×10^{-2}
TTSHW	1 107.88	830.38	791.45	759.88	832.22	$< 1 \times 10^{-17}$	4.2943×10^{-2}
TTSOR	33.92	203.54	199.50	158.79	108.95	$< 1 \times 10^{-17}$	9.9920×10^{-16}
TISHW Mean	7.08	5.31	5.08	4.87	5.34	$< 1 \times 10^{-17}$	4.9796×10^{-2}
TISOR Mean	1.58	9.61	9.48	7.47	5.12	$< 1 \times 10^{-17}$	1.1102×10^{-16}
TISHW Max	19.45	16.17	15.57	13.70	15.85	1.3212×10^{-14}	8.5139×10^{-2}
TISOR Max	2.13	25.40	24.38	20.72	13.27	$< 1 \times 10^{-17}$	6.6569×10^{-10}



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.10: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation with queue limits in Scenario 2.*

implementations at a 5% level of significance in respect of the maximum TISHW. ALINEA, PI-ALINEA and kNN-TD, on the other hand, were all found to perform statistically similarly at a 5% level of significance in respect of both the mean and maximum TISHW PMIs.

When considering the mean and maximum travel times for vehicles joining the highway from the on-ramp, kNN-TD RM again outperformed all three other RM implementations at a 5% level of significance in respect of both the mean and maximum TISOR PMIs, as may be seen from the p-values in Tables 6.53 and 6.55. Furthermore, Q-Learning was also able to outperform both ALINEA and PI-ALINEA at a 5% level of significance in respect of both of these PMIs, while the latter two were found to perform statistically similarly at a 5% level of significance. These trends are also clearly visible in the box plots of Figures 6.10(e) and 6.10(g). As may be seen in Table 6.53, kNN-TD RM was able to achieve a mean TISOR-value of 5.12 minutes, compared with 7.42 minutes for Q-Learning, 9.48 minutes for PI-ALINEA and 9.61 minutes for ALINEA. Furthermore, kNN-TD RM was able to limit the maximum TISOR to 13.27 minutes, while this value increased to 20.72 minutes for Q-Learning, 24.38 minutes for PI-ALINEA and 25.40 minutes for ALINEA, as may be seen in Table 6.55.

TABLE 6.49: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.2861×10^{-5}	3.5115×10^{-8}	1.0658×10^{-14}	1.4756×10^{-12}
ALINEA		—	9.9067×10^{-2}	1.6884×10^{-5}	4.6347×10^{-4}
PI-ALINEA			—	5.9510×10^{-3}	5.6480×10^{-2}
Q-Learning				—	3.8630×10^{-1}
kNN-TD					—
Mean	1 141.80	1033.93	990.95	918.68	941.17

TABLE 6.50: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.5033×10^{-12}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	5.0061×10^{-1}	2.4069×10^{-2}	9.9998×10^{-1}
PI-ALINEA			—	5.6885×10^{-1}	3.1744×10^{-1}
Q-Learning				—	3.8353×10^{-3}
kNN-TD					—
Mean	1 107.88	830.38	791.45	759.88	832.22

Scenario 3

As in Scenarios 1 and 2, an ANOVA test revealed that there are, again, statistical differences at a 5% level of significance between the means returned in Scenario 3 by at least some pair of algorithms in respect of all seven PMIs, as may be seen from the p-values presented in Table 6.56. As may be seen in the Table, the Levene test revealed that there are statistically significant differences in the variances of at least some pair of algorithms' output in respect of all PMIs except the maximum TISHW. Therefore, the Fisher LSD *post hoc* test is employed in order to ascertain between which pair of algorithmic output data the differences occur in respect

TABLE 6.51: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance..*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	4.3299×10^{-14}	8.2159×10^{-14}	3.2199×10^{-14}	6.8830×10^{-14}
ALINEA		—	8.5583×10^{-1}	2.4536×10^{-5}	$< 1 \times 10^{-17}$
PI-ALINEA			—	9.1325×10^{-5}	8.3522×10^{-12}
Q-Learning				—	2.6205×10^{-6}
kNN-TD					—
Mean	33.92	203.54	199.50	158.79	108.95

TABLE 6.52: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.6019×10^{-12}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
ALINEA		—	5.1209×10^{-1}	1.4585×10^{-2}	9.9977×10^{-1}
PI-ALINEA			—	4.2172×10^{-1}	2.7067×10^{-1}
Q-Learning				—	1.2400×10^{-3}
kNN-TD					—
Mean	7.08	5.31	5.08	4.87	5.34

TABLE 6.53: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	2.5540×10^{-15}	5.5509×10^{-15}	$< 1 \times 10^{-17}$
ALINEA		—	9.1040×10^{-1}	5.9401×10^{-6}	1.0439×10^{-11}
PI-ALINEA			—	1.874×10^{-5}	9.7837×10^{-12}
Q-Learning				—	9.7581×10^{-7}
kNN-TD					—
Mean	1.58	9.61	9.48	7.47	5.12

TABLE 6.54: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	3.7864×10^{-7}	3.5214×10^{-9}	2.2204×10^{-16}	3.2730×10^{-8}
ALINEA		—	3.3522×10^{-1}	1.0184×10^{-17}	6.0562×10^{-1}
PI-ALINEA			—	2.8974×10^{-3}	6.5381×10^{-1}
Q-Learning				—	6.6415×10^{-4}
kNN-TD					—
Mean	19.45	16.17	15.57	13.70	15.85

TABLE 6.55: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	7.7700×10^{-16}	$< 1 \times 10^{-17}$	2.6650×10^{-15}	$< 1 \times 10^{-17}$
ALINEA		—	3.3285×10^{-1}	1.3576×10^{-6}	4.8058×10^{-12}
PI-ALINEA			—	2.4739×10^{-4}	$< 1 \times 10^{-17}$
Q-Learning				—	1.1754×10^{-11}
kNN-TD					—
Mean	2.13	25.40	24.38	20.72	13.27

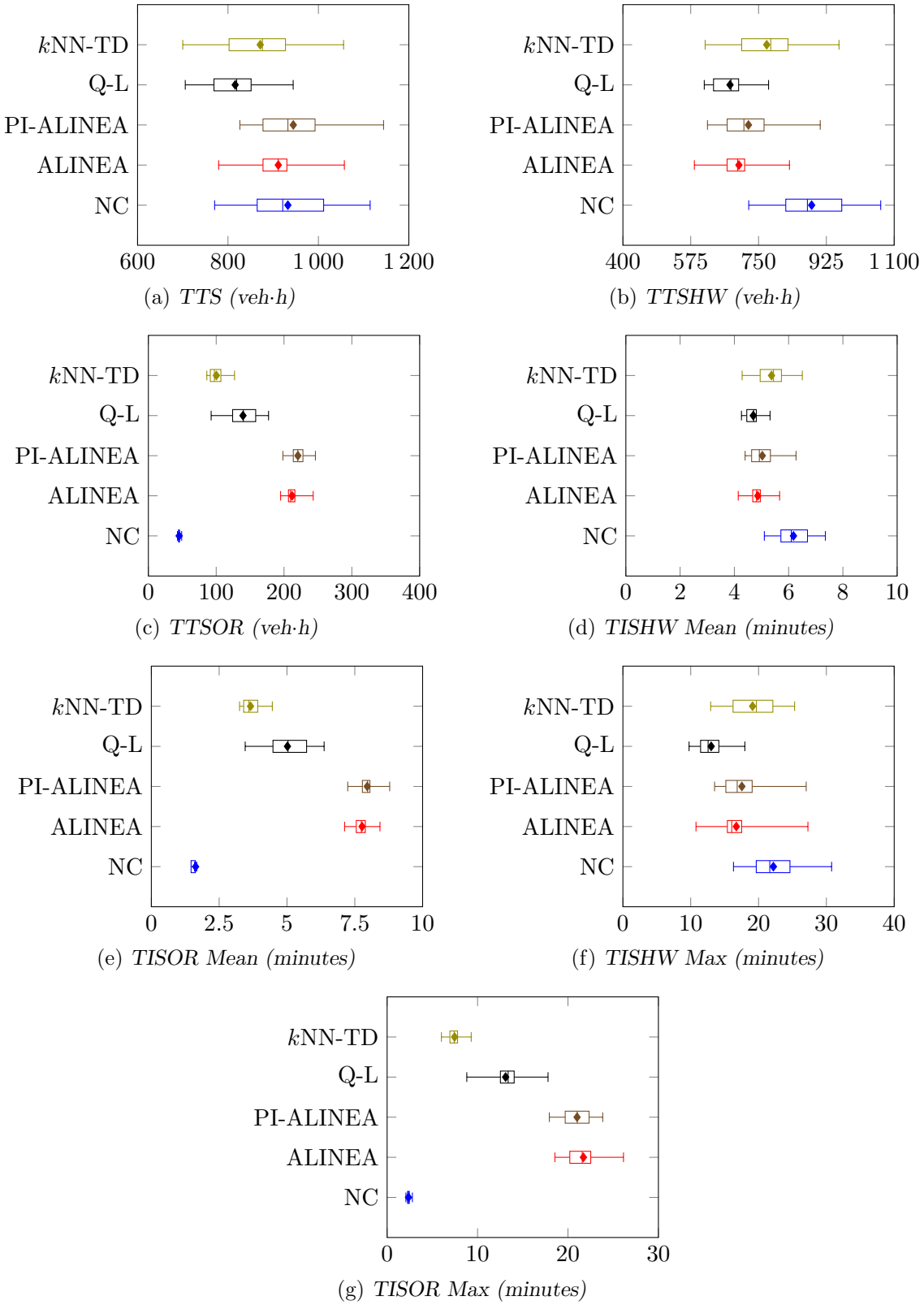
TABLE 6.56: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 3. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	932.46	911.23	944.46	815.96	871.34	2.8263×10^{-9}	2.4019×10^{-2}
TTSHW	887.07	699.32	723.97	676.46	771.09	$< 1 \times 10^{-17}$	6.7529×10^{-4}
TTSOR	45.40	211.91	220.49	139.50	100.25	$< 1 \times 10^{-17}$	1.2178×10^{-12}
TISHW Mean	6.18	4.86	5.03	4.70	5.67	$< 1 \times 10^{-17}$	1.0893×10^{-4}
TISOR Mean	1.63	7.77	7.96	5.02	3.66	$< 1 \times 10^{-17}$	1.8097×10^{-14}
TISHW Max	22.19	16.72	17.54	13.02	19.12	$< 1 \times 10^{-17}$	1.0556×10^{-1}
TISOR Max	2.37	21.71	21.01	13.10	7.46	$< 1 \times 10^{-17}$	1.3517×10^{-9}

of the maximum TISHW, while the Games-Howell *post hoc* test is employed for this purpose in respect of the other six PMIs.

In Scenario 3, for the first time, only the Q-Learning implementation was able to outperform the no-control case in respect of the TTS at a 5% level of significance, as may be seen in Table 6.57. Q-Learning returned the smallest TTS-value of 815.96 veh·h, which was, in fact, small enough to be able to outperform both ALINEA and PI-ALINEA, which achieved TTS-values of 911.23 veh·h and 944.46 veh·h, respectively, at a 5% level of significance, while its performance was found to be statistically similar to that of kNN-TD RM, which achieved a TTS-value of 871.34 veh·h. Q-Learning was followed in the order of relative algorithmic performances by kNN-TD which was able to outperform PI-ALINEA at a 5% level of significance while its performance was found to be statistically indistinguishable from that of ALINEA at a 5% level of significance. This similarity in the algorithmic performances is also clearly visible in the box plots of Figure 6.11(a).

Similarly to Scenarios 1 and 2, Q-Learning achieved the smallest TTSHW-value, returning a value of 676.46 veh·h. As may be seen in Table 6.58, this value was small enough to allow the algorithm to outperform both PI-ALINEA and kNN-TD learning at a 5% level of confidence, as they achieved TTSHW-values of 723.97 veh·h and 771.09 veh·h, respectively. ALINEA, on the other hand, returned a TTSHW-value of 699.32 veh·h, which placed its performance statistically on par with that of Q-Learning at a 5% level of significance. Furthermore, ALINEA was found to perform statistically indistinguishably from PI-ALINEA, while it was able to outperform kNN-TD RM at a 5% level of significance. Finally, PI-ALINEA and kNN-TD RM were found to perform statistically similarly at a 5% level of significance, while all of the RM implementations were able to outperform the no-control case. This ordering of the relative algorithmic performances may also be seen in the box plots of Figure 6.11(b).



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.11: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation with queue limits in Scenario 3.*

As may be seen in Table 6.59, statistical differences exist between all RM implementations in respect of their TTSOR-values at a 5% level of significance. From the box plots in Figure 6.11(c) the order of relative algorithmic performances may easily be established. As expected, the no-control case returned the smallest TTSOR-value of 45.40 veh·h, followed by kNN-TD RM and Q-Learning, which achieved a values of 100.25 veh·h and 139.50 veh·h, respectively. Q-Learning was followed by ALINEA, which returned a TTS-value of 211.91 veh·h, while PI-ALINEA achieved the largest TTSOR-value of 220.49 veh·h.

The ordering of the relative algorithmic performances in respect of the mean and maximum TISHW PMIs is similar to that in respect of the TTSH, as may be seen in the box plots of Figures 6.11(d) and 6.11(f). Q-Learning achieved an improvement of 23.95% over the no-control case in respect of the mean TISHW, followed by ALINEA, which achieved a 21.36% reduction. As a result, Q-Learning was able to outperform PI-ALINEA, which achieved a reduction of 18.61%, and kNN-TD RM, which was able to achieve a reduction of only 8.25%, while the performances of Q-Learning and ALINEA were found to be statistically indistinguishable at a 5% level of significance. Owing to its small TISHW-value, ALINEA was found to perform statistically similarly to PI-ALINEA, while it outperformed kNN-TD RM at a 5% level of significance. Furthermore, PI-ALINEA and kNN-TD RM were also found to perform statistically on par, while all of the RM implementations were able to outperform the no-control case at a 5% level of significance, as may be seen from the p-values in Table 6.60. The Fisher LSD test performed in respect of the maximum TISHW PMI, presented in Table 6.62, shows that the ordering of the relative algorithmic performances is the same as that in respect of the mean and maximum TISHW PMIs, except that in respect of the maximum TISHW, Q-Learning was also able to outperform ALINEA.

Interestingly, although the performances of all RM implementations were found to differ statistically in respect of the TTSOR, this was not the case for both the mean and maximum TISOR PMIs, as may be seen in Tables 6.61 and 6.63. As for the TTSOR, kNN-TD was again able to outperform all the other RM implementations at a 5% level of significance as it achieved mean and maximum TISOR-values of 3.66 minutes and 7.46 minutes, respectively. Q-Learning achieved the next best performance, returning mean and maximum TISOR-values of 5.02 minutes and 13.10 minutes, respectively, thereby outperforming both ALINEA and PI-ALINEA at a 5% level of significance. Unlike for the TTSOR, however, ALINEA and PI-ALINEA were found to perform statistically indistinguishably in respect of both the mean and maximum TISOR-values at a 5% level of significance, as they returned values of 7.77 minutes and 7.96 minutes, respectively, for the mean TISOR, while in respect of the maximum TISOR, ALINEA and PI-ALINEA achieved values of 21.71 minutes and 21.01 minutes, respectively. These trends are also clearly visible in the box plots of Figures 6.12(e) and 6.12(g).

TABLE 6.57: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	8.3021×10^{-1}	9.8346×10^{-1}	2.9345×10^{-6}	8.4639×10^{-2}
ALINEA		—	4.1224×10^{-1}	7.9135×10^{-7}	2.9445×10^{-1}
PI-ALINEA			—	5.2642×10^{-8}	1.6056×10^{-2}
Q-Learning				—	5.2339×10^{-2}
kNN-TD					—
Mean	932.46	911.23	944.46	815.96	871.34

TABLE 6.58: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	4.7430×10^{-12}	3.1285×10^{-9}	1.3231×10^{-12}	4.2385×10^{-5}
ALINEA		—	5.8330×10^{-1}	3.9681×10^{-1}	3.0398×10^{-3}
PI-ALINEA			—	3.2666×10^{-2}	1.7079×10^{-2}
Q-Learning				—	3.3559×10^{-5}
kNN-TD					—
Mean	887.07	699.32	723.97	676.46	771.09

TABLE 6.59: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.3846×10^{-14}	4.2411×10^{-14}	1.2990×10^{-14}	4.8850×10^{-14}
ALINEA		—	2.6579×10^{-2}	3.9113×10^{-13}	1.4559×10^{-11}
PI-ALINEA			—	1.0879×10^{-12}	1.4040×10^{-11}
Q-Learning				—	2.1421×10^{-9}
kNN-TD					—
Mean	45.40	211.91	220.49	139.50	100.25

TABLE 6.60: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	4.1245×10^{-10}	8.7152×10^{-13}	1.6417×10^{-5}
ALINEA		—	5.5283×10^{-1}	2.5756×10^{-1}	1.5042×10^{-3}
PI-ALINEA			—	1.7887×10^{-2}	1.1859×10^{-1}
Q-Learning				—	1.0359×10^{-5}
kNN-TD					—
Mean	6.18	4.86	5.03	4.70	5.67

TABLE 6.61: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.9979×10^{-15}	$< 1 \times 10^{-17}$	1.3320×10^{-15}	$< 1 \times 10^{-17}$
ALINEA		—	1.2160×10^{-1}	$< 1 \times 10^{-17}$	1.1696×10^{-11}
PI-ALINEA			—	$< 1 \times 10^{-17}$	1.4551×10^{-11}
Q-Learning				—	1.1808×10^{-9}
kNN-TD					—
Mean	1.63	7.77	7.96	5.02	3.66

TABLE 6.62: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	6.3505×10^{-10}	9.0554×10^{-8}	$< 1 \times 10^{-17}$	2.8464×10^{-4}
ALINEA		—	3.2136×10^{-1}	1.4505×10^{-5}	4.2464×10^{-3}
PI-ALINEA			—	1.8026×10^{-7}	5.8097×10^{-2}
Q-Learning				—	1.0496×10^{-11}
kNN-TD					—
Mean	22.19	16.72	17.54	13.01	19.12

TABLE 6.63: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	8.3269×10^{-15}	1.3323×10^{-14}	1.1435×10^{-14}	9.9365×10^{-14}
ALINEA		—	5.9488×10^{-1}	1.4886×10^{-11}	$< 1 \times 10^{-17}$
PI-ALINEA			—	1.0708×10^{-11}	3.5805×10^{-13}
Q-Learning				—	$< 1 \times 10^{-17}$
kNN-TD					—
Mean	2.37	21.71	21.01	13.10	7.46

Scenario 4

As for Scenarios 1, 2 and 3, the ANOVA performed on the PMI-values returned by the algorithms in the case of Scenario 4 revealed that there are, in fact, statistical differences at a 5% level of significance between at least some pair of algorithmic output data for all seven PMIs, as may be seen from the results presented in Table 6.64. The results of the Levene test indicated that the variances of the algorithmic output data are statistically indistinguishable only for the TTS, while statistical differences occur between the variances of at least some pair of algorithmic output data in respect of the other six PMIs at a 5% level of significance. Therefore, the Fisher LSD test was again employed in order to ascertain between which pairs of algorithmic output these differences occur in respect of the TTS, while the Games-Howell test was employed for this purpose in respect of all other PMIs.

TABLE 6.64: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 4. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

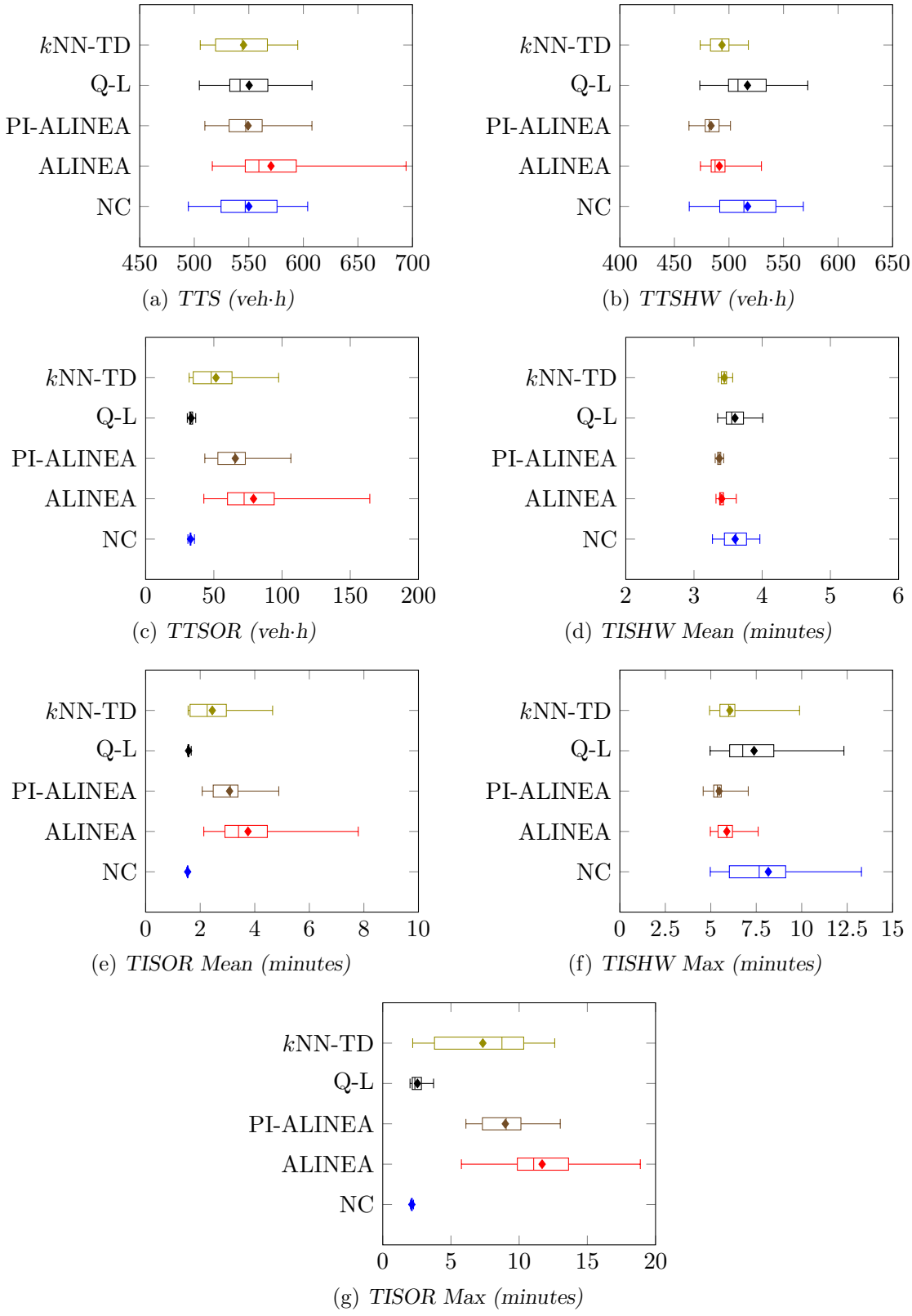
			Mean value			p-value	
PMI	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	550.00	570.31	549.32	550.31	545.19	1.6563×10^{-2}	5.4449×10^{-2}
TTSHW	517.07	491.21	483.57	516.95	493.60	2.1148×10^{-11}	3.5305×10^{-14}
TTSOR	32.93	79.10	65.75	33.36	51.59	$< 1 \times 10^{-17}$	9.1038×10^{-15}
TISHW Mean	3.60	3.41	3.37	3.60	3.45	1.8652×10^{-14}	$< 1 \times 10^{-17}$
TISOR Mean	1.54	3.75	3.08	1.57	2.44	$< 1 \times 10^{-17}$	2.2204×10^{-16}
TISHW Max	8.16	5.88	5.46	7.38	6.05	1.2454×10^{-10}	7.4829×10^{-13}
TISOR Max	2.13	11.68	8.99	2.55	7.34	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$

Due to the low traffic demand in Scenario 4, it was expected that the RM would be the least effective in this scenario. This expectation is corroborated by the results in Table 6.65, from which it is clear that that PI-ALINEA, Q-Learning and kNN-TD RM all perform statistically indistinguishably when compared with each other and the no-control case at a 5% level of confidence, as they achieved TTS-values of 549.32 veh·h, 550.31 veh·h and 545.19 veh·h, respectively, compared with the 550.00 veh·h returned by the no-control case. Interestingly, however, ALINEA was outperformed by all three other RM implementations, as well as the no-control case, as it achieved an increase in TTS to 570.31 veh·h. As may be seen from the box plots in Figure 6.12(a), this increase is largely attributed to an increase in the variance of the TTS-values corresponding to the ALINEA implementation.

Interestingly, PI-ALINEA achieved the smallest TTSHW-value of 483.57 veh·h, outperforming Q-Learning, which achieved a TTSHW-value of 516.95 veh·h, kNN-TD RM, which achieved a TTSHW-value of 493.60 veh·h, and the no-control case, which returned a TTSHW-value of 517.07 veh·h, at a 5% level of significance. As may be seen from the p-values in Table 6.66, PI-ALINEA and ALINEA were found to perform statistically similarly at a 5% level of significance, as ALINEA returned a TTSHW-value of 491.21 veh·h. PI-ALINEA is followed in the order of relative algorithmic performances by ALINEA and kNN-TD RM, which were also able to outperform the no-control case and Q-Learning at a 5% level of significance, while performing statistically on par with one another. As may be inferred from the box plots of Figure 6.12(b), the order of relative algorithmic performances is completed by Q-Learning, the performance of which was found to be statistically indistinguishable from that of the no-control case at a 5% level of significance.

As expected, the no-control case again returned the smallest TTSOR-value. Interestingly, however, Q-Learning returned a TTSOR-value which represented an increase of only 1.31% over the no-control case, resulting in the fact that the algorithmic performance of Q-Learning was found to be statistically indistinguishable from the no-control case at a 5% level of significance, as may be seen from the p-values presented in Table 6.67. Owing to this small TTSOR-value, Q-Learning was able to outperform ALINEA, PI-ALINEA and kNN-TD RM at a 5% level of significance, as these RM implementations resulted in 240.21%, 199.67% and 156.67% increases over the no-control case, respectively. Q-Learning was followed in the order of relative algorithmic performances by kNN-TD RM, which was able to outperform both ALINEA and PI-ALINEA at a 5% level of significance, while the performances of the latter two control strategies were once again found to be statistically indistinguishable at a 5% level of significance. These trends in respect of the TTSOR-values are also evident in the box plots of Figure 6.12(c)

Similarly to the TTSHW, PI-ALINEA again returned the smallest values in respect of both the mean and maximum TISHW-values. As may be seen from the p-values in Tables 6.68 and 6.70, PI-ALINEA was able to outperform all three other RM implementations at a 5% level of significance in respect of the mean TISHW, while it was able to outperform both Q-Learning and kNN-TD RM in respect of the maximum TISHW. PI-ALINEA was followed by ALINEA in the order of relative algorithmic performances, as ALINEA was able to outperform Q-Learning at a 5% level of significance in respect of both the mean and maximum TISHW, while it was found to perform statistically indistinguishably from kNN-TD in respect of both of these PMIs. Q-Learning was also outperformed by kNN-TD RM at a 5% level of significance in respect of both the mean and maximum TISHW, while it was the only implementation that did not outperform the no-control case. This order of relative algorithmic performances, as well as the similarity in performance between Q-Learning and the no-control case, is clearly visible in the box plots of Figures 6.12(d)and 6.12(f).



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 6.12: *PMI results for the no-control case (NC), the ALINEA and PI-ALINEA control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation with queue limits in Scenario 4.*

From the box plots in Figures 6.12(e) and 6.12(g), it is evident that the order of relative algorithmic performance for the mean and maximum TISOR PMIs is the same as that for the TTSOR. As expected, the no-control case returned the smallest mean and maximum TISOR-values, achieving values of 1.54 minutes and 2.13 minutes, respectively, and outperforming all of the RM implementations at a 5% level of significance. Although Q-Learning was outperformed by the no-control case at 5% level of significance, returning mean and maximum TISOR-values of 1.57 minutes and 2.55 minutes, one may argue that these increases of approximately 1.8 seconds and 25 seconds will hardly be noticeable by drivers in a real world scenario. From the p-values in Tables 6.69 and 6.71, it is evident that Q-Learning is followed by kNN-TD in the order of relative algorithmic performances, as kNN-TD was able to outperform both ALINEA and PI-ALINEA at a 5% level of significance in respect of the mean TISOR, achieving a value of 2.44 minutes, while kNN-TD RM outperformed ALINEA in respect of the maximum TISOR, achieving a value of 6.05 minutes. The performances of ALINEA and PI-ALINEA were found to be statistically similar at a 5% level of significance in respect of the mean TISOR, as these control strategies returned values of 3.75 minutes and 3.08 minutes, respectively, while in respect of the maximum TISOR, PI-ALINEA managed to outperform ALINEA as the maximum values increased to 8.99 minutes and 11.68 minutes, respectively.

TABLE 6.65: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.1064×10^{-2}	9.3125×10^{-1}	9.6862×10^{-1}	5.4309×10^{-1}
ALINEA		—	8.6903×10^{-3}	1.2329×10^{-2}	1.7821×10^{-3}
PI-ALINEA			—	9.0004×10^{-1}	6.0166×10^{-1}
Q-Learning				—	5.1737×10^{-1}
kNN-TD					—
Mean	550.00	570.31	549.32	550.31	545.19

TABLE 6.66: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.1218×10^{-3}	1.8472×10^{-5}	9.9999×10^{-1}	3.2701×10^{-3}
ALINEA		—	6.5917×10^{-2}	5.2346×10^{-4}	9.3928×10^{-1}
PI-ALINEA			—	5.9754×10^{-6}	4.0344×10^{-3}
Q-Learning				—	1.6748×10^{-3}
kNN-TD					—
Mean	517.07	491.21	483.57	516.95	493.60

Discussion

When queue limits are imposed on the RM strategies, Q-Learning achieved the smallest TTS-values in Scenarios 1–3, while it was not outperformed by any other algorithm at a 5% level of significance in respect of the TTS in Scenario 4. The second-best algorithm in respect of the TTS is the kNN-TD RM implementations, which achieved the second-smallest TTS-value in Scenarios 1–3 and the smallest TTS-value in Scenario 4, being outperformed only twice by

TABLE 6.67: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.5159×10^{-8}	4.6194×10^{-11}	7.1375×10^{-1}	6.6125×10^{-5}
ALINEA		—	2.0156×10^{-1}	1.8308×10^{-8}	6.4548×10^{-4}
PI-ALINEA			—	5.9053×10^{-11}	2.1524×10^{-2}
Q-Learning				—	9.4166×10^{-5}
kNN-TD					—
Mean	32.93	79.10	65.75	33.36	51.59

TABLE 6.68: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.7871×10^{-5}	3.0089×10^{-6}	9.9999×10^{-1}	1.2249×10^{-3}
ALINEA		—	4.6653×10^{-2}	4.1497×10^{-5}	1.1437×10^{-5}
PI-ALINEA			—	1.9675×10^{-6}	4.2116×10^{-6}
Q-Learning				—	9.8416×10^{-4}
kNN-TD					—
Mean	3.60	3.41	3.37	3.60	3.45

TABLE 6.69: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	4.8407×10^{-9}	8.3349×10^{-12}	1.5008×10^{-4}	4.0233×10^{-5}
ALINEA		—	1.1685×10^{-1}	6.6547×10^{-9}	3.6390×10^{-4}
PI-ALINEA			—	1.3582×10^{-11}	2.4677×10^{-2}
Q-Learning				—	7.0563×10^{-5}
kNN-TD					—
Mean	1.54	3.75	3.08	1.57	2.44

TABLE 6.70: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	7.3847×10^{-4}	6.5964×10^{-5}	6.9115×10^{-1}	2.2710×10^{-3}
ALINEA		—	5.2194×10^{-2}	2.0518×10^{-3}	9.1933×10^{-1}
PI-ALINEA			—	6.3216×10^{-5}	2.9631×10^{-2}
Q-Learning				—	1.0814×10^{-2}
kNN-TD					—
Mean	8.16	5.88	5.46	7.38	6.05

TABLE 6.71: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.3260×10^{-16}	6.9939×10^{-16}	1.5587×10^{-3}	7.0862×10^{-8}
ALINEA		—	1.5015×10^{-3}	6.9167×10^{-14}	4.7295×10^{-5}
PI-ALINEA			—	8.8800×10^{-16}	1.7717×10^{-1}
Q-Learning				—	3.4482×10^{-7}
kNN-TD					—
Mean	2.13	11.68	8.99	2.55	7.34

Q-Learning in Scenarios 1 and 3. ALINEA and PI-ALINEA consistently returned the weakest performances of the RM control strategies in respect of the TTS, only outperforming the no-control case at a 5% level of significance in respect of the TTS in Scenario 2.

In respect of the TTSHW, Q-Learning again consistently returned the best performance, achieving the smallest TTSHW-values in Scenarios 1–3, and being outperformed in respect of the TTSHW only once by PI-ALINEA in Scenario 4. The kNN-TD RM implementation, on the other hand, achieved the largest TTSHW-values in Scenarios 1–3, while still outperforming the no-control case at a 5% level of significance in each of these scenarios. Similarly to the case where queue limitations were not imposed in the RM control strategies, the feedback controllers were successful in protecting the highway traffic flow when queue limits are in place, as PI-ALINEA was only outperformed in respect of the TTSHW in Scenario 3, while ALINEA was outperformed at a 5% level of significance only by Q-Learning in Scenario 2.

The kNN-TD implementation was the most successful in reducing the TTSOR when queue limitations were in place, consistently achieving the best performance in Scenarios 1–3. A possible explanation for these reduced metering rates is that the punishment due to long on-ramp queues may be applied to queue length centres which are significantly smaller than the actual queue length, if these centres form part of the k-nearest neighbours of the current state. If the queue length then reaches a value near this centre, those actions which already reduce the queue length are chosen, due to the punishment which has propagated down to these centres. This leads to maximum on-ramp queues which are significantly shorter than the maximum allowable queue length, as may be seen in Table 6.38. This phenomenon is not experienced in any of the other RM implementations. The kNN-TD implementation is followed by Q-Learning in the order of algorithmic performances in respect of the TTSOR, as Q-Learning achieved the second smallest TTSOR-value in Scenarios 1–3 and the smallest TTSOR-value in Scenario 4, while both Q-Learning and kNN-TD RM outperformed ALINEA and PI-ALINEA in respect of the TTSOR in all four scenarios. The superiority of the RL approaches in managing the queue length and the on-ramp waiting times may be due to the fact that a direct action selection policy is employed rather than the incremental approach followed by the feedback controllers. This direct action selection, allows the controller to adjust the red phase times at the on-ramp faster, resulting in a more responsive controller, which may better manage the on-ramp queue around the maximum allowable queue length.

6.7 Chapter Summary

This chapter opened in §6.1 with a brief description of how the ALINEA and PI-ALINEA RM control laws were implemented in the microscopic traffic simluation model of §5.1.2. Thereafter, the RM problem was formulated as an RL problem in §6.2. This formulation was accompanied by descriptions of the state and action spaces as well as the reward function employed. This was followed in §6.3 and §6.4 by descriptions of the Q-Learning and kNN-TD implementations adopted for solving the RM problem, respectively.

In §6.5.1, a complete parameter evaluation was conducted in order to find combinations of parameters that yield the best performance for each of the RM implementations. Once these parameter combinations had been found, the relative performances of the three RM implementations were compared in §6.5.2 in the context of the four different scenarios of traffic demand simulated in the benchmark simulation model of §5.1.2, as described in §5.3.2. It was found that the kNN-TD implementation is generally the best-performing algorithm over all of the traffic scenarios simulated.

Thereafter, an on-ramp queue limit was introduced in order to prevent the excessively long on-ramp queues for which RM is notorious. The implementation of these queue limitations was outlined in §6.6, together with a thorough algorithmic performance comparison while taking into account the queue limit in each of the four scenarios of traffic demand in §5.3.2. It was found that the Q-Learning implementation generally yielded the most favourable results when a queue limit is imposed.

CHAPTER 7

Reinforcement Learning for Variable Speed Limits

Contents

The purpose of this chapter is to provide a detailed description of the implementation of RL in the context of VSLs. The chapter opens in §7.1 with a description of a feedback-based VSL implementation, which, like ALINEA for the RM implementations, is employed as a benchmark against which to measure the performance of the RL VSL implementations. This is followed in §7.2 with a description of the VSL problem in the context of RL, which then serves as the blueprint for the algorithmic implementations of Q-Learning and the kNN-TD reinforcement learning algorithms in §7.3 and §7.4, respectively. Computational results of an algorithmic parameter evaluation are presented in §7.5.1. This is followed by a thorough algorithmic performance comparison in §7.5.2 using suitable algorithmic parameter values. The chapter finally closes in §7.6 with a brief summary of the work included in the chapter.

7.1 The Feedback-based VSL Controller Implementation

The *mainline traffic flow control* (MTFC) controller of Müller *et al.* [105] was chosen as the benchmark VSL controller against which the performance of the RM VSL implementations is measured. This controller was chosen due to its proven performance in the context of a microscopic traffic simulation model [105], as well as its relative ease of implementation. A graphical illustration of the working of this implementation is shown in Figure 7.1.

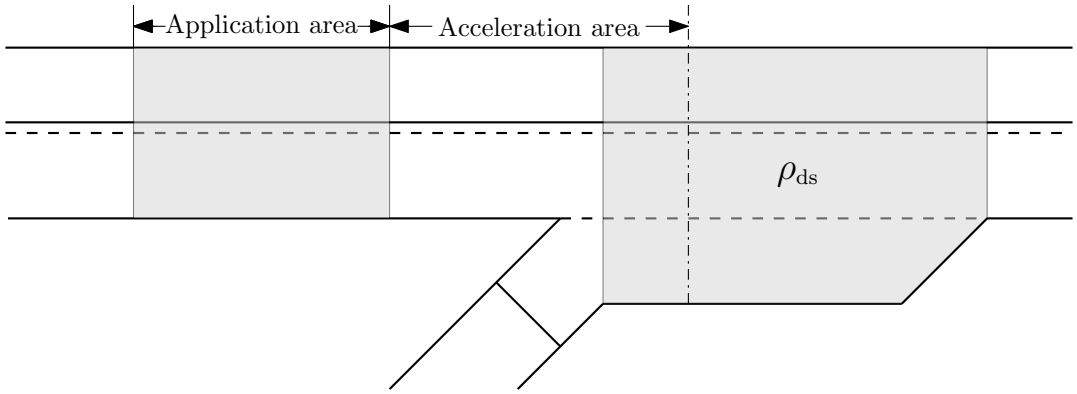


FIGURE 7.1: *The feedback-based MTFC VSL implementation of Müller* et al. *[105].*

As may be seen in the figure, the VSL determined according to (3.32)–(3.33) and based on the downstream density measured at the area, is applied for a relatively short section of the highway, denoted as the application area. This application area is followed by the so-called acceleration area. The motivation behind introducing the short application area is that possibly very small VSL values[1] are applied in order to create a controlled bottleneck at a location upstream of the true bottleneck, thereby controlling the traffic flow that enters the true bottleneck location from the mainline [105]. Vehicles then accelerate in the acceleration area after the VSL application area in order to reach the nominal speed limit such that, by the time the vehicles travelling along the mainline and the on-ramp merge, both these groups of vehicles travel at approximately the same speed, which may result in a smoother merging of the two traffic flows. In order to prevent a scenario where the speed limit drops from the nominal speed limit to the specified VSL value, VSLs are displayed at 100 metre intervals upstream of the application area. Each of these speed limits indicates a speed limit value which is 10 km/h higher than the following downstream speed limit, in order to ensure a gradual reduction in speed aimed at preventing the formation of shockwaves propagating backwards along the highway.

7.2 Formulation as a Reinforcement Learning Problem

Walraven *et al.* [166], as well as Zhu and Ukkusuri [179], have shown that the VSL problem may be formulated as an RL problem and may subsequently be solved using RL techniques. In the benchmark model of §5.1.2 considered in this study, VSLs are applied from the start of $S_{1.3}$ through $S_{1.4}$ until the start of $S_{2.1}$, where the normal speed limit of 120 km/h is restored after the bottleneck at the on-ramp, as shown in Figure 7.2.

7.2.1 The State Space

As for the state space in the RM application, the state space in the VSL implementation comprises three main components, as illustrated graphically in Figure 7.3. The first state is the density ρ_{ds} directly downstream of the on-ramp. This state is chosen so as to provide the learning agent with information on the state of traffic flow at the bottleneck.

The second component of the state space is the vehicle density on $S_{1.4}$ at the application area, denoted by ρ_{app}. This state is chosen since it is expected to give the agent an indication of the

[1] Recall from (3.33) that the VSLs to be applied reside within the interval VSL $\in$ [20,120] km/h.

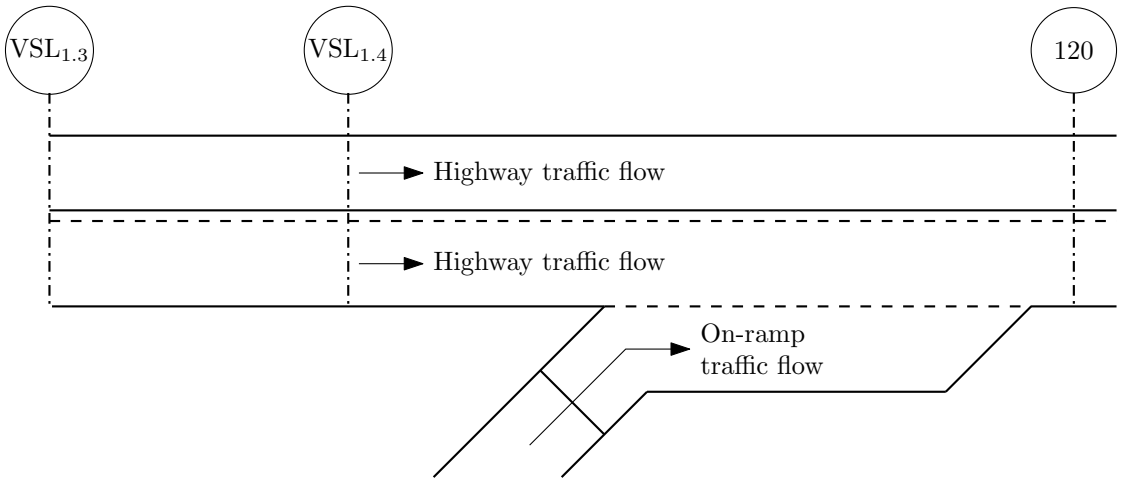


FIGURE 7.2: *The VSL implementation adopted in the context of the benchmark model of §5.1.2.*

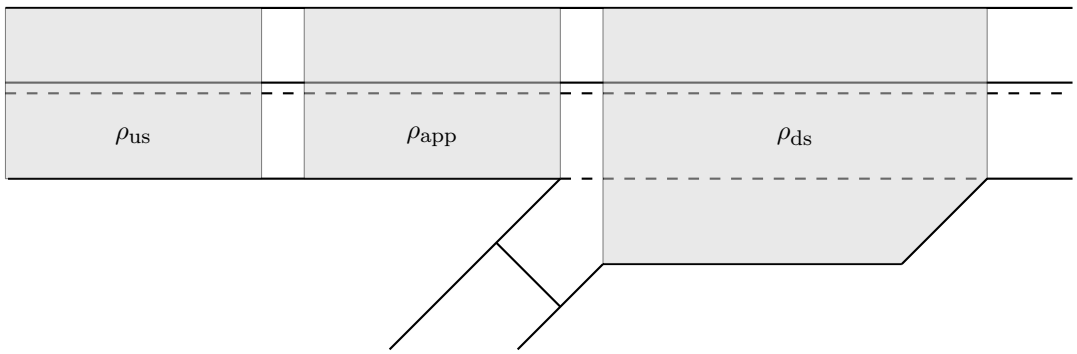


FIGURE 7.3: *A representation of the state space for the VSL problem in the context of the benchmark model of §5.1.2.*

effectiveness of the action chosen, as the most immediate response to the action will be reflected on this section of the highway.

The third and final component of the state space is the upstream density ρ_{us}. In the case of VSLs, the upstream density is the density on $S_{1.3}$. This state is chosen so as to provide the learning agent with a predictive component in terms of highway demand, as well as an indication of the severity of congestion, should it have spilled back beyond the application area.

7.2.2 The Action Space

As in the RM implementation, a direct action selection policy is adopted for the VSL problem in pursuit of a fast learning speed. The VSL to be applied is determined as

$$VSL_{1.4} = 90 + 10a, \tag{7.1}$$

where $a \in \{0, 1, 2, 3\}$. This results in minimum and maximum variable speed limits of 90 km/h and 120 km/h, respectively. As may be seen in (7.1), the learning agent adjusts the speed limit directly at $S_{1.4}$. In order to reduce the difference in speed limit from 120 km/h at $S_{1.2}$ to $VSL_{1.4}$, the speed limit at $S_{1.3}$ is adjusted as

$$VSL_{1.3} = \max\left[(VSL_{1.4} + \delta), 120\right], \tag{7.2}$$

where δ is an empirically determined parameter. This more gradual reduction in the speed limit is introduced in order to reduce the probability of shock-waves propagating backwards along the highway as a result of sudden, sharp reductions in the speed limit.

7.2.3 The Reward Function

As was the case in RM, the objective of the VSL implementation remains to minimise the total time spent in the system by all transportation network users. This may be achieved by maximising the system throughput. As a result, the reward function chosen for the VSL RL agent is the flow out of the bottleneck location, as shown in Figure 7.4. The goal of the agent is then to maximise the outflow out of the bottleneck location, thereby maximising the system throughput.

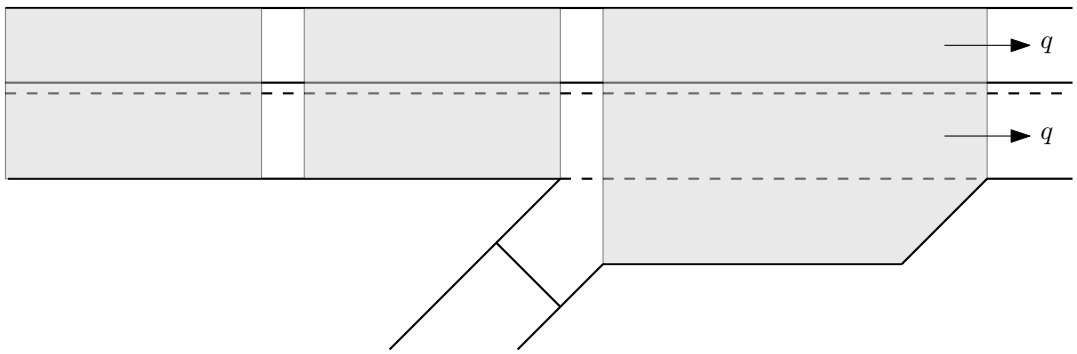


FIGURE 7.4: *The reward function employed for the VSL agent in the context of the benchmark model of §5.1.2.*

The control interval employed for the VSL agent is 5 minutes in length. This is the same length of control interval employed by Walraven *et al.* [166]. This length is chosen since it is expected that five minutes will be sufficient to notice differences in flow after a distance of 1 kilometre (the length of $S_{1.3}$ and $S_{1.4}$), even in slow-moving traffic. As a result, the flow is measured over 5-minute intervals. In order to amplify the differences in flow between subsequent intervals, the flow q measured over the 5-minute interval is multiplied by 12 so as to obtain a flow value measured in vehicles per hour. The reward function for the VSL agent is thus given by

$$r = 12q. \tag{7.3}$$

7.3 Q-Learning for Variable Speed Limits

As was the case in the Q-Learning implementation for RM in §6.3, the state space is discretised into $n_{\rho_{\text{ds}}} = n_{\rho_{\text{app}}} = n_{\rho_{\text{us}}} = 10$ equi-spaced intervals for the upstream, application and downstream densities, respectively. This results in a total state space comprising $|n_{\rho_{\text{ds}}}| \times |n_{\rho_{\text{app}}}| \times |n_{\rho_{\text{us}}}| = 1\,000$ states. A table-based approach to Q-value approximation is again adopted, employing AnyLogic's built-in Microsoft SQL Server functionality, Q-Learning is implemented for the benchmark model of §5.1.2 as outlined in Algorithm 2.3. In order to find an effective trade-off between exploration of the state-action space and exploitation of that which has already been learnt by the agent, the same rules for determining an adaptive α-value and adaptive ϵ-value as given in (6.5) and (6.6), respectively, are employed in the Q-Learning VSL implementation.

7.4 kNN-TD Learning for Variable Speed Limits

Due to the fact that maximum throughput is achieved at the critical density, the centres for the downstream, application and upstream densities should be clustered around the critical density value so as to be able to provide more detailed approximations of the Q-values around this point. Due to the fact that the density at the bottleneck is expected to be generally higher than the critical density, the centres for the downstream density are clustered around a density of 35 veh/km, which is above the typical critical density, in order to provide the agent with more detailed feedback regarding the situation at the bottleneck. As a result, the downstream centres are placed at $\{12, 19, 24, 29, 32, 35, 38, 45, 55, 60\}$. In pursuit of finding a balanced approximation for various densities around the expected critical density of approximately 28 veh/km [130], the centres for the application and the upstream density are chosen at $\{12, 20, 26, 30, 35, 40, 45, 55\}$. The lookup table used to store the approximated Q-values for all of the centre-action pairs is again implemented using Anylogic's built-in database functionality. In order to achieve the trade-off between exploration and exploitation, the rules for finding an adaptive α-value and ϵ-value in the kNN-TD VSL implementation are the same as those in the kNN-TD RM implementation, given in (6.5) and (6.8), respectively.

7.5 Computational Results

In this section, the performance of MTFC, Q-Learning and kNN-TD learning are fine-tuned by means of an algorithmic parameter evaluation in §7.5.1. Thereafter, their respective algorithmic performances are compared in each of the four scenarios of traffic demand outlined in §5.3.2 and implemented within the benchmark simulation model described in §5.1.2, adopting suitable parameter values found in the previous section. The results of this relative algorithmic performance comparison are presented in §7.5.2.

7.5.1 Parameter Evaluation

This section is devoted to determining good parameter values for the MTFC, Q-Learning and the kNN-TD VSL implementations described in §7.1, §7.3 and §7.4, respectively. The focus of this parameter evaluation is to find a suitable target density and controller parameter K_I for MTFC, as well as a suitable value for the parameter δ in (7.2) in the RL implementations.

MTFC Parameter Evaluation

A process similar to that of finding the best-performing target density value in the RM implementations was followed for the MTFC implementation. Three values, judged to be low, average and high for the controller parameter K_I were considered. The low value was chosen as 0.0025, while the medium and high values were set as 0.005 and 0.0075, respectively. Müller *et al.* [105] suggested setting the controller parameter value to 0.005. The initial parameter evaluation of target density values between 24 veh/km and 34 veh/km indicated that setting the target density to 32 veh/km yielded the best results. Furthermore, setting K_I to 0.005 consistently returned the smallest TTS-values. Therefore, target density values between 31.5 veh/km and 32.5 veh/km were considered in increments of 0.1 veh/km for the case where $K_I = 0.005$, as may be seen in Table 7.1. Finally, the lengths of the application and acceleration areas were set to 100 and 175 metres, respectively, as suggested by Müller *et al.* [105]. As may be seen in

the table, the combination of setting $K_I = 0.005$ and $\hat{\rho} = 32$ yielded the smallest TTS-value. Therefore, this combination of parameters is adopted in all further comparisons in this chapter.

TABLE 7.1: *Parameter evaluation results for the MTFC VSL implementation, measured in terms of the TTS in veh·h.*

	Target density $\hat{\rho}$						
K_I	31.0	31.5	31.6	31.7	31.8	31.9	32.0
0.0025	1 098.38	—	—	—	—	—	1 072.77
0.0050	1 074.57	1 071.40	1 073.86	1 066.66	1 065.12	1 062.31	1 058.05
0.0075	1 095.49	—	—	—	—	—	1 092.77
	Target density $\hat{\rho}$						
K_I	32.1	32.2	32.3	32.4	32.5	33.0	
0.0025	—	—	—	—	—	1 094.14	
0.0050	1 063.37	1 061.06	1 065.03	1 064.32	1 065.44	1 076.00	
0.0075	—	—	—	—	—	1 089.68	

RL Parameter Evaluations

For the RL implementations, three different values of δ are considered, as may be seen in Table 7.2. In the first of these cases, a value of $\delta = 10$ is employed. In the second case, a value of $\delta = 20$ is considered. Finally, in the third case, the speed limit at $S_{1.3}$ is adjusted so that it is always at the mid-point between the speed limits of 120 km/h at $S_{1.2}$ and $VSL_{1.4}$ at $S_{1.4}$. It is envisioned that this will yield the smoothest transition from the standard speed limit of 120 km/h to the VSL at $S_{1.4}$. As a result, the expression in (7.2) becomes

$$VSL_{1.3} = VSL_{1.4} + (120 - VSL_{1.4})/2. \tag{7.4}$$

Due to the fact that the ultimate goal is once again to reduce the TTS as much as possible, TTS is the performance measure according to which the best-performing value of δ is chosen.

TABLE 7.2: *Parameter evaluation results for VSLs, measured in terms of the TTS in veh·h.*

	Scenario 1		Scenario 2	
δ	Q-Learning	kNN-TD	Q-Learning	kNN-TD
Case 1	1 735.58	1 727.60	1 112.72	1 087.50
Case 2	1 746.55	1 738.33	1 067.70	1 052.84
Case 3	1 740.92	1 764.01	1 068.28	1 103.69
	Scenario 3		Scenario 4	
δ	Q-Learning	kNN-TD	Q-Learning	kNN-TD
Case 1	913.19	889.95	543.98	542.68
Case 2	888.97	877.73	547.41	533.67
Case 3	894.92	892.84	553.00	537.34

As may be seen in Table 7.2, the case where $\delta = 20$ yielded the best performance for all four scenarios, except for Scenario 1 in the kNN-TD VSL implementation. In Scenario 1, however, $\delta = 10$ resulted in the best performance. For the Q-Learning implementation, $\delta = 20$ again resulted in the best performance in Scenarios 2 and 3, while $\delta = 10$ performed best in Scenarios 1 and 4.

Interestingly, employing the more sophisticated law in (7.4) never led to the best performance in any of the four scenarios. Due to its consistently good performance, the value of $\delta = 20$ is employed in all further comparisons conducted in this chapter. As was the case in the kNN-TD RM implementation of Chapter 6, $k = 4$ nearest neighbours are employed in the kNN-TD VSL implementation.

7.5.2 Algorithmic Comparison

In this section, the simulation results and relative algorithmic performances are analysed for the three VSL implementations described above. This comparison is performed in each of the four different scenarios of traffic demand described in §5.3.2. The results are presented and interpreted through the use of box plots in which the means, medians and interquartile ranges of the PMIs are indicated, as well as tables indicating whether or not statistical differences exist between the PMI values for each pair of algorithms at a 5% level of significance.

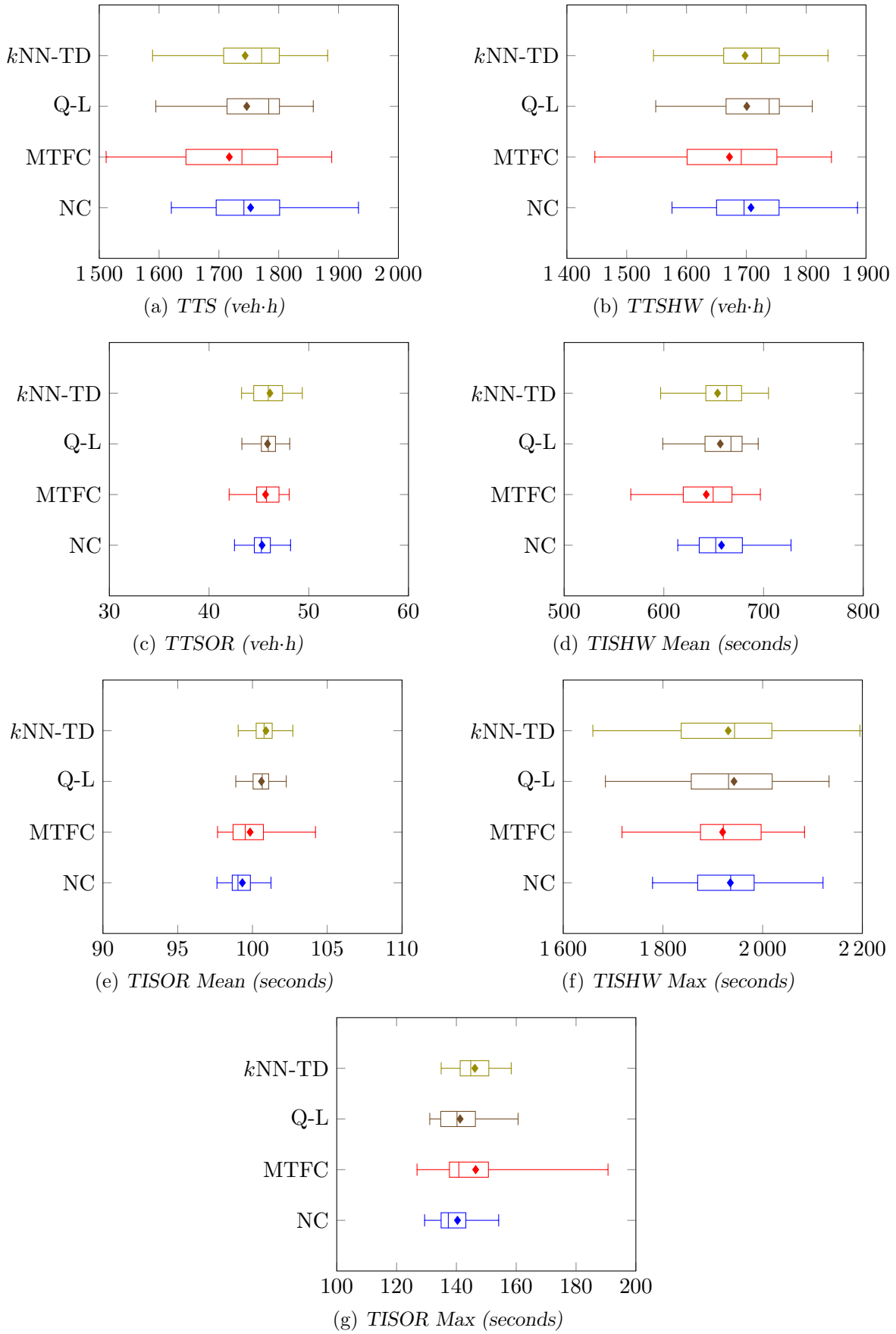
Scenario 1

As may be seen from the p-values of the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms in Scenario 1, presented in Table 7.3, the ANOVA test revealed that there are only differences between at least some pair of algorithmic output data in respect of the mean TISOR and maximum TISOR PMIs at a 5% level of significance. Furthermore, Levene's test revealed that the variances of the algorithmic output data sets are statistically indistinguishable at a 5% level of significance in respect of all PMIs, except for the mean and maximum TISOR. Hence the Games-Howell test was employed to determine where the differences in algorithmic output lie in respect of both the mean and maximum TISOR.

TABLE 7.3: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 1. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value				p-value	
PMI	No Control	MTFC	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 753.01	1 717.43	1 746.55	1 738.33	4.4791×10^{-1}	3.2564×10^{-1}
TTSHW	1 707.70	1 671.76	1 700.68	1 697.82	4.3747×10^{-1}	3.4943×10^{-1}
TTSOR	45.31	45.67	45.86	46.22	1.2055×10^{-1}	1.7823×10^{-3}
TISHW Mean	657.83	642.54	656.61	653.05	2.2311×10^{-1}	3.1855×10^{-1}
TISOR Mean	99.32	99.84	100.60	100.90	2.6559×10^{-7}	1.4751×10^{-2}
TISHW Max	1 935.24	1 919.76	1 942.55	1 930.96	6.2356×10^{-1}	3.0864×10^{-1}
TISOR Max	140.37	146.45	141.25	148.01	1.4707×10^{-2}	7.4637×10^{-4}

As may be seen in Figure 7.5(a), the VSL implementations were not able to achieve significant improvements over the no-control case in terms of the TTS in Scenario 1. This was confirmed by the p-values of the ANOVA, presented in Table 7.3, which show that neither of the VSL implementations performed statistically better than the no-control case at a 5% level of significance. There is also no substantial evidence of any homogenisation of traffic flow due to VSLs in Scenario 1 as the Levene test revealed that variances in respect of the TTS are homogeneous. From Figure 7.5(a) one may, however, argue that the implementation of VSLs may improve upon the worst-case scenario in terms of the TTS, as may be seen from the fact that the upper whiskers of all three VSL implementations occur at a smaller TTS-value than the no-control case.



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (seconds)*

(e) *TISOR Mean (seconds)*

(f) *TISHW Max (seconds)*

(g) *TISOR Max (seconds)*

FIGURE 7.5: *PMI results for the no-control case (NC), MTFC, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the VSL implementation in Scenario 1.*

As may be seen in Figure 7.5(b), a trend similar to that for the TTS emerges when considering the TTSHW. Again, all three of the VSL implementations were able to reduce the value at the upper whisker of the box plots compared to the no-control case. As for the TTS, however, no statistical differences in respect of the mean TTSHW-values could be identified at a 5% level of significance, although the VSL implementations were all able to achieve marginally lower mean TTSHW values than the no-control case. This is again evident from the results presented in Table 7.3.

Interestingly, MTFC, Q-Learning and kNN-TD for VSLs resulted in increases in the total time spent in the system by vehicles joining the highway from the on-ramp. As may be seen in Table 7.3, none of these increases were large enough for the algorithmic performances to be classified as being statistically different from one another at a 5% level of significance. The no-control case achieved a TTSOR-value of 45.31 veh·h, while MTFC, Q-Learning and kNN-TD returned TTSOR-values of 45.67 veh·h, 45.86 veh·h and 46.22 veh·h, respectively. These results are summarised in the box plots of Figure 7.5(c).

For both the mean and maximum TISHW, again no statistical differences were identified between any of the VSL implementations and the no-control case at a 5% level of significance, as may be seen from the results presented in Table 7.3. In respect of the mean TISHW, again all three of the VSL implementations reduced the maximum values at the upper whiskers of the box plots, as may be seen in Figure 7.5(d). Interestingly, although Levene's test revealed that the variances in respect of the maximum TISHW are homogeneous at a 95% level of confidence, based on the box plots in Figure 7.5(f), it may be seen that these variances increased for both Q-Learning and kNN-TD for VSLs when compared with both the no-control case and MTFC for VSLs. One may argue that this provides evidence against the homogenisation of traffic flow due to VSLs when these are implemented on long sections of the highway in which such heavy traffic conditions prevail.

When considering the mean and maximum travel times for vehicles joining the highway from the on-ramp, it is the no-control case that achieved the smallest travel times, returning values of 99.32 seconds and 140.37 seconds, respectively. As may be seen in Table 7.4, the no-control case outperformed both Q-Learning and kNN-TD for VSLs in respect of the mean TISOR at a 5% level of significance, while its performance was statistically indistinguishable from that of MTFC for VSLs. This difference is also clear in the box plots of Figure 7.5(e). Unlike the case of the mean TISOR, however, the no-control case was unable to outperform either MTFC for VSLs or Q-Learning in respect of the maximum TISOR, while the no-control case and Q-Learning both outperformed kNN-TD learning for VSLs in this respect, as may be seen in Table 7.5. These differences are clear in the box plots of Figure 7.5(g). Although the means of the no-control case, MTFC for VSLs and Q-Learning were found to be statistically indistinguishable at a 5% level of significance, a significant increase in the variances of travel times experienced by travellers is observed when MTFC for VSLs is employed, as may be seen in Figure 7.5(g). This increase in variance was also confirmed by Levene's test, as may be seen in Table 7.3.

Scenario 2

As may be seen in Table 7.6, the p-values returned by the ANOVA test performed for Scenario 2 revealed that there are statistical differences between the means of at least some pair of algorithmic output data in respect of all PMIs except the TTSOR at a 5% level of significance. Furthermore, the results of Levene's test revealed that the variances of the PMI data sets returned by the algorithms in respect of the TTSOR, the mean TTSOR and the maximum TISHW are statistically indistinguishable, while the variances of at least some pair of algorithmic

TABLE 7.4: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	4.3463×10^{-1}	2.0252×10^{-5}	1.5999×10^{-6}
MTFC		—	8.7747×10^{-1}	6.1512×10^{-3}
Q-Learning			—	3.0960×10^{-1}
kNN-TD				—
Mean	99.32	99.84	100.60	100.90

TABLE 7.5: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISOR Max			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	3.0419×10^{-1}	9.7942×10^{-2}	6.6051×10^{-3}
MTFC		—	3.9204×10^{-1}	9.6337×10^{-1}
Q-Learning			—	4.8751×10^{-3}
kNN-TD				—
Mean	140.37	146.45	141.25	148.01

output data sets were found to be statistically different at a 5% level of significance for all other PMIs, as may be seen in the table. Hence the Fisher LSD test was subsequently performed in respect of the mean TISOR and maximum TISHW, while the Games-Howell test was employed in respect of all other PMIs in order to determine between which pairs of algorithmic output the differences occur in respect of these PMIs.

TABLE 7.6: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 2. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value				p-value	
PMI	No Control	MTFC	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 141.80	1 058.05	1 067.70	1 052.83	7.8261×10^{-4}	2.3434×10^{-2}
TTSHW	1 107.88	1 024.70	1 033.87	1 019.16	7.6608×10^{-4}	2.2711×10^{-2}
TTSOR	33.92	33.35	33.82	33.68	1.7150×10^{-1}	9.7952×10^{-1}
TISHW Mean	424.86	395.26	398.65	393.36	1.0671×10^{-3}	4.1497×10^{-2}
TISOR Mean	94.68	94.73	96.38	97.21	$< 1 \times 10^{-17}$	1.4789×10^{-1}
TISHW Max	1 166.96	1 109.58	1 103.57	1 087.60	1.0595×10^{-2}	8.0094×10^{-2}
TISOR Max	127.85	128.88	135.98	131.82	2.2715×10^{-13}	3.5134×10^{-7}

In contrast to what was observed in Scenario 1, the results obtained from the VSL implementations in Scenario 2 provide strong evidence of the homogenisation effect of traffic flow due to VSLs, especially in the case of the RL implementations, where the VSLs are applied for significantly longer sections than in the MTFC implementation. As may be seen from the reduced width of the interquartile ranges of the box plots corresponding to the VSL implementations in Figure 7.6(a), the variances in respect of the TTS are significantly reduced in the case of both

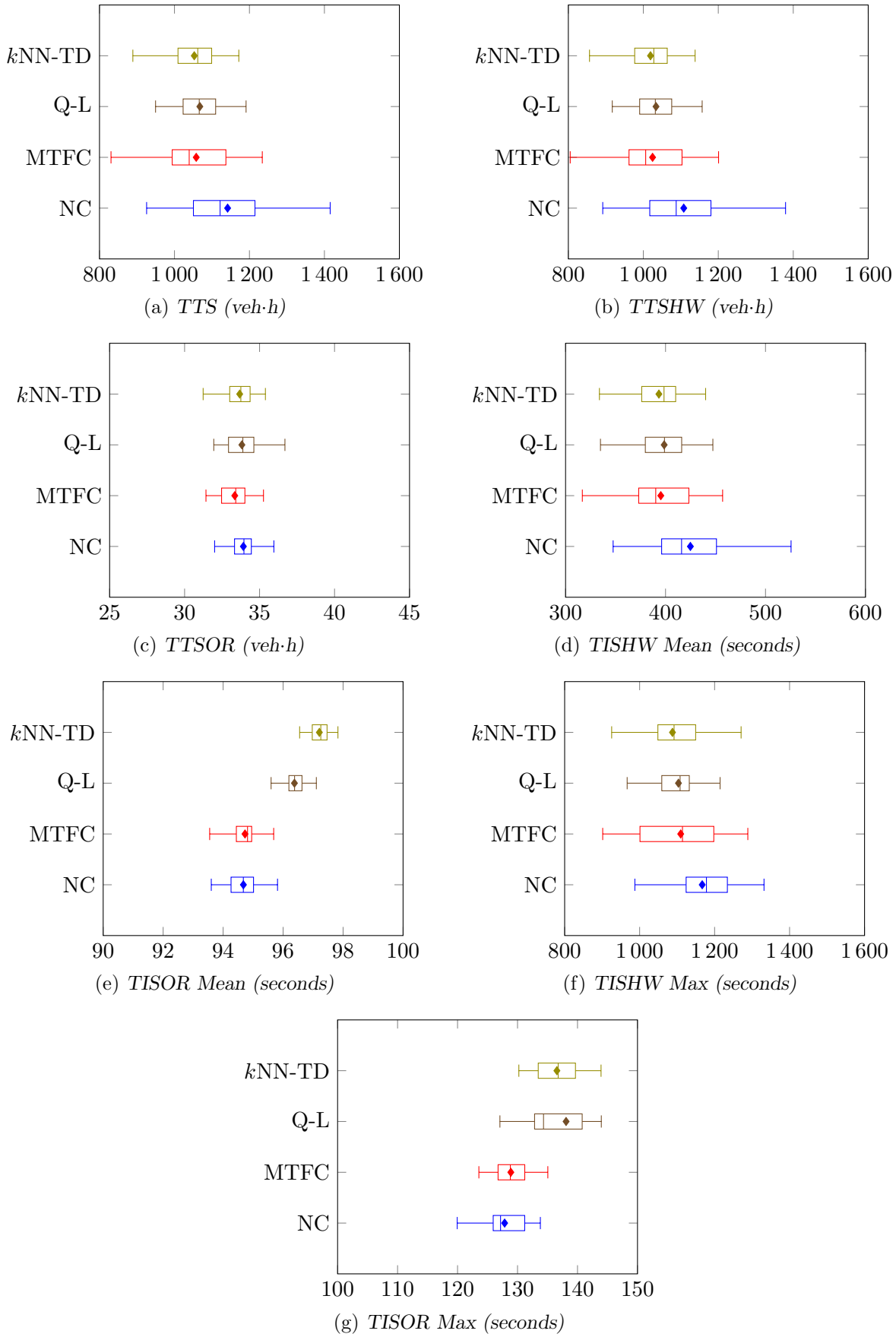
of the RL VSL implementations when compared with both the no-control case and MTFC for VSLs. This was confirmed by the Levene test and, as may be seen in Table 7.7, the Games-Howell test was subsequently performed. As shown in the table, all three VSL implementations outperformed the no-control case in respect of the TTS at a 5% level of significance, as MTFC for VSLs, Q-Learning and kNN-TD achieved values of 1 058.05 veh·h, 1 067.70 veh·h and 1 052.83 veh·h, respectively, compared with the value 1 141.80 veh·h of the no-control case.

In respect of the TTSHW, a trend very similar to that of the TTS emerged, as all three of the VSL implementations again outperformed the no-control case at a 5% level of significance, as may be seen in Table 7.8. MTFC and Q-Learning achieved improvements of 7.51% and 6.68%, respectively over the no-control case while kNN-TD learning achieved an improvement of 8.01% in respect of the TTSHW. Again the reason for the improvement in respect of the RL implementations may be the homogenisation effect, since, as may be seen in Figure 7.6(b), the lower whiskers of the box plots are located at similar positions as for the no-control case while the upper whiskers for both of the RL VSL implementations achieve significantly smaller values than those of the no-control case. For the MTFC implementation, on the other hand, an absolute improvement without the reduction in variance is observed, as may be seen from the box plot in Figure 7.6(b).

As may be seen from the results presented in Table 7.6, no statistical differences were found between the performances of the no-control case and either MTFC, Q-Learning or kNN-TD learning in respect of the TTSOR at a 5% level of significance, as the four cases achieved values of 33.92 veh·h, 33.52 veh·h, 33.82 veh·h and 33.68 veh·h, respectively. These similarities between all four cases in respect of the TTSOR are also evident in the box plots of Figure 7.6(c). This implies that, as may have been expected, the gains that are to be made due to VSL implementations are achieved through improved flow along the highway, while there is little to no trade-off with respect to increased travel times for vehicles joining the highway from the on-ramp.

For the mean and maximum travel times of the vehicles travelling along the highway only, a trend similar to that observed for both the TTS and TTSHW emerged. This is evident from the box plots in Figures 7.6(d) and 7.6(f). For the mean TISHW, MTFC, Q-Learning and kNN-TD learning again outperformed the no-control case, achieving savings of 29.60 seconds, 26.21 seconds and 31.50 seconds, respectively, as may be calculated from the results in Table 7.9. As shown in Table 7.11, these differences are further amplified for the maximum TISHW, where MTFC, Q-Learning and kNN-TD learning achieved savings of 57.38 seconds, 63.39 seconds and 79.36 seconds, respectively, over the no-control case. For both the mean and maximum TISHW, no statistical differences were identified between MTFC, Q-Learning and kNN-TD at a 5% level of significance.

Interestingly, although no statistical differences were found between the TTSOR-values for any of the four cases, the RL implementations were found to perform statistically different from one antoher, as well as both the no-control case and MTFC in respect of the mean TISOR-values at a 5% level of significance, as may be seen in Table 7.10. The differences in respect of the mean TISOR are clear in the box plots of Figure 7.6(e). From the figure it is evident that the no-control case and MTFC exhibited the best performance, achieving values of 94.68 seconds and 94.73 seconds, respectively, followed by Q-Learning with a value of 96.38 seconds. Finally, kNN-TD was outperformed by both the no-control case and Q-Learning at a 5% level of significance, as it achieved a value of 97.21 seconds in respect of the mean TISOR. Perhaps unexpectedly, the order of relative performances is not the same in respect of the maximum TISOR, as may be seen in Table 7.12. The no-control case and MTFC again yielded the best performance, outperforming both Q-Learning and kNN-TD, with values of 127.85 seconds and 128.88 seconds, respectively. MTFC was followed by kNN-TD, the performance of which was



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (seconds)*

(e) *TISOR Mean (seconds)*

(f) *TISHW Max (seconds)*

(g) *TISOR Max (seconds)*

FIGURE 7.6: *PMI results for the no-control case (NC), MTFC, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the VSL implementation in Scenario 2.*

found to be statistically indistinguishable from that of Q-Learning, as these two algorithms achieved maximum TISOR values of 131.82 seconds and 135.98 seconds, respectively. These results are summarised in the box plots of Figure 7.6(g).

TABLE 7.7: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTS			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	2.2942×10^{-2}	2.5372×10^{-2}	5.7953×10^{-3}
MTFC		—	9.7273×10^{-1}	9.9572×10^{-1}
Q-Learning			—	8.5756×10^{-1}
kNN-TD				—
Mean	1 141.80	1 058.05	1 067.70	1 052.83

TABLE 7.8: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTSHW			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	2.3264×10^{-2}	2.4486×10^{-2}	5.6554×10^{-3}
MTFC		—	9.7600×10^{-1}	9.9480×10^{-1}
Q-Learning			—	8.5890×10^{-1}
kNN-TD				—
Mean	1 107.88	1 024.70	1 033.87	1 019.16

TABLE 7.9: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	2.5232×10^{-2}	2.7766×10^{-2}	6.5086×10^{-3}
MTFC		—	9.7556×10^{-1}	9.9570×10^{-1}
Q-Learning			—	8.7018×10^{-1}
kNN-TD				—
Mean	424.86	395.26	398.65	393.36

Scenario 3

The ANOVA test performed on the PMI-values returned by the algorithms for Scenario 3 revealed that the TTSOR and the maximum TISOR are the only two PMIs for which the means of at least some pair of algorithmic output are statistically indistinguishable at a 5% level of significance, as may be seen in Table 7.13. Interestingly, the Levene test revealed that the variances of the sets of output data returned by the algorithms in respect of all PMIs associated purely with vehicles joining the highway from the on-ramp are statistically indistinguishable, while the variances of the output data sets returned by at least some pair of algorithms were found to be

TABLE 7.10: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TISOR Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	6.4893×10^{-1}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
MTFC		—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Q-Learning			—	1.1824×10^{-10}
kNN-TD				—
Mean	94.68	94.73	96.38	97.21

TABLE 7.11: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	2.2243×10^{-2}	1.1755×10^{-2}	1.7460×10^{-3}
MTFC		—	8.0870×10^{-1}	3.7665×10^{-1}
Q-Learning			—	5.2030×10^{-1}
kNN-TD				—
Mean	1 166.96	1 109.58	1 103.57	1 087.60

TABLE 7.12: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISOR Max			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	6.4253×10^{-1}	9.3126×10^{-6}	2.1277×10^{-11}
MTFC		—	4.5753×10^{-5}	2.7005×10^{-11}
Q-Learning			—	8.3101×10^{-1}
kNN-TD				—
Mean	127.85	128.88	135.98	131.82

statistically different at a 5% level of significance for all those PMIs based on data emanating from the vehicles travelling along the highway only, as may be seen in the table. Therefore, the Games-Howell test was performed in respect of the TTS, TTSHW, mean TISHW and maximum TISHW, while the Fisher LSD test was employed for the mean TISOR in order to determine between which sets of algorithmic output the differences occur in respect of these PMIs.

As for Scenario 2, the results obtained for Scenario 3 provide yet more evidence of a successful reduction in travel times due to the homogenisation effect of the RL VSL implementations on traffic flow. As may be seen from the smaller interquartile ranges of the box plots corresponding to the RL VSL implementations in Figure 7.7(a), the variances of both the RL VSL implementations are again significantly smaller than those of the no-control case. Although MTFC also returned smaller variances than the no-control case, the variances returned by the RL VSL implementations are smaller than that of the MTFC implementation. The result of

TABLE 7.13: *The mean values of all PMIs, as well as the results for the ANOVA and Levene statistical tests in Scenario 3. A P-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

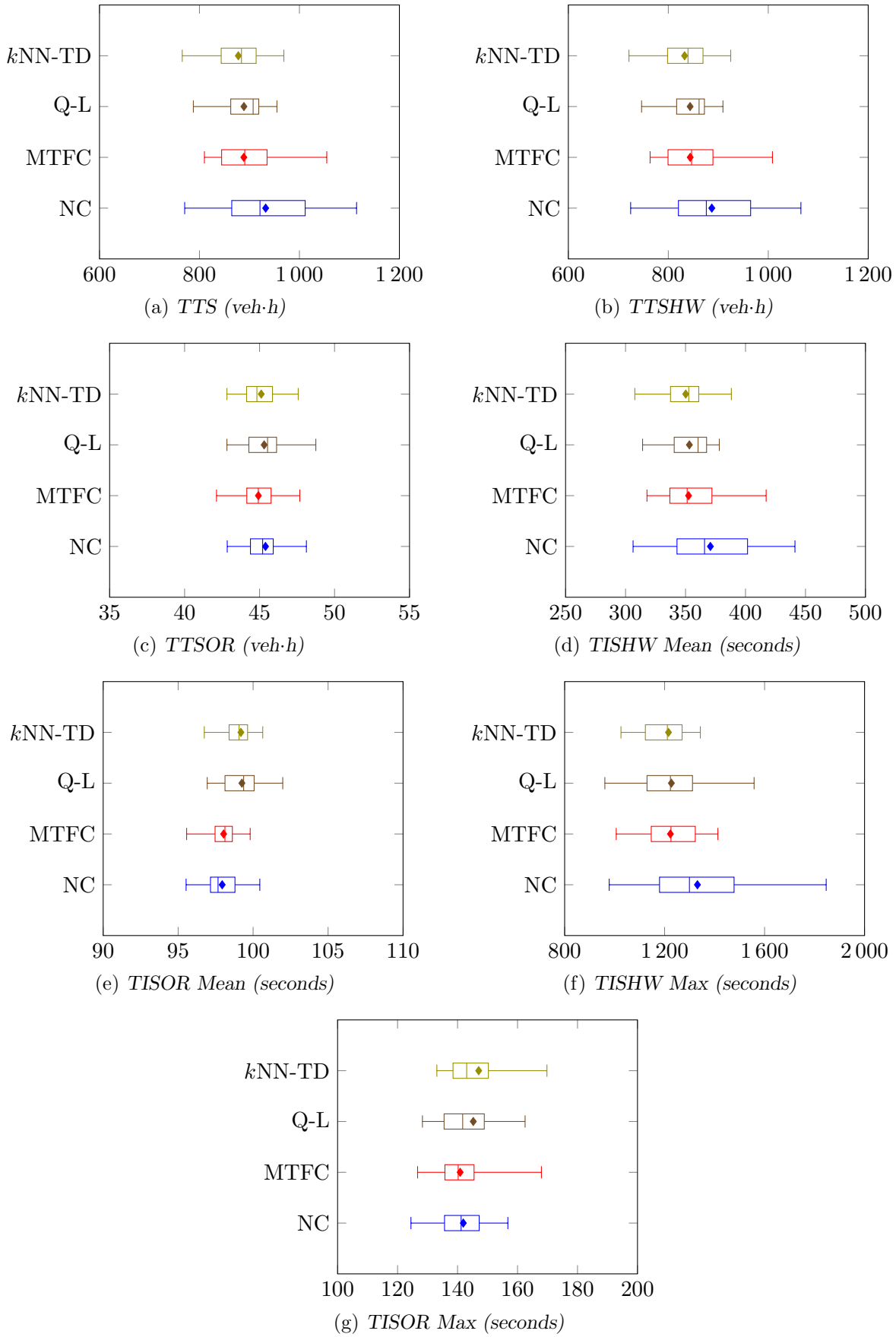
	Mean value				*p*-value	
PMI	No Control	MTFC	Q-Learning	*k*NN-TD	ANOVA	Levene's Test
TTS	932.46	888.79	888.97	877.59	1.1557×10^{-2}	2.7572×10^{-3}
TTSHW	887.07	843.87	843.66	832.62	1.1508×10^{-2}	2.7097×10^{-3}
TTSOR	45.40	44.92	45.31	45.11	6.4332×10^{-1}	9.2018×10^{-1}
TISHW Mean	370.72	352.53	353.21	350.16	1.2166×10^{-2}	1.7680×10^{-3}
TISOR Mean	97.94	98.03	99.25	99.19	1.1484×10^{-5}	2.3286×10^{-1}
TISHW Max	1 331.29	1 223.28	1 227.12	1 215.50	3.0662×10^{-2}	1.9290×10^{-2}
TISOR Max	141.92	140.86	145.24	147.05	1.3413×10^{-1}	1.8487×10^{-1}

these reduced variances was that *k*NN-TD outperformed the no-control case in respect of the TTS at a 5% level of significance, as may be seen in Table 7.14. Meanwhile no statistical differences were detected between the performances of MTFC, *k*NN-TD and Q-Learning at a 5% level of significance. Although MTFC and Q-Learning achieved smaller means of 888.79 veh·h and 888.97 veh·h, compared with the value 932.46 veh·h of the no-control case, these three cases are statistically indistinguishable at a 5% level of significance, as shown in Table 7.14.

As may have been expected, the trend that emerged in respect of the TTSHW is very similar to that for the TTS in Scenario 3, as may be seen in Figure 7.7(b). As may be seen in Table 7.15, the *k*NN-TD VSL implementation outperformed the no-control case, achieving a reduction of 6.14% in respect of the TTSHW, while MTFC, Q-Learning and the no-control case performed statistically similar at a 5% level of significance, although MTFC and Q-Learning were able to achieve reductions of 4.87% and 4.89%, respectively, over the no-control case in respect of the TTSHW. Finally, as was the case in respect of the TTS, no statistical differences were detectable between the performances of the *k*NN-TD, Q-Learning and MTFC VSL implementations in respect of the TTSHW at a 5% level of significance.

The differences between the performances of the algorithms in respect of the TTSOR were, as in Scenario 2, not large enough to prove statistically different at a 5% level of significance, as may be seen in Table 7.13. The MTFC implementation returned the smallest TTSOR-value of 44.92 veh·h, followed by the *k*NN-TD implementation which achieved a value of 45.11 veh·h. The *k*NN-TD implementation was followed by Q-Learning, which achieved a value of 45.31 veh·h, while the no-control case achieved a value of 45.40 veh·h. The similarity in performance of all three implementations in respect of the TTSOR is also evident in the box plots of Figure 7.7(c).

As is clear in Figures 7.7(d) and 7.7(f), all three of the VSL implementations again achieved improvements in respect of both the mean and maximum travel times of vehicles travelling along the highway only. As may be seen in Table 7.16, the *k*NN-TD VSL implementation outperformed the no-control case, achieving a saving of 20.56 seconds in the mean TISHW. Although Q-Learning and MTFC achieved savings of 17.51 seconds and 18.19 seconds, respectively, they could not be classified as statistically different from the no-control case at a 5% level of significance. As shown in Table 7.18, the ranking in respect of the maximum TISHW is the same as that for the mean TISHW, as the *k*NN-TD VSL implementation outperformed the no-control case, achieving a saving of 115.79 seconds, while it performed statistically indistinguishable from both MTFC and Q-Learning at a 5% level of significance. In respect of the maximum TISHW, Q-Learning achieved a reduction in the maximum TISHW of 104.17 seconds, while MTFC achieved a reduction of 108.01 seconds, but these reductions were again not enough for these three cases to be classified as statistically different at a 5% level of significance.



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (seconds)*

(e) *TISOR Mean (seconds)*

(f) *TISHW Max (seconds)*

(g) *TISOR Max (seconds)*

FIGURE 7.7: *PMI results for the no-control case (NC), MTFC, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the VSL implementation in Scenario 3.*

Similarly to what was observed in Scenario 2, the RL VSL implementations led to increases in both the mean and maximum TISOR-values, as is evident from the box plots in Figures 7.7(e) and 7.7(g). This is substantiated by the results presented in Table 7.17, where it is shown that the no-control case and MTFC outperformed both of the RL VSL implementations in respect of the mean TISOR at a 5% level of significance. The RL VSL implementations resulted in approximately a 2-second increase in the time spent in the system by the vehicles joining the highway from the on-ramp. Although MTFC achieved the smallest maximum TISOR-value, followed by the no-control case and Q-Learning, while kNN-TD learning returned the largest average maximum TISOR-value, the performances of all the algorithms were statistically indistinguishable at a 5% level of significance, as shown in Table 7.13.

TABLE 7.14: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS		
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	1.6532×10^{-1}	1.0718×10^{-1}	3.6384×10^{-2}
MTFC		—	9.9999×10^{-1}	9.0322×10^{-1}
Q-Learning			—	8.3642×10^{-1}
kNN-TD				—
Mean	932.46	888.79	888.97	877.59

TABLE 7.15: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW		
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	1.6851×10^{-1}	1.0545×10^{-1}	3.6499×10^{-2}
MTFC		—	9.9999×10^{-1}	8.9682×10^{-1}
Q-Learning			—	8.4104×10^{-1}
kNN-TD				—
Mean	887.07	843.87	843.66	832.62

TABLE 7.16: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean		
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	1.3101×10^{-1}	9.6805×10^{-2}	4.5159×10^{-2}
MTFC		—	9.9948×10^{-1}	9.8199×10^{-1}
Q-Learning			—	9.3607×10^{-1}
kNN-TD				—
Mean	370.72	352.53	353.21	350.16

TABLE 7.17: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TISOR Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	7.6995×10^{-1}	1.0525×10^{-4}	2.2114×10^{-4}
MTFC		—	3.0499×10^{-4}	6.1688×10^{-4}
Q-Learning			—	8.3952×10^{-1}
kNN-TD				—
Mean	97.94	98.03	99.25	99.19

TABLE 7.18: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISHW Max			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	1.2060×10^{-1}	1.3169×10^{-2}	1.3223×10^{-2}
MTFC		—	9.9949×10^{-1}	9.9765×10^{-1}
Q-Learning			—	9.9174×10^{-1}
kNN-TD				—
Mean	1 331.29	1 223.28	1 227.12	1 215.50

Scenario 4

As may be seen in Table 7.19, the p-values returned by the ANOVA test performed in respect of the PMI-values returned by the algorithms for Scenario 4 indicate that there are, in fact, statistical differences at a 5% level of significance between the means of at least some pair of algorithmic output for all mean and maximum TISHW and TISOR PMIs, while the relative algorithmic performances were found to be statistically indistinguishable at a 5% level of significance for the TTS, TTSHW and TTSOR PMIs. As in Scenario 3, the Levene test again revealed that the variances of the PMI data sets returned by at least some pair of algorithms were found to differ statistically at a 5% level of significance in respect of all PMIs for vehicles travelling along the highway only, while these variances in respect of the PMIs associated only with vehicles joining the highway from the on-ramp were found to be statistically indistinguishable, as may be seen in the table. Therefore, the Fisher LSD test was performed in order to establish between which pairs of algorithmic output the differences occur in respect of the mean and maximum TISOR, while the Games-Howell test was employed for this purpose in respect of the mean and maximum TISHW.

The results obtained by the RL VSL implementations in Scenario 4 provide yet further evidence in support of the notion that VSLs lead to a homogenisation of traffic flow if these are applied for relatively long highway sections. This is again evident from the reduced width of the interquartile ranges of the box plots corresponding to the RL VSL implementations in Figure 7.8(a). Interestingly, however, MTFC returned a significant increase in the variance corresponding to the TTS, as is evident from the figure. The statistical significance of the differences in variances of the algorithmic output is evident from the result of Levene's test, as may be seen in Table 7.19. The kNN-TD VSL implementation reduced the TTS to 533.23 veh·h, compared with the values

of 550.00 veh·h, 548.39 veh·h and 547.41 veh·h for the no-control case, MTFC and Q-Learning, respectively. Although all three VSL implementations were once again able to improve on the no-control case, these improvements were not large enough to classify these performances as statistically different at a 5% level of significance.

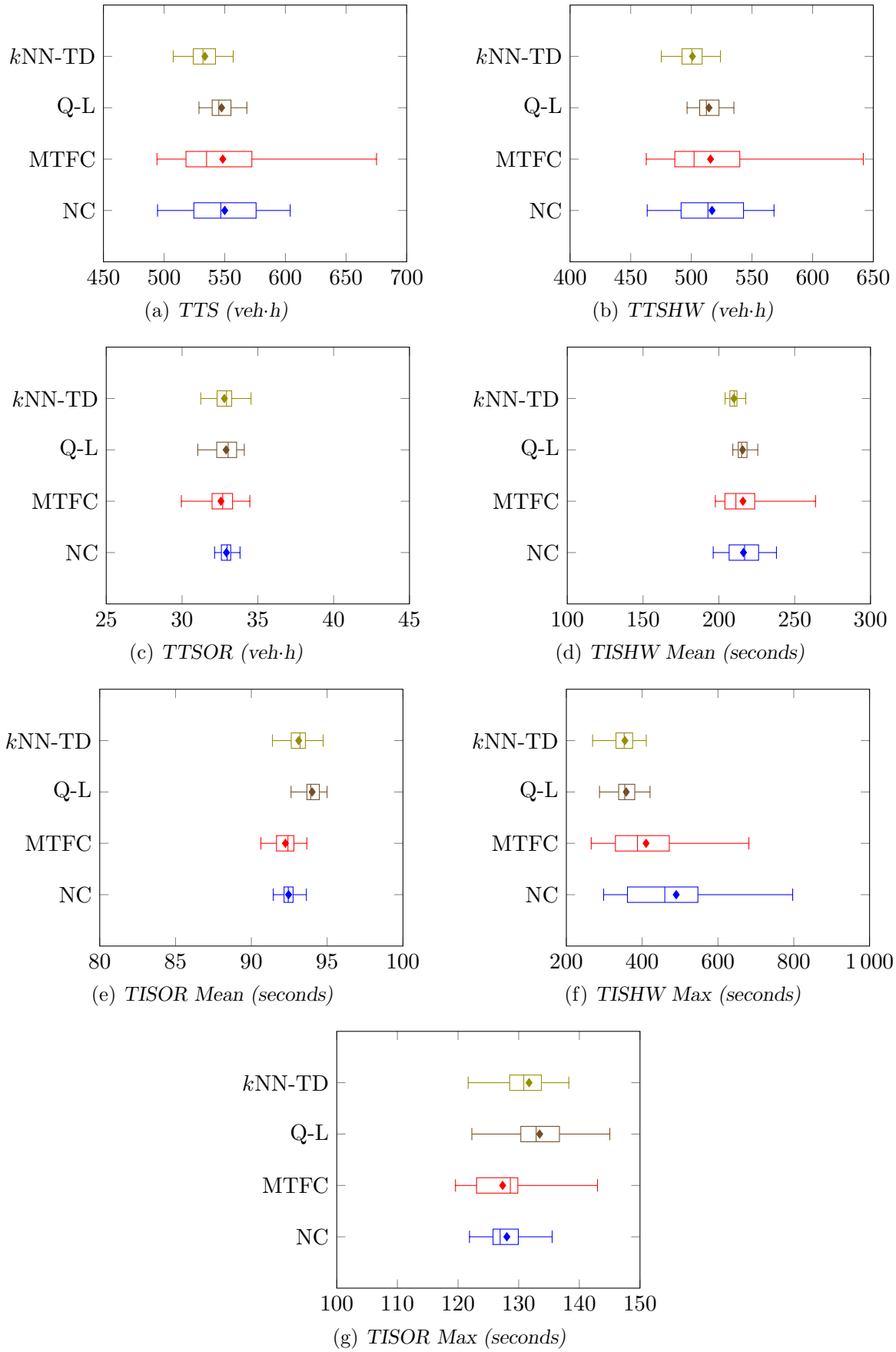
TABLE 7.19: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 4. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

PMI	Mean value				p-value	
	No Control	MTFC	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	550.00	548.39	547.41	533.23	7.83556×10^{-2}	2.2395×10^{-11}
TTSHW	517.07	515.82	514.50	500.88	7.1299×10^{-2}	1.4262×10^{-11}
TTSOR	32.93	32.57	32.91	32.79	4.1885×10^{-1}	5.2780×10^{-1}
TISHW Mean	216.21	215.89	215.65	209.89	4.6530×10^{-2}	3.1871×10^{-12}
TISOR Mean	92.46	92.25	94.01	93.12	1.1102×10^{-16}	2.3170×10^{-1}
TISHW Max	489.89	410.39	358.04	354.46	9.6053×10^{-7}	1.3882×10^{-9}
TISOR Max	128.05	127.35	133.46	131.71	1.7922×10^{-6}	5.3830×10^{-1}

In respect of the TTSHW, a trend very similar to that of the TTS again emerged, as may be seen in Figure 7.8(b). From the results presented in Table 7.19 it was determined that the kNN-TD implementation returned the smallest TTSHW-value, improving on the no-control case by 3.13%, while Q-Learning reduced the TTSHW by 0.50%, and MTFC achieved a reduction of only 0.24%. Similarly to the TTS, however, these differences were not large enough for the algorithmic performance to be classified as statistically different at a 5% level of significance.

It is interesting to note that there seemed to be a marginal increase in the variance of the TTSOR values for both the RL VSL implementations, indicated by the larger interquartile ranges in the box plots corresponding to the RL VSL implementations when compared with that of the no-control case in Figure 7.8(c). As may be seen in the figure, MTFC did, however, again return an even larger variance in respect of the TTSOR-values when compared with the RL implementations. These differences were, however, found not to be of statistical significance when applying Levene's test as shown in Table 7.19. As was the case in both Scenario 2 and Scenario 3, the TTSOR values for the four cases were found not to be statistically different at a 5% level of significance, with the no-control case, MTFC, Q-Learning and kNN-TD achieving TTSOR-values of 32.93 veh·h, 32.57 veh·h, 32.91 veh·h and 32.79 veh·h, respectively.

The kNN-TD VSL implementation exhibited the best performance in respect of the mean TISHW, outperforming the no-control case at a 5% level of significance, as may be seen from the results in Table 7.20. As was the case for both the TTS and TTSHW, MTFC, Q-Learning and the no-control case performed statistically indistinguishably at a 5% level of significance. Furthermore, kNN-TD, Q-Learning and MTFC were also found to perform statistically indistinguishable at a 5% level of significance in respect of the mean TISHW. In Figure 7.8(d), the box plots corresponding to the mean TISHW are shown. From these box plots it is evident that there is homogenisation of traffic flow due to the RL VSLs, as suggested by the reduced width of the interquartile ranges of the box plots for both the RL VSL implementations. A similar trend is seen for the maximum TISHW, apart from the fact that now both Q-Learning and kNN-TD outperformed the no-control case, while MTFC and the no-control case were, again, found to perform statistically indistinguishably, as shown in Table 7.22. From the box plots in Figure 7.8(f), one may argue that these improvements by the RL VSL implementations are due a combination of an absolute improvement in performance in respect of the maximum TISHW values, as well as more stable traffic flow due to the homogenisation of traffic flow. Finally, the



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (seconds)*

(e) *TISOR Mean (seconds)*

(f) *TISHW Max (seconds)*

(g) *TISOR Max (seconds)*

FIGURE 7.8: *PMI results for the no-control case (NC), MTFC, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the VSL implementation in Scenario 4.*

two RL VSL implementations were proven not to perform statistically different from one another or the MTFC implementation in respect of the maximum TISHW at a 5% level of significance.

In respect of the mean TISOR, the MTFC implementation and the no-control case again resulted in the best performance, outperforming both Q-Learning and kNN-TD, as may be seen in Table 7.21. MTFC and the no-control case were followed by kNN-TD, for which the vehicles joining the highway from the on-ramp typically took 0.66 seconds longer than in the no-control case to travel through the system. The kNN-TD implementation outperformed Q-Learning, by which the vehicles took 1.55 seconds longer to travel through the system than they would in the no-control case. These differences are illustrated graphically in the box plots of Figure 7.8(e). Although these values were found to be statistically different at a 5% level of significance, one may argue that they do not have much practical significance, as a 1-second delay may not be noticeable by a travelling motorist. As may be expected, in respect of the maximum TISOR, these differences were amplified. Again MTFC and the no-control case exhibited the best performance, outperforming both Q-Learning and kNN-TD, while the two RL VSL implementations could not be proven to perform statistically different at a 5% level of significance, as shown in Table 7.23. These differences are also evident in the box plots of Figure 7.8(g).

TABLE 7.20: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	9.9973×10^{-1}	9.9943×10^{-1}	3.5093×10^{-2}
MTFC		—	9.9979×10^{-1}	1.8569×10^{-1}
Q-Learning			—	5.0783×10^{-7}
kNN-TD				—
Mean	216.21	215.89	215.65	209.89

TABLE 7.21: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TISOR Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	2.5829×10^{-1}	1.2546×10^{-13}	4.6665×10^{-4}
MTFC		—	3.3307×10^{-16}	6.1779×10^{-5}
Q-Learning			—	4.7495×10^{-6}
kNN-TD				—
Mean	92.46	92.25	94.01	93.12

Discussion

As was the case in the RM implementations of Chapter 6, the kNN-TD VSL implementation consistently achieved the best performance, never once being outperformed in terms of the TTS. Q-Learning, although not quite as effective as kNN-TD, was also consistently able to perform at least as well as the no-control case, while it outperformed the no-control case in respect of the TTS in Scenario 2. Furthermore, both of the RL VSL implementations consistently performed

TABLE 7.22: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISHW Max			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	1.2824×10^{-1}	7.2813×10^{-4}	5.4420×10^{-4}
MTFC		—	7.5548×10^{-2}	5.5325×10^{-2}
Q-Learning			—	9.8418×10^{-1}
kNN-TD				—
Mean	489.89	410.39	358.04	354.46

TABLE 7.23: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TISOR Max			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	5.7608×10^{-1}	2.7522×10^{-5}	3.7812×10^{-3}
MTFC		—	2.7911×10^{-6}	6.2554×10^{-4}
Q-Learning			—	1.6076×10^{-1}
kNN-TD				—
Mean	128.05	127.35	133.46	131.71

at least on par with the MTFC benchmark implementation, as they were never outperformed in respect of either the TTS or TTSHW PMIs. From the results presented, it may be concluded that VSLs (when implemented as in both of the RL implementations) can be effective in improving highway traffic flow, through traffic flow homogenisation, while the improvements achieved by MTFC are generally down to an absolute reduction in the travel times along the highway. As may be seen in all four scenarios, the minimum TTS-values in the box plots are roughly the same for both of the RL VSL implementations and the no-control case. The improvements achieved by the TL VSL algorithms are thus largely due to the reduced variance in travel times by the vehicles travelling along the highway only. These reduced variances are, however, not reflected in the algorithmic performance of the MTFC implementation.

It was furthermore found that none of the VSL implementations have practically significant effects on the traffic flow entering the highway stream from the on-ramp, as corroborated by the fact that the TTSOR-values were statistically indistinguishable at a 5% level of significance in all four scenarios. This implies that the increases in the travel time of vehicles joining the highway from the on-ramp, as reflected by the mean and maximum TISOR-values in the RL VSL implementations, are usually not large enough to have a practical effect on system level, while the travel times for vehicles joining the highway from the on-ramp were generally statistically similar to the no-control case in the MTFC VSL implementation. The fact that there is no statistically significant increase in the travel times due to the MTFC implementation may be down to the fact that an acceleration area was employed in the MTFC implementation, resulting in the fact that the vehicles entering the highway from the on-ramp are never subjected to VSLs, directly or indirectly.

7.6 Chapter Summary

This chapter opened in §7.1 with a description of the implementation of the MTFC VSL controller by Müller *et al.* [105]. This was followed by a formulation of the VSL problem as an RL problem, containing descriptions of the state and action spaces as well as the reward function employed in §7.2. This formulation was followed by a description in §7.3 of the Q-Learning implementation for solving the RL problem adopted in this dissertation. Thereafter, the kNN-TD learning approach adopted for solving the VSL RL problem was discussed in §7.4. The computational results of the implementations of feedback-based MTFC and both the Q-Learning algorithm and the kNN-TD algorithm were presented in §7.5 within the context of the benchmark model of §5.1.2. A complete parameter evaluation was carried out in §7.5.1 in order to identify the best combination of target density and controller parameter K_I for MTFC and finding the best speed limit adjustment rule. Finally, the relative algorithmic performances were compared in §7.5.2 for each of the four scenarios of varying traffic demand described in §5.3.2.

CHAPTER 8

Multi-Agent Reinforcement Learning

Contents

The purpose of this chapter is to provide the reader with a detailed description of the implementation of *multi-agent reinforcement learning* (MARL) for integrated RM and VSL control adopted in this dissertation. The chapter opens in §8.1 with a brief description of the working of a feedback-based controller for integrating RM and VSLs. This is followed by a brief introduction to the expansive field of MARL in §8.2, with a specific focus on the techniques implemented in this dissertation. Thereafter, the adaptations of these techniques required for their application within the highway traffic control problem are discussed in §8.3. The relative algorithmic performances of the MARL implementations are thoroughly evaluated in §8.4. Queueing limitations are again implemented within the RM component of the MARL approaches in §8.5, and an algorithmic comparison, taking these queue limits into account is performed in §8.5.2. The chapter finally closes in §8.6 with a brief summary of the work included in the chapter.

8.1 An Integrated RM and VSL Feedback Controller

Similarly to the individual RM and VSL implementations, a feedback controller is implemented in this dissertation as a benchmark against which the performances of the various MARL ap-

proaches implemented in this dissertation can be measured. The integrated feedback controller of Carlson *et al.* [24] is implemented for this purpose. In this controller, a combination of separate RM and MTFC controllers is employed in order to regulate the traffic flow around a bottleneck. These controllers are linked by means of a so-called split block. The operation of this split block is relatively simple. RM is applied until one of two restrictions applies: 1) the lower bound of the RM controller has been reached (*i.e.* the longest allowable red phase time is applied), or 2) the queue management orders a higher value (*i.e.* the queue management component dictates that the red phase time should be shortened due to long on-ramp queue build-up), in which case the RM-ordered flow is set equal to the flow determined according to the queue management controller so as to prevent the formation of excessively long on-ramp queues. In either of these two cases, the MTFC controller is subsequently engaged in order to provide additional metering of the traffic flow into the bottleneck [24]. Due to their proven performance in the individual cases, PI-ALINEA and the MTFC controller of Müller *et al.* [105] were chosen as the two controllers to be implemented for integrated RM and MTFC. Furthermore, Carlson *et al.* [24] suggested the use of the queue management rule of Smaragdis and Papageorgiou [150] in (6.9) in order to prevent the build-up of excessively long on-ramp queues when the integrated controller is employed.

8.2 An Introduction to Multi-Agent Reinforcement Learning

In the literature review on the reinforcement learning problem in Chapter 2, a single agent was assumed to interact with its immediate environment in search of an optimal policy. In theory, this approach may be applied in order to find optimal actions for both RM and VSLs combined, using a single agent. In such a scenario, the agent's action would be the selection of a combination of a red phase duration for the traffic signal at the on-ramp and a VSL for the vehicles on the highway. The problem with this approach, however, is that the state space of such a single agent is very large compared with the state spaces of the single agents, as implemented in Chapters 6 and 7. As a result of this increase in the number of states and actions, the learning process requires significantly more learning time [130]. Although this approach is sound in theory, it is not practical for solving large problems. Employing a multi-agent approach is a viable alternative solution to this problem. Buşoniu *et al.* [21] defined a multi-agent system as "a group of autonomous, interacting entities sharing a common environment, which they perceive with sensors and upon which they act with actuators." MARL problems are those problems in which reinforcement learning is applied within a multi-agent context. Several approaches toward solving MARL problems have been proposed in the literature. A few of these approaches are briefly discussed in this section.

8.2.1 Independent Learners

Perhaps the simplest approach towards solving MARL problems is that of employing independent learners. In this paradigm, the environment is partitioned into a number of smaller sub-environments (each typically only containing a single reinforcement learning agent). Each of these independent reinforcement learning agents then observes the state space in its immediate environment and chooses an action so as to maximise its local reward, irrespective of the choices of the other agents. Although the agents' actions are locally optimal, there is no guarantee of global optimality for the entire network of agents. Furthermore, the lack of coordination between agents limits the opportunities for effective cooperation (*e.g.* reducing the speed limit

on the highway directly upstream of an on-ramp so as to create gaps between vehicles in order to allow more vehicles to enter the highway stream from the on-ramp).

8.2.2 Cooperative Reinforcement Learning

Considering that a decentralised structure is the only practical solution for applying reinforcement learning to larger networks which include multiple agents, there is often a need for coordination among these agents. This need for coordination stems from the fact that the effect of any agent's action also inherently depends on the actions taken by other agents [157]. Therefore, the choice of actions taken by different agents should be mutually consistent in order to achieve the desired effect on the environment. This desired effect of course depends on the specific type of problem at hand (*e.g.* competitive, cooperative or mixed problems). As a result, MARL algorithms are usually tailored to specific problem types [130]. Buşoniu *et al.* [21] provided an overview of MARL techniques employed for various different problem types.

All approaches towards solving cooperative MARL problems (*i.e.* problems in which multiple agents work together towards achieving a common goal) involve some sort of communication between the agents. This communication may involve transmitting state, action or reward information, or combinations thereof. Finding an effective communication protocol between agents is a challenge due to the typically dramatic increase in the size of the state-action space as the amount of communicated information increases. Consider, for example, the case where two agents, with state spaces $\mathcal{S}^1$ and $\mathcal{S}^2$, respectively, communicate state information with each other. Then the joint state space has size $|\mathcal{S}^1| \times |\mathcal{S}^2|$. Furthermore, the size of the state-action space of agent 1 is $|\mathcal{S}^1| \times |\mathcal{A}^1| \times |\mathcal{A}^2|$, where $\mathcal{A}^1$ and $\mathcal{A}^2$ denote the sets of actions for agents 1 and 2, respectively. It is therefore imperative to keep this rapid increase in the size of state-action space in mind when designing MARL systems.

A fairly simple approach towards joint action selection in cooperative MARL problems is that of employing so-called "social conventions" in the action selection process [21]. According to this approach, the agents are ordered. The binary relation *agent 1* $<$ *agent 2*, for example, would indicate that agent 1 precedes agent 2 in the ordering. In order to facilitate coordination, the first agent in the ordering (agent 1) chooses an optimal action. This action is then communicated to agent 2. Thereafter, the agent in the second position of the ordering chooses its optimal action, taking into account the action that agent 1 has chosen. The third agent of the ordering then chooses its optimal action, taking into account the actions chosen by agents 1 and 2, and so forth. This approach is henceforth referred to as *hierarchical MARL*.

The most notable application of a MARL methodology to a traffic control optimisation problem is attributed to El-Tantawy *et al.* [157] who employed the so-called *multi-agent reinforcement learning for an integrated network of adaptive traffic signal controllers* (MARLIN-ATSC) algorithm in a case study involving an urban network of signalised traffic intersections in downtown Toronto. The working of the MARLIN-ATSC algorithm is based on that of the Q-learning algorithm which has been adapted so as to be applicable within a MARL context. As a result, the MARLIN-ATSC approach is, as Q-learning, a table-based approach. According to the MARLIN-ATSC approach, each signalised intersection (agent) plays a game with all the adjacent intersections in its neighbourhood. The state and action spaces are distributed in such a manner that an agent learns a joint optimal policy with one of its neighbours at a time. The learning approach designed for the MARLIN-ATSC algorithm is as follows:

1. Suppose the set of neighbours of agent i is denoted by $\mathcal{N}^i$. Then there exist $|\mathcal{N}^i|$ partial state and action spaces for agent i. These partial state and action spaces consist of the

state and action space of agent i as well as the state and action space of each neighbour $N \in \mathcal{N}^i$. Consequently, the partial state space has a size $|\mathcal{S}|^2$, assuming that the state spaces for the agents are identical.

2. A model is built by each agent for the purpose of estimating the policy for each of its neighbours. The model of agent i and one of its neighbours $N \in \mathcal{N}^i$ is represented by a matrix $\boldsymbol{M}^{i,N}$ whose rows represent the joint states $\mathcal{S}^i \times \mathcal{S}^N$ and whose columns represent the neighbour's actions $\mathcal{A}^N$. Each entry in the matrix $\boldsymbol{M}^{i,N}$ represents the probability that the neighbouring agent N takes action a^N when in joint state $[s^i, s^N]$. This probability is estimated based on the number of visits by the neighbouring agent to each action a^N when the system is in the joint state $[s^i, s^N]$.

3. Each agent learns an optimal joint policy for itself and each of its neighbours by updating a matrix of joint Q-values. This matrix for agent i and one of its neighbours $N \in \mathcal{N}^i$ is denoted by $\boldsymbol{Q}^{i,N}$ and consists of $|\mathcal{S}^i| \times |\mathcal{S}^N|$ rows and $|\mathcal{A}^i| \times |\mathcal{A}^N|$ columns, where the entry in row $[s^i, s^N]$ and column $[a^i, a^n]$ of $\boldsymbol{Q}^{i,N}$ represents the Q-value for a state-action pair in the partial state space of agent i and its neighbour N.

4. Agent i updates the Q-values in $\boldsymbol{Q}^{i,N}$ based on the best-response action taken in the next state at time $t+1$. This best-response value is calculated by determining the most likely neighbour action a^N_{t+1} in the next joint state $[s^i_{t+1}, s^N_{t+1}]$ from $\boldsymbol{M}^{i,N}$, and subsequently identifying the action a^i_{t+1} which yields the maximum expected Q-value from $\boldsymbol{Q}^{i,N}$. The updated Q-value is then determined using the standard update rule employed in Q-learning, as described in Algorithm 2.3.

5. This process is repeated at every time step t for all agents i and all of their neighbours $N \in \mathcal{N}^i$.

According to this approach, each agent decides upon its action without explicit interaction with its neighbours, but rather implicitly through the use of estimated models for the neighbouring agent's behaviour. Invoking the so-called "principle of locality of interaction among agents," Nair *et al.* [106] showed that through the estimation of a local neighbourhood utility, a mapping of an agent's effect on the global value function is created while only considering the interaction of an agent and its neighbour. Hence it was deemed sufficient by El-Tantawy *et al.* [157] to consider only a single neighbour's policy in order to find the best policy for each agent.

8.3 MARL for Highway Traffic Control

Three approaches towards solving the MARL problem for the RM and VSL problems are adopted in this dissertation. The first and simplest of these approaches is that of employing independent learners. The second approach is that of employing social conventions for joint action selection. This approach requires explicit communication between the agents in terms of action selection. The third, and most sophisticated approach is, as the MARLIN-ATSC approach, based on the principle of locality of interaction among agents. As with the approach of adopting social conventions, agent actions are communicated explicitly by each agent with its neighbour. The joint actions of the agent and its neighbour are then updated iteratively from an initial random action in search of an optimal joint action. This process, which is similar to the one described by Rezaee [130], is as follows for each agent in turn:

1. Agent i chooses an initial action which is communicated to its neighbour j.

2. Agent i finds the action a^i_{t+1} which results in the maximum joint Q-value given by

$$\max_{a^i_{t+1}} \left[Q^i \left(s^i_{t+1}, a^i_{t+1}, a^j_t \right) + Q^j \left(s^j_{t+1}, a^j_t, a^i_{t+1} \right) \right] \tag{8.1}$$

   with its neighbour.

3. The *gain* in the joint Q-value is calculated for each agent i if it were to change its action.

4. Only the agent who is able to achieve the largest gain in the joint Q-value is allowed to change its action, while the action of that agent's neighbour remains unchanged. The process is then repeated from Step 2 until no agent is able to achieve an increase in the joint gain by changing its action.

The advantage of this approach over the approach implemented by El-Tantawy *et al.* [157] is that the partial state-action space only increases by a factor of $|\mathcal{A}^N|$ compared with the increase by a factor of $|\mathcal{S}^N| \times |\mathcal{A}^N|$ in the MARLIN-ATSC algorithm. Due to the fact that according to the above approach both agents aim to find the maximum gain, this approach is henceforth referred to as the *maximax MARL approach*.

8.3.1 Independent MARL for RM and VSL

As stated above, multiple agents learn in parallel without any form of explicit communication according to the independent learner approach towards solving the MARL problem. Thus the implementations for both the RM and the VSL agents of Chapters 6 and 7 remain unchanged, apart from the fact that the control interval t is reduced from 5 minutes in the original VSL implementation to 2 minutes in the independent MARL solution approach. The reason for this reduction is that the agents should make decisions at the same points in time such that an update to a Q-value is only affected by two actions — a single action from the agent itself, and one from the neighbouring agent. It is anticipated that if this is not the case, large variations in reward (due to multiple actions taken by the neighbouring agent during a single control interval) may result in unstable Q-values which may, in turn, negatively affect the learning process.

8.3.2 Hierarchical MARL for RM and VSL

Due to the fact that the RM agent has on its own previously been able to achieve greater improvements in terms of the PMIs of §5.1.4, as reported in Chapter 6, the agent ranking is chosen in such a way that the RM agent gets to choose its action first in the hierarchical control approach towards solving the MARL problem. In this ordering, the RM agent is denoted by agent i, while the VSL agent is denoted by agent j. A graphical illustration of the flow of information when executing the hierarchical MARL approach is shown in the flow chart of Figure 8.1.

In this hierarchical MARL for RM and VSL implementation, the kNN-TD learning algorithm (Algorithm 2.6) is employed as the underlying learning algorithm. As may be seen in the figure, the states s^i_{t+1} and s^j_{t+1}, as well as the rewards r^i_t and r^j_t, are observed from the environment. Thereafter, the k nearest neighbours are determined for both agents, and the Q-values of the centre-action pairs are updated, as described in Algorithm 2.6. Next agent i (the RM agent) chooses its action a^i_{t+1}, which is then communicated to its neighbouring agent j (the VSL

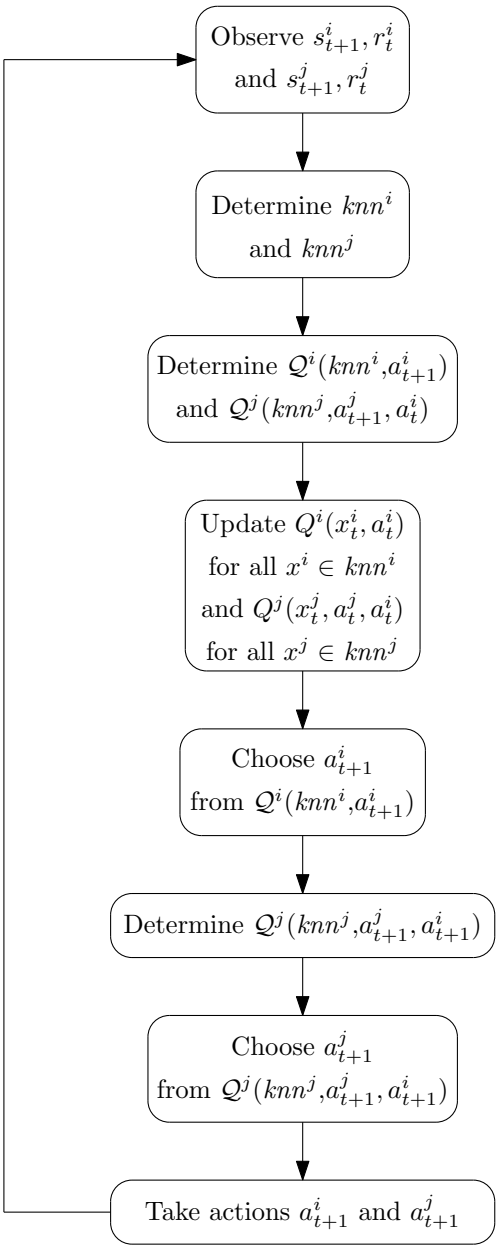


FIGURE 8.1: *The flow of information, as well as the sequence of events, during execution of the hierarchical MARL algorithm.*

agent). After receiving information about the action chosen by agent i, an updated $\mathcal{Q}$-value[1] is determined for agent j, which takes the action of agent i into account. Taking into account this updated $\mathcal{Q}$-value, agent j then chooses its action.

Note that, due to the fact that there is only a one-way communication of agent i's action with agent j, the $\mathcal{Q}^i$-value is calculated using only the k nearest neighbours knn^i as well as the available actions a^i_{t+1} of agent i, while agent i's action a^i_t has to be incorporated in the calculation of these for agent j, resulting in the amended notation $\mathcal{Q}^j(knn^j, a^j_{t+1}, a^i_t)$. The same applies for the updating of the centre-action values Q^i and Q^j for agents i and j, respectively.

[1]Recall from Algorithm 2.6 that the $\mathcal{Q}$-value denotes the approximated state-action value, calculated using the tabulated Q-values corresponding to each of the centre-action pairs $x \in knn$ as well as the corresponding weights.

In order to still effectively manage the balance between exploration of the state-action space, and exploitation of what has already been learned, the same adaptive rules, given in (6.5) and (6.6), for determining the learning rate α, as well as the ϵ-value employed in the ϵ-greedy action selection, are employed.

8.3.3 Maximax MARL for RM and VSL

As was the case in the hierarchical MARL approach, the RM agent is, for the sake of consistency, denoted by agent i and the VSL agent is denoted by agent j in this description of the maximax MARL approach. A graphical illustration of the process for the maximax MARL, as suggested by Rezaee [130], is shown in Figure 8.2, once again employing the kNN-TD learning algorithm as the underlying reinforcement learning algorithm.

As may be seen in the figure, the states s^i_{t+1} and s^j_{t+1}, as well as the rewards r^i_t and r^j_t for both agents, are observed from the environment. Thereafter, the k nearest neighbours are once again determined for both agents, according to the method outlined in Algorithm 2.6. Once these k nearest neighbours have been determined, the Q-values of all the centre-action pairs $(x^i_t, a^i_t, a^j_t) \in knn^i$ and $(x^j_t, a^j_t, a^i_t) \in knn^j$ are updated before action selection is performed.

Due to the fact that the RM agent receives negative rewards, while the VSL agent receives positive rewards, the joint gain for the agents cannot simply be taken as the sum of the respective $\mathcal{Q}$-values as suggested by Rezaee [130]. In order to compensate for this difference, as well as the differences in magnitude of rewards, the joint gain is calculated as a proportional increase in the sum of the $\mathcal{Q}$-values. Therefore, the joint gain for the RM agent is

$$G_{a^i_{t+1}} = \max_{a^i_{t+1}} \left[\frac{\mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_t) - \mathcal{Q}^i(s^i_{t+1}, a^i_{t+1}, a^j_t)}{\mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_t)} + \frac{\mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_{t+1}) - \mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_t)}{\mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_{t+1})} \right], \tag{8.2}$$

while the joint gain for the VSL agent is

$$G_{a^j_{t+1}} = \max_{a^j_{t+1}} \left[\frac{\mathcal{Q}^j(s^j_{t+1}, a^j_{t+1}, a^i_t) - \mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_t)}{\mathcal{Q}^j(s^j_{t+1}, a^j_{t+1}, a^i_t)} + \frac{\mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_t) - \mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_{t+1})}{\mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_t)} \right]. \tag{8.3}$$

For the sake of completeness, however, the cases where the agents receive the same reward are also investigated in this dissertation. In these cases, the combined gain achieved by each agent when changing its action is still calculated as a proportional increase in the sum of the $\mathcal{Q}$-values. Therefore, the combined gain in the case where both agents receive the negative reward based on density is

$$G'_{a^i_{t+1}} = \max_{a^i_{t+1}} \left[\frac{\mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_t) - \mathcal{Q}^i(s^i_{t+1}, a^i_{t+1}, a^j_t)}{\mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_t)} + \frac{\mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_t) - \mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_{t+1})}{\mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_t)} \right], \tag{8.4}$$

while the joint gain in the case where both agents receive the positive reward based on flow is

$$G''_{a^i_{t+1}} = \max_{a^i_{t+1}} \left[\frac{\mathcal{Q}^i(s^i_{t+1}, a^i_{t+1}, a^j_t) - \mathcal{Q}^i(s^i_{t+1}, a^i_t, a^j_t)}{\mathcal{Q}^i(s^i_{t+1}, a^i_{t+1}, a^j_t)} + \frac{\mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_{t+1}) - \mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_t)}{\mathcal{Q}^j(s^j_{t+1}, a^j_t, a^i_{t+1})} \right]. \tag{8.5}$$

In the latter two cases, the same expression is employed to determine the combined gain for both the RM and VSL agents.

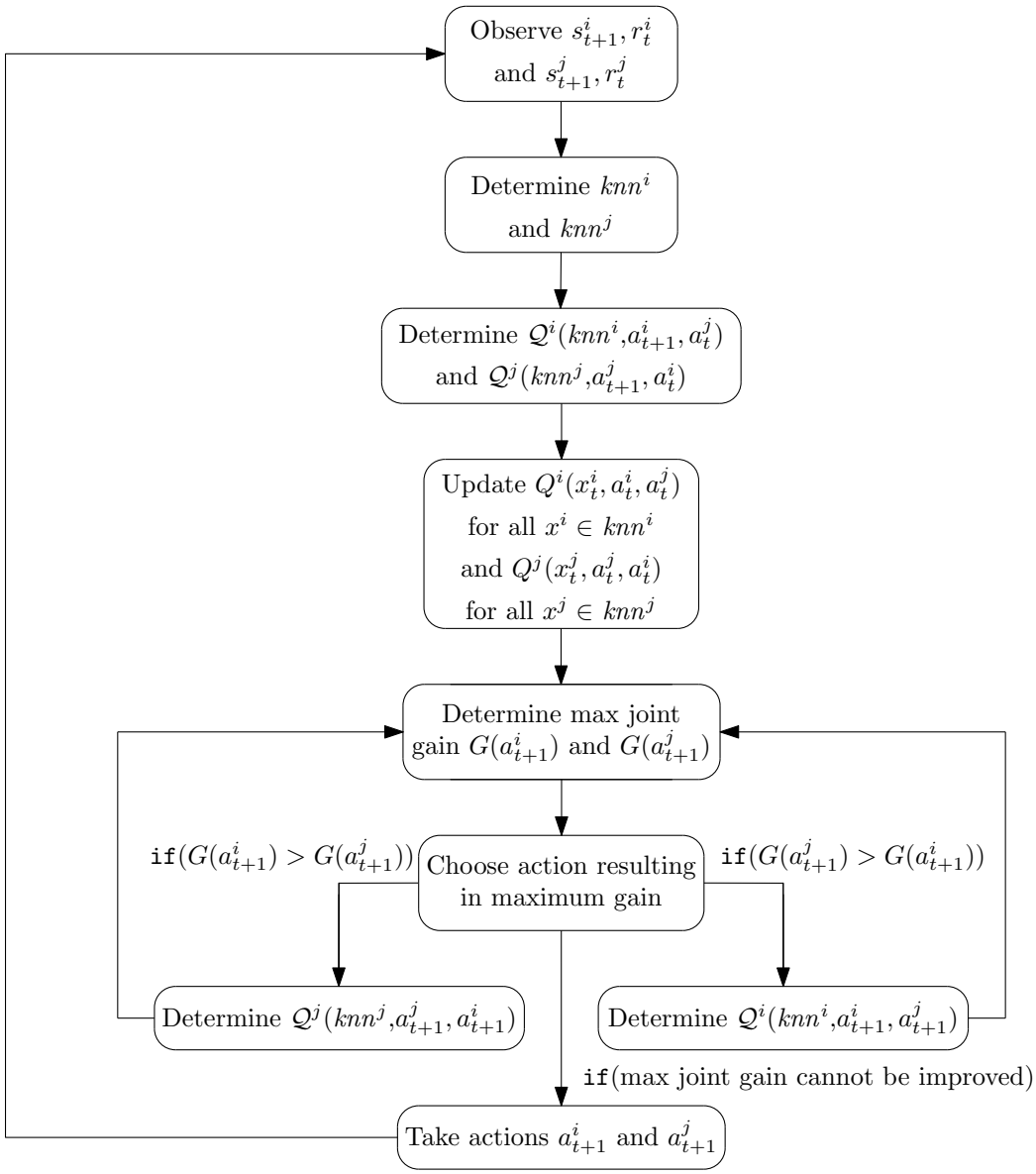


FIGURE 8.2: *The flow of information, as well as the sequence of events, during execution of the maximax MARL algorithm.*

Once the action a^i_{t+1} or a^j_{t+1} resulting in the maximum joint gain has been found, a new $\mathcal{Q}$-value is calculated for either agent j or agent i, respectively. Using the information about the neighbouring agent's action, a new maximum joint gain is determined. This process is repeated until no further improvements in the joint gain are achieved.

As was the case in all previous implementations, the trade-off between exploration and exploitation is taken into account through the use of the adaptive α and ϵ-value calculations, based on the number of visits to each state-action or centre-action pair. Due to the increase in the size of the centre-action space, when compared with that of the standard kNN-TD algorithm, it is expected that the algorithms will take longer to converge. In pursuit of good results, however, the parameters of the update rules in (6.5) and (6.6) remain unchanged for the maximax MARL implementation.

8.4 Computational Results

In this section, the performance of the kNN-TD learning algorithm for RM, as the algorithm which has thus far been able to achieve the greatest improvements in terms of the TTS across all four traffic scenarios of §5.3.2, is compared with the performances of each of the three MARL approaches implemented in the benchmark simulation model of §5.1.2. The kNN-TD RM implementation was chosen above the integrated feedback controller of Carlson *et al.* [24] due to the fact that the integrated controller was designed with a queueing consideration, while the on-ramp queue length is not considered in the MARL approaches evaluated in this section. This comparison is again performed for each of the four scenarios of varying traffic demand of §5.3.2. Initially, the performances of the MARL approaches, employing different combinations of the reward functions, are fine-tuned in §8.4.1. Thereafter, an algorithmic comparison is performed in §8.4.2, adopting the combination of reward functions found to yield the best performance for each of the MARL implementations.

8.4.1 Reward Function Evaluation

This section is devoted to determining which combination of reward functions yields the best performance when implemented in each of the MARL approaches described in §8.2. As was the case for the parameter evaluation completed for the RM implementation of Chapter 6, the evaluation in respect of the reward functions is performed in Scenario 2. Three different cases are investigated, as may be seen in Table 8.1. In the first of these cases, the reward function for the RM agent is as in (6.2) and the reward for the VSL agent is

$$r_t = 30q, \tag{8.6}$$

where q denotes the flow of vehicles out of the bottleneck location during the control interval. The change in the reward function for the VSL agent from the one in (7.3) is due to the fact that the VSL agent now chooses an action every two minutes (not every five minutes as in the original VSL implementation), as stated in §8.2, and the subsequent reward should still be measured in units of veh/h. In the second case, both agents are rewarded based on density, according to (6.2), while in the third case, both agents are rewarded based on the flow out of the bottleneck location according to (8.6).

TABLE 8.1: *Reward function evaluation results for MARL, measured in terms of the total time spent in the system (TTS) by the vehicles (in veh·h).*

	MARL Approach		
Reward	Independent	Hierarchical	Maximax
Case 1	877.32	840.55	1 022.60
Case 2	882.34	873.53	852.53
Case 3	1 092.90	1 115.77	962.94

As may be seen in Table 8.1, the case where the original reward functions are employed for both the RM and VSL agents consistently yields the best results in the independent and the hierarchical MARL implementations. For the maximax MARL implementation, however, the case where both agents are rewarded based on the density downstream of the on-ramp yields the best results. Therefore, for the comparisons conducted in the following section, the RM agent is rewarded according to (6.2), while the VSL agent is rewarded according to (8.6) in the

independent and hierarchical MARL implementations. In the maximax MARL implementation, however, both agents are rewarded based on the downstream density, according to (6.2).

The learning progression of the three different MARL approaches in Scenario 2 is shown in Figure 8.3. As may be seen in the figure, independent MARL exhibits the fastest learning rate, as the TTS-values decrease steadily until convergence begins at approximately the 180^{th} learning episode. This relatively fast learning speed is to be expected, as the state spaces of both the RM and VSL agents have not changed from the single agent implementations and are thus still relatively small. Due to the fact that the size of the state space of the VSL agent increases as described in §8.2.2, it may be expected that the initial learning speed exhibited by the hierarchical MARL agent will be slower. This is confirmed by the results presented in the figure, where it may be seen that the hierarchical MARL agent requires approximately 300 learning episodes until convergence of the TTS-value achieved. Finally, the maximax MARL agent exhibits the longest time until convergence is achieved, requiring approximately 400 learning episodes, as may be seen in Figure 8.3. This increase in learning rate over and above the increase exhibited by the hierarchical MARL agent was again to be expected, as the state spaces of both the RM and VSL agents have increased for the maximax MARL approach, as described in §8.2.3. Due to the fact, however, that all three MARL approaches achieve convergence within the learning time of 1 000 episodes, 1 000 training episodes were considered to be a sufficient training time for the agents before the algorithmic comparison of the following section was conducted.

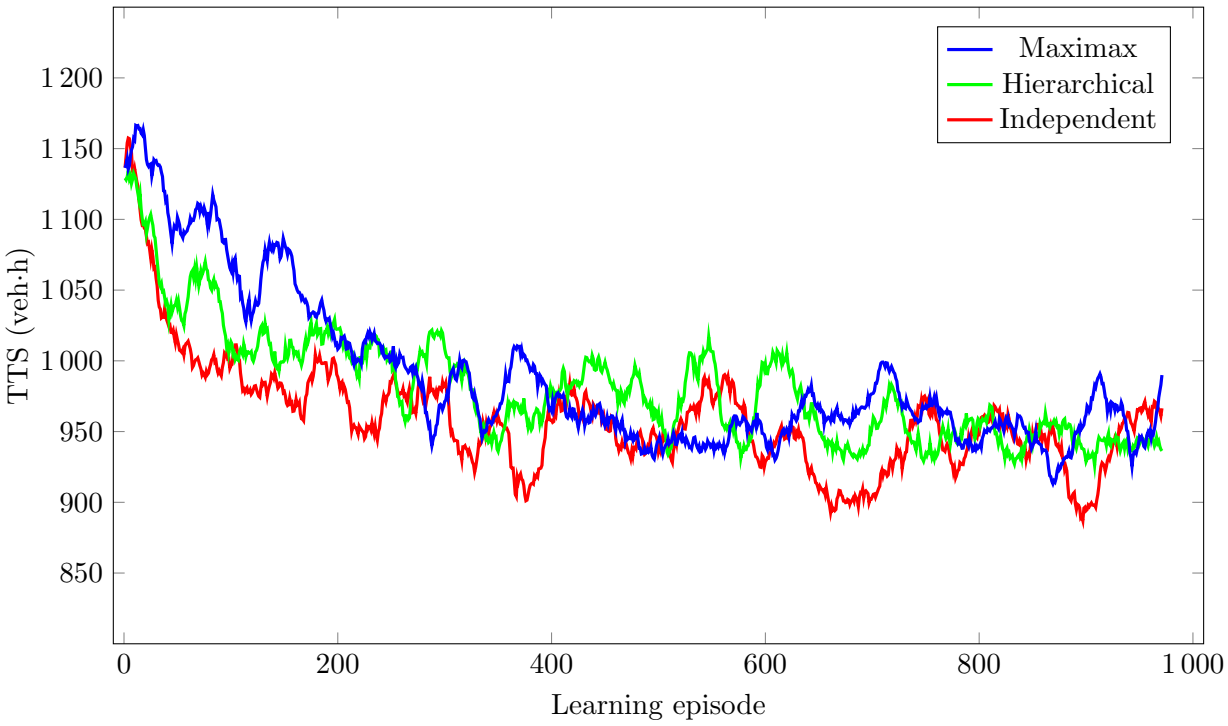


FIGURE 8.3: *The learning progression over the course of 1 000 training episodes for Scenario 2 of §5.3.2, shown for the various MARL approaches. In order to filter out some simulation noise, a moving average over 30 episodes is shown.*

8.4.2 Algorithmic Comparison

In this section, the simulation results and relative algorithmic performances are analysed for each of the MARL implementations of §8.2, which are compared with the kNN-TD RM implementation, the best-performing highway control measure identified thus far. These comparisons are conducted in each of the four scenarios of varying traffic demand introduced in §5.3.2. The results are presented and interpreted through the use of box plots in which the means, medians and interquartile ranges of the PMIs are indicated, as well as tables indicating whether or not statistical differences exist between the PMI-values for each pair of algorithms at a 5% level of significance.

Scenario 1

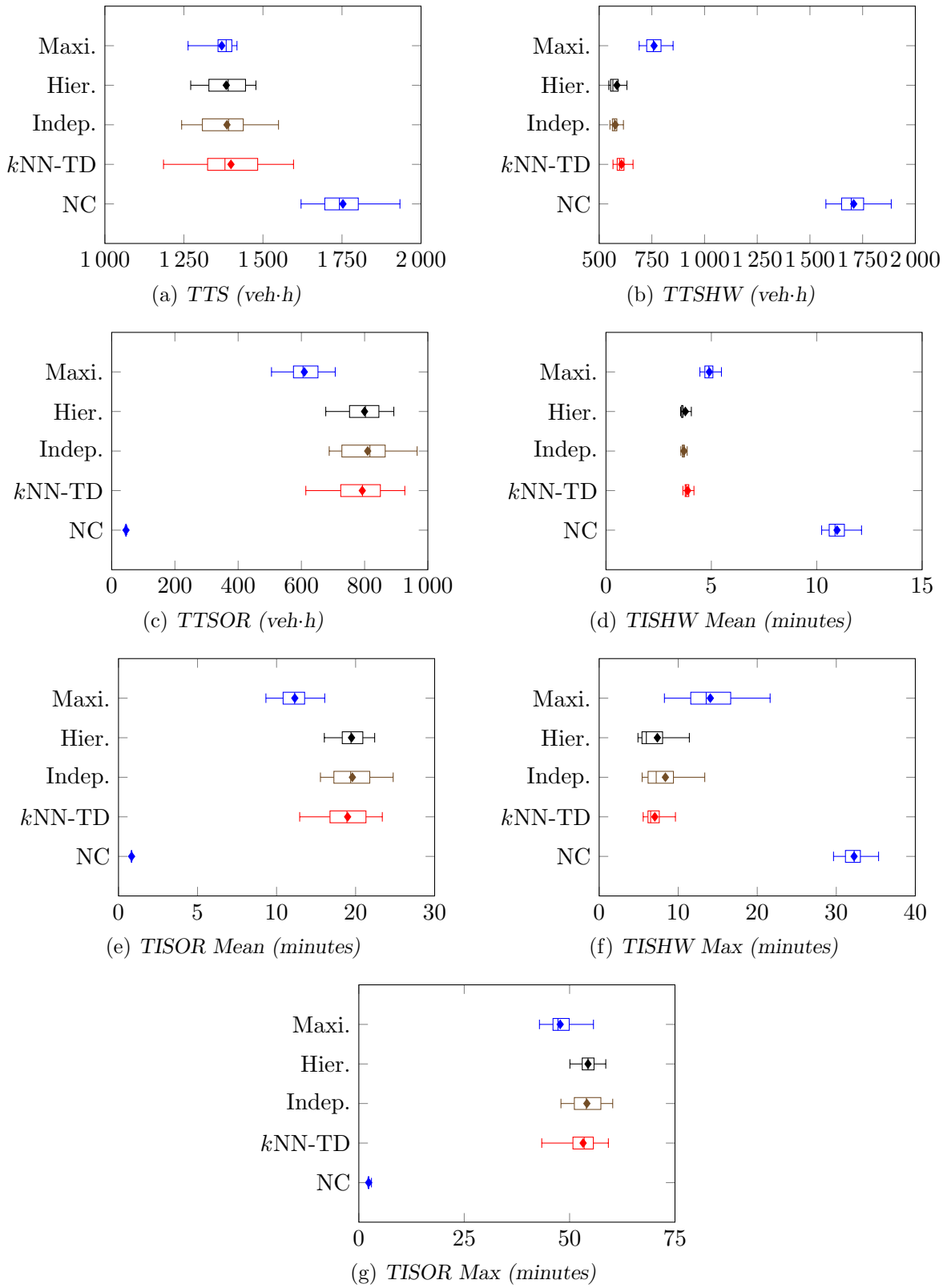
As may be seen from the p-values of the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms for Scenario 1, presented in Table 8.2, the ANOVA test revealed that there are, in fact, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all seven PMIs. Furthermore, Levene's test revealed that the variances between the PMI-values returned by at least some pair of algorithms also differ at a 5% level of significance for all seven PMIs. Therefore, the Games-Howell test was employed to determine between which pairs of algorithmic outputs differences occur in respect of all seven PMIs.

TABLE 8.2: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 1. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	kNN-TD	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	1 753.01	1 398.80	1 386.17	1 384.02	1 369.74	$< 1 \times 10^{-17}$	5.3869×10^{-5}
TTSHW	1 707.70	606.16	576.45	584.22	760.45	$< 1 \times 10^{-17}$	1.7827×10^{-8}
TTSOR	45.31	792.64	809.72	799.80	609.29	$< 1 \times 10^{-17}$	3.2085×10^{-14}
TISHW Mean	10.96	3.88	3.70	3.76	4.90	$< 1 \times 10^{-17}$	1.7987×10^{-8}
TISOR Mean	1.66	28.99	29.59	29.49	22.30	$< 1 \times 10^{-17}$	1.5432×10^{-14}
TISHW Max	32.25	7.04	8.38	7.38	14.07	$< 1 \times 10^{-17}$	3.3276×10^{-5}
TISOR Max	2.34	53.21	54.10	54.37	47.77	$< 1 \times 10^{-17}$	2.5828×10^{-11}

As may be seen in Figure 8.4(a), all the MARL implementations were able to achieve statistically significant improvements over the no-control case in respect of the TTS. This is corroborated by the p-values presented in Table 8.3. As may be seen in the table, all of the MARL implementations were able to achieve further improvements over the kNN-TD RM implementation as the independent MARL, hierarchical MARL and the maximax MARL implementations achieved reductions in the TTS of 20.93%, 21.05% and 21.86%, respectively, over the no-control case compared with the improvement of 20.21% achieved by the kNN-TD RM implementation. All of the MARL implementations were, however, found to perform statistically indistinguishably from one another and the kNN-TD RM implementation at a 5% level of significance, as may be seen in Table 8.3.

Because of the RM component of the MARL implementations, the savings in terms of travel times were achieved on the highway due to protection from over-utilisation of the highway provided by RM, as expected. This trend is clearly visible in the box plots of Figure 8.4(b), as all algorithms achieved significant improvements over the no-control case in respect of the



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 8.4: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) in Scenario 1.*

TABLE 8.3: *Differences in respect of the total time spent in the system by all vehicles (TTS) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	7.9768×10^{-12}	1.3710×10^{-11}	$< 1 \times 10^{-17}$	8.8118×10^{-13}
kNN-TD		—	9.8635×10^{-1}	9.6013×10^{-1}	6.0775×10^{-1}
Independent			—	9.9997×10^{-1}	8.9977×10^{-1}
Hierarchical				—	8.6794×10^{-1}
Maximax					—
Mean	1 753.01	1 398.80	1 386.17	1 384.02	1 369.74

TABLE 8.4: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	5.1992×10^{-13}	1.0447×10^{-13}	$< 1 \times 10^{-17}$	4.0890×10^{-13}
kNN-TD		—	3.4936×10^{-3}	3.3805×10^{-1}	7.1310×10^{-13}
Independent			—	9.3792×10^{-1}	3.4694×10^{-13}
Hierarchical				—	1.1676×10^{-11}
Maximax					—
Mean	1 707.70	606.16	576.45	584.22	760.45

TABLE 8.5: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.3319×10^{-15}	5.6000×10^{-16}	5.6000×10^{-16}	1.0000×10^{-15}
kNN-TD		—	9.3224×10^{-1}	9.9519×10^{-1}	$< 1 \times 10^{-17}$
Independent			—	9.8527×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	5.4782×10^{-12}
Maximax					—
Mean	45.31	792.64	809.72	799.80	609.29

TABLE 8.6: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	5.6266×10^{-13}	9.4920×10^{-14}	$< 1 \times 10^{-17}$	9.8140×10^{-14}
kNN-TD		—	4.0981×10^{-3}	5.0932×10^{-1}	6.2714×10^{-12}
Independent			—	8.3143×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	1.2227×10^{-11}
Maximax					—
Mean	10.96	3.88	3.70	3.76	4.90

TABLE 8.7: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.9980×10^{-15}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
kNN-TD		—	9.2092×10^{-1}	9.2286×10^{-1}	$< 1 \times 10^{-17}$
Independent			—	9.9979×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	1.4297×10^{-11}
Maximax					—
Mean	1.66	28.99	29.59	29.49	22.30

TABLE 8.8: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.3021×10^{-11}	1.0354×10^{-12}	8.6675×10^{-13}	2.0661×10^{-13}
kNN-TD		—	2.1622×10^{-1}	9.8365×10^{-1}	2.1151×10^{-11}
Independent			—	7.3629×10^{-1}	1.2235×10^{-7}
Hierarchical				—	2.0689×10^{-9}
Maximax					—
Mean	32.25	7.04	8.38	7.38	14.07

TABLE 8.9: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	7.7719×10^{-15}	7.7700×10^{-15}	8.4400×10^{-15}	$< 1 \times 10^{-17}$
kNN-TD		—	8.6685×10^{-1}	5.6643×10^{-1}	4.0121×10^{-7}
Independent			—	9.9642×10^{-1}	3.9099×10^{-9}
Hierarchical				—	3.4939×10^{-13}
Maximax					—
Mean	2.34	53.21	54.10	54.37	44.77

TTSHW. The best performances in respect of the TTSHW were achieved by independent MARL and hierarchical MARL, which performed statistically indistinguishably from one another at a 5% level of significance — achieving improvements of 66.24% and 65.79%, respectively, over the no-control case. Independent MARL was also able to outperform the kNN-TD learning algorithm at a 5% level of significance, as kNN-TD managed a 64.50% improvement over the no-control case. Interestingly, although the maximax MARL implementation achieved the smallest TTS, it was outperformed by all other algorithms in respect of the TTSHW, as may be seen from the p-values in Table 8.4. The maximax MARL implementation did, however, outperform the no-control case at a 5% level of significance, achieving a reduction of 55.47% in respect of the TTSHW.

As may be seen in Figure 8.4(c), the reductions in respect of travel times on the highway are offset by increases in respect of travel times for vehicles joining the highway from the on-ramp.

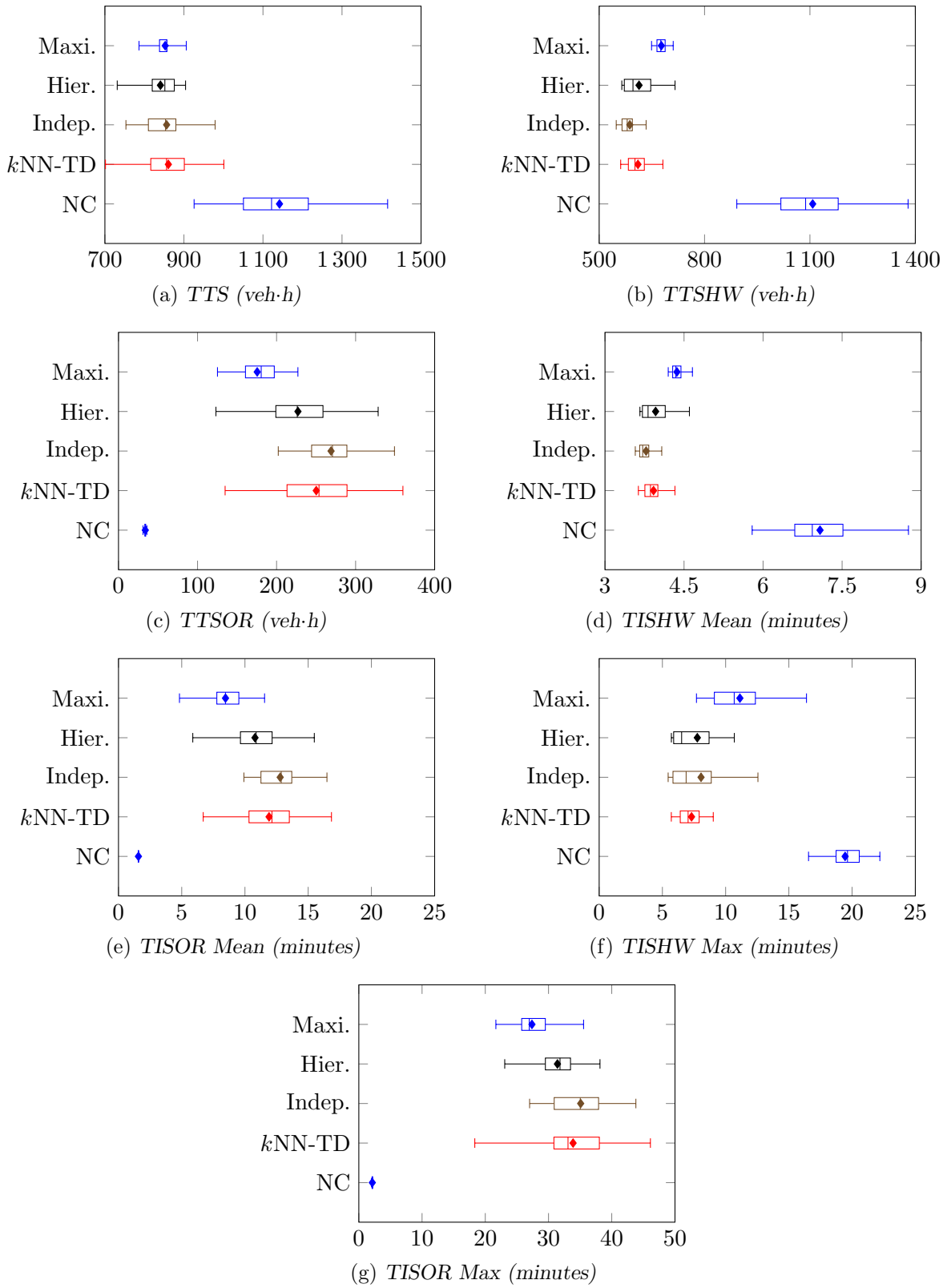
Naturally, the no-control case achieved the smallest travel times for vehicles joining the highway from the on-ramp, since it is the only case in which RM is not applied. Taking the natural increase in travel times for vehicles entering the network from the on-ramp due to RM into account, it is the maximax MARL implementation that achieved the smallest TTSOR-value of 609.29 veh·h, outperforming kNN-TD RM, independent MARL and hierarchical MARL at a 5% level of significance, while these algorithms achieved TTSOR-values of 792.64 veh·h, 809.72 veh·h and 799.80 veh·h, respectively. As may be seen from the p-values in Table 8.5, kNN-TD RM, independent MARL and hierarchical MARL returned results that are statistically indistinguishable at a 5% level of significance.

The trends in respect of the mean and maximum travel times for vehicles travelling along the highway only are very similar to that observed for the TTSHW, as is evident in Figures 8.4(d) and 8.4(f), respectively. Once again, independent MARL returned the best performance in respect of the mean TISHW, outperforming kNN-TD RM, maximax MARL and the no-control case at a 5% level of significance, as may be seen from the p-values in Table 8.6. Independent MARL is followed in the order of relative algorithmic performances by hierarchical MARL and kNN-TD RM, which were found to perform statistically similarly in respect of the mean TISHW, as may be seen in Table 8.6. Maximax MARL achieved the largest mean TISHW value, as it was outperformed by all three other algorithmic implementations at a 5% level of significance. The ordering of relative algorithmic performances in respect of the maximum TISHW is similar to that in respect of the mean TISHW, except that kNN-TD RM, independent MARL and hierarchical MARL were all found to perform statistically indistinguishably at a 5% level of significance, as may be seen in Table 8.8. Maximax MARL was again outperformed by kNN-TD, independent MARL and hierarchical MARL, in respect of the maximum TISHW, but it outperformed the no-control case at a 5% level of significance in respect of both of the mean and maximum TISHW PMIs, achieving improvements of 55.29% and 56.37%, respectively.

As in the case of the TTSOR, increases were again to be expected in respect of both the mean and maximum travel times for vehicles joining the highway from the on-ramp. This trend is clearly visible in the box plots of Figures 8.4(e) and 8.4(g). Similarly to what was observed for the TTSOR, the maximax MARL implementation outperformed kNN-TD RM, independent MARL and hierarchical MARL at a 5% level of significance in respect of both these performance measures, as may be seen from the p-values presented in Tables 8.7 and 8.9. For the maximax MARL implementation the mean TISOR was 22.30 minutes, while vehicles required 28.99 minutes, 29.59 minutes and 29.49 minutes in the kNN-TD RM, independent MARL and hierarchical MARL implementations, respectively. Furthermore, maximax MARL was able to limit the maximum TISOR to 44.77 minutes, while this value increased to 53.21 minutes for kNN-TD, 54.10 minutes for independent MARL and 54.37 minutes for hierarchical MARL.

Scenario 2

As in Scenario 1, the p-values returned by the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms in Scenario 2, presented in Table 8.10, revealed that there are, again, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all seven PMIs. Furthermore, Levene's test revealed that the variances returned by at least some pair of algorithms are again statistically different at a 5% level of significance for all seven PMIs. Hence the Games-Howell test was performed to ascertain between which pairs of algorithms the differences between the algorithmic output data occur in respect of all seven PMIs.



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 8.5: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) in Scenario 2.*

TABLE 8.10: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 2. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					*p*-value	
PMI	No Control	*k*NN-TD	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	1 141.79	860.61	877.32	840.55	852.53	$< 1 \times 10^{-17}$	1.8335×10^{-9}
TTSHW	1 107.88	610.40	612.11	613.49	677.02	$< 1 \times 10^{-17}$	1.5711×10^{-12}
TTSOR	33.92	250.21	265.22	227.05	175.50	$< 1 \times 10^{-17}$	1.2843×10^{-12}
TISHW Mean	7.08	3.92	3.94	3.96	4.37	$< 1 \times 10^{-17}$	4.3681×10^{-12}
TISOR Mean	1.58	11.91	12.50	10.79	8.42	$< 1 \times 10^{-17}$	1.3850×10^{-12}
TISHW Max	19.45	7.31	7.83	7.76	11.13	$< 1 \times 10^{-17}$	6.5087×10^{-3}
TISOR Max	2.13	33.89	34.35	31.41	27.38	$< 1 \times 10^{-17}$	3.0768×10^{-10}

All three MARL implementations were again able to achieve significant improvements over the no-control case in respect of the TTS, as may be seen in Figure 8.5(a). Hierarchical MARL achieved the best performance, returning a value of 840.55 veh·h, and was followed by maximax MARL, which returned a TTS-value of 852.53 veh·h. Maximax MARL was followed by *k*NN-TD RM, which returned a TTS-value of 860.61 veh·h, while independent MARL was the worst performing algorithm with a TTS-value of 877.32 veh·h. As may be seen from the *p*-values in Table 8.11, hierarchical MARL outperformed independent MARL at a 5% level of significance, while all other algorithms were, however, found to yield statistically indistinguishable results at a 5% level of significance.

In a trend similar to that for Scenario 1, *k*NN-TD RM, independent MARL and hierarchical MARL were able to achieve the greatest reduction in travel time for vehicles travelling along the highway only over the no-control case, with neither of these algorithms outperforming one another. All three of these algorithms were, however, able to outperform maximax MARL at a 5% level of significance in respect of the TTSHW, as may be seen in Table 8.12. This trend is clearly visible in the box plots of Figure 8.5(b). Interestingly, the largest reduction of 44.90% was achieved by the single *k*NN-TD RM agent. This algorithm was followed by independent MARL and hierarchical MARL with reductions in the TTSHW of 44.75% and 44.62%, respectively. Finally, maximax MARL outperformed the no-control case at a 5% level of significance, achieving a reduction in the TTSHW of 38.89%.

Interestingly, in respect of the TTSOR, *k*NN-TD RM was found to perform statistically indistinguishably from both independent MARL and hierarchical MARL at a 5% level of significance, while the performances of all other implementations were found to differ statistically from one another at a 5% level of significance. Taking the natural increase in travel times for vehicles joining the highway from the on-ramp into account, maximax MARL yielded the best performance, achieving a TTSOR-value of 175.50 veh·h and thereby outperforming all other algorithms, as may be seen in Table 8.13. Maximax MARL was followed by hierarchical MARL, which was able to outperform independent MARL, achieving a TTSOR-value of 227.05 veh·h. Finally, independent MARL and *k*NN-TD complete the order of relative algorithmic performance as they returned statistically indistinguishable TTSOR-values of 265.22 veh·h and 250.21 veh·h, respectively, at a 5% level of significance. This ordering is clearly visible in the box plots of Figure 8.5(c).

From the box plots in Figures 8.5(d) and 8.5(f), it is clear that *k*NN-TD RM, independent MARL and hierarchical MARL were, again, able to achieve the largest reductions in respect of both the mean and maximum TISHW PMIs. This is confirmed by the *p*-values presented in Tables 8.14 and 8.16. As may be seen in the tables, *k*NN-TD RM, independent MARL and hierarchical

TABLE 8.11: *Differences in respect of the total time spent in the system by all vehicles (TTS) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	1.1083×10^{-12}	$< 1 \times 10^{-17}$	1.1258×10^{-13}
kNN-TD		—	8.0320×10^{-1}	6.9232×10^{-1}	9.8133×10^{-1}
Independent			—	1.1018×10^{-2}	9.8432×10^{-2}
Hierarchical				—	7.6909×10^{-1}
Maximax					—
Mean	1 141.79	860.61	877.32	840.55	852.53

TABLE 8.12: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	1.0515×10^{-12}	$< 1 \times 10^{-17}$	6.3620×10^{14}
kNN-TD		—	9.9994×10^{-1}	9.9879×10^{-1}	2.6189×10^{-9}
Independent			—	9.9998×10^{-1}	2.5490×10^{-5}
Hierarchical				—	6.4527×10^{-7}
Maximax					—
Mean	1 107.88	610.40	612.11	613.49	677.02

TABLE 8.13: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	3.3309×10^{-15}	6.1100×10^{-15}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
kNN-TD		—	8.2282×10^{-1}	3.5660×10^{-1}	2.0941×10^{-7}
Independent			—	4.3253×10^{-2}	1.7171×10^{-8}
Hierarchical				—	4.8001×10^{-5}
Maximax					—
Mean	33.92	250.21	265.22	227.05	175.50

TABLE 8.14: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	9.1040×10^{-14}	3.1863×10^{-13}	5.5840×10^{-14}
kNN-TD		—	9.9947×10^{-1}	9.8041×10^{-1}	1.0307×10^{-10}
Independent			—	9.9930×10^{-1}	1.5601×10^{-5}
Hierarchical				—	4.9604×10^{-7}
Maximax					—
Mean	7.08	3.92	3.94	3.96	4.37

TABLE 8.15: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test: TISOR Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	2.0000×10^{-15}	1.5500×10^{-15}	$< 1 \times 10^{-17}$
kNN-TD		—	8.8824×10^{-1}	3.0322×10^{-1}	1.5845×10^{-7}
Independent			—	5.2857×10^{-2}	3.3028×10^{-8}
Hierarchical				—	5.0813×10^{-5}
Maximax					—
Mean	1.58	11.91	12.50	10.79	8.42

TABLE 8.16: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
kNN-TD		—	8.9919×10^{-1}	9.2471×10^{-1}	5.0280×10^{-8}
Independent			—	9.9998×10^{-1}	2.0794×10^{-4}
Hierarchical				—	9.2991×10^{-4}
Maximax					—
Mean	19.50	7.31	7.83	7.76	11.13

TABLE 8.17: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	8.9000×10^{-16}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
kNN-TD		—	9.9608×10^{-1}	2.7989×10^{-2}	5.2863×10^{-5}
Independent			—	2.0171×10^{-2}	2.5247×10^{-8}
Hierarchical				—	7.9231×10^{-4}
Maximax					—
Mean	2.13	33.89	34.35	31.41	27.38

MARL outperformed maximax MARL at a 5% level of significance, as the algorithms achieved reductions in the mean TISHW of 44.63%, 44.35% and 44.07%, respectively, over the no-control case, compared with the 38.28% improvement returned by maximax MARL. Similarly, kNN-TD RM, independent MARL and hierarchical MARL achieved reductions of 62.51%, 59.85% and 60.21%, respectively, over the no-control case in respect of the maximum TISHW, outperforming maximax MARL which achieved an improvement of 42.92%.

Maximax MARL again exhibited the best performance in respect of the mean TISOR, as may be seen in Table 8.15. Independent MARL, hierarchical MARL and kNN-TD RM, on the other hand, were all found to perform statistically similarly at a 5% level of significance in respect of the mean TISOR. This trend is clearly visible in the box plots of Figure 8.5(e). When considering the maximum travel time for vehicles joining the highway from the on-ramp, maximax MARL again outperformed all other algorithms at a 5% level of significance. Maximax MARL was followed

by hierarchical MARL, which outperformed both kNN-TD RM and independent MARL, while the latter two were found to perform statistically indistinguishable at a 5% level of significance, as may be seen in Table 8.17. This trend is also evident in Figure 8.5(g).

Scenario 3

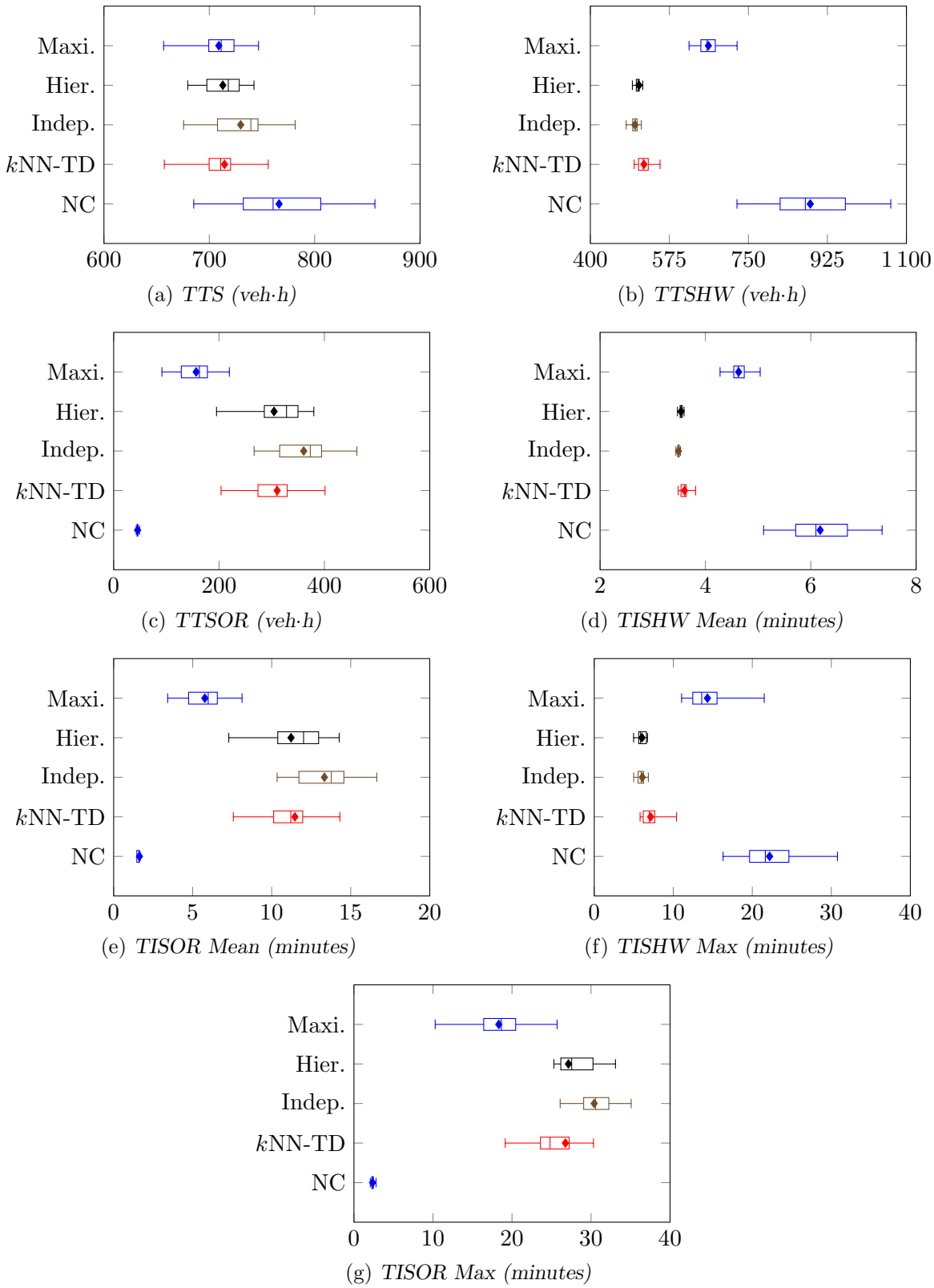
As in Scenarios 1 and 2, an ANOVA test revealed that there are, again, statistical differences at a 5% level of significance in the case of Scenario 3 between the means returned by at least some pair of algorithms in respect of all seven PMIs, as may be seen from the p-values presented in Table 8.18. The Levene test revealed that the variances returned by at least some pair of algorithms are statistically different at a 5% level of significance for all seven PMIs. Therefore, the Games-Howell *post hoc* test was employed in order to determine between which pairs of algorithms the differences between the algorithmic output occur in respect of all seven PMIs.

TABLE 8.18: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 3. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	kNN-TD	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	932.46	829.02	859.57	825.65	818.06	3.8231×10^{-12}	2.7102×10^{-5}
TTSHW	887.07	518.66	498.78	521.22	661.42	$< 1 \times 10^{-17}$	3.9968×10^{-15}
TTSOR	45.40	310.36	360.79	304.44	156.64	$< 1 \times 10^{-17}$	2.6338×10^{-7}
TISHW Mean	6.18	3.60	3.49	3.65	4.63	$< 1 \times 10^{-17}$	2.2204×10^{-16}
TISOR Mean	1.63	11.47	13.33	11.22	5.77	$< 1 \times 10^{-17}$	3.4049×10^{-6}
TISHW Max	22.19	7.14	6.08	6.89	14.33	$< 1 \times 10^{-17}$	4.7492×10^{-8}
TISOR Max	2.37	26.76	30.40	27.14	18.33	$< 1 \times 10^{-17}$	5.6712×10^{-3}

In Scenario 3, all of the algorithms were again able to achieve significant improvements over the no-control case in respect of the TTS at a 5% level of significance, as may be seen in Table 8.19. Maximax MARL achieved the smallest TTS-value of 818.06 veh·h, followed by hierarchical MARL with 825.65 veh·h and kNN-TD RM with 829.02 veh·h. Maximax MARL was able to outperform independent MARL, which achieved a TTS-value of 859.57 veh·h, at a 5% level of significance, while its performance was found to be statistically indistinguishable from those of kNN-TD RM and hierarchical MARL. Finally, kNN-TD RM, independent MARL and hierarchical MARL did not perform statistically differently at a 5% level of significance, as may be seen in the table. These results are summarised in the box plots of Figure 8.6(a).

As may be seen in Table 8.20, there exist statistical differences at a 5% level of significance between the performances of all algorithms, except independent MARL and hierarchical MARL, in respect of the TTSHW-values, while hierachical MARL and kNN-TD RM were also found to perform statistically indistinguishably from one another at a 5% level of significance. Independent MARL returned the best performance, outperforming maximax MARL and kNN-TD RM — achieving an improvement of 43.77% over the no-control case. Independent MARL is followed in the order of relative algorithmic performances by kNN-TD RM and hierarchical MARL, both of which outperformed maximax MARL as they achieved improvements of 45.53% and 41.24%, respectively over the no-control case. Finally, maximax MARL was able to outperform only the no-control case at a 5% level of significance, as it achieved an improvement of 24.44% in respect of the TTSHW. This ordering of relative algorithmic performances is clearly visible in the box plots of Figure 8.6(b).



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 8.6: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) Scenario 3.*

As may have been expected, the ordering of the relative performances in respect of the TTSOR is exactly opposite to that in respect of the TTSHW, as may be seen in Figure 8.6(c). The performances of all the algorithms, except kNN-TD RM and hierarchical MARL, were found to be statistically different at a 5% level of significance in respect of the TTSOR, as may be seen in Table 8.21. Naturally, the no-control case achieved the smallest TTSOR-value of 45.40 veh·h, followed by maximax MARL, which achieved a TTSOR-value of 156.64 veh·h. Maximax MARL was followed by hierarchical MARL and kNN-TD RM, which returned values of 304.44 veh·h and 310.36 veh·h, respectively. Finally, hierarchical MARL and kNN-TD RM outperformed independent MARL, which achieved a TTSOR-value of 360.79 veh·h.

As was the case for the TTSHW, all the algorithms, except independent MARL and hierarchical MARL, as well as hierarchical MARL and kNN-TD RM performed statistically differently at a 5% level of significance in respect of both the mean and maximum TISHW, as may be seen in Tables 8.22 and 8.24. From the box plots in Figures 8.6(d) and 8.6(f), it is evident that the ordering of the relative algorithmic performances in respect of these two PMIs is also the same as it was in respect of the TTSHW. Independent MARL outperformed both maximax MARL and kNN-TD RM in respect of the mean TISHW, as it achieved a reduction of 43.53% over the no-control case. Similarly, independent MARL yielded the largest reduction over the no-control case of 72.60% in respect of the maximum TISHW. Independent MARL was followed by hierarchical MARL and kNN-TD RM, which were able to achieve reductions of 40.94% and 41.75%, respectively, over the no-control case in respect of the mean TISH, while these implementations achieved reductions of 68.95% and 67.82%, respectively, in respect of the maximum TISHW. Finally, maximax MARL completes the order of relative algorithmic performances with reductions of 17.15% and 20.23%, respectively, in respect of the mean and maximum TISHW over the no-control case.

As for the TTSOR, the ordering of relative algorithmic performances in respect of both the mean and maximum time spent in the system by vehicles joining the highway from the on-ramp is exactly opposite to what it was in the case of the mean and maximum travel times for vehicles travelling along the highway only, as may be seen in Figures 8.6(e) and 8.6(g). The maximax MARL implementation achieved the smallest mean and maximum TISOR-values of all algorithmic implementations, returning values of 5.77 minutes and 18.33 minutes, respectively, and outperforming all other algorithms at a 5% level of significance, as may be seen in Tables 8.23 and 8.25. Hierarchical MARL and kNN-TD RM achieved the second place in the ordering of relative algorithmic performances as they achieved mean TISOR-values of 11.22 minutes and 11.47 minutes, respectively, while returning maximum TISHW-values of 27.14 minutes and 26.76 minutes, respectively, thereby outperforming independent MARL. Finally, independent MARL concludes the order of relative algorithmic performances as it returned values of 13.33 minutes and 30.40 minutes, respectively, in respect of the mean and maximum TISOR PMIs.

TABLE 8.19: *Differences in respect of the total time spent in the system by all vehicles (TTS) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	3.1995×10^{-5}	4.4285×10^{-3}	6.1705×10^{-6}	2.2208×10^{-6}
kNN-TD		—	2.4979×10^{-1}	9.9883×10^{-1}	9.2643×10^{-1}
Independent			—	5.9084×10^{-2}	2.2412×10^{-2}
Hierarchical				—	9.5371×10^{-1}
Maximax					—
Mean	932.46	829.02	859.57	825.65	818.06

TABLE 8.20: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	6.9611×10^{-14}	1.0550×10^{-14}	$< 1 \times 10^{-17}$	2.2038×10^{-13}
kNN-TD		—	1.0542×10^{-6}	9.9908×10^{-1}	$< 1 \times 10^{-17}$
Independent			—	1.8846×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	5.3960×10^{-14}
Maximax					—
Mean	887.07	518.66	498.78	521.22	661.42

TABLE 8.21: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	2.5500×10^{-15}	2.3300×10^{-15}	5.8800×10^{-15}
kNN-TD		—	7.6016×10^{-14}	9.9649×10^{-1}	$< 1 \times 10^{-17}$
Independent			—	9.1933×10^{-3}	$< 1 \times 10^{-17}$
Hierarchical				—	6.7810×10^{-12}
Maximax					—
Mean	45.40	310.36	360.79	304.44	156.64

TABLE 8.22: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	4.3632×10^{-14}	5.2200×10^{-15}	$< 1 \times 10^{-17}$	8.8150×10^{-14}
kNN-TD		—	6.7558×10^{-8}	9.6176×10^{-1}	1.1272×10^{-11}
Independent			—	1.4369×10^{-1}	4.6410×10^{-14}
Hierarchical				—	3.1097×10^{-13}
Maximax					—
Mean	6.18	3.60	3.49	3.65	4.63

TABLE 8.23: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	4.6600×10^{-15}	3.4400×10^{-15}	$< 1 \times 10^{-17}$
kNN-TD		—	1.8422×10^{-2}	9.9617×10^{-1}	2.4052×10^{-12}
Independent			—	5.2138×10^{-3}	$< 1 \times 10^{-17}$
Hierarchical				—	5.0748×10^{-12}
Maximax					—
Mean	1.63	11.47	13.33	11.22	5.77

TABLE 8.24: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	1.1224×10^{-13}	2.1058×10^{-12}	2.9532×10^{-12}
kNN-TD		—	1.6756×10^{-3}	9.9215×10^{-1}	3.3035×10^{-13}
Independent			—	5.7681×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	1.3741×10^{-11}
Maximax					—
Mean	22.19	7.14	6.08	6.89	14.33

TABLE 8.25: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	4.7700×10^{-15}	6.8800×10^{-15}	$< 1 \times 10^{-17}$
kNN-TD		—	1.6187×10^{-1}	9.9946×10^{-1}	5.8735×10^{-5}
Independent			—	2.7144×10^{-2}	3.6006×10^{-12}
Hierarchical				—	2.4620×10^{-9}
Maximax					—
Mean	2.37	26.76	30.40	27.14	18.33

Scenario 4

Interestingly, the ANOVA test performed on the PMI-values returned by the algorithms in Scenario 4 revealed that the means returned by the algorithms in respect of the TTS are statistically indistinguishable at a 5% level of significance, as may be seen from the p-values presented in Table 8.26. For the other six PMIs, however, the p-values returned by the ANOVA test revealed that there are, in fact, statistical differences between at least some pair of algorithms at a 5% level of significance. Furthermore, the Levene test revealed that the variances returned by at least some pair of algorithms are statistically different at a 5% level of significance in respect of all seven PMIs. Therefore, the Games-Howell test was employed in respect of the six PMIs for which the ANOVA indicated that statistical differences exist between at least some pair of algorithmic outputs so as to determine between which pairs of algorithms these differences occur.

As may have been expected, the MARL implementations were least effective in Scenario 4 due to the low traffic demand. This expectation is confirmed in the box plots of Figure 8.7(a), as all the means in the box plots lie relatively close to one another. This is corroborated by the p-values in Table 8.26, revealing that the performances of all algorithms are statistically indistinguishable at a 5% level of significance. From the p-values returned by the Levene test, as conducted for Scenario 4, however, it may be seen that the variances of the algorithms' output data are statistically different for at least some pair of algorithms. This difference in the variances is clearly visible in the figure, as the interquartile ranges of the box plots in respect of the TTS corresponding to the hierarchical MARL and maximax MARL are significantly smaller than those corresponding to the other algorithms. This implies that, although the algorithms

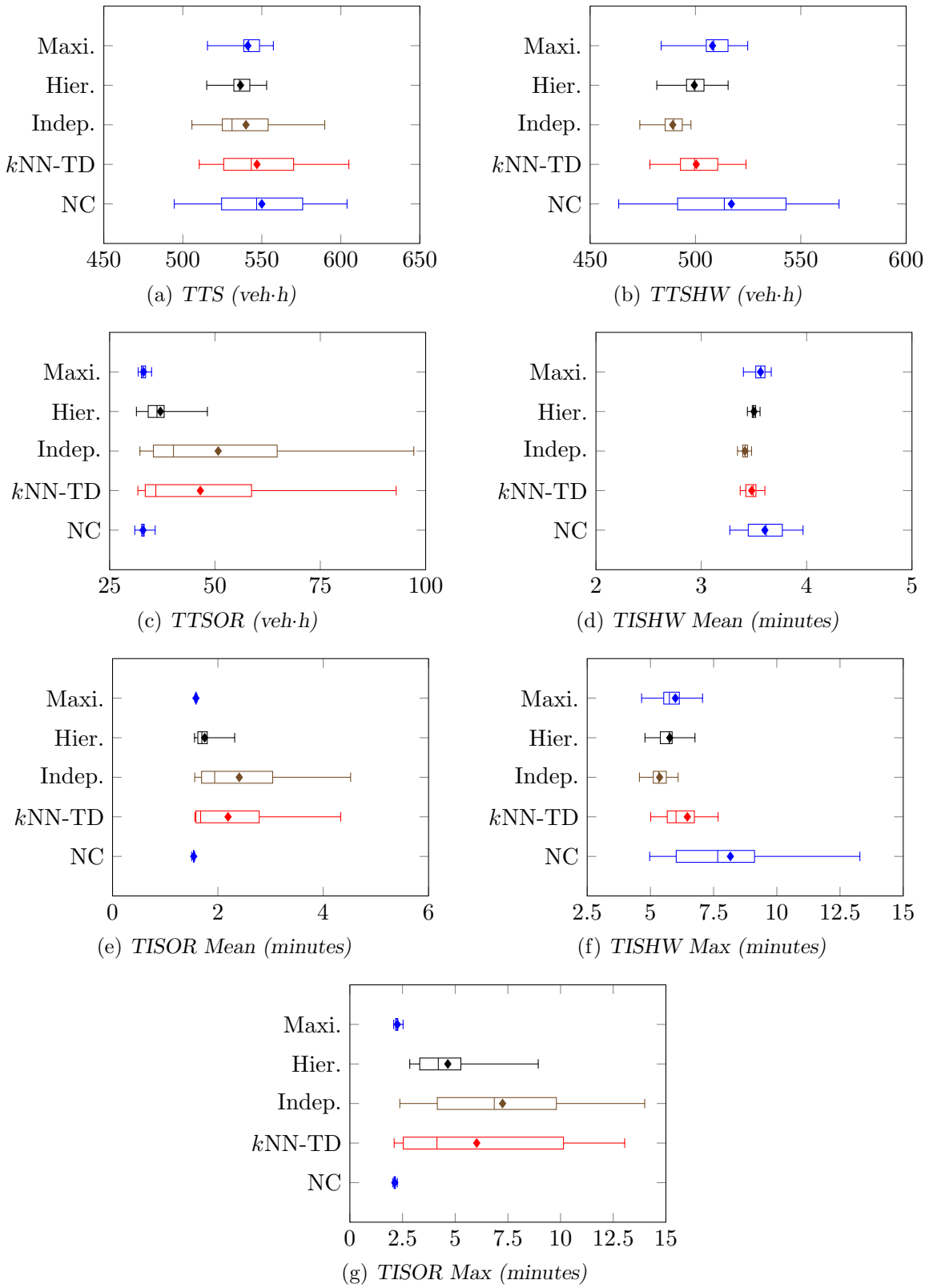


FIGURE 8.7: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) in Scenario 4.*

TABLE 8.26: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 4. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	kNN-TD	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	550.00	546.93	539.98	536.52	541.14	1.0136×10^{-1}	3.1752×10^{-14}
TTSHW	517.07	500.40	489.21	499.46	508.08	7.0896×10^{-9}	$< 1 \times 10^{-17}$
TTSOR	32.93	46.53	50.77	37.05	33.06	6.0726×10^{-9}	$< 1 \times 10^{-17}$
TISHW Mean	3.48	3.52	3.41	3.50	3.56	1.2474×10^{-11}	$< 1 \times 10^{-17}$
TISOR Mean	1.54	2.19	2.41	1.75	1.58	1.3917×10^{-9}	$< 1 \times 10^{-17}$
TISHW Max	8.16	6.46	5.36	5.77	5.99	4.6406×10^{-11}	3.6006×10^{-11}
TISOR Max	2.13	6.02	7.25	4.65	2.25	6.6613×10^{-16}	$< 1 \times 10^{-17}$

were unable to achieve significant improvements in respect of the TTS, the traffic flow was more stable in the situation where hierarchical MARL or maximax MARL was employed.

Interestingly, although no statistical differences could be identified between any of the algorithmic output sets in respect of the TTS at a 5% level of significance, this is not the case in respect of the TTSHW. As may be seen in Table 8.27, independent MARL and hierarchical MARL were able to outperform not only the no-control case, but also maximax MARL at a 5% level of significance; these algorithms achieved TTSHW-values of 489.21 veh·h and 499.46 veh·h, respectively. Furthermore, independent MARL was also able to outperform kNN-TD RM, while kNN-TD RM and hierarchical MARL were found to be statistically indistinguishable at a 5% level of significance. As may be seen in Figure 8.7(b), these improvements in respect of the TTSHW are not only down to an absolute reduction in the TTSHW, but also due to lower variances from the minimum TTSHW. These differences in the variances of the algorithms' output were confirmed by the Levene test at a 5% level of significance, as may be seen in Table 8.26. Finally, kNN-TD, maximax MARL and the no-control case were found to perform statistically indistinguishable at a 5% level of significance, although the maximax MARL implementation returned a reduced variance compared with the no-control case and kNN-TD RM; the algorithms achieved mean TTSHW-values of 500.40 veh·h, 508.08 veh·h and 517.07 veh·h, respectively.

As may have been expected, the improvements in terms of the TTSHW achieved by independent MARL and hierarchical MARL are offset by a deterioration in respect of the TTSOR. A similar deterioration due to an increased variance is observed for kNN-TD RM. This deterioration is clearly visible in the box plots of Figure 8.7(c). Interestingly, the deterioration in the mean is due to a significant increase in the variance of the TTSOR values returned by the two algorithms. The result is that, while the no-control case and maximax MARL were found to be statistically indistinguishable at a 5% level of significance, these two methods were both able to outperform kNN-TD RM, independent MARL and hierarchical MARL, as may be deduced from the p-values presented in Table 8.28. Furthermore, hierarchical MARL also outperformed independent MARL at a 5% level of significance in respect of the TTSOR, while independent MARL and kNN-TD RM were found to perform statistically indistinguishably.

Independent MARL was able to achieve the smallest mean travel times for vehicles travelling along the highway only, outperforming all other algorithms in respect of the mean TISHW, as may be seen in Table 8.29. The second-best performing algorithm in respect of the mean TISHW was hierarchical MARL as it outperformed maximax MARL, while hierarchical MARL was found to perform statistically indistinguishably at a 5% level of significance from the no-control case and kNN-TD RM in respect of the mean TISHW. Although kNN-TD RM and maximax MARL returned smaller mean TISHW values than the no-control case, the differences

TABLE 8.27: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	5.9648×10^{-2}	2.8422×10^{-4}	3.4919×10^{-2}	5.5626×10^{-1}
kNN-TD		—	2.0993×10^{-4}	9.9569×10^{-1}	5.4484×10^{-2}
Independent			—	1.2108×10^{-1}	2.9714×10^{-10}
Hierarchical				—	4.9525×10^{-3}
Maximax					—
Mean	517.07	500.40	489.21	499.46	508.08

TABLE 8.28: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	3.2264×10^{-3}	4.2369×10^{-4}	3.1426×10^{-4}	9.7902×10^{-1}
kNN-TD		—	9.1662×10^{-1}	6.9900×10^{-2}	3.5678×10^{-3}
Independent			—	8.7942×10^{-3}	4.6615×10^{-4}
Hierarchical				—	5.0154×10^{-5}
Maximax					—
Mean	32.93	46.53	50.77	37.05	33.06

TABLE 8.29: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.4588×10^{-2}	9.0345×10^{-5}	5.1969×10^{-2}	7.9949×10^{-1}
kNN-TD		—	1.3049×10^{-4}	4.0492×10^{-1}	2.0074×10^{-5}
Independent			—	9.1109×10^{-12}	$< 1 \times 10^{-17}$
Hierarchical				—	9.8780×10^{-5}
Maximax					—
Mean	3.60	3.52	3.41	3.50	3.56

TABLE 8.30: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	2.1368×10^{-3}	1.5088×10^{-4}	5.8677×10^{-6}	4.7545×10^{-12}
kNN-TD		—	8.8472×10^{-1}	6.4442×10^{-2}	4.3507×10^{-3}
Independent			—	4.7958×10^{-3}	3.0038×10^{-4}
Hierarchical				—	1.9586×10^{-4}
Maximax					—
Mean	1.54	2.19	2.41	1.75	1.58

TABLE 8.31: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	2.7965×10^{-2}	3.6865×10^{-5}	3.8707×10^{-4}	1.9926×10^{-3}
kNN-TD		—	1.2656×10^{-3}	9.7282×10^{-2}	5.8741×10^{-1}
Independent			—	1.9285×10^{-2}	4.8477×10^{-2}
Hierarchical				—	8.8006×10^{-1}
Maximax					—
Mean	8.16	6.46	5.36	5.77	5.99

TABLE 8.32: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	5.0068×10^{-5}	1.1786×10^{-7}	7.4810×10^{-8}	3.1813×10^{-3}
kNN-TD		—	7.0323×10^{-1}	3.9503×10^{-1}	8.0429×10^{-5}
Independent			—	7.4209×10^{-3}	1.8784×10^{-7}
Hierarchical				—	1.9448×10^{-7}
Maximax					—
Mean	2.13	6.02	7.25	4.65	2.25

were not large enough for the algorithms to be classified as performing statistically different. In respect of the maximum TISHW, independent MARL was again able to outperform all other algorithms, as may be seen in Table 8.31. Hierarchical MARL, maximax MARL and kNN-TD RM, on the other hand, were found to perform statistically similarly, while they were all able to outperfrom the no-control case at a 5% level of significance in respect of the maximum TISHW. As may be seen in Figures 8.7(d) and 8.7(f), the improvements achieved by the algorithms in respect of both these PMIs are, again, largely down to a reduction in the variances of the corresponding PMI-values.

As may be deduced from the box plots in Figures 8.7(e) and 8.7(g), a trend very similar to that observed for the TTSOR emerges for the mean and maximum travel times of vehicles joining the highway from the on-ramp. The no-control case was, as expected, the best-performing algorithm, achieving a mean TISOR-value of 1.54 minutes, followed by maximax MARL, which achieved a mean TISOR-value of 1.58 minutes, thereby outperforming all other algorithms. Maximax MARL was followed by hierarchical MARL, which achieved a mean TISOR-value of 1.75 minutes, thereby outperforming independent MARL, while its perfromance was found to be statistically indistinguishable from that of kNN-TD RM at a 5% level of significance. Finally, as may be seen from the p-values in Table 8.7, the kNN-TD RM implementation, which achieved a mean TISOR-value of 2.19 minutes, was found to perform statistically on par with independent MARL, which returned a value of 2.41 minutes in respect of the mean TISOR. This ordering of the algorithmic performances is clearly visible in the box plots in Figure 8.7(e). As may be seen in Figure 8.7(g), a very similar trend emerges in respect of the maximum TISOR. From the p-values in Table 8.32, it is evident that the no-control case outperformed all algorithms as it returned a maximum TISOR-value of 2.13 minutes. The no-control case was followed by maximax MARL which achieved a

maximum TISOR-value of 2.25 minutes, outperforming all three other algorithms. Maximax MARL was, again, followed by hierarchical MARL, which achieved a maximum TISOR-value of 4.65 minutes, outperforming independent MARL, while its performance was found to be statistically indistinguishable from that of kNN-TD RM, which achieved a maximum TISOR-value of 6.02 minutes. Finally, although independent MARL returned the largest maximum TISOR-value of 7.25 minutes, kNN-TD RM and independent MARL were again found to perform statistically indistinguishably at a 5% level of significance in respect of the maximum TISOR.

Discussion

Although maximax MARL was never able to outperform all of the other algorithms in respect of the TTS-values, it was never outperformed at a 5% level of significance in any of the four scenarios by any of the other algorithms in terms of the TTS. Furthermore, maximax MARL achieved the smallest TTS-value in Scenarios 1 and 3, while achieving the second-smallest TTS-value in Scenario 2. Apart from this consistently good performance in respect of the TTS, maximax MARL seemed to find a better balance than all other algorithms between protecting the highway flow and achieving acceptable queue lengths at the on-ramp, without compromising gains in respect of the TTS. This may be favourable, since it represents a fairer distribution of travel times for the road users, and it may prevent on-ramp queues spilling back into the arterial road networks feeding into the highway. Maximax MARL was also the most effective in utilising VSLs for homogenisation of traffic flow, as may be seen from the smaller variances achieved in respect of the TTS, as indicated by the smaller interquartile ranges in the corresponding box plots.

The hierarchical MARL implementation also proved to be very effective as, similarly to maximax MARL, it was never outperformed by another algorithm at a 5% level of significance in respect of the TTS. Hierarchical MARL, in fact, returned the smallest TTS-value in Scenarios 2 and 4, while it returned the second smallest TTS-value in Scenarios 1 and 3. Apart from the small TTS-values returned by hierarchical MARL, it also consistently achieved smaller values for the TTSOR, mean TISOR and maximum TISOR than did independent MARL and kNN-TD RM. This indicates that if heavy traffic conditions prevail on the highway, hierarchical MARL is able to find a better balance between protection of the highway flow and achieving an acceptable queue length on the on-ramp than kNN-TD and independent MARL, while it is outperformed in this regard by maximax MARL. Furthermore, hierarchical MARL consistently returned smaller variances in respect of the TTS than did independent MARL and kNN-TD RM, indicating that VSLs may have been more effectively utilised by hierarchical MARL for homogenisation of traffic flow.

Independent MARL was consistently the worst-performing of the MARL approaches in respect of the TTS, except in Scenario 4 where it achieved a smaller TTS-value than maximax MARL. The performance of independent MARL turned out to be very similar to that of kNN-TD RM in all four scenarios. In Scenario 3, independent MARL achieved the smallest travel times for vehicles travelling along the highway only. Independent MARL was, however, unable to outperform hierarchical MARL at a 5% level of significance in this regard, while hierarchical MARL did, in fact, outperform independent MARL in respect of all PMIs based on vehicles joining the highway from the on-ramp.

Although the reductions in the TTS achieved by the MARL approaches were never large enough to be of statistical significance when compared with the results returned by a single RM agent, there may be other benefits to employing a multi-agent approach, such as finding better balances between protecting the highway flow and achieving an acceptable on-ramp queue length.

Furthermore, the performances of the MARL approaches, especially hierarchical MARL and maximax MARL, were typically more consistent than that of the kNN-TD RM agent, as indicated by the smaller interquartile ranges visible in most of the box plots presented above. Finally, although communication between agents did not necessarily lead to further absolute reductions in the TTS, the control measures may be utilised more effectively when agents do communicate, as may be seen from the fact that, generally, hierarchical MARL and maximax MARL exhibited more consistent performances than independent MARL.

8.5 MARL with a Queueing Consideration

Although the hierarchical and maximax MARL approaches presented above were typically able to find a better balance than the single agent RM approaches between protecting the highway flow and achieving acceptable on-ramp queue lengths, these approaches still result in undesirably large on-ramp travel times. Therefore, queueing limitations were also introduced in these MARL approaches, as was done in the case of the single agent RM. Due to the fact that, in the single-agent RM implementations of Chapter 6, Q-Learning generally achieved the most favourable performance when on-ramp queue limitations were implemented, the Q-Learning algorithm was chosen for the RM component in the MARL implementations with an on-ramp queue consideration. For the VSL component, however, the kNN-TD RM algorithm was again employed due to its superior performance in the single-agent paradigm. The same three approaches towards integrating the RM and VSL problems by means of MARL, as outlined in §8.3 were again implemented, with the exception that a punishment for excessively long on-ramp queues was employed as in the reward function in (6.11).

In the implementation of the integrated feedback controller of Carlson *et al.* [24], the same controller parameters, as employed in the individual implementations of PI-ALINEA and MTFC for VSLs by Müller *et al.* [105] were maintained. It is expected that retaining the parameters as identified during the parameter evaluations of §6.5.1 and §7.5.1 will yield the best performance in the integrated case.

8.5.1 Reward Function Evaluation

This section is devoted to determining which combination of reward functions yields the best performance when implemented in each of the three MARL approaches when a queue limitation is applied. As was the case for the reward function evaluation conducted in §8.4.1, the evaluation in respect of the adapted reward functions is performed for the case of Scenario 2. The same three different cases as for the implementations without queueing considerations are again investigated, as may be seen in Table 8.33, except that a punishment as in (6.11) is awarded to the RM agent if the maximum allowable queue length, which was again set to 100 vehicles, is exceeded.

TABLE 8.33: *Reward function evaluation results for MARL with an on-ramp queue limit, measured in terms of the total time spent in the system (TTS) by the vehicles (in veh·h).*

	MARL Approach		
Reward	Independent	Hierarchical	Maximax
Case 1	1 021.48	897.33	912.42
Case 2	1 058.63	1 046.35	960.73
Case 3	1 203.45	1 138.66	1 159.37

As may be seen in Table 8.33, the case where the original reward functions are employed for both the RM and VSL agents consistently yields the best results in all three MARL implementations. Therefore, for the comparisons conducted in the following section, the RM agent is rewarded according to (6.11), while the VSL agent is rewarded according to (8.6).

As may have been expected, the introduction of the queue limitation in the MARL approaches was again detrimental to their performances, as may be seen in Table 8.34. This expectation holds in Scenarios 1, 2 and 4, while surprisingly, in Scenario 3, the introduction of the queue limitation resulted in a decrease in respect of the TTS. This phenomenon may be attributed to the fact that the on-ramp demand in Scenario 3 is large, while the demand on the highway is relatively small. As a result, the gains achieved by RM along the highway are not as pronounced as in Scenarios 1 and 2 where there exists a large traffic demand along the highway, while due to the large on-ramp demand in Scenario 3, the on-ramp queues and travel times grow quickly, resulting in a situation where managing the on-ramp queue allows the learning agent to find a better trade off between protecting the highway flow and managing a reasonable on-ramp queue.

TABLE 8.34: *The effect of employing queue limitations in the MARL implementations on their overall performance.*

	Scenario 1					
	Independent		Hierarchical		Maximax	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS (veh·h)	1 673.31	1 386.17	1 510.53	1 384.02	1 612.73	1 369.74
TTSHW (veh·h)	1 570.74	576.45	1 260.33	584.22	1 372.67	760.45
TTSOR (veh·h)	120.71	809.72	250.20	799.80	240.06	609.29
	Scenario 2					
	Independent		Hierarchical		Maximax	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS (veh·h)	1 021.48	877.32	897.33	840.55	912.42	852.53
TTSHW (veh·h)	922.63	612.11	749.67	613.49	764.01	677.02
TTSOR (veh·h)	98.85	265.22	147.66	227.05	148.41	175.50
	Scenario 3					
	Independent		Hierarchical		Maximax	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS (veh·h)	838.54	859.57	813.89	825.65	795.12	818.06
TTSHW (veh·h)	766.74	498.78	725.24	521.22	641.68	664.42
TTSOR (veh·h)	71.80	360.79	88.65	304.44	153.44	156.64
	Scenario 4					
	Independent		Hierarchical		Maximax	
PMI	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$	$\hat{w} = 100$	$\hat{w} = \infty$
TTS (veh·h)	546.90	539.98	568.20	536.52	549.38	541.14
TTSHW (veh·h)	508.67	489.21	511.30	499.46	514.84	508.08
TTSOR (veh·h)	38.23	50.77	56.90	37.05	34.54	33.06

8.5.2 Algorithmic Comparison

In this section, the simulation results and relative algorithmic performances are analysed for each of the MARL implementations of §8.3 with an added queue length restriction, which are compared with the feedback controller of Carlson *et al.* [24], employed as a benchmark strategy.

These comparisons are conducted in each of the four scenarios of varying traffic demand introduced in §5.3.2. The results are again presented and interpreted through the use of box plots in which the means, medians and interquartile ranges of the PMIs are indicated, as well as tables indicating whether or not statistical differences exist between the PMI-values for each pair of algorithms at a 5% level of significance.

Scenario 1

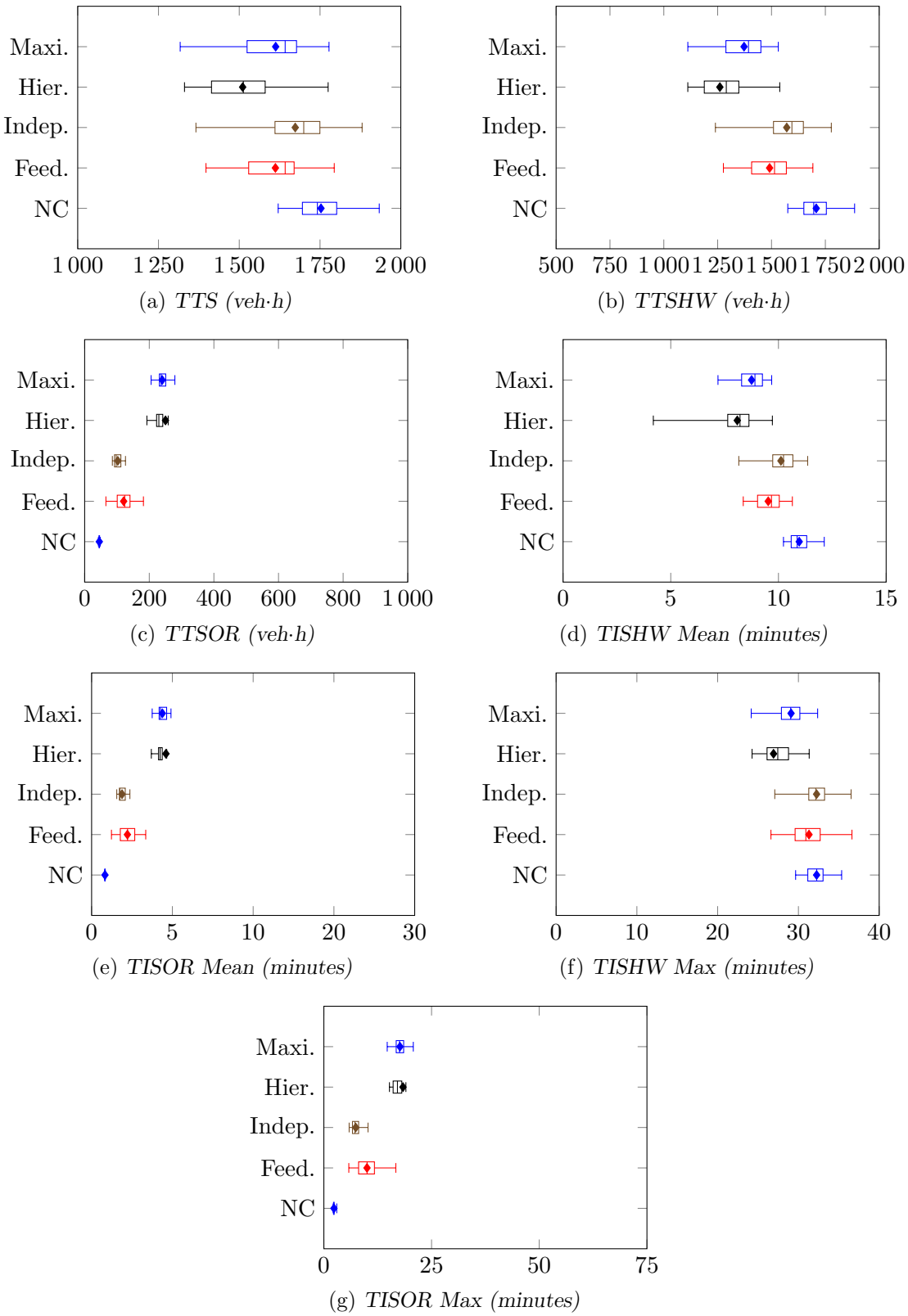
As may be seen from the p-values of the ANOVA and Levene statistical tests conducted on the PMI-values returned by the algorithms in Scenario 1, presented in Table 8.35, the ANOVA test revealed that there are, in fact, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all seven PMIs. Furthermore, Levene's test revealed that the variances of the algorithmic output data are statistically indistinguishable at a 5% level of significance in respect of the TTS, TTSHW, mean TISHW and maximum TISHW, while the variances of at least some pair of algorithmic output data differ statistically in respect of the TTSOR, mean and maximum TISOR. As a result, the Fisher LSD test is employed in order to ascertain between which pairs of algorithmic output data these differences occur in respect of the TTS, TTSHW, and mean and maximum TISHW, while the Games-Howell *post-hoc* test is employed for this purpose in respect of the other three PMIs.

TABLE 8.35: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 1. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	Feedback	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	1 753.01	1 611.90	1 673.31	1 510.53	1 612.73	1.9218×10^{-13}	6.3601×10^{-1}
TTSHW	1 707.70	1 491.19	1 570.74	1 260.33	1 372.67	$< 1 \times 10^{-17}$	4.0081×10^{-1}
TTSOR	45.31	120.71	102.56	250.20	240.06	$< 1 \times 10^{-17}$	1.7518×10^{-2}
TISHW Mean	10.96	9.53	10.12	8.09	8.75	$< 1 \times 10^{-17}$	3.5722×10^{-1}
TISOR Mean	1.66	4.43	3.76	9.20	8.76	$< 1 \times 10^{-17}$	1.6816×10^{-2}
TISHW Max	32.25	31.31	32.25	26.92	29.09	3.3307×10^{-16}	1.3583×10^{-1}
TISOR Max	2.34	10.02	7.40	18.37	17.67	$< 1 \times 10^{-17}$	1.9902×10^{-2}

As is evident from the box plots in Figure 8.8(a), all of the integrated control approaches were able to achieve improvements over the no-control case in respect of the TTS. This is corroborated by the p-values in Table 8.36. As may be seen in the table, hierarchical MARL returned the best performance, achieving a TTS-value of 1 510.53 veh·h. Hierarchical MARL is followed in the order of algorithmic performances by the feedback controller and maximax MARL, which achieved TTS-values of 1 611.90 veh·h and 1 612.73 veh·h, respectively, as they both outperformed independent MARL, while their performances were found to be statistically indistinguishable at a 5% level of significance. The order of algorithmic performances is finally completed by independent MARL, which returned a value of 1 673.31 veh·h in respect of the TTS, thereby only outperforming the no-control case.

Interestingly, the performances of all the algorithmic implementations were found to be statistically distinguishable at a 5% level of significance in respect of the TTSHW, as may be seen in Table 8.37. As for the TTS, hierarchical MARL returned the smallest TTSHW-value, achieving a 26.20% improvement over the no-control case. Hierarchical MARL was followed by maximax MARL, which outperformed the feedback controller, independent MARL and the no-control case as it returned an improvement of 19.62% over the no-control case. The feedback controller



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 8.8: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) with queue limits in Scenario 1.*

was able to achieve a reduction of 12.68% over the no-control case, which was large enough to enable it to outperform independent MARL for which an improvement of only 8.02% was recorded. This ordering of relative algorithmic performances is clearly visible in the box plots of Figure 8.8(b).

As expected, the no-control case returned the smallest TTSOR-value of 45.31 veh·h, outperforming all algorithmic implementations. Independent MARL and the feedback controller returned the next-best algorithmic performances, achieving TTSOR-values of 102.56 veh·h and 120.71 veh·h, respectively, thereby outperforming both hierarchical and maximax MARL at a 5% level of significance, as may be deduced from the p-values in Table 8.38. Although maximax MARL achieved a marginally smaller TTSOR-value of 240.06 veh·h than the 250.20 veh·h of hierarchical MARL, this difference was not large enough for the performances of these algorithms to be classified as statistically distinguishable at a 5% level of significance. These similarities in performance between independent MARL and the feedback controller, as well as hierarchical and maximax MARL, are also evident in the box plots of Figure 8.8(c).

The order of relative algorithmic performances in respect of the mean TISHW is the same as that in respect of the TTSHW, as all algorithms were again found to perform statistically differently at a 5% level of significance, as may be seen in Table 8.39. Hierarchical MARL again outperformed all other algorithms, as vehicles took, on average, 8.09 minutes to travel along the length of the highway. This value increased to 8.75 minutes for maximax MARL, which outperformed both independent MARL and the feedback controller. Independent MARL again returned the largest mean TISHW-value of 10.12 minutes, as it was outperformed by the feedback controller, which achieved a mean TISHW-value of 9.53 minutes. This ordering of the relative algorithmic performances is also evident in the box plots of Figure 8.8(d). As may be seen in Figure 8.8(f), a similar ordering of the algorithmic performances emerged in respect of the maximum TISHW. From the p-values in Table 8.41, it is evident that hierarchical MARL again returned the best performance, limiting the maximum travel time along the highway only to 26.92 minutes, outperforming all the other algorithmic implementations. Hierarchical MARL was again followed by maximax MARL for which this value increased to 29.09 minutes. Maximax MARL was thus able to outperform both the feedback controller and independent MARL at a 5% level of significance, while the latter two were found to perform statistically indistinguishably, as they returned maximum TISHW-values of 31.31 minutes and 32.25 minutes, respectively.

From the box plots in Figures 8.8(e) and 8.8(g), it is evident that the order of relative algorithmic performances in respect of the mean and maximum TISOR PMIs is exactly opposite to that in respect of the mean and maximum TISHW PMIs. As expected, the no-control case returned the smallest values in respect of both of these PMIs, achieving mean and maximum TISOR-values of 1.66 minutes, and 2.34 minutes, respectively, and outperforming all other algorithms at a 5% level of significance, as may be seen in Tables 8.40 and 8.42. In respect of the mean TISOR, the no-control case was followed by the feedback controller and independent MARL, which were found to perform statistically indistinguishably at a 5% level of significance, as these algorithms returned values of 4.43 minutes and 3.76 minutes, thereby outperforming both hierarchical and maximax MARL, which returned mean TISOR-values of 9.20 minutes and 8.76 minutes, respectively. Finally, hierarchical and maximax MARL were found to perform statistically similarly at a 5% level of significance in respect of the mean TISOR. This ordering of the relative algorithmic performances changes only slightly in respect of the maximum TISOR, as independent MARL was able to outperform all other algorithms at a 5% level of significance, having achieved a maximum TISOR value of 7.40 minutes. As may be seen in Table 8.42, the feedback controller outperformed both hierarchical and maximax MARL at a 5% level of significance, while the performances of the latter two were again found to be statistically on par

with one another, as these algorithms returned maximum TISOR-values of 10.02 minutes, 18.37 minutes and 17.67 minutes, respectively.

TABLE 8.36: *Differences in respect of the total time spent in the system by all vehicles (TTS) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.2008×10^{-6}	4.8146×10^{-3}	6.2172×10^{-15}	1.3705×10^{-6}
Feedback		—	2.8975×10^{-2}	3.7619×10^{-4}	9.7636×10^{-1}
Independent			—	3.1759×10^{-8}	3.1170×10^{-2}
Hierarchical				—	3.3838×10^{-4}
Maximax					—
Mean	1 753.01	1 611.90	1 673.31	1 510.53	1 612.73

TABLE 8.37: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTSHW			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	3.7354×10^{-11}	1.2278×10^{-5}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Feedback		—	9.4376×10^{-3}	2.7948×10^{-12}	1.3639×10^{-4}
Independent			—	$< 1 \times 10^{-17}$	9.3381×10^{-10}
Hierarchical				—	2.8930×10^{-4}
Maximax					—
Mean	1 707.70	1 491.19	1 570.74	1 260.33	1 372.67

TABLE 8.38: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	3.5091×10^{-12}	2.3648×10^{-14}	3.3644×10^{-10}	$< 1 \times 10^{-17}$
Feedback		—	5.7289×10^{-2}	4.1090×10^{-6}	9.5324×10^{-13}
Independent			—	3.3644×10^{-7}	$< 1 \times 10^{-17}$
Hierarchical				—	9.8647×10^{-1}
Maximax					—
Mean	45.31	120.71	102.56	250.20	240.06

Scenario 2

As for Scenario 1, the ANOVA test performed on the algorithmic output data in the case of Scenario 2 revealed that there are again statistical differences between at least some pair of algorithmic outputs in respect of all seven PMIs at a 5% level of significance, as may be seen in Table 8.43. Furthermore, the Levene test revealed that the variances in the algorithmic output data are only statistically indistinguishable in respect of the maximum TISHW, while statistical differences exist between the variances of at least some pair of algorithmic output data at a 5% level of significance in respect of all six other PMIs. In order to ascertain between which

TABLE 8.39: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.0982×10^{-12}	9.2439×10^{-6}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Feedback		—	1.6597×10^{-3}	1.1483×10^{-12}	4.7973×10^{-5}
Independent			—	$< 1 \times 10^{-17}$	1.0327×10^{-11}
Hierarchical				—	4.3155×10^{-4}
Maximax					—
Mean	10.96	9.53	10.12	8.09	8.75

TABLE 8.40: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	4.2734×10^{-12}	$< 1 \times 10^{-17}$	1.9376×10^{-10}	3.1089×10^{-13}
Feedback		—	6.1337×10^{-2}	2.6312×10^{-6}	1.0133×10^{-12}
Independent			—	2.0988×10^{-7}	3.1153×10^{-13}
Hierarchical				—	9.7258×10^{-1}
Maximax					—
Mean	1.66	4.43	3.76	9.20	8.76

TABLE 8.41: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.4547×10^{-1}	9.9466×10^{-1}	7.7827×10^{-14}	2.4740×10^{-6}
Feedback		—	1.4732×10^{-1}	2.4334×10^{-10}	7.5544×10^{-4}
Independent			—	8.0824×10^{-14}	2.5475×10^{-6}
Hierarchical				—	9.7529×10^{-4}
Maximax					—
Mean	32.25	31.31	32.25	26.92	29.09

TABLE 8.42: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.4433×10^{-14}	5.3291×10^{-14}	5.9590×10^{-11}	2.3093×10^{-14}
Feedback		—	1.0804×10^{-4}	6.9839×10^{-6}	2.1849×10^{-13}
Independent			—	2.9610×10^{-8}	$< 1 \times 10^{-17}$
Hierarchical				—	9.8456×10^{-1}
Maximax					—
Mean	2.34	10.02	7.40	18.37	17.67

pairs of algorithms the differences in algorithmic output data occur, the Fisher LSD test was performed in respect of the maximum TISHW, while the Games-Howell test was performed for this purpose in respect of the other six PMIs.

TABLE 8.43: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 2. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					*p*-value	
PMI	No Control	Feed.	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	1 141.79	950.92	1 021.48	897.33	912.42	$< 1 \times 10^{-17}$	3.5305×10^{-4}
TTSHW	1 107.88	908.93	922.63	749.67	764.01	$< 1 \times 10^{-17}$	2.2059×10^{-4}
TTSOR	33.92	41.99	98.85	147.66	148.41	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Mean	7.08	5.83	5.91	4.80	4.92	$< 1 \times 10^{-17}$	4.2140×10^{-4}
TISOR Mean	1.58	2.01	4.64	6.97	7.03	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Max	19.45	16.85	17.53	13.18	12.65	$< 1 \times 10^{-17}$	6.3153×10^{-1}
TISOR Max	2.13	5.20	11.65	20.34	21.17	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$

In respect of the TTS, hierarchical MARL, maximax MARL and the feedback controller returned the best performances, achieving 21.41%, 20.09% and 16.72% improvements over the no-control case, respectively. As may be seen in Table 8.44, these improvements were large enough to outperform independent MARL, which achieved a reduction of only 10.54% over the no-control case, while their performances were found to be statistically indistinguishable at a 5% level of significance. This similarity in performance of the feedback controller, hierarchical and maximax MARL, as well as the improvement achieved by these implementations over both independent MARL and the no-control case, is also visible in the box plots of Figure 8.9(a).

Hierarchical MARL and maximax MARL also returned the best performances in respect of the TTSHW, as is clearly visible in the box plots of Figure 8.9(b). This is corroborated by the *p*-values in Table 8.45, from which it is evident that hierarchical and maximax MARL outperformed all three other implementations at a 5% level of significance, achieving TTSHW-values of 749.67 veh·h and 764.01 veh·h, respectively. The feedback controller achieved the next smallest TTSHW-value of 908.93 veh·h. This value was not, however, small enough to allow it to outperform independent MARL, which achieved a TTSHW-value of 922.63 veh·h. All of the algorithmic implementations were, however, able to outperform the no-control case at a 5% level of significance, which returned a TTSHW-value of 1 107.88 veh·h.

Taking the natural increase in respect of the travel times for vehicles joining the highway from the on-ramp due to RM into account, it is the feedback controller which returned the best performance in respect of the TTSOR, as may be seen in Table 8.46. The feedback controller returned a value of 41.99 veh·h in respect of the TTSOR, thereby outperforming independent, hierarchical and maximax MARL which achieved values of 98.85 veh·h, 147.67 veh·h and 148.41 veh·h, respectively, at a 5% level of significance. Owing to its relatively small TTSOR-value, independent MARL was able to outperform both hierarchical and maximax MARL, while no statistically significant differences could be found between the performances of the latter two at a 95% level of confidence. As may be seen in the box plots of Figure 8.9(c), hierarchical and maximax MARL interestingly did not only result in an increase in the TTSOR-value, but also exhibited a comparatively large variance, indicating that due to the increased level of RM employed in these strategies, the travel times for vehicles joining the highway from the on-ramp also display more variability.

The order of relative algorithmic performances in respect of both the mean and maximum TISHW is the same as that in respect of the TTSHW, as may be seen in the box plots of

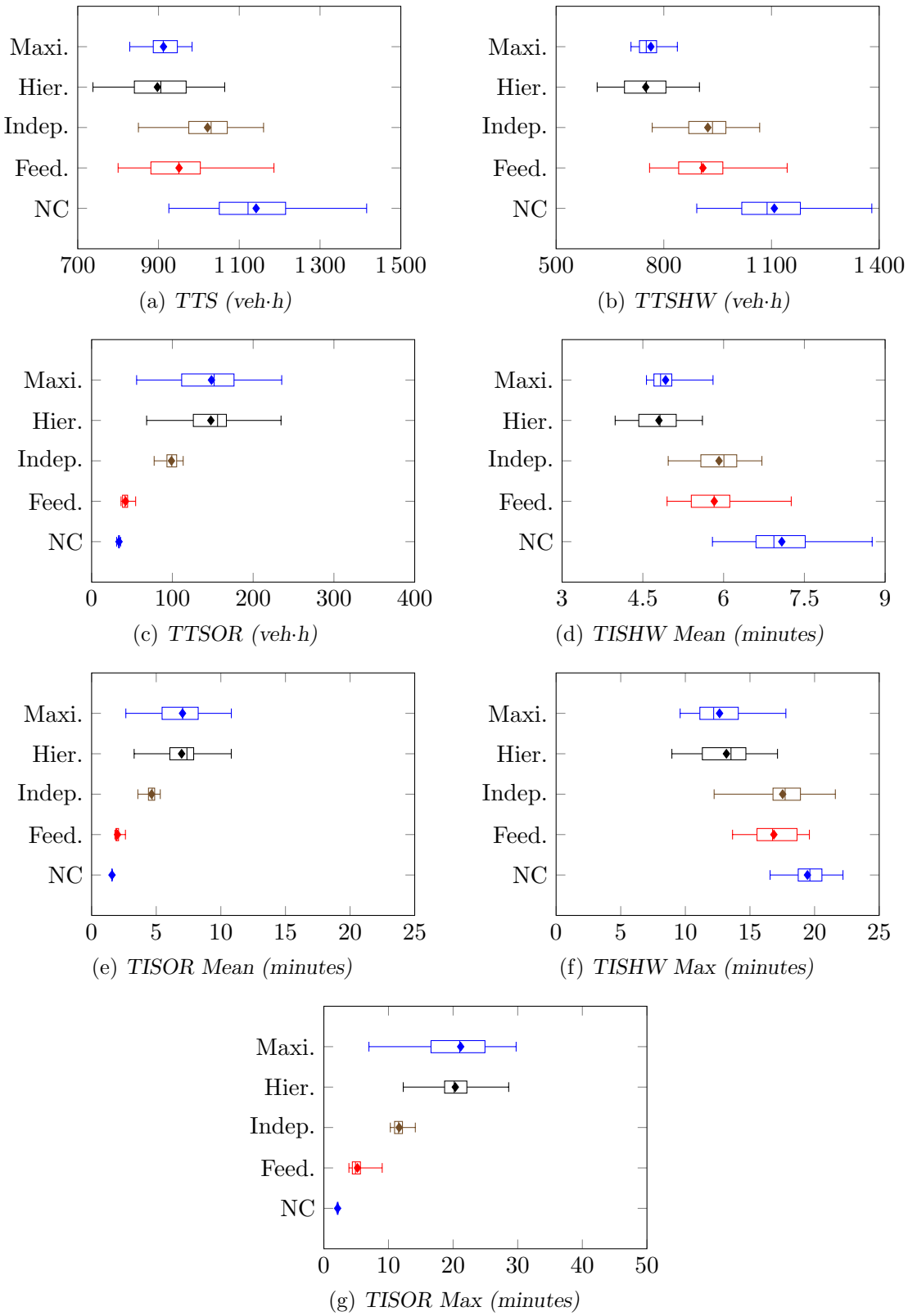


FIGURE 8.9: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) with queue limits in Scenario 2.*

Figures 8.9(d) and 8.9(f). As may be seen in Tables 8.47 and 8.49, hierarchical and maximax MARL again returned the best performance, outperforming all other algorithmic implementations at a 5% level of significance in respect of both of these PMIs, while their performances were found to be statistically indistinguishable from one another at a 5% level of significance. Hierarchical and maximax MARL are followed in the order of relative algorithmic performances by independent MARL and the feedback controller, whose performances were also found to be statistically on par with one another at a 5% level of significance in respect of both the mean and maximum TISHW PMIs, while they were both able to outperform the no-control case at a 5% level of significance in respect of both of these PMIs.

As may have been expected, the order of relative algorithmic performances in respect of the mean and maximum TISOR PMIs is the same as that in respect of the TTSOR. These trends are again clearly visible in the box plots of Figures 8.9(e) and 8.9(g). Taking the natural increase in travel times for those vehicles joining the highway from the on-ramp due to RM into account, it is the feedback controller which achieved the best performance, limiting the mean and maximum TISOR-values to 2.01 minutes and 5.20 minutes, respectively, thereby outperforming all the other algorithmic implementations, as may be seen from the p-values presented in Tables 8.48 and 8.50. The feedback controller is followed by independent MARL, which returned mean and maximum TISOR-values of 4.64 minutes and 11.65 minutes, respectively. These values were small enough to allow the algorithm to outperform both hierarchical and maximax MARL at a 5% level of significance. Finally, the performances of hierarchical and maximax MARL were again found to be statistically indistinguishable at a 5% level of significance in respect of both of these PMIs, as the latter algorithms achieved mean TISOR values of 6.97 minutes and 7.03 minutes, respectively, while these values increased to 20.34 minutes and 21.17 minutes, respectively, in respect of the maximum TISOR.

TABLE 8.44: *Differences in respect of the total time spent in the system by all vehicles (TTS) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	5.1248×10^{-8}	2.3896×10^{-4}	1.4081×10^{-11}	5.7429×10^{-11}
Feedback		—	1.9803×10^{-2}	1.5160×10^{-1}	2.7030×10^{-1}
Independent			—	1.9628×10^{-6}	3.1253×10^{-7}
Hierarchical				—	9.0998×10^{-1}
Maximax					—
Mean	1 141.79	950.92	1 021.48	897.33	912.42

TABLE 8.45: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.3817×10^{-8}	2.8117×10^{-8}	$< 1 \times 10^{-17}$	$< 1 \times 10^{17}$
Feedback		—	9.7005×10^{-1}	8.7225×10^{-9}	1.3631×10^{-8}
Independent			—	3.2228×10^{-11}	2.9107×10^{-12}
Hierarchical				—	8.9608×10^{-1}
Maximax					—
Mean	1 107.88	908.93	922.63	749.67	764.01

TABLE 8.46: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	3.9705×10^{-11}	2.4203×10^{-14}	$< 1 \times 10^{-17}$	2.4314×10^{-14}
Feedback		—	1.0265×10^{-12}	3.6748×10^{-15}	1.4133×10^{-14}
Independent			—	1.4330×10^{-8}	3.3146×10^{-4}
Hierarchical				—	9.9999×10^{-1}
Maximax					—
Mean	33.92	41.99	98.85	147.67	148.41

TABLE 8.47: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	2.7832×10^{-9}	7.2885×10^{-9}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Feedback		—	9.6290×10^{-1}	8.4020×10^{-10}	5.1192×10^{-9}
Independent			—	1.5143×10^{-11}	1.4659×10^{-12}
Hierarchical				—	6.8469×10^{-1}
Maximax					—
Mean	7.08	5.83	5.91	4.80	4.92

TABLE 8.48: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test: TISOR Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	3.6784×10^{-12}	8.8800×10^{-16}	$< 1 \times 10^{-17}$	1.3989×10^{-14}
Feedback		—	7.7083×10^{-13}	2.5091×10^{-14}	7.2720×10^{-14}
Independent			—	6.2115×10^{-8}	1.2241×10^{-6}
Hierarchical				—	9.9988×10^{-1}
Maximax					—
Mean	1.58	2.01	4.64	6.79	7.03

TABLE 8.49: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.3777×10^{-6}	2.8273×10^{-4}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Feedback		—	1.8998×10^{-1}	4.6470×10^{-11}	1.5743×10^{-3}
Independent			—	3.0420×10^{-14}	1.1102×10^{-16}
Hierarchical				—	3.0547×10^{-1}
Maximax					—
Mean	19.50	16.85	17.53	13.18	12.65

TABLE 8.50: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	6.8057×10^{-14}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Feedback		—	1.0821×10^{-11}	$< 1 \times 10^{-17}$	9.9365×10^{-14}
Independent			—	2.8599×10^{-13}	1.8971×10^{-9}
Hierarchical				—	9.5764×10^{-1}
Maximax					—
Mean	2.13	5.20	11.65	20.34	21.17

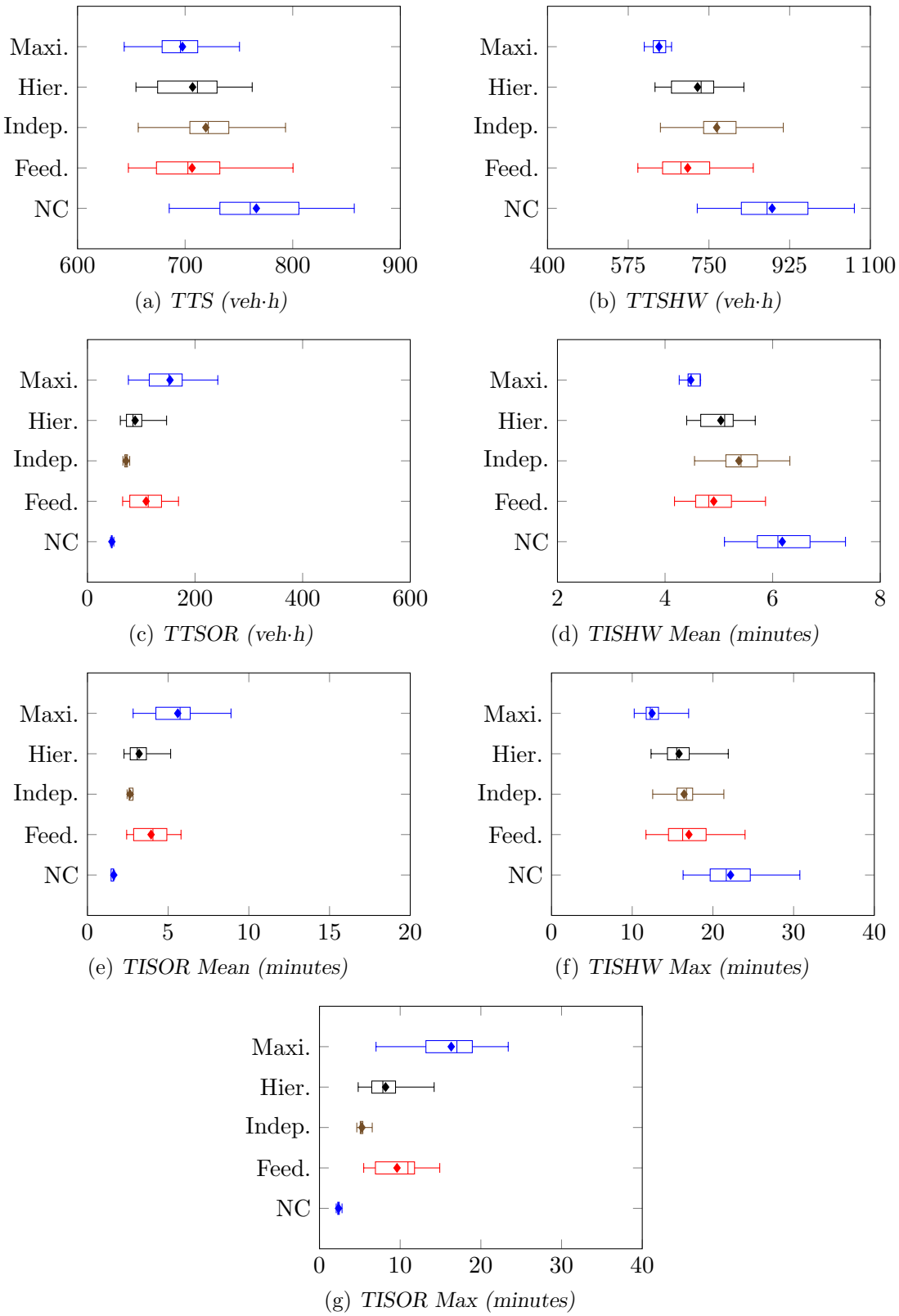
Scenario 3

As was the case in Scenarios 1 and 2, the ANOVA test performed on the algorithmic output data in Scenario 3 revealed that there are again statistical differences at a 5% level of significance between at least some pair of algorithms output in respect of all seven PMIs, as may be seen from the p-values presented in Table 8.51. The Levene test for homogeneity of variances furthermore revealed that the variances of the algorithmic outputs are statistically indistinguishable at a 5% level of significance only in respect of the TTS, while statistical differences exist between the variances of at least some pair of algorithms output in respect of the other six PMIs. The Fisher LSD test was thus performed in order to ascertain between which pairs of algorithms the differences occur in respect of the TTS, while the Games-Howell test was performed for this purpose in respect of the other six PMIs.

TABLE 8.51: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 3. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	Feed.	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	932.46	812.98	838.54	813.89	795.12	2.0731×10^{-11}	5.9125×10^{-2}
TTSHW	887.07	703.92	766.74	725.24	641.68	$< 1 \times 10^{-17}$	4.9341×10^{-9}
TTSOR	45.40	109.06	71.80	88.65	153.44	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Mean	6.18	4.91	5.37	5.04	4.48	$< 1 \times 10^{-17}$	6.5362×10^{-10}
TISOR Mean	1.63	3.94	2.64	3.20	5.60	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Max	22.19	17.02	16.43	15.80	12.46	$< 1 \times 10^{-17}$	7.9541×10^{-7}
TISOR Max	2.37	9.61	5.26	8.20	16.33	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$

All of the algorithmic implementations were once again able to improve on the no-control case in respect of the TTS in Scenario 3, as may be seen in the box plots of Figure 8.10(a). From the figure it is evident that the performances of the four algorithmic implementations are very similar in respect of the TTS. This is corroborated by the p-values in Table 8.52, from which it may be seen that the performances of the four implementations were found to be statistically indistinguishable at a 5% level of significance, except that maximax MARL, which returned the smallest TTS-value, was able to outperform independent MARL, which returned the largest TTS-value of the four algorithmic implementations. Maximax MARL achieved a TTS-value of 795.12 veh·h, compared with 812.98 veh·h, 813.89 veh·h and 838.54 veh·h for the feedback controller, hierarchical MARL and independent MARL, respectively, while the no-control case returned a TTS-value of 932.46 veh·h.



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 8.10: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) with queue limits Scenario 3.*

Maximax MARL also achieved the smallest TTSHW-value of 641.68 veh·h in Scenario 3, outperforming all other algorithmic implementations, as may be seen from the p-values in Table 8.53. Maximax MARL was followed by the feedback controller, which achieved a TTSHW-value of 703.92 veh·h, outperforming independent MARL at a 5% level of significance, while its performance was found to be statistically indistinguishable from that of hierarchical MARL, which achieved a TTSHW-value of 725.24 veh·h. Furthermore, the performances of hierarchical MARL and independent MARL were also found to be statistically indistinguishable at a 5% level of significance, as independent MARL returned a value of 838.54 veh·h in respect of the TTSHW. As is evident from the box plot in Figure 8.10(b), all of the algorithms were again able to outperform the no-control case in respect of the TTHSW.

Interestingly, in respect of the TTSOR, statistical differences were found between all of the algorithmic implementations at a 5% level of significance, as may be seen in Table 8.54. As expected, the no-control case returned the smallest TTSOR-value as it is the only implementation in which RM is not applied. The no-control case was followed by independent MARL, which returned a TTSOR-value of 71.80 veh·h compared with the 45.40 veh·h of the no-control case. The next-best performance was achieved by hierarchical MARL, for which the TTSOR-value increased to 88.65 veh·h. Hierarchical MARL is followed in the order of relative algorithmic performances by the feedback controller, which achieved a value of 109.06 veh·h in respect of the TTSOR. Finally, the good performance of maximax MARL in respect of the TTSHW, was compromised by its worst performance in respect of the TTSOR, as maximax MARL returned the largest TTSOR-value of 153.44 veh·h. These trends in the relative algorithmic performances are also clear in the box plots in Figure 8.10(c).

As is evident from the box plots in Figure 8.10(d), the order of relative algorithmic performances in respect of the mean TISHW is the same as that in respect of the TTSHW. This is corroborated by the results of the Games-Howell *post hoc* test in Table 8.55. Maximax MARL again achieved the best performance, outperforming all other implementations at a 5% level of significance in respect of the mean TISHW, as it achieved a 27.51% improvement over the no-control case. The feedback controller, which was able to achieve a reduction of 20.55% over the no-control case, returned the next-best performance, outperforming independent MARL, while its performance was found not to differ statistically from that of hierarchical MARL, which achieved a 18.45% improvement over the no-control case. Finally, the performances of independent MARL, which was able to reduce the mean TISHW by 13.11%, and hierarchical MARL were also found to be statistically on par with one another at a 5% level of significance. A similar trend emerged in respect of the maximum TISHW, as maximax MARL again returned the best performance, as an improvement of 43.85% over the no-control case was recorded, outperforming all other algorithms at a 5% level of significance, as may be seen in Table 8.57. Unlike for the mean TISHW, however, the performances of independent MARL, hierarchical MARL and the feedback controller were all found to be statistically similar at a 5% level of significance in respect of the maximum TISHW as they achieved 25.96%, 28.80% and 23.30% improvement over the no control case, respectively. The similarity in the performances of these three algorithms is also evident in the box plots of Figure 8.10(f).

As for the TTSOR, the performances of all the algorithmic implementations were again found to differ statistically at a 5% level of significance in respect of the mean TISOR, as is evident from the p-values in Table 8.56. As may be seen in Figure 8.10(e), the ordering of the relative algorithmic performances in respect of the mean TISOR is also the same as that in respect of the TTSOR, as independent MARL was the best-performing algorithm, achieving a mean TISOR-value of 2.64 minutes. Independent MARL was followed by hierarchical MARL, which achieved a value of 3.20 minutes. Hierarchical MARL was followed by the feedback controller

with a mean TISOR-value of 3.94 minutes, while the largest mean TISOR-value of 5.60 minutes was recorded for maximax MARL. From the box plots in Figure 8.10(g), it is evident that the ordering of the relative algorithmic performances in respect of the maximum TISOR is the same as that in respect of both the TTSOR and mean TISOR. This is corroborated by the p-values in Table 8.58, as independent MARL was again able to outperform all other algorithms at a 5% level of significance, having returned a maximum TISOR-value of 5.26 minutes. Although hierarchical MARL achieved a smaller maximum TISOR-value of 8.20 minutes, compared with 9.61 minutes for the feedback controller, this difference was not large enough for the performances of these algorithms to be classified as being statistically different at a 5% level of significance, while they were both able to outperform maximax MARL, which achieved a maximum TISOR-value of 16.33 minutes, at a 5% level of significance.

TABLE 8.52: *Differences in respect of the total time spent in the system by all vehicles (TTS) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	2.5421×10^{-9}	1.6659×10^{-6}	3.2534×10^{-9}	1.7060×10^{-11}
Feedback		—	1.7607×10^{-1}	9.6130×10^{-1}	3.4382×10^{-1}
Independent			—	1.9194×10^{-1}	2.2337×10^{-2}
Hierarchical				—	3.1977×10^{-1}
Maximax					—
Mean	932.46	812.98	838.54	813.89	795.12

TABLE 8.53: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	6.3617×10^{-11}	2.3635×10^{-6}	6.9849×10^{-10}	9.5590×10^{-14}
Feedback		—	5.7431×10^{-3}	6.9749×10^{-1}	4.3518×10^{-4}
Independent			—	7.2415×10^{-2}	7.4107×10^{-11}
Hierarchical				—	4.3194×10^{-8}
Maximax					—
Mean	887.07	703.92	766.74	725.24	641.68

TABLE 8.54: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.0579×10^{-10}	7.9781×10^{-13}	1.1275×10^{-10}	2.5211×10^{-12}
Feedback		—	6.2360×10^{-6}	4.6131×10^{-2}	7.6816×10^{-4}
Independent			—	2.1702×10^{-3}	1.7726×10^{-9}
Hierarchical				—	2.5342×10^{-7}
Maximax					—
Mean	45.40	109.06	71.80	88.65	153.44

TABLE 8.55: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	8.3908×10^{-12}	1.8298×10^{-5}	7.8285×10^{-11}	6.2172×10^{-14}
Feedback		—	1.2911×10^{-3}	7.3311×10^{-1}	1.6111×10^{-4}
Independent			—	1.7202×10^{-2}	2.3215×10^{-11}
Hierarchical				—	3.0138×10^{-8}
Maximax					—
Mean	6.18	4.91	5.37	5.04	4.48

TABLE 8.56: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	5.1279×10^{-11}	9.8699×10^{-14}	3.2949×10^{-11}	9.7911×10^{-13}
Feedback		—	6.6284×10^{-6}	3.4899×10^{-2}	3.4541×10^{-4}
Independent			—	2.8118×10^{-3}	9.8928×10^{-10}
Hierarchical				—	8.5806×10^{-8}
Maximax					—
Mean	1.63	3.94	2.64	3.20	5.60

TABLE 8.57: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	4.6105×10^{-6}	1.8628×10^{-8}	1.9485×10^{-9}	$< 1 \times 10^{-17}$
Feedback		—	9.1735×10^{-1}	4.7902×10^{-1}	2.8207×10^{-7}
Independent			—	7.9394×10^{-1}	2.8819×10^{-11}
Hierarchical				—	2.1738×10^{-7}
Maximax					—
Mean	22.19	17.02	16.43	15.80	12.46

TABLE 8.58: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	6.0839×10^{-14}	$< 1 \times 10^{-17}$	3.2785×10^{-13}	5.1070×10^{-15}
Feedback		—	9.9235×10^{-9}	2.0704×10^{-1}	1.7562×10^{-8}
Independent			—	1.3956×10^{-6}	8.9262×10^{-14}
Hierarchical				—	5.0261×10^{-11}
Maximax					—
Mean	2.37	9.61	5.26	8.20	16.33

Scenario 4

As may be seen in Table 8.59, the results of the ANOVA test performed on the algorithmic output data in Scenario 4, indicate that statistical differences again exist between at least some pair of algorithms output at a 5% level of significance in respect of all seven PMIs. Furthermore, Levene's test revealed that there are also statistical differences between variances of the algorithmic output of at least some pair of algorithms in respect of each of the seven PMIs. As a result, the Games-Howell test was performed in respect of all seven PMIs in order to ascertain between which pair of algorithmic output these statistical differences occur.

TABLE 8.59: *The mean values of all seven PMIs, as well as the corresponding p-values for the ANOVA and Levene statistical tests in Scenario 4. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	Feed.	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	550.00	531.62	546.90	568.20	549.38	1.1692×10^{-5}	3.5072×10^{-5}
TTSHW	517.07	492.35	508.67	511.30	514.84	7.0896×10^{-9}	6.9126×10^{-6}
TTSOR	32.93	39.27	38.23	56.90	34.54	$< 1 \times 10^{-17}$	1.5581×10^{-10}
TISHW Mean	3.48	3.44	3.56	3.56	3.60	1.9417×10^{-4}	2.5626×10^{-5}
TISOR Mean	1.54	1.85	1.82	2.75	1.63	$< 1 \times 10^{-17}$	1.1037×10^{-12}
TISHW Max	8.16	6.11	6.06	6.72	7.17	6.4850×10^{-6}	4.5783×10^{-8}
TISOR Max	2.13	4.65	4.29	9.09	2.68	$< 1 \times 10^{-17}$	1.3868×10^{-10}

Interestingly, in Scenario 4 none of the algorithmic implementations was able to outperform the no-control case at a 5% level of significance in respect of the TTS, as may be seen in Table 8.60. The feedback controller, which achieved the smallest TTS-value of 531.62 veh·h, was, however, able to outperform both hierarchical MARL and maximax MARL, which achieved TTS-values of 568.20 veh·h and 549.38 veh·h, respectively, while the performances of the feedback controller and independent MARL were found to be statistically indistinguishable at a 5% level of significance. Independent MARL, which achieved a TTS-value of 546.90 veh· h, was also able to outperform hierarchical MARL at a 5% level of significance, while its performance was found not to differ statistically from that of maximax MARL. Finally, the performances of maximax MARL and hierarchical MARL were found to be statistically on par at a 5% level of significance. These trends, and the weak performance of hierarchical MARL are clearly visible in the box plots of Figure 8.11(a).

In respect of the TTSHW, only the feedback controller outperformed the no-control case, as well as all other algorithmic implementations at a 5% level of significance, while independent MARL, hierarchical MARL, maximax MARL and the no-control case were all found to perform statistically indistinguishably from one another at 5% level of significance, as may be seen in Table 8.61. This improvement in respect of the TTSHW by the feedback controller is clearly visible in Figure 8.11(b). The similarity between hierarchical MARL, maximax MARL and the no-control case is also very clear in the figure. Interestingly, independent MARL resulted in a smaller variance than the other MARL approaches and the no-control case, indicating a more stable traffic flow along the highway if independent MARL is employed.

As may have been expected, the superior performance of the feedback controller in respect of the TTSHW is compromised by a significant increase in the TTSOR, as may be seen in Figure 8.11(c). Interestingly, hierarchical MARL resulted in an even larger increase in the TTSOR than the feedback controller. This is corroborated by the p-values in Table 8.62. The no-control case naturally returned the smallest TTSOR-value of 32.93 veh·h, outperforming all of the algorithmic implementations at a 5% level of significance. The no-control case is followed in the

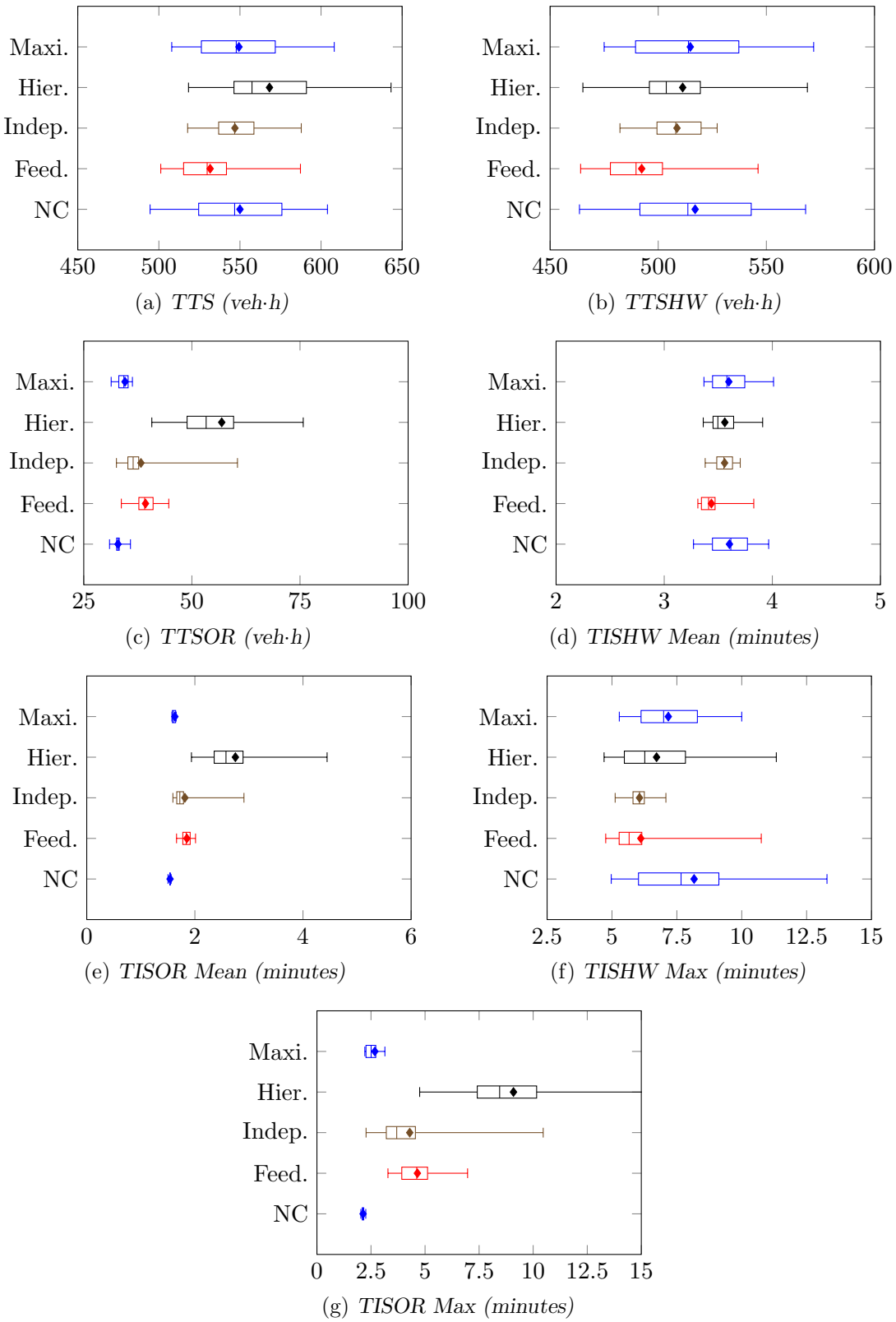


FIGURE 8.11: *PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) with queue limits in Scenario 4.*

order of relative algorithmic performances by maximax MARL, which achieved a value of 34.54 veh·h in respect of the TTSOR, thereby outperforming the feedback controller and hierarchical MARL while its performance was found to be statistically indistinguishable from that of independent MARL, which achieved a TTSOR-value of 38.23 veh·h. Independent MARL and the feedback controller were also found to perform statistically on par at a 5% level of significance, while they were both able to outperform hierarchical MARL, as the feedback controller and hierarchical MARL returned TTSOR-values of 29.27 veh·h and 56.90 veh·h, respectively.

The order of relative algorithmic performances in respect of the mean TISHW is the same as that in respect of the TTSHW, as may be seen in Figure 8.11(d). The feedback controller achieved the smallest mean TISHW-value, improving upon the no-control case by 4.44% and outperforming all of the other algorithmic implementations at a 5% level of significance. Independent MARL and hierarchical MARL were both able to reduce the mean TISHW by 1.11%, while the same average mean TISHW-values were recorded for maximax MARL and the no-control case. As may have been expected, independent MARL, hierarchical MARL, maximax MARL and the no-control case were all found to perform statistically indistinguishably at a 5% level of significance, as may be seen in Table 8.63. In respect of the maximum TISHW, however, both the feedback controller and independent MARL were able to outperform both the no-control case and maximax MARL, as they achieved reductions in the maximum TISHW of 25.12% and 25.74%, respectively. Hierarchical MARL, which achieved a reduction in the maximum TISHW of 17.65% was found to perform statistically on par with all the other implementations, not outperforming any other algorithm, but also not being outperformed by any other algorithm. Finally, although maximax MARL was able to improve on the no-control case by 12.13%, the performances of these two cases were found to be statistically indistinguishable at a 5% level of significance, as may be seen from the p-values in Table 8.65. These improvements by all of the algorithms are also visible in the box plots of Figure 8.11(f).

As may be seen in Figures 8.11(e) and 8.11(g) the order of relative algorithmic performances in respect of the mean and maximum TISOR PMIs is very similar to that in respect of the TTSOR. Maximax MARL again returned the best performance of all the algorithms in respect of both of these PMIs, limiting the mean and maximum TISOR to 1.63 minutes and 2.68 minutes, respectively, and thereby outperforming all the other algorithms at a 5% level of significance in respect of both of these PMIs, as may be seen from the p-values in Tables 8.64 and 8.66. Maximax MARL was followed by the feedback controller and independent MARL, which achieved mean TISOR-values of 1.85 minutes and 1.82 minutes, respectively, while limiting the maximum TISOR-values to 4.65 minutes and 4.29 minutes, respectively, as no statistical differences were found between these implementations at a 5% level of significance. The feedback controller and independent MARL were, however, both able to outperform hierarchical MARL at a 5% level of significance in respect of both the mean and maximum TISOR PMIs, as the mean and maximum TISOR-values increased to 2.75 minutes and 9.09 minutes, respectively, for the hierarchical MARL implementation.

Discussion

The hierarhcical MARL implementation again performed consistently well, outperforming all other implementations in respect of the TTS in Scenario 1, and only being outperformed once in respect of the TTS by the feedback controller and independent MARL in Scenario 4. This success in respect of the TTS may largely be attributed to exploiting the available on-ramp space and maximum allowable queue length well, thereby protecting the flow along the highway. This is evident from the fact that hierarchical MARL achieved the smallest TTSHW-values

TABLE 8.60: *Differences in respect of the total time spent in the system by all vehicles (TTS) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 95% level of confidence.*

		Games-Howell test: TTS			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	6.8161×10^{-2}	9.8813×10^{-1}	1.6670×10^{-1}	9.9999×10^{-1}
Feedback		—	1.4086×10^{-1}	1.3659×10^{-5}	4.3497×10^{-2}
Independent			—	1.1664×10^{-2}	9.9219×10^{-1}
Hierarchical				—	9.9085×10^{-2}
Maximax					—
Mean	550.00	531.62	546.90	568.20	549.38

TABLE 8.61: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	4.3846×10^{-3}	6.4083×10^{-1}	9.2879×10^{-1}	9.9818×10^{-1}
Feedback		—	2.8487×10^{-3}	1.3507×10^{-2}	4.3923×10^{-3}
Independent			—	9.8469×10^{-1}	7.8798×10^{-1}
Hierarchical				—	9.8397×10^{-1}
Maximax					—
Mean	517.07	492.35	508.67	511.30	514.84

TABLE 8.62: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	1.1774×10^{-3}	6.2728×10^{-11}	3.4677×10^{-2}
Feedback		—	9.2589×10^{-1}	5.0940×10^{-8}	6.6047×10^{-8}
Independent			—	1.7193×10^{-8}	5.2953×10^{-2}
Hierarchical				—	2.1267×10^{-7}
Maximax					—
Mean	32.93	39.27	38.23	56.90	34.54

TABLE 8.63: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	2.0772×10^{-3}	7.6963×10^{-1}	8.6988×10^{-1}	9.9995×10^{-1}
Feedback		—	5.3281×10^{-4}	8.1908×10^{-3}	1.9062×10^{-3}
Independent			—	9.9999×10^{-1}	8.2169×10^{-1}
Hierarchical				—	9.0832×10^{-1}
Maximax					—
Mean	3.60	3.44	3.56	3.56	3.60

TABLE 8.64: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	2.3207×10^{-10}	4.1722×10^{-4}	5.8262×10^{-11}	9.1923×10^{-4}
Feedback		—	9.7288×10^{-1}	3.8116×10^{-8}	2.3207×10^{-10}
Independent			—	1.6038×10^{-8}	3.2287×10^{-2}
Hierarchical				—	2.5077×10^{-10}
Maximax					—
Mean	1.54	1.85	1.82	2.75	1.63

TABLE 8.65: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	5.7674×10^{-3}	1.8310×10^{-3}	1.0463×10^{-1}	3.7907×10^{-1}
Feedback		—	9.9975×10^{-1}	5.5478×10^{-1}	3.4742×10^{-2}
Independent			—	2.2859×10^{-1}	7.7975×10^{-4}
Hierarchical				—	7.5391×10^{-1}
Maximax					—
Mean	8.16	6.11	6.06	6.72	7.17

TABLE 8.66: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.4655×10^{-14}	1.8631×10^{-5}	3.6826×10^{-13}	1.1391×10^{-3}
Feedback		—	8.9856×10^{-1}	8.8186×10^{-9}	4.2314×10^{-11}
Independent			—	4.1657×10^{-8}	1.5279×10^{-3}
Hierarchical				—	1.2172×10^{-12}
Maximax					—
Mean	2.13	4.65	4.29	9.09	2.68

in Scenario 1 and 2, while it was outperformed in respect of the TTSHW only by maximax MARL in Scenario 3 and the feedback controller in Scenario 4. This protection of the highway flow through effective RM did, however, result in relatively large TTSOR-values, as hierarchical MARL was outperformed by independent MARL in respect of the TTSOR in all four scenarios. Although hierarchical MARL did result in these increased travel times for vehicles joining the highway from the on-ramp, the maximum allowable queue length was only marginally exceeded in Scenarios 1 and 2, where maximum on-ramp queues of 115 vehicles and 108 vehicles were experienced, respectively, while the maximum on-ramp queue lengths in Scenarios 3 and 4 were restricted to 59 vehicles and 24 vehicles, respectively.

The maximax MARL implementation again also proved to be effective in reducing the TTS, being outperformed in this regard only by hierarchical MARL in Scenario 1 and by independent MARL and the feedback controller in Scenario 4. Similarly to the hierarchical MARL imple-

mentation, maximax MARL reduced the TTS mainly by reducing the TTSHW, outperforming all other algorithms in respect of the TTSHW in Scenario 3, while it was outperformed only by hierarchical MARL in Scenario 1 and the feedback controller in Scenario 4, in respect of the TTSHW. Futhermore, the traffic flow was typically more stable when employing maximax MARL than when employing hierarchical MARL, as is particularly evident in the box plots corresponding to the TTS and TTSHW in Scenarios 2 and 4. Similarly to the hierarchical MARL implementation, the improvements achieved along the highway are compromised by increases in the travel times of vehicles joining the highway from the on-ramp. Maximax MARL was, however, also effective in limiting the on-ramp queue as maximum queue lengths of 114 vehicles, 109 vehicles, 108 vehicles and 2 vehicles were recorded for Scenarios 1–4, respectively.

The feedback controller generally exhibited very consistent performance, being outperformed by hierarchical MARL in respect of the TTS in Scenario 1, while being outperformed by hierarchical MARL and maximax MARL in respect of the TTSHW in Scenarios 1–3. The feedback controller did, however outperform hierarchical MARL and maximax MARL in respect of the TTSOR in Scenarios 1 and 2, while it was outperformed in this respect by independent MARL in Scenarios 2–4.

Interestingly, independent MARL consistently achieved the best performance in respect of the TTSOR, while not fully utilising the available on-ramp queueing space. This good performance in respect of the TTSOR did, however, result in poor performance in respect of the TTSHW, as independent MARL was consistently outperformed in this regard by both hierarchical MARL and maximax MARL in Scenarios 1–3. The poor performance in respect of the TTSHW also meant that independent MARL was consistently outperformed by both hierarchical and maximax MARL in respect of the TTS in Scenarios 1–3, thus illustrating the value of communication between agents, especially when additional constraints, such as on-ramp queue limits are enforced.

8.6 Chapter Summary

This chapter opened in §8.3.1 with a brief description of an integrated feedback controller for simultaneously solving the RM and VSL control problems. This was followed by a brief introduction to the paradigm of MARL §8.2, introducing the notions of employing either independent or cooperative learners. Thereafter, a detailed description of the three approaches to MARL adopted in this dissertation followed in §8.3, namely independent learners (§8.3.1), hierarchical MARL (§8.3.2) and maximax MARL (§8.3.3) was provided.

An evaluation was carried out in §8.4.1 of the best combination of reward functions to be employed within each of these MARL implementations. Once these combinations of the reward functions had been found, the relative performances of the three MARL implementations were compared with one another in §8.4.2, as well as with kNN-TD RM, the best-performing single-agent RL algorithm. These comparisons were again conducted in the context of the four varying scenarios of traffic demand described in §5.3.2 within the benchmark simulation model of §5.1.2. It was found that the maximax MARL algorithm generally returned the most favourable results out of all the algorithms over all the traffic scenarios simulated.

Thereafter, a queueing limitation was implemented within the RM components of the MARL agents in §8.5, and a comparison of the three MARL approaches with the queueing limitation and the integrated feedback controller was performed. The hierarchical MARL implementation was found to return the most favourable performance when queue limitations are implemented.

CHAPTER 9

The N1: The Simulation Model

Contents

This chapter is devoted to a detailed description of an agent-based microscopic traffic simulation model for a case study in which the applicability of the RL implementations of Chapters 6–8 may be evaluated in a real-world scenario. In §9.1, the study area and corresponding simulation model are described. Then the focus shifts in §9.2 to a description of the input data employed within this case study, as well as a thorough description of how these data were gathered. This is followed in §9.3 by a description of the output data gathered from the simulation model execution. Thereafter, the validation of the simulation model is discussed in respect of real-world measurements in §9.4. In §9.5, the adopted experimental design is described, with a specific focus on the simulation warm-up period as well as some general specifications of the simulation framework. The chapter finally closes in §9.6 with a brief summary of the work included in the chapter.

9.1 Model Description

As was the case for the simulation benchmark model of §5.1.2, the case study simulation model was developed in the AnyLogic [5] software suite, making specific use of the built-in Road Traffic and Process Modelling Libraries. The highway section considered for the real-world case study comprises a stretch of the N1 national road outbound from Cape Town in South Africa's Western Cape province, as shown in Figure 9.1. As may be seen in the figure, the study area comprises the stretch of the N1 from before the R300 off-ramp (denoted by O_1) up to a section after the on-ramp at the Okavango Road interchange (denoted by D_3). Five on- and off-ramps fall within this study area, namely the off-ramp at the R300 interchange (denoted by D_1), the on-ramp at the R300 interchange (denoted by O_2), the on-ramp at the Brackenfell Boulevard interchange

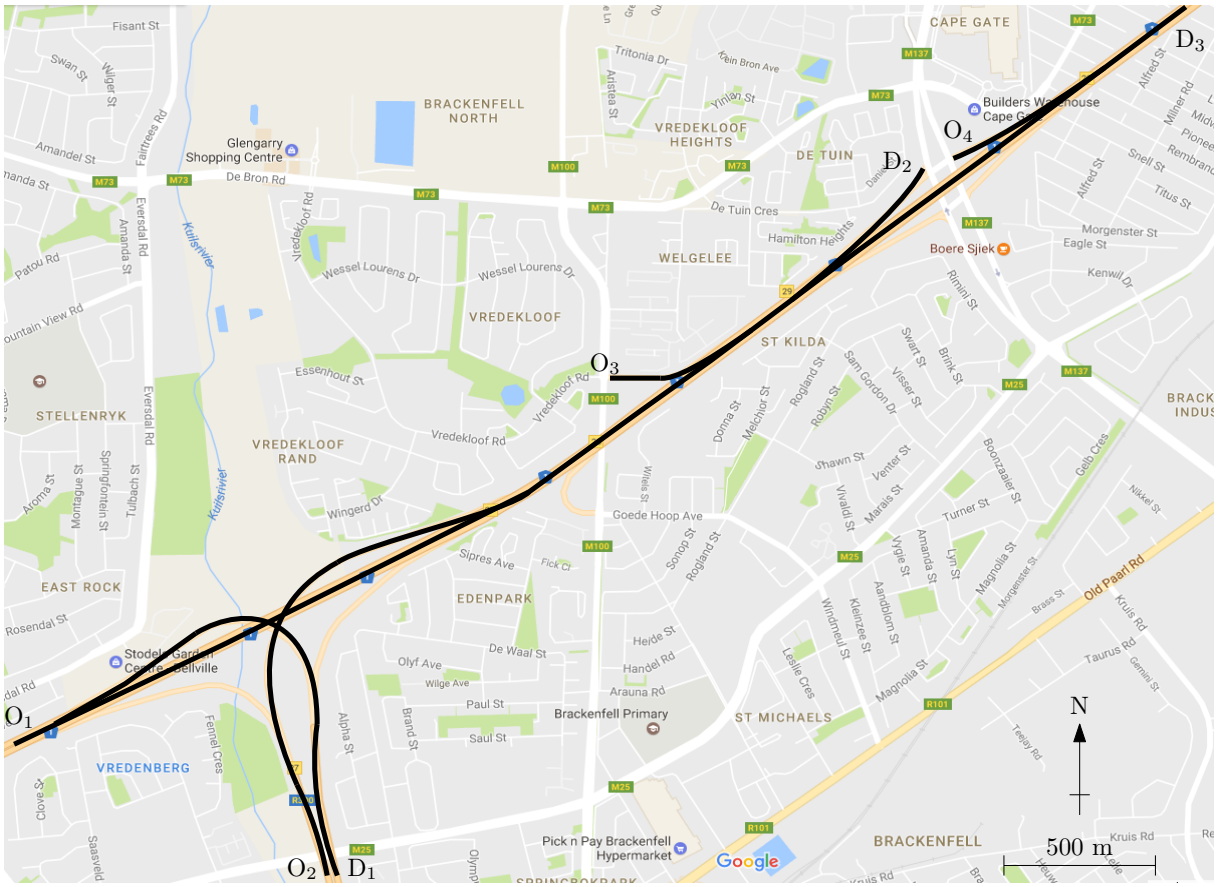


FIGURE 9.1: *The stretch of highway considered in this case study. The focus in this area is the N1 outbound from Cape Town, stretching from before the R300 off-ramp at O_1 to just past the Okavango Road on-ramp at D_3, including the R300 on-ramp at O_2, the R300 off-ramp at D_1, the Brackenfell Boulevard on-ramp at O_3, the Okavango Road off-ramp at D_2 and the Okavango Road on-ramp at O_4.*

(denoted by O_3), the off-ramp at the Okavango Road interchange (denoted by D_2), and the on-ramp at the Okavango Road interchange (denoted by O_4). The reason for investigation of this stretch of the N1 is due to high traffic volumes and significant congestion problems often observed there, especially during the afternoon peak. These problems may be attributed to large traffic volumes entering the N1 from the R300 and leaving the N1 at the Okavango Road off-ramp.

In order to ensure that the simulation model is an accurate representation of the real-world system in respect of the scale and shape of the road network, AnyLogic's [5] built-in GIS functionality was employed, as described in §5.1.1. All major routes were created by defining `GISPoints` and subsequently generating the routes between these points based on the existing infrastructure. Once these roads had been modelled, the connections at the intersections were added manually, as the intersections are not created automatically when the GIS routes are converted to road mark-up elements. This was followed by ensuring that the number of lanes, and the lane connectors joining the available routes through the intersections were correctly specified, as in the corresponding real-world system.

In the simulation model of the case study area, vehicles are generated at one of a number of `source` nodes, as shown in Figure 9.2. Vehicle arrivals in this simulation model are determined according to a Poisson distribution with an input mean equal to the desired traffic volume (measured in veh/h). The input mean is varied throughout the execution of the simulation

model as the demand profiles vary. In the case of insufficient space for a vehicle to enter the road network, the vehicle is stored in one of the various `queue` blocks until such time that sufficient space becomes available for the vehicle to enter the road network, passing through one of the `carEnter` blocks. Thereafter, the vehicle's destination is assigned as it passes through one of the `selectOutput` blocks. In these blocks, vehicles are randomly assigned to move either to the Okavango off-ramp, or continue along the N1 highway, according to a fixed probability. The `selectOutput` blocks, however, only perform the action of making a choice in respect of the route to be followed by a vehicle, while a final destination is assigned to the vehicle as it enters a `carMoveTo` block. Finally, once a vehicle reaches its destination, it is removed from the simulation environment by the `carDispose` block.

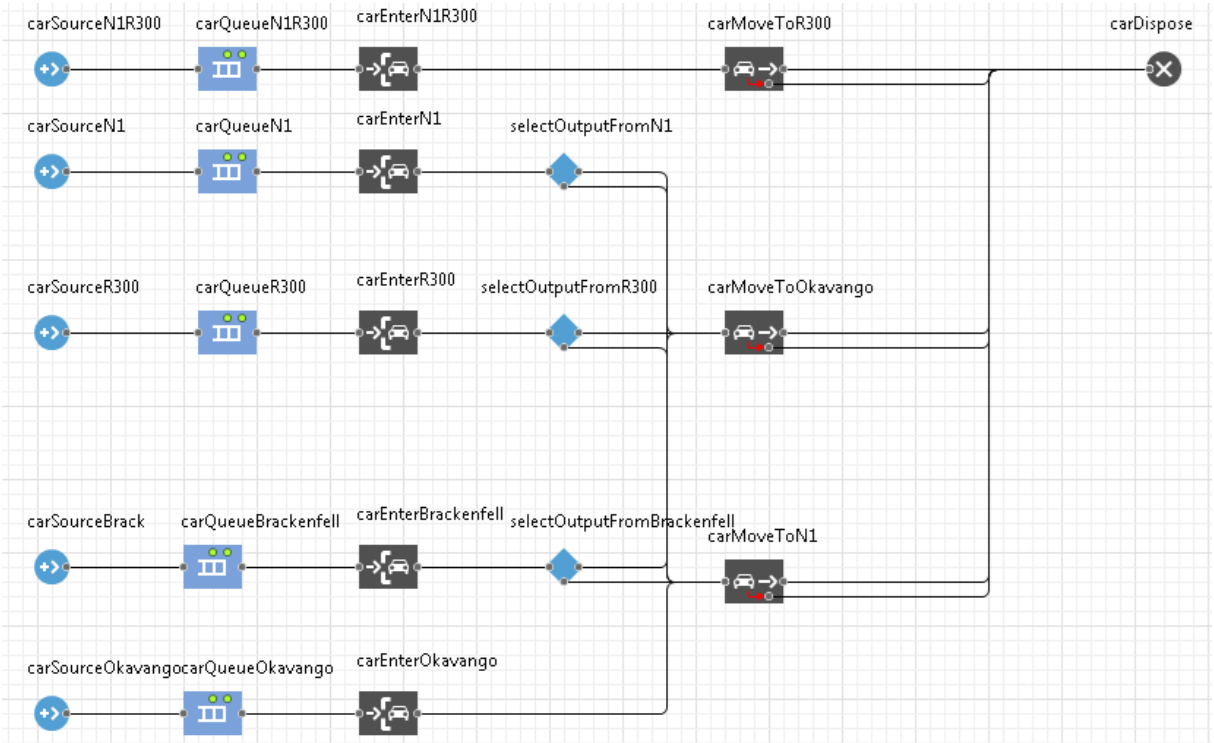


FIGURE 9.2: *A state chart depicting a number of connected blocks in the simulation model for the case study road network, illustrating the logical process followed by vehicles from the time that they are generated, until they reach their destinations.*

As may be seen in the figure, there are two `carSource` blocks generating vehicles on the N1 highway at O_1, namely `carSourceN1R300` and `carSourceN1`. The reason for this is that it is expected that vehicles travelling along the R300 off-ramp, generated by `carSourceN1R300`, will already have moved into the left-most lane by the time they enter the simulated environment. These vehicles therefore enter the network in the left-most lane, while the vehicles generated by the `carSourceN1` block enter the network in any one of three randomly assigned lanes. Light delivery vehicles and trucks, which are also simulated, follow the same logic as the passenger vehicles presented above.

9.2 Input Data

The input data for this special case study were obtained from the *South African National Roads Agency Limited* (SANRAL). Two major types of sources were employed in obtaining the

required input data. The primary sources are Wavetronix® [172] smart sensor devices installed at various locations along the major highways throughout the Cape Town metropole. The working of such a device, as well as the data collected by the device, is illustrated in Figure 9.3. The sensor employs two radar beams in order to detect individual vehicles as they pass the sensor, measuring individual vehicle data such as vehicle speed, length and the lane in which the vehicle is currently travelling. As may be seen in the figure, these data are then aggregated into lane data. For the classification category displayed under lane data, vehicles are classified into three major classes, based on their respective lengths. These classes are (1) passenger vehicles, (2) light delivery vehicles and (3) trucks. For the purposes of this case study it is assumed that the respective speed limits for vehicles in each of these classes are 120 km/h, 100 km/h and 80 km/h, respectively. The data from the Wavetronix® smart sensors were obtained in the form of *comma separated value* (.CSV) files in both hourly, and 10-minute intervals over the entire study period. Not all the data listed in Figure 9.3 were obtained. The data received only provided information on lane volumes, average speeds and vehicle classification.

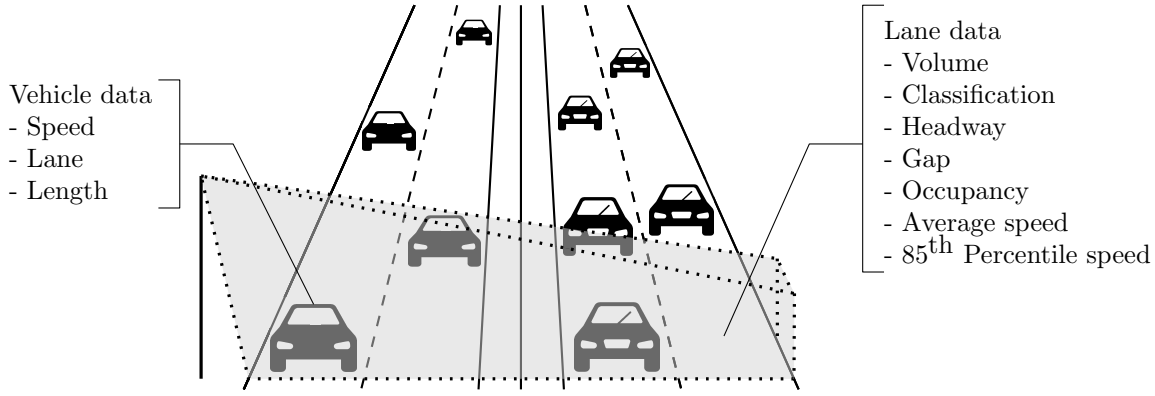


FIGURE 9.3: *The working and data collected by Wavetronix® smart sensor devices installed at various locations along the N1 highway in the Western Cape province of South Africa.*

The secondary sources of vehicle demand data are video recordings from *closed circuit television* (CCTV) cameras which are installed at all major intersections along the major routes in the Cape Town metropole. The CCTV footage was used to estimate the on- and off-ramp flows at intersections in cases where these flows could not be derived from the sensor data. For the sake of consistency with the data obtained from the Wavetronix® smart sensors, the vehicle flows that were estimated by counting vehicles from the CCTV footage were aggregated into the same 10-minute intervals. Vehicles were similarly classified into the same three vehicle classes, while vehicle speeds could naturally not be estimated from the video footage. A graphical illustration of the physical locations of the Wavetronix® smart sensors and the CCTV cameras may be found in Figure 9.4. Both the Wavetronix® smart sensor data and the video recordings were received for three Friday afternoon peaks. More specifically, data were received for the first three Fridays of March 2017. None of these days was a public holiday, and as a result, the recorded traffic flows should reflect the typical traffic situation during a Friday afternoon peak. Furthermore, the data were obtained for the time period spanning 15:30 to 18:30, as it is expected that this time window encapsulates the afternoon peak sufficiently.

9.3 Model Output Data

As was the case with the benchmark simulation model of §5.1.2, the performance data recorded throughout the model execution of each simulation run were written to an excel file at the end of

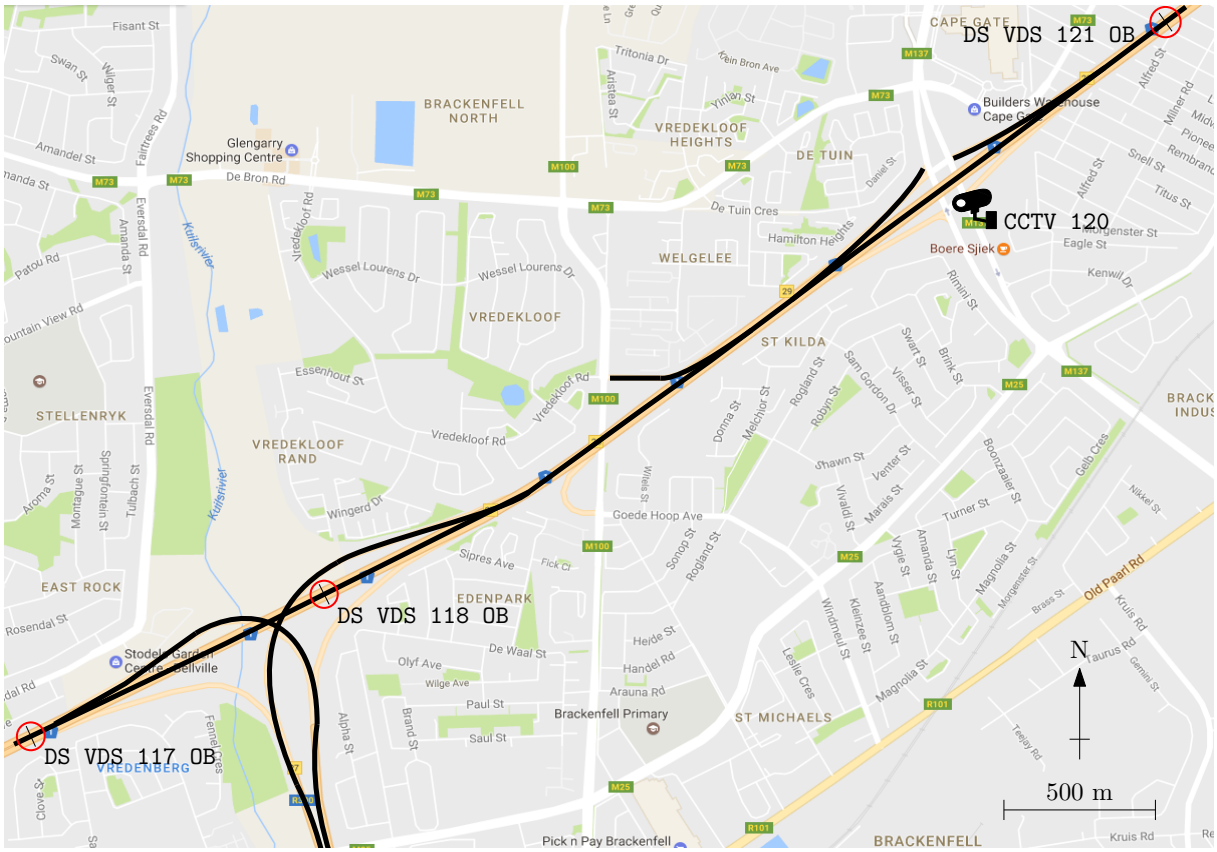


FIGURE 9.4: *Wavetronix® smart sensor locations (circled in red) and a CCTV camera location along the stretch of the N1 highway considered in the case study.*

the simulation run. These performance data were again partitioned into three classes of PMIs, according to which the relative algorithmic performance comparison is carried out.

The first class of PMIs was again the *total time spent in the system* (TTS) by the vehicles, which is simply the sum of the total times spent in the system by the various vehicles. This PMI was then broken down into four further PMIs, according to vehicle origin, namely the *total time spent in the system by vehicles coming from the N1* (TTSN1), the *total time spent in the system by vehicles coming from the R300* (TTSR300), the *total time spent in the system by vehicles coming from the Brackenfell Boulevard on-ramp* (TTSBB), and the *total time spent in the system by vehicles coming from the Okavango road on-ramp* (TTSO). The reason for this breakdown is again that increases in travel time due to ramp metering at the various locations may thus be captured more effectively as these are not captured sufficiently in the single TTS measure.

The second class of PMIs was the mean travel time. In order to account for differences in destinations of the vehicles and thus differences in their distances travelled, however, the mean travel time was normalised by dividing the time taken by a vehicle by the distance over which the vehicle had travelled. As with the total travel times, the normalised mean travel times were classified according to vehicle origin.

Similarly, for the third PMI class, normalised maximum travel times of the vehicles were recorded, again classified according to vehicle origin. In order to obtain the normalised value, the maximum travel time was divided by the distance over which the vehicle had travelled so as to take differences and distance travelled to the various destinations into account.

As was the case for the benchmark simulation model, additional information, such as the minimum values, maximum values, standard deviations, confidence intervals and the number of sample points, were also recorded for all the aforementioned output data. Furthermore, the same types of statistical analyses as performed for the benchmark simulation model described in §5.3.3 were performed during the analysis of the output data of the case study model.

9.4 Simulation Model Validation

Throughout the model building process, the same verification and validation techniques as outlined in §5.2 were employed. Furthermore, due to the availability of real-world data for the case study, simulation outputs were compared with real measurements from sensors installed along the study area. The real-world traffic flows in Tables 9.1, 9.2 and 9.6 were measured by the corresponding Wavetronix® sensors installed at the locations shown in Figure 9.4, while the real-world flows in Tables 9.3, 9.4 and 9.5 were derived by the author through a process of counting vehicles from video footage recorded by the CCTV camera located at the Okavango intersection.

For the model validation by means of the real-world measurements, the simulation model was executed for a period of three hours and forty minutes, so as to include a 40-minute warm-up period, before starting to record vehicle counts over the subsequent three hours. This process was replicated thirty times. The average output results of these thirty replications were compared against the real-world measurements and the absolute errors were recorded, as shown in Tables 9.1–9.6. Note that the values for the vehicle counts presented in the tables have been rounded to the nearest integer, while the simulation error percentages displayed in the tables were calculated from the vehicle data before rounding. As may be seen in the tables, the errors in respect of the flow of passenger vehicles, abbreviated in the tables as PV, after the three simulation hours never exceeds 2%. In terms of the light delivery vehicles, abbreviated in the tables as LDV, the maximum errors after three simulation hours rises to 4.90%. The reason for this is that the number of light delivery vehicles travelling through the system is significantly smaller than that of light passenger vehicles, resulting in the phenomenon that even a small deviation in terms of the number of vehicles is reflected as a relatively large error when expressed as a percentage. Finally, the largest error after three hours of simulation in terms of trucks, abbreviated in the table as T, travelling through the system is 2.86%. As in the case of light delivery vehicles, however, relatively few trucks travelled through the system, and as a result, a small error in terms of number is reflected as a relatively large error when expressed as a percentage. Due to the fact that the total error in respect of the number of vehicles that passed any of the six counting stations never exceeded 2%, the simulation model is deemed to be a sufficiently accurate representation of the underlying real-world system.

9.5 Experimental Design

This section is devoted to a discussion on various aspects pertaining to the experimental design according to which the algorithmic comparison of the various RL algorithms is performed in the following chapter. This includes the determination of a suitable simulation warm-up period, as well as some of the general specifications pertaining to the road network, such as vehicle and road attributes.

TABLE 9.1: *Validation of simulated traffic flow at DS VDS 117 OB.*

Time period	Measured flow			Simulated flow			Simulation error		
	PV	LDV	T	PV	LDV	T	PV	LDV	T
15:30 – 15:40	487	100	17	477	91	17	2.01%	8.86%	1.37%
15:40 – 15:50	985	188	28	951	183	33	3.48%	2.83%	16.79%
15:50 – 16:00	1 441	274	49	1 418	270	49	1.57%	1.30%	0.95%
16:00 – 16:10	1 914	374	67	1 907	363	71	0.37%	3.01%	5.47%
16:10 – 16:20	2 409	458	89	2 392	458	92	0.69%	0.01%	3.78%
16:20 – 16:30	2 912	551	106	2 875	558	114	1.27%	1.25%	4.15%
16:30 – 16:40	3 382	631	133	3 370	644	129	0.36%	2.03%	3.26%
16:40 – 16:50	3 877	735	144	3 869	729	144	0.21%	0.78%	0.12%
16:50 – 17:00	4 373	820	157	4 367	821	158	0.14%	0.13%	0.68%
17:00 – 17:10	4 856	894	178	4 880	887	177	0.49%	0.80%	0.82%
17:10 – 17:20	5 353	957	195	5 385	953	196	0.60%	0.44%	0.67%
17:20 – 17:30	5 808	1 019	212	5 882	1 017	215	1.28%	0.23%	1.24%
17:30 – 17:40	6 234	1 095	233	6 316	1 092	239	1.31%	0.25%	2.76%
17:40 – 17:50	6 689	1 181	257	6 743	1 169	264	0.80%	1.01%	2.84%
17:50 – 18:00	7 112	1 253	286	7 167	1 248	289	0.78%	0.40%	1.10%
18:00 – 18:10	7 550	1 322	308	7 599	1 323	311	0.65%	0.04%	1.13%
18:10 – 18:20	7 981	1 391	332	8 023	1 395	334	0.52%	0.31%	0.52%
18:20 – 18:30	8 420	1 478	349	8 448	1 468	356	0.33%	0.66%	2.05%
Total			10 247			10 272			0.24%

TABLE 9.2: *Validation of simulated traffic flow at DS VDS 118 OB.*

Time period	Measured flow			Simulated flow			Simulation error		
	PV	LDV	T	PV	LDV	T	PV	LDV	T
15:30 – 15:40	340	24	7	355	19	9	4.52%	22.08%	31.90%
15:40 – 15:50	698	45	16	708	37	17	1.36%	17.33%	8.13%
15:50 – 16:00	1 050	56	25	1 050	56	26	0.04%	0.77%	2.27%
16:00 – 16:10	1 415	73	38	1 415	77	38	0.01%	4.93%	0.61%
16:10 – 16:20	1 786	93	48	1 763	97	51	1.27%	3.87%	5.97%
16:20 – 16:30	2 140	109	61	2 103	116	63	1.71%	6.61%	3.39%
16:30 – 16:40	2 487	123	73	2 450	131	73	1.50%	6.29%	0.14%
16:40 – 16:50	2 839	138	84	2 804	141	82	1.24%	2.34%	2.66%
16:50 – 17:00	3 170	145	89	3 154	151	90	0.51%	4.11%	0.64%
17:00 – 17:10	3 522	149	103	3 513	160	103	0.25%	7.16%	0.10%
17:10 – 17:20	3 867	159	119	3 863	167	120	0.11%	5.28%	0.59%
17:20 – 17:30	4 202	165	133	4 207	175	135	0.12%	6.12%	1.60%
17:30 – 17:40	4 532	179	147	4 555	186	150	0.50%	4.17%	1.77%
17:40 – 17:50	4 878	192	161	4 897	198	163	0.40%	3.07%	1.16%
17:50 – 18:00	5 206	205	172	5 240	211	177	0.65%	2.89%	2.83%
18:00 – 18:10	5 570	218	185	5 602	222	190	0.58%	1.74%	2.70%
18:10 – 18:20	5 916	224	200	5 977	233	203	1.03%	3.85%	1.67%
18:20 – 18:30	6 283	237	211	6 355	242	217	1.15%	2.29%	2.81%
Total			6 731			6 814			1.23%

TABLE 9.3: *Validation of simulated traffic flow at Brackenfell Boulevard.*

Time period	Measured flow			Simulated flow			Simulation error		
	PV	LDV	T	PV	LDV	T	PV	LDV	T
15:30 – 15:40	63	1	1	71	1	1	12.54%	43.33%	26.67%
15:40 – 15:50	136	3	2	141	3	2	3.87%	3.33%	11.67%
15:50 – 16:00	219	6	3	212	6	3	3.42%	8.33%	5.56%
16:00 – 16:10	286	8	3	280	7	4	1.99%	15.41%	27.78%
16:10 – 16:20	352	9	4	346	8	5	1.63%	9.26%	15.83%
16:20 – 16:30	417	10	5	411	10	5	1.37%	2.67%	7.33%
16:30 – 16:40	492	12	6	482	11	7	1.94%	6.94%	8.33%
16:40 – 16:50	563	14	7	552	13	7	1.98%	9.05%	4.29%
16:50 – 17:00	631	15	8	624	14	8	1.16%	4.89%	0.42%
17:00 – 17:10	690	16	10	691	15	9	0.18%	7.29%	7.67%
17:10 – 17:20	754	17	11	755	15	11	0.09%	9.01%	1.21%
17:20 – 17:30	822	17	12	821	16	12	0.09%	5.09%	2.22%
17:30 – 17:40	874	17	12	885	16	12	1.22%	4.90%	3.06%
17:40 – 17:50	930	17	12	945	16	12	1.56%	4.90%	3.06%
17:50 – 18:00	990	17	12	1 003	16	12	1.29%	4.90%	3.06%
18:00 – 18:10	1 041	17	13	1 055	16	13	1.30%	4.90%	3.08%
18:10 – 18:20	1 091	17	13	1 107	16	13	1.49%	4.90%	1.28%
18:20 – 18:30	1 141	17	13	1 160	16	13	1.68%	4.90%	1.03%
Total			1 171			1 189			1.54%

TABLE 9.4: *Validation of simulated traffic flow at Okavango Road off-ramp.*

Time period	Measured flow			Simulated flow			Simulation error		
	PV	LDV	T	PV	LDV	T	PV	LDV	T
15:30 – 15:40	90	8	5	101	5	3	11.78%	40.41%	30.67%
15:40 – 15:50	185	15	10	212	9	7	14.50%	41.78%	32.33%
15:50 – 16:00	285	21	14	314	13	11	10.29%	36.98%	24.76%
16:00 – 16:10	391	27	17	418	19	14	7.16%	31.48%	17.45%
16:10 – 16:20	499	32	20	526	24	18	5.42%	24.38%	11.33%
16:20 – 16:30	609	35	23	640	30	21	5.01%	15.04%	8.84%
16:30 – 16:40	713	42	28	751	35	25	5.38%	17.14%	11.31%
16:40 – 16:50	825	47	32	867	38	28	5.09%	18.15%	13.33%
16:50 – 17:00	947	50	35	981	42	31	3.62%	16.40%	10.67%
17:00 – 17:10	1 059	52	40	1 093	45	35	3.26%	13.27%	13.08%
17:10 – 17:20	1 172	55	44	1 196	49	39	2.05%	11.39%	10.68%
17:20 – 17:30	1 287	58	46	1 301	53	44	1.15%	9.37%	4.86%
17:30 – 17:40	1 390	61	51	1 405	56	48	1.07%	7.70%	5.36%
17:40 – 17:50	1 497	66	56	1 507	62	52	0.68%	6.81%	6.90%
17:50 – 18:00	1 607	72	61	1 610	66	56	0.18%	7.78%	8.91%
18:00 – 18:10	1 712	75	66	1 711	71	60	0.05%	5.15%	9.55%
18:10 – 18:20	1 813	78	69	1 808	75	64	0.24%	3.59%	7.58%
18:20 – 18:30	1 910	81	70	1 906	78	68	0.20%	3.25%	2.86%
Total			2 061			2 052			0.44%

TABLE 9.5: *Validation of simulated traffic flow on the N1 after Okavango Road off-ramp.*

Time period	Measured flow PV	LDV	T	Simulated flow PV	LDV	T	Simulation error PV	LDV	T
15:30 – 15:40	417	14	16	359	15	13	13.88%	4.28%	18.75%
15:40 – 15:50	823	30	31	736	31	27	10.60%	4.44%	13.98%
15:50 – 16:00	1 218	48	46	1 119	48	41	8.12%	0.76%	11.74%
16:00 – 16:10	1 617	67	63	1 496	66	55	7.46%	1.00%	12.06%
16:10 – 16:20	2 022	83	79	1 874	83	72	7.30%	0.28%	8.86%
16:20 – 16:30	2 433	96	94	2 253	99	87	7.40%	3.23%	7.94%
16:30 – 16:40	2 731	110	107	2 634	117	100	3.54%	6.48%	6.45%
16:40 – 16:50	3 129	120	122	3 033	129	112	3.06%	7.89%	8.47%
16:50 – 17:00	3 523	133	134	3 443	141	123	2.27%	5.73%	8.18%
17:00 – 17:10	3 904	145	146	3 836	152	138	1.73%	4.67%	5.73%
17:10 – 17:20	4 260	157	168	4 229	162	157	0.73%	2.97%	6.45%
17:20 – 17:30	4 651	171	187	4 606	171	176	0.97%	0.19%	5.65%
17:30 – 17:40	5 044	188	204	4 992	183	194	1.03%	2.50%	4.80%
17:40 – 17:50	5 392	199	218	5 371	197	210	0.40%	0.84%	3.48%
17:50 – 18:00	5 788	212	232	5 755	212	226	0.57%	0.14%	2.74%
18:00 – 18:10	6 169	229	248	6 139	225	242	0.49%	1.54%	2.23%
18:10 – 18:20	6 523	238	265	6 545	238	260	0.34%	0.13%	1.97%
18:20 – 18:30	6 886	255	280	6 948	247	279	0.90%	2.97%	0.36%
Total			7 421			7 474			0.71%

TABLE 9.6: *Validation of simulated traffic flow at DS VDS 121 OB.*

Time period	Measured flow PV	LDV	T	Simulated flow PV	LDV	T	Simulation error PV	LDV	T
15:30 – 15:40	436	60	26	377	60	29	13.43%	0.39%	10.26%
15:40 – 15:50	846	116	57	772	124	56	8.72%	6.90%	1.17%
15:50 – 16:00	1 269	179	83	1 173	188	85	7.57%	5.07%	2.85%
16:00 – 16:10	1 711	220	114	1 591	242	114	7.00%	9.86%	0.41%
16:10 – 16:20	2 162	283	138	2 017	295	144	6.69%	4.36%	4.40%
16:20 – 16:30	2 612	337	169	2 442	347	174	6.51%	2.99%	2.88%
16:30 – 16:40	3 024	381	194	2 875	400	197	4.92%	4.97%	1.75%
16:40 – 16:50	3 440	425	219	3 332	447	218	3.14%	5.07%	0.56%
16:50 – 17:00	3 864	471	237	3 793	492	239	1.84%	4.47%	0.70%
17:00 – 17:10	4 303	519	265	4 252	535	259	1.18%	3.10%	2.20%
17:10 – 17:20	4 723	555	289	4 702	576	284	0.45%	3.86%	1.74%
17:20 – 17:30	5 160	601	315	5 146	618	310	0.27%	2.86%	1.52%
17:30 – 17:40	5 579	646	345	5 572	663	342	0.13%	2.65%	1.00%
17:40 – 17:50	5 943	690	372	5 975	710	369	0.53%	2.95%	0.68%
17:50 – 18:00	6 375	736	398	6 390	758	398	0.24%	3.05%	0.05%
18:00 – 18:10	6 794	784	424	6 813	800	425	0.28%	2.06%	0.24%
18:10 – 18:20	7 193	824	448	7 255	839	453	0.86%	1.88%	1.20%
18:20 – 18:30	7 585	863	477	7 697	877	482	1.48%	1.68%	1.04%
Total			8 925			9 056			1.47%

9.5.1 The Simulation Warm-up Period

At commencement of the simulation model, there are initially no vehicles present in the road network. As vehicles are generated at the source nodes, and begin to travel through the road network, the number of vehicles present in the road network gradually increases until the number of vehicles in the network reaches a steady state. The recording of vehicle travel times and delays during this initial period may potentially yield misleading results, due to the lower traffic demand implied by the comparatively small number of vehicles present in the network. For this reason it is necessary to determine a simulation warm-up period of a suitable length, which is long enough to ensure consistency in the recorded results, yet short enough in order to avoid wasted computation time during model execution. In order to determine the required length of this warm-up period for the case study simulation model, the same method as outlined in §5.3.1, which was previously employed in order to determine a suitable warm-up period for the benchmark model of §5.1.2, was again employed.

For the determination of the length of the warm-up period for the case study simulation model, the value of ω was chosen to be 30 replications as it is expected that this value will give a sufficiently accurate indication of the steady state of the system. Each iteration was run for 3 600 seconds, and observations regarding the number of vehicles present in the system were made every second, resulting in 3 600 observations for each simulation run. It was found that for the initial traffic flows shown in Table 9.7, a warm-up period of 2 400 seconds is sufficient. A graph depicting the convergence to the steady traffic state for these initial traffic conditions is shown in Figure 9.5.

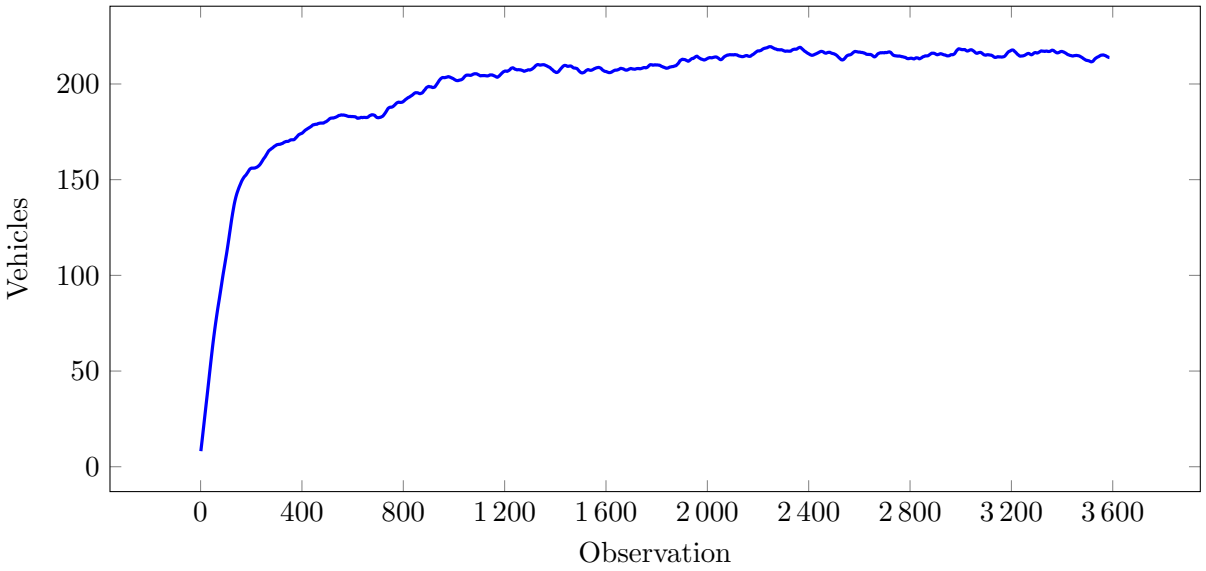


FIGURE 9.5: *An indication of the simulation warm-up time under the initial traffic conditions. A suitable warm-up time is approximately 2 400 seconds. At the steady state there are approximately 215 vehicles present in the network.*

9.5.2 General Specifications of the Simulation Framework

As stated above, the vehicle arrivals follow a Poisson distribution with an input mean equal to the desired traffic volume (measured in veh/h). These desired traffic volumes are adjusted in the simulation model in 30-minute intervals according to the arrival rates shown in Table 9.8.

TABLE 9.7: *Initial traffic flows for each vehicle class at each of the origins in the case study simulation model.*

Vehicle Class	Source node $O_{1.1}$	$O_{1.2}$	O_2	O_3	O_4
Passenger	888	2 240	500	50	100
Light delivery	436	112	14	10	225
Truck	48	50	50	5	60

TABLE 9.8: *Arrival rates employed as input data at each of the vehicle sources in the case study simulation model.*

Time period	Passenger vehicle arrivals $O_{1.1}$	$O_{1.2}$	O_2	O_3	O_4
15:30 – 16:00	890	2 150	738	438	110
16:00 – 16:30	808	2 290	850	396	256
16:30 – 17:00	1 140	2 065	644	428	324
17:00 – 17:30	1 121	2 071	450	382	336
17:30 – 18:00	675	2 058	350	356	152
18:00 – 18:30	462	2 254	300	312	224
Time period	**Delivery vehicle arrivals $O_{1.1}$**	**$O_{1.2}$**	**O_2**	**O_3**	**O_4**
15:30 – 16:00	436	112	20	12	285
16:00 – 16:30	448	126	21	8	215
16:30 – 17:00	466	60	31	10	204
17:00 – 17:30	358	48	48	4	196
17:30 – 18:00	388	75	30	0	198
18:00 – 18:30	386	60	40	0	168
Time period	**Truck arrivals $O_{1.1}$**	**$O_{1.2}$**	**O_2**	**O_3**	**O_4**
15:30 – 16:00	48	50	74	6	90
16:00 – 16:30	48	80	45	4	85
16:30 – 17:00	38	50	45	6	56
17:00 – 17:30	20	92	35	8	42
17:30 – 18:00	65	75	47	0	76
18:00 – 18:30	48	78	34	2	62

These arrival rates were initially estimated from the vehicle flows at each of the various counting stations, and subsequently adjusted empirically during the model validation process.

As part of the calibration of the simulation model (so that it would accurately reflect the corresponding real-world scenario), the vehicle properties were adjusted as these parameters have an influence on the car following behaviour which, in turn, affects the vehicle throughput. Passenger vehicle lengths were fixed at 5 metres, while light delivery vehicle lengths were taken as 10 metres, and trucks were assumed to be 15 metres in length. The initial speeds for passenger vehicles entering the network at O_1 and O_2 were set to 100 km/h, while the corresponding initial speeds at O_3 and O_4 were set to 60 km/h. Similarly, light delivery vehicles entering the network at O_1 or O_2 were assumed to have an initial speed of 100 km/h, while light delivery vehicles

entering the network at O_3 or O_4 were given an initial speed of 60 km/h. Finally, the initial speed of trucks entering the network at O_1 or O_2 was taken as 80 km/h, with trucks entering the network at a speed of 60 km/h at O_3 and O_4. In order to account for different driving styles and variation in driver aggressiveness, the preferred speeds of passenger vehicles were distributed uniformly between 110 km/h and 130 km/h, while the preferred speeds of light delivery vehicles were uniformly distributed between 90 km/h and 110 km/h. Finally, the preferred speeds of trucks were distributed uniformly between 70 km/h and 90 km/h. The maximum acceleration and deceleration values for passenger vehicles were taken as 2.7 m/s^2 and -4.4 m/s^2, respectively. For light delivery vehicles these values were set to 1.5 m/s^2 and -3.1 m/s^2, respectively, while the maximum acceleration and deceleration values for trucks were set at 1.5 m/s^2 and -2.8 m/s^2, respectively. Throughout the process of adjusting these values empirically, care was taken to stay within the reasonable bounds of 1.5 m/s^2 to 4 m/s^2 for the maximum acceleration and -1 m/s^2 to -6 m/s^2 for the maximum deceleration, respectively, as suggested by Amirjamshidi and Roorda [4] in their multi-objective approach to traffic microsimulation model calibration.

The probabilities that vehicles of given classifications, given their origins, will turn off from the N1 highway at the Okavango road interchange are as shown in Table 9.9. These probabilities were, just as the arrival rates at the various vehicle sources, adjusted empirically during the model validation process so as to achieve the most realistic representation of the underlying real-world system. As may be seen from the turning probability of vehicles generated at O_2 at the R300 on-ramp, 75% of vehicles which join the N1 from the R300 leave the N1 at the Okavango road interchange, which is in line with the earlier statement that there are large traffic volumes joining the N1 from the R300 which then leave the N1 at the Okavango road interchange.

TABLE 9.9: *The probabilities that vehicles which have entered the network from specific sources will turn off from the N1 highway at the Okavango Road interchange.*

	Vehicle type		
Source	PV	LDV	T
O_1	0.050	0.225	0.200
O_2	0.750	0.325	0.200
O_3	0.450	0.200	0.200

9.6 Chapter Summary

This chapter opened in §9.1 with a description of the area under consideration for the practical case study conducted in this dissertation, as well as a detailed description of the simulation model developed as testbed for the evaluation of the relative algorithmic performances in the following chapter. This was followed by a description of the input data obtained for the purpose of this case study in §9.2. Thereafter, the model output data were described briefly in §9.3. In §9.4, a model validation, carried out based on real-world measurements, was then presented, ensuring that the simulation model reflects the real-world situation sufficiently accurately. Finally, an experimental design was described in §9.5, with a specific focus on the simulation warm-up period as well as certain general parameter specifications employed in the simulation model.

CHAPTER 10

The N1: Computational Results

Contents

The purpose of this chapter is to provide a detailed description of the implementations of RM, VSLs and MARL for RM and VSLs in the context of the case study simulation model of Chapter 9. The chapter opens in §10.1 with a description of the implementations of the various algorithms for RM within the case study simulation model. In §10.1.1, the algorithmic implementations are discussed, while the focus shifts in §10.1.2 to the parameter evaluations conducted in order to determine the best-performing target density values in each of these implementations. Thereafter, a thorough algorithmic performance comparison is performed in §10.1.3. The

section on RM closes with a brief discussion of the results in §10.1.4. Queue limits are again incorporated in the RM implementations in §10.2, after which a thorough algorihmic performance comparison again follows. This is followed by a description of the VSL implementations in §10.3. More specifically, the algorithmic implementations, parameter evaluations and algorithmic performance comparisons are presented in §10.3.1–§10.3.3, respectively, and the section again closes with a brief discussion of the results obtained. This process is repeated for the MARL implementations, for which the algorithmic implementations are described in §10.4.1, followed by a reward function evaluation in §10.4.2 and a statistical performance comparison in §10.4.3. A brief discussion of the findings in respect of the MARL implementations is provided in §10.4.4. This process is again repeated for MARL agents with the addition of a queue limitation in §10.5. The chapter finally closes in §10.6 with a brief summary of the work included in the chapter.

10.1 Ramp Metering

This section is devoted to a thorough description of the parameter evaluation and algorithmic performance comparison performed in respect of the RM implementations within the case study simulation model of Chapter 9. The best-performing target densities for the ALINEA and PI-ALINEA implementations are determined first. Thereafter, the focus shifts to identifying the target densities which yield the best performance for the Q-Learning RM implementations. Finally, the best-performing target density values for the kNN-TD learning RM implementations are determined. Once these densities have been determined, the relative algorithmic performances are compared. The results of this comparison are presented and interpreted through the use of box plots in which the means, medians and interquartile ranges of the PMIs of §9.3 are indicated, as well as tables indicating whether or not statistical differences exist between the PMI-values for each pair of algorithms at a 5% level of significance.

10.1.1 Algorithmic Implementations

RM may be applied at all three on-ramps of the case study stretch of the N1 highway, namely the R300 on-ramp at O_2, the Brackenfell Boulevard on-ramp at O_3 and the Okavango Road on-ramp at O_4, as may be seen in Figure 10.1. The state spaces for the RM agents, comprising the downstream density, upstream density and on-ramp queue-length, remain unchanged from the implementation in the benchmark simulation model discussed in Chapter 6.

The R300 RM agent thus receives information on the downstream density ρ_{ds} at the section of highway directly downstream of the on-ramp where vehicles joining the highway from the on-ramp enter the highway traffic flow. The upstream density ρ_{us} is measured on the section of highway between the R300 off-ramp at D_1 and the R300 on-ramp at O_2, while the queue length w is the sum of the number of vehicles present on the R300 on-ramp and those in the queue buffer (in cases where there is not sufficient space available on the on-ramp for vehicles to enter the highway network).

The downstream density for the Brackenfell Boulevard RM agent is again measured at the section directly downstream of the on-ramp where the traffic flows from the on-ramp and the highway merge. The upstream density is measured on the section of highway between the R300 on-ramp at O_2 and the Brackenfell Boulevard on-ramp at O_3. Finally, the queue length is again the sum of the number of vehicles present on the on-ramp and the number of vehicles present in the queue buffer waiting to enter the road network as soon as sufficient space becomes available.

Similarly, for the Okavango Road RM agent, the downstream density is measured at the section where the on-ramp and highway traffic flows merge, while the upstream density is measured on the section of highway between the Okavango Road off-ramp at D_2 and the Okavango Road on-ramp at O_4. Finally, as was the case for both the other RM agents, the queue length is the sum of the number of vehicles present on the on-ramp and the number of vehicles in the queue buffer waiting to enter the road network.

The action space of the RM agents also remains unchanged from that employed in the benchmark simulation model for the implementations in the case study, where RM is again enforced by traffic lights placed at the on-ramps, with a fixed green phase time of 3 seconds, while the RM agents vary the red phase time in order to control the inflow of traffic onto the highway. Finally, the reward function for all three RM agents remains unchanged from that presented in (6.2).

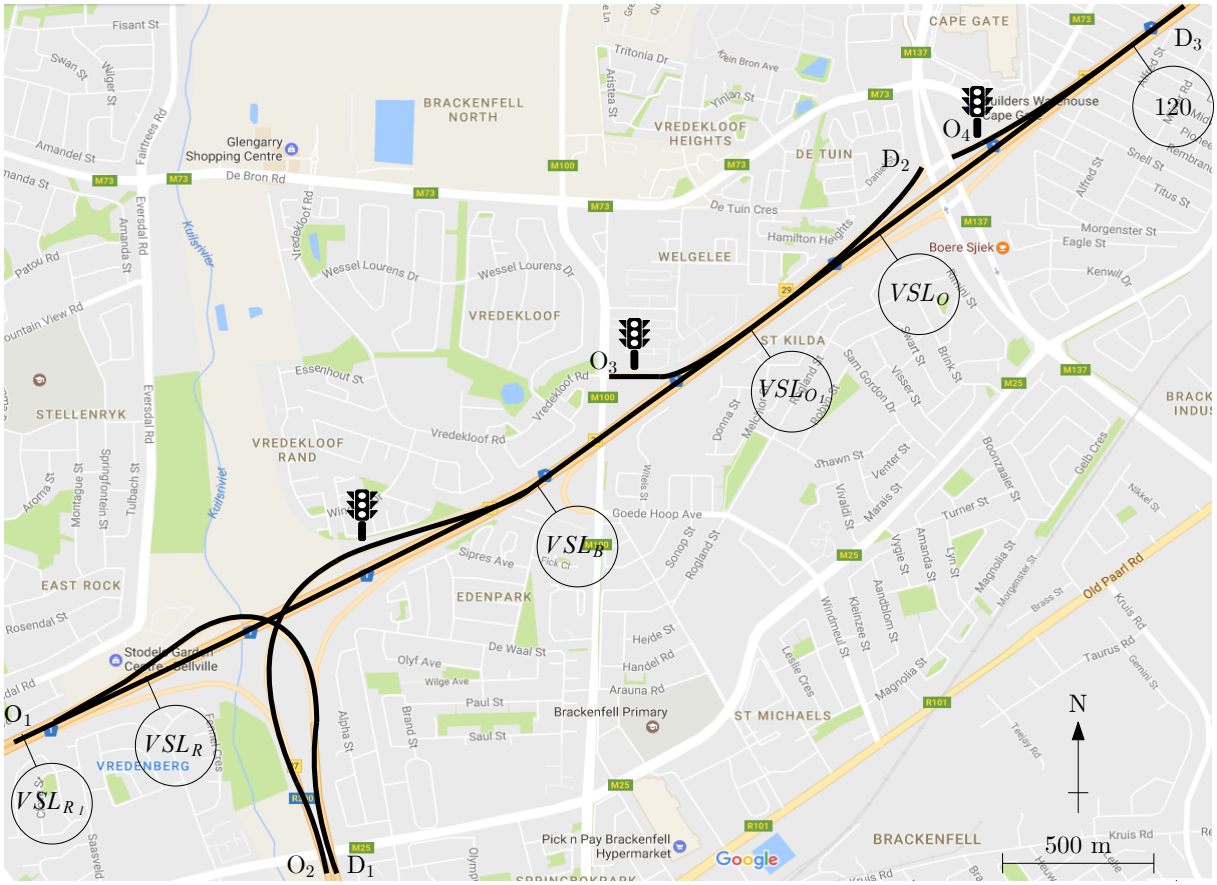


FIGURE 10.1: *The locations at which RM (indicated by the traffic lights) is applied in the case study area.*

10.1.2 Parameter Evaluations

This section is devoted to a thorough parameter evaluation with the aim of finding the best-performing target densities in respect of the ALINEA, PI-ALINEA, Q-Learning and kNN-TD RM implementations, measured according to the total time spent in the system by all vehicles. Furthermore, the aim in this section is to find the best-performing combinations of on-ramps in the case study area at which RM should be applied.

ALINEA parameter evaluation

Recall from §6.5.1 that for the ALINEA control strategy, a good combination of two parameter values has to be determined. In the parameter evaluation conducted in the context of this case study, the value of 40 for the nonnegative control parameter K_R is retained, while the aim of the parameter evaluation is finding the best target density $\hat{\rho}$ at each of the on-ramps considered. Due to the large number of combinations available when determining suitable target density values for each of the three on-ramps, a step-wise approach to determining good target densities was adopted in this dissertation. In this approach, the target density for the R300 on-ramp, which is the first on-ramp at which vehicles may enter the N1 in the study area, was determined first. Similarly to the parameter evaluation conducted for the benchmark simulation model, target densities between 24 veh/km and 34 veh/km were initially investigated in unit intervals. After it was found that setting the density to 31 veh/km yielded the best performance, the unit interval around 31 veh/km was examined more closely in intervals of 0.1 veh/km. The results of this parameter evaluation are presented in Table 10.1. As may be seen in the table, setting the target density to a value of 30.9 veh/km yielded the best performance. As a result, the target density was set to 30.9 veh/km for all further comparisons conducted including the ALINEA implementation at the R300 on-ramp.

TABLE 10.1: *Parameter evaluation results for the ALINEA RM control policy at the R300 on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	30.0	30.5	30.6	30.7	30.8	30.9	31.0
—	2 332.73	2 339.62	2 357.77	2 384.77	2 307.81	2 301.85	2 322.42
	Target density $\hat{\rho}$						
Combination	31.1	31.2	31.3	31.4	31.5	32.0	
—	2 432.30	2 375.70	2 393.66	2 381.68	2 340.45	2 340.15	

Once the best-performing target density for the R300 on-ramp had been determined, the focus shifted to the second on-ramp in the study area, namely the Brackenfell Boulevard on-ramp. For this on-ramp, an initial rough parameter evaluation from 24 veh/km to 34 veh/km was conducted in order to determine the best-performing target density as if it were the only RM implementation. Then the same parameter evaluation was repeated in order to evaluate the performance when RM is applied at both the Brackenfell Boulevard on-ramp and the R300 on-ramp. As stated above, the target density for ALINEA at the R300 on-ramp was kept at 30.9 veh/km. These results are presented in Table 10.2. As may be seen in the table, the case where RM is only applied at the Brackenfell Boulevard on-ramp consistently outperformed the combined case. As a result, for the finer parameter evaluation, only the case where RM is applied at the Brackenfell Boulevard on-ramp was considered. As may be seen in the table, setting the target density to 28.5 veh/km resulted in the best performance.

As in the case of the Brackenfell Boulevard on-ramp, a rough parameter evaluation from 24 veh/km to 34 veh/km was again conducted in combination with the previous best-performing RM implementation, as well as in the case where RM is only applied at the Okavango Road on-ramp. The results of this investigation may be seen in Table 10.3. It is clear that the case where RM is applied only at the Okavango Road on-ramp consistently outperformed the case where ALINEA is applied at both the Brackenfell Boulevard and Okavango Road on-ramps. The best-performing density for the Okavango Road on-ramp was found to be 31 veh/km. As may be seen from the results of the finer investigation in 0.1 veh/km increments around 31

TABLE 10.2: *Parameter evaluation results for the ALINEA RM control policy at the Brackenfell Boulevard on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	27	27.5	27.6	27.7	27.8	27.9	27.0
Alone	2 106.57	2 086.07	2 151.44	2 117.72	2 086.32	2 162.99	2 102.00
R300	2 369.14	—	—	—	—	—	2 430.72
	Target density $\hat{\rho}$						
Combination	28.1	28.2	28.3	28.4	28.5	28.6	29.0
Alone	2 163.24	2 096.87	2 162.69	2 185.59	2 063.73	2 088.41	2 119.48
R300	—	—	—	—	—	—	2 377.83

veh/km, shown in Table 10.3, the final best-performing target density was 31.2 veh/km. As the smallest TTS-value of all ALINEA implementations was achieved when RM is applied only at the Okavango Road on-ramp with a target density of 31.2 veh/km; this is the configuration employed for all comparisons involving ALINEA performed later in this chapter.

TABLE 10.3: *Parameter evaluation results for the ALINEA RM control policy at the Okavango Road on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	30	30.5	30.6	30.7	30.8	30.9	31.0
Alone	1 924.68	1 981.28	1 953.45	1 967.09	1 893.38	1 887.71	1 897.87
Brackenfell	2 082.89	—	—	—	—	—	1 975.11
	Target density $\hat{\rho}$						
Combination	31.1	31.2	31.3	31.4	31.5	32	
Alone	1 896.79	1 881.50	1 913.77	1 890.96	1 907.49	1 902.50	
Brackenfell	—	—	—	—	—	1 941.46	

PI-ALINEA parameter evaluation

As for the ALINEA implementation, the values of the nonnegative control parameters K_P and K_R were retained at 60 and 40, respectively. The same step-wise approach towards determining the best-performing target density values as that adopted for ALINEA was again performed for this purpose in respect of PI-ALINEA. As may be seen in Table 10.4, the initial rough investigation of target densities between 24 veh/km and 34 veh/km indicated that the smallest TTS-value could be achieved when setting the target density to 33 veh/km. Therefore, the unit interval around 33 veh/km was investigated in intervals of 0.1 veh/km. As may be seen in the table, setting the target density to 32.9 veh/km yielded the smallest TTS-value. Therefore, the target density is set to 32.9 veh/km for all further comparisons conducted involving PI-ALINEA for RM at the R300 on-ramp.

Once the best-performing target density at the R300 on-ramp had been found, the focus shifted to the Brackenfell Boulevard on-ramp. Again, the rough parameter evaluation between densities of 24 veh/km and 34 veh/km was conducted for the cases where RM is applied only at the Brackenfell Boulevard on-ramp, and where RM is applied at both the R300 and Brackenfell Boulevard on-ramps. Note that the target density at the R300 on-ramp was kept at 32.9 veh/km

TABLE 10.4: *Parameter evaluation results for the PI-ALINEA RM control policy at the R300 on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	32.0	32.5	32.6	32.7	32.8	32.9	33.0
—	2 399.43	2 349.23	2 396.26	2 352.48	2 390.19	2 290.47	2 335.51
	Target density $\hat{\rho}$						
Combination	33.1	33.2	33.3	33.4	33.5	34.0	
—	2 314.70	2 320.14	2 335.44	2 387.45	2 332.77	2 355.38	

throughout this parameter evaluation. As may be seen in Table 10.5, the case where RM is applied only at the Brackenfell Boulevard on-ramp consistently achieved smaller TTS-values than the case where RM is applied at both the R300 and Brackenfell Boulevard on-ramps, the smallest TTS-value being achieved when setting the target density to 31 veh/km. The finer investigation of the unit interval around 31 veh/km subsequently revealed that setting the target density to 30.9 veh/km at the Brackenfell Boulevard on-ramp yielded the best performance. As a result this is the target density employed in all further comparisons involving PI-ALINEA at the Brackenfell Boulevard on-ramp in this chapter.

TABLE 10.5: *Parameter evaluation results for the PI-ALINEA RM control policy at the Brackenfell Boulevard on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	30	30.5	30.6	30.7	30.8	30.9	31.0
Alone	2 012.73	2 033.58	2 135.58	1 975.46	2 002.20	1 932.19	1 949.73
R300	2 500.59	—	—	—	—	—	2 499.57
	Target density $\hat{\rho}$						
Combination	31.1	31.2	31.3	31.4	31.5	31.0	
Alone	2 091.81	2 013.64	1 944.39	2 096.03	2 069.26	2 027.05	
R300	—	—	—	—	—	2 511.30	

The same process for determining the best-performing target density at the Okavango Road on-ramp revealed that, again, applying RM only at the Okavango Road on-ramp and not in combination with RM at the Brackenfell Boulevard on-ramp, consistently yielded the best performance, with the initial parameter evaluation revealing that the best-performing target density at the Okavango Road on-ramp is 29 veh/km. As may be seen in Table 10.6, the unit interval around 29 veh/km was subsequently investigated in intervals of 0.1 veh/km, indicating that the smallest TTS-value was achieved when setting the target density at the Okavango Road on-ramp to 28.8 veh/km. Due to the fact that employing a target density of 28.8 veh/km with RM only applied at the Okavango Road on-ramp yielded the smallest TTS-value in all of the PI-ALINEA-related parameter evaluations, this is the combination employed in all further comparisons involving PI-ALINEA in this chapter.

Q-Learning parameter evaluation

The parameter evaluation conducted for the Q-Learning implementations followed the same step-wise approach, first determining the best-performing target density at the R300 on-ramp. The results of the initial investigation of the target densities between 24 veh/km and 34 veh/km

TABLE 10.6: *Parameter evaluation results for the PI-ALINEA RM control policy at the Okavango Road on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	28	28.5	28.6	28.7	28.8	28.9	29.0
Alone	1 925.67	1 899.80	2 008.02	1 945.64	1 851.21	1 923.77	1 903.44
Brackenfell	2 027.00	—	—	—	—	—	2 071.51
	Target density $\hat{\rho}$						
Combination	29.1	29.2	29.3	29.4	29.5	30	
Alone	1 916.89	1 953.46	2 001.50	1 915.77	1 989.46	1 908.19	
Brackenfell	—	—	—	—	—	2 059.89	

revealed that setting the target density to 34 veh/km yielded the best performance. As a result, the target densities of 35 veh/km and 36 veh/km were also investigated, and it was found that setting the target density to 35 veh/km resulted in the best performance. Therefore, the unit interval around 35 veh/km was subsequently investigated in 0.1 veh/km increments. The results of this investigation are presented in Table 10.7. As may be seen in the table, setting the target density to 34.9 veh/km resulted in the overall-smallest TTS-value. Therefore, the target density is set to 34.9 veh/km for all further investigations and comparisons including a Q-Learning RM agent at the R300 conducted in this chapter.

TABLE 10.7: *Parameter evaluation results for Q-Leaning RM at the R300 on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	34.0	34.5	34.6	34.7	34.8	34.9	35.0
—	2 305.19	2 302.06	2 312.62	2 295.41	2 346.31	2 314.73	2 268.57
	Target density $\hat{\rho}$						
Combination	35.1	35.2	35.3	35.4	35.5	36.0	
—	2 374.26	2 283.64	2 272.64	2 371.28	2 272.27	2 311.52	

Once the best-performing target density for the agent at the R300 on-ramp had been found, the focus shifted to the Brackenfell Boulevard on-ramp. Two scenarios were again investigated. In the first of these, there is only a single RM agent at the Brackenfell Boulevard on-ramp, and in the second there are two RM agents, one at the R300 on-ramp and the other at the Brackenfell Boulevard on-ramp. The results of the initial investigation in respect of target densities for the Brackenfell Boulevard RM agent are presented in Table 10.8. As may be seen in the table, the single RM agent consistently outperformed the combination of RM agents, achieving the smallest TTS-value at a target density of 34 veh/km, as was the case in the ALINEA and PI-ALINEA implementations. The surrounding unit interval was subsequently considered, and the results showed that the best performance is achieved when setting the target density for the Brackenfell Boulevard RM agent to 33.9 veh/km. The target density is therefore set to this value for all further investigations and comparisons conducted in this chapter involving a Q-Learning RM agent at the Brackenfell Boulevard on-ramp.

A rough parameter evaluation from 24 veh/km to 34 veh/km was again conducted in combination with the previously identified best-performing RM implementation, as well as the case where RM is only applied at the Okavango Road on-ramp. The results of this initial investigation are shown in Table 10.9. As may be seen in the table, the case where RM is applied only at

TABLE 10.8: *Parameter evaluation results for Q-Learning RM at the Brackenfell Boulevard on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	33	33.5	33.6	33.7	33.8	33.9	34.0
Alone	2 037.11	2 124.22	2 024.30	2 082.28	2 044.75	1 976.75	2 002.28
R300	2 403.43	—	—	—	—	—	2 431.19
	Target density $\hat{\rho}$						
Combination	34.1	34.2	34.3	34.4	34.5	35	
Alone	2 078.84	2 041.75	2 057.42	2 058.35	1 981.83	2 010.79	
R300	—	—	—	—	—	2 465.43	

the Okavango Road on-ramp consistently outperformed the case where RM is applied at both the Brackenfell Boulevard and Okavango Road on-ramps. The best-performing density for the Okavango Road on-ramp was found to be 32 veh/km. As may be seen from the results of the finer investigation in 0.1 veh/km increments around 32 veh/km, shown in Table 10.3, the final best-performing target density was 31.6 veh/km. As the smallest TTS-value of all Q-Learning RM implementations was achieved when RM is applied only at the Okavango Road on-ramp with a target density of 31.6 veh/km, this is the configuration employed for all comparisons involving the Q-Learning RM agents performed in this chapter.

TABLE 10.9: *Parameter evaluation results for Q-Learning RM at the Okavango Road on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	31	31.5	31.6	31.7	31.8	31.9	32.0
Alone	1 959.82	1 930.53	1 822.85	1 949.95	1 867.25	1 993.49	1 905.24
Brackenfell	2 160.71	—	—	—	—	—	2 054.78
	Target density $\hat{\rho}$						
Combination	32.1	32.2	32.3	32.4	32.5	33	
Alone	1 856.57	1 907.27	1 905.38	1 961.11	1 879.07	1 925.60	
Brackenfell	—	—	—	—	—	2 112.20	

kNN-TD learning parameter evaluation

A step-wise approach was again followed to determine the best-performing target densities at each of the on-ramps. As with all prior RM implementations, the effectiveness of the kNN-TD algorithm was investigated in unit intervals for target densities ranging from 24 veh/km to 34 veh/km when applied to the RM problem at the R300 on-ramp. As may be seen in Table 10.10, this initial investigation indicated that the smallest TTS-value is achieved when employing a target density of 28 veh/km. Hence, the unit interval around 28 veh/km was investigated more closely in steps of 0.1 veh/km. The results of this finer investigation indicated that setting the target density to 28 veh/km indeed resulted in the best performance. Therefore, the target density for the kNN-TD RM agent at the R300 on-ramp is set to 28 veh/km for all further investigations and comparisons in this chapter.

TABLE 10.10: *Parameter evaluation results for the kNN-TD RM implementation at the R300 on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	27.0	27.5	27.6	27.7	27.8	27.9	28.0
—	2 293.59	2 557.23	2 588.20	1 859.09	2 567.00	2 617.21	1 814.58
	Target density $\hat{\rho}$						
Combination	28.1	28.2	28.3	28.4	28.5	29.0	
—	2 411.512	2 005.02	2 555.28	2 510.05	2 563.21	2 510.83	

Once the best-performing target density for the R300 on-ramp was found, target density values for the Brackenfell Boulevard on-ramp were investigated. This investigation was, again, performed in unit intervals for densities between 24 veh/km and 34 veh/km for the RM agent at the Brackenfell Boulevard on-ramp alone, as well as for the combination of RM agents at the R300 and Brackenfell Boulevard on-ramps, as may be seen in Table 10.11. From the results in the table it is evident that the agent at the Brackenfell Boulevard on-ramp consistently performed better alone than when combined with the R300 on-ramp RM agent, achieving the smallest TTS-value at a target density of 25 veh/km. The unit interval around 25 veh/km was then investigated more closely in increments of 0.1 veh/km. As may be seen in the table, the best-performing target density for the RM agent at the Brackenfell Boulevard on-ramp was 24.9 veh/km. Due to the fact, however, that the RM agent at the R300 achieved a smaller TTS value by itself, the *k*NN-TD RM agent at the Brackenfell Boulevard on-ramp is not considered for further comparisons in this chapter.

TABLE 10.11: *Parameter evaluation results for the kNN-TD RM implementation at the Brackenfell Boulevard on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	24	24.5	24.6	24.7	24.8	24.9	25.0
Alone	2 041.58	2 039.37	2 047.73	2 007.43	2 166.07	2 006.28	2 015.04
R300	2 364.12	—	—	—	—	—	2 332.25
	Target density $\hat{\rho}$						
Combination	25.1	25.2	25.3	25.4	25.5	26	
Alone	1 856.57	1 907.27	1 905.38	1 961.11	1 879.07	2 100.56	
R300	—	—	—	—	—	2 390.10	

Finally, the parameter evaluation concluded with an investigation of the best-performing target density at the Okavango Road on-ramp. Target densities between 24 veh/km and 34 veh/km were yet again investigated for the scenario where there is only an RM agent at the Okavango on-ramp as well as for the scenario where there are RM agents at both the Okavango Road and R300 on-ramps; these results are shown in Table 10.12. As may be seen in the table, the results of this initial investigation suggested that the combination of the RM agents at the R300 and Okavango Road on-ramp performed better than the single RM agent at the Okavango Road on-ramp. The initial investigation, however, also revealed that the best-performing target density at the Okavango Road on-ramp was 34 veh/km, and as a result, the target densities 35 veh/km and 36 veh/km were also investigated, achieving TTS-values of 1 819.38 veh·h and 1 960.77 veh·h, respectively. Subsequently, the unit interval around the target density of 35 veh/km was considered in 0.1 veh/km increments. The results of this investigation indicated that the best-

performing combination of kNN-TD RM agents in the case study area is an RM agent at the R300 on-ramp, with a target density of 28 veh/km, and an RM agent at the Okavango Road on-ramp with a target density of 35.5 veh/km. This parameter combination is used in all further comparisons involving kNN-TD RM agents conducted in this chapter.

TABLE 10.12: *Parameter evaluation results for the kNN-TD RM implementation at the Okavango Road on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	24	24.5	24.6	24.7	24.8	24.9	25.0
Alone	2 043.28	—	—	—	—	—	2 021.37
R300	1 892.96	1 798.74	1 841.81	1 856.54	1 794.58	1 939.92	1 819.38
	Target density $\hat{\rho}$						
Combination	35.1	35.2	35.3	35.4	35.5	35.6	36
Alone	—	—	—	—	—	—	2 065.49
R300	1 863.70	1 812.85	1 901.62	1 774.78	1 768.29	1 876.64	1 960.77

10.1.3 Algorithmic Comparison

The p-values of the ANOVA and Levene statistical tests conducted on the PMI-values returned by the RM algorithms are presented in Table 10.13. The ANOVA test revealed that there are, in fact, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all PMIs, except the maximum TISBB. Furthermore, Levene's test revealed that the variances of the PMI-values returned by the algorithms were statistically indistinguishable for the TTS, TTSR300, mean TISR300 and maximum TISR300 PMIs. Therefore, the Fisher LSD test was performed in order to ascertain between which pairs of algorithmic outputs significant differences occur in respect of these PMIs. The Games-Howell test was performed for this purpose in respect of all the other PMIs (except for the maximum TISBB, of course).

TABLE 10.13: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests associated with RM. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 960.01	1 881.50	1 851.21	1 822.85	1 768.29	4.1611×10^{-2}	3.7681×10^{-1}
TTSN1	884.11	926.03	930.27	904.21	606.44	1.7764×10^{-15}	2.6224×10^{-6}
TTSR300	992.19	838.96	811.55	823.43	1 014.18	8.6241×10^{-6}	9.8702×10^{-1}
TTSBB	69.71	67.74	72.71	73.56	59.69	7.7642×10^{-3}	3.1779×10^{-3}
TTSO	14.00	48.06	35.78	20.80	86.27	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISN1 Mean	1.24	1.29	1.31	1.27	0.89	1.1102×10^{-16}	2.4680×10^{-6}
TISN1 Max	5.30	8.69	11.16	7.94	3.88	8.3908×10^{-8}	5.2898×10^{-7}
TISR300 Mean	8.84	7.49	7.17	7.26	14.32	$< 1 \times 10^{-17}$	3.6410×10^{-1}
TISR300 Max	25.42	22.16	23.28	21.68	42.03	$< 1 \times 10^{-17}$	3.2675×10^{-1}
TISBB Mean	2.01	1.98	2.08	2.10	1.72	6.9534×10^{-3}	5.3302×10^{-3}
TISBB Max	5.05	4.98	4.77	5.06	4.46	2.8878×10^{-1}	9.5416×10^{-2}
TISO Mean	0.82	2.81	2.11	1.23	5.03	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISO Max	1.50	12.37	6.89	5.42	18.43	$< 1 \times 10^{-17}$	2.2877×10^{-9}

As may be seen in the box plots in Figure 10.2(a), all of the RM implementations were able to achieve smaller mean TTS-values than the no-control case. These findings are corroborated

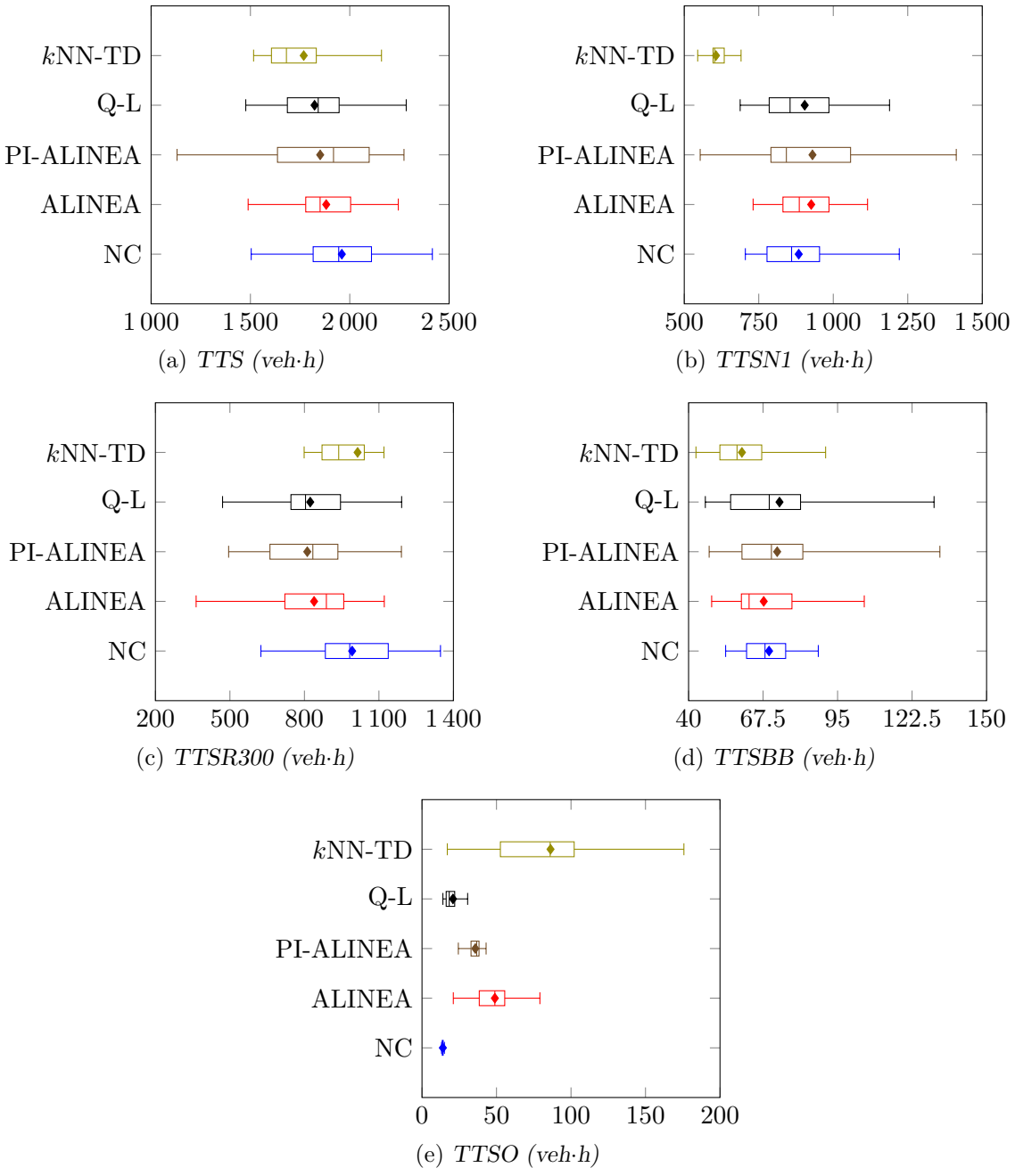


FIGURE 10.2: *Total time spent in the system PMI results for the no-control case (NC), the ALINEA control strategy, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm in the case of RM applied to the case study model of Chapter 9.*

by the results presented in Table 10.14. The kNN-TD RM and Q-Learning implementations, achieving 9.78% and 7.00% improvements over the no-control case, respectively, returned the best performance, outperforming the no-control case, while their performances were found to be statistically indistinguishable from that of ALINEA and PI-ALINEA at a 5% level of significance. ALINEA and PI-ALINEA were also found to perform on par statistically at a 5% level of significance. Although ALINEA and PI-ALINEA were able to achieve reductions of 4.01% and 5.55% over the no-control case, respectively, in respect of the TTS, this improvement was not large enough for these algorithms to be classified as statistically distinguishable from the no-control case at a 5% level of significance. This ordering of the relative algorithmic performances in respect of the TTS is also evident in the box plots of Figure 10.2(a).

In respect of the TTSN1, kNN-TD again returned the best performance, achieving a TTSN1-value of 606.44 veh·h, thereby outperforming the no-control case, Q-Learning, ALINEA and PI-ALINEA at a 5% level of significance, as may be inferred from the p-values in Table 10.15. Interestingly, the no-control case returned the second-best performance, achieving a TTSN1-value of 884.11 veh·h. The performances of the no-control case, ALINEA, PI-ALINEA and Q-Learning were found to be statistically indistinguishable at a 5% level of significance, as ALINEA, PI-ALINEA and Q-Learning returned TTSN1-values of 926.03 veh·h, 930.27 veh·h and 904.21 veh·h, respectively. This order of relative algorithmic performances is also evident from the box plots of Figure 10.2b.

Interestingly, the ordering of relative algorithmic performances in respect of the TTSR300 is almost exactly the opposite of that for the TTSN1, as may be seen from the box plots in Figure 10.2(c). PI-ALINEA returned the smallest TTSR300-value, achieving an 18.21% improvement over the no-control case, thereby outperforming both the no-control case and kNN-TD RM, while it was found to perform statistically on par with ALINEA and Q-Learning at a 5% level of significance, as may be seen from the p-values in Table 10.16. PI-ALINEA was followed by Q-Learning and ALINEA, achieving 17.00% and 15.44% improvements over the no-control case, respectively, also outperforming both the no-control case and kNN-TD RM at a 5% level of significance. Finally, kNN-TD RM returned a 2.22% increase in the TTSR300 when compared to the no-control case. This increase was not, however, large enough for the algorithmic performances to be classified as statistically distinguishable at a 5% level of significance. An increase in travel times for the kNN-TD RM implementation was, however, to be expected, as the kNN-TD RM implementation was the only RM implementation in which RM is applied at the R300 on-ramp.

In respect of the TTSBB, kNN-TD again returned the best performance, achieving a TTSBB-value of 59.69 veh·h and outperforming PI-ALINEA, Q-Learning and the no-control case at a 5% level of significance, while its performance was found to be statistically indistinguishable from that of ALINEA, as may be deduced from the p-values in Table 10.17. ALINEA takes second place in the order of relative algorithmic performances, having achieved a TTSBB-value of 64.71 veh·h. Although this value is smaller than the 69.71 veh·h achieved by the no-control case, the 73.56 veh·h returned by Q-Learning and the 72.71 veh·h achieved by PI-ALINEA, these four performances were found to be statistically indistinguishable at a 5% level of significance. This order of relative algorithmic performances is also clear in the box plots of Figure 10.2(d).

As may have been expected, due to the fact that all four RM implementations employ an RM agent at the Okavango Road on-ramp, the no-control case returned the smallest TTSO-value, outperforming all of the RM implementations at a 5% level of significance, as may be seen from the p-values in Table 10.18. As is also evident from the box plots in Figure 10.2(e), Q-Learning achieved a smaller TTSO-value than any of the other RM implementations, outperforming ALINEA, PI-ALINEA and kNN-TD RM at a 5% level of significance. PI-ALINEA outperformed

both ALINEA and kNN-TD RM, at a 5% level of significance, while ALINEA was able to outperform kNN-TD RM, which returned the largest TTSO-value.

TABLE 10.14: *Differences in respect of the total time spent in the system (TTS) by all vehicles in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.1499×10^{-1}	8.6510×10^{-2}	3.1195×10^{-2}	2.7965×10^{-3}
ALINEA		—	6.3166×10^{-1}	3.5375×10^{-1}	7.4597×10^{-2}
PI-ALINEA			—	6.5344×10^{-1}	1.9043×10^{-1}
Q-Learning				—	3.8817×10^{-1}
Mean	1 960.01	1 881.50	1 851.21	1 822.85	1 768.29

TABLE 10.15: *Differences in respect of the total time spent in the system by vehicles entering the system from the N1 (TTSN1) in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSN1			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	7.9648×10^{-1}	8.4832×10^{-1}	9.8506×10^{-1}	9.1958×10^{-12}
ALINEA		—	9.9999×10^{-1}	9.8473×10^{-1}	2.8819×10^{-11}
PI-ALINEA			—	9.8385×10^{-1}	2.6781×10^{-8}
Q-Learning				—	6.3810×10^{-10}
Mean	884.11	926.03	930.27	904.21	606.44

TABLE 10.16: *Differences in respect of the total time spent in the system by vehicles entering the system from the R300 (TTSR300) in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTSR300			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.4003^{-3}	3.7498×10^{-4}	8.6136×10^{-4}	6.5824×10^{-1}
ALINEA		—	5.8135×10^{-1}	7.5459×10^{-1}	5.5107×10^{-4}
PI-ALINEA			—	8.1108×10^{-1}	7.2385×10^{-5}
Q-Learning				—	1.7899×10^{-4}
Mean	992.19	838.96	811.55	823.42	1 014.18

TABLE 10.17: *Differences in respect of the total time spent in the system by vehicles entering the system from Brackenfell Boulevard (TTSBB) in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSBB			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.7071×10^{-1}	9.2787×10^{-1}	9.1740×10^{-1}	4.6529×10^{-3}
ALINEA		—	7.5327×10^{-1}	7.6514×10^{-1}	1.1689×10^{-1}
PI-ALINEA			—	9.9985×10^{-1}	1.1474×10^{-2}
Q-Learning				—	3.8544×10^{-2}
Mean	69.71	67.74	72.71	73.56	59.69

In a trend similar to that in respect of the TTSN1, kNN-TD RM achieved the best performance in respect of both the mean and maximum TISN1, outperforming all other algorithms

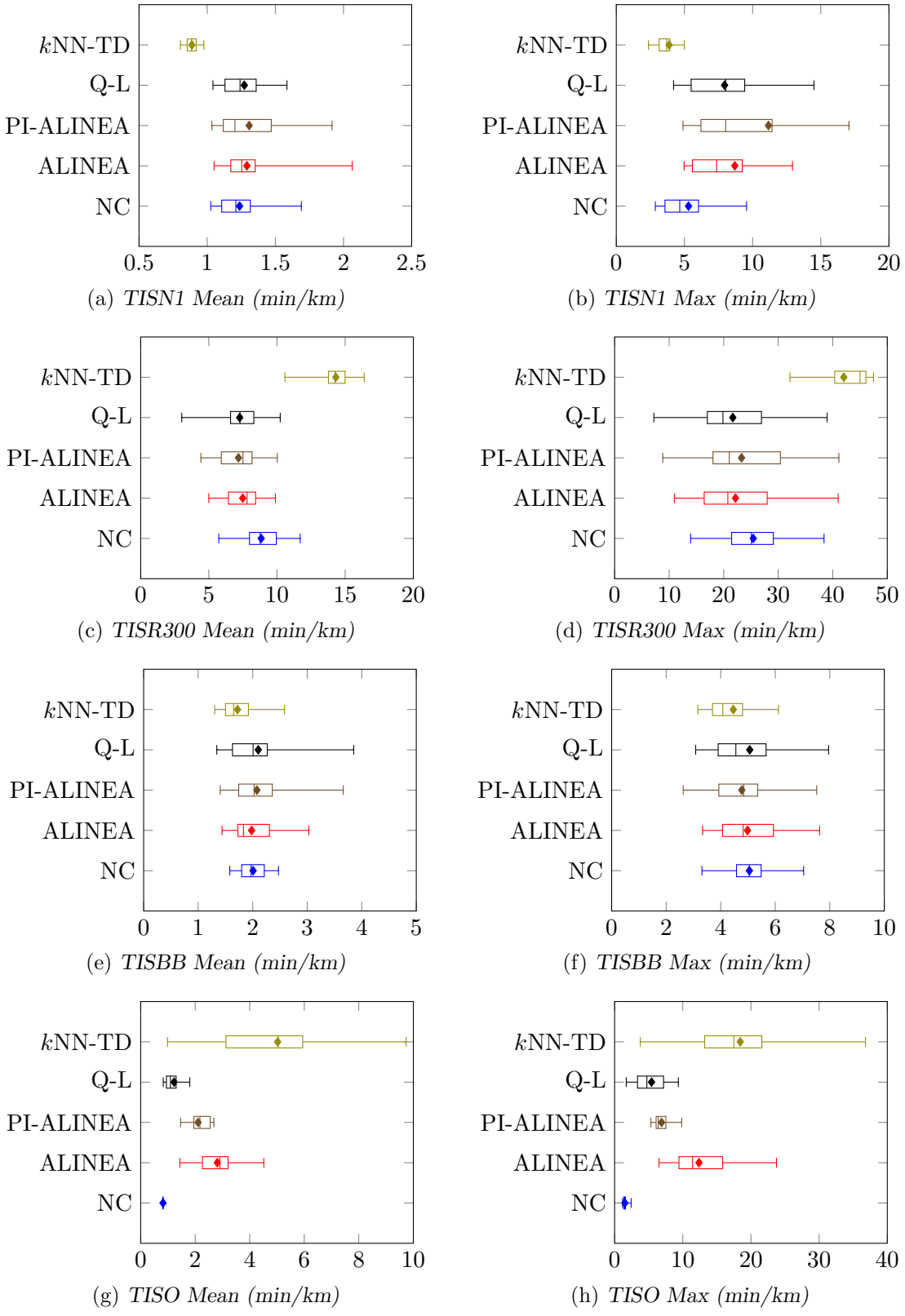
TABLE 10.18: *Differences in respect of the total time spent in the system by vehicles entering the system from Okavango Road (TTSO) in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSO			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.0891×10^{-13}	4.9294×10^{-14}	4.1273×10^{-4}	5.7257×10^{-10}
ALINEA		—	1.9032×10^{-4}	2.9092×10^{-12}	1.2816×10^{-4}
PI-ALINEA			—	5.4877×10^{-11}	9.5037×10^{-7}
Q-Learning				—	4.0105×10^{-9}
Mean	14.00	48.06	35.78	20.80	86.27

at a 5% level of significance in respect of both of these PMIs, as may be inferred from the p-values in Tables 10.19 and 10.20. The no-control case returned the second-best performance, outperforming ALINEA, PI-ALINEA and Q-Learning at a 5% level of significance in respect of the maximum TISN1, while the performances of the no-control case, ALINEA, PI-ALINEA and Q-Learning were found to be statistically indistinguishable from one another in repsect of the mean TISN1. Although Q-Learning achieved a smaller mean TISN1-value and a smaller maximum TISN1-value than both ALINEA and PI-ALINEA, these algorithms were found to perform statistically indistinguishably in respect of both these PMIs. This order of algorithmic performances is clearly visible in the box plots of Figures 10.3(a) and 10.3(b) corresponding to the mean and maximum TISN1, respectively.

As may also have been expected, the order of relative algorithmic performance in respect of both the mean and maximum TISR300 PMIs is similar to that for the TTSR300. PI-ALINEA achieved the smallest mean TISR300-value of 7.17 min/km, thereby outperforming the no-control case and kNN-TD RM, while its performance was found to be statistically indistinguishable from that of ALINEA and Q-Learning at a 5% level of significance, as may be deduced from the p-values in Table 10.21. ALINEA and Q-Learning were, however, also both able to outperform both the no-control case and kNN-TD RM at a 5% level of significance, as they achieved mean TISR300-values of 7.49 minutes and 7.26 minutes, respectively. Due to the fact that RM is applied at the R300 on-ramp in the kNN-TD RM implementation, the no-control case outperformed kNN-TD RM at a 5% level of significance in respect of the mean TISR300, as these implementations returned values of 8.84 minutes and 14.32 minutes, respectively. This ordering of the relative algorithmic performances is also evident from the box plots in Figure 10.3(c). A similar trend emerged in respect of the maximum TISR300. Q-Learning, which achieved the smallest maximum TISR300-value of 21.68 minutes, was able to outperform both the no-control case and kNN-TD RM at a 5% level of significance, as may be seen in Table 10.22. The no-control case, ALINEA and PI-ALINEA, which returned maximum TISR300-values of 25.42 minutes, 22.16 minutes and 23.28 minutes, were found to perform statistically on par, while all outperforming kNN-TD RM at a 5% level of significance, for which a maximum TISR300-value of 42.03 minutes was recorded. This order of relative algorithmic performances is again evident in the box plots of Figure 10.3d.

In respect of the mean TISBB, kNN-TD returned the best performance, outperforming PI-ALINEA, Q-Learning and the no-control case at a 5% level of significance, as may be seen from the p-values in Table 10.23. The performances of the kNN-TD RM and ALINEA implementations were found to be statistically indistinguishable at a 5% level of significance as they returned mean TISBB-values of 1.72 min/km and 1.98 min/km, respectively. Furthermore, ALINEA, PI-ALINEA, the no-control case and Q-Learning were all found to perform statistically on par at



(a) *TISN1 Mean (min/km)* (b) *TISN1 Max (min/km)*

(c) *TISR300 Mean (min/km)* (d) *TISR300 Max (min/km)*

(e) *TISBB Mean (min/km)* (f) *TISBB Max (min/km)*

(g) *TISO Mean (min/km)* (h) *TISO Max (min/km)*

FIGURE 10.3: *Mean and maximum time spent in the system PMI results for the no-control case (NC), the ALINEA control strategy, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm in the case of RM applied to the case study model of Chapter 9.*

a 5% level of significance. This closeness in performance of the RM algorithms is also evident from the box plots in Figure 10.3(e). Interestingly, PI-ALINEA and Q-Learning resulted in an increase in the mean TISBB, as they achieved values of 2.08 min/km and 2.10 min/km, respectively, compared to the 2.01 min/km returned by the no-control case. This increase in respect of the mean TISBB may be attributed to a significant increase in the variances of the results, as may be seen in Figure 10.3(e).

As may have been expected, the no-control case achieved the smallest mean and maximum TISO-values due to the fact that RM is applied at the Okavango Road on-ramp in all of the algorithmic implementations, thus outperforming all of the RM implementations at a 5% level of significance in respect of both of these PMIs. This is evident from the p-values in Tables 10.24 and 10.25. Q-Learning returned the second-best performance in respect of both of these PMIs, as it returned mean and maximum TISO-values of 1.23 min/km and 5.42 min/km, respectively, thereby outperforming ALINEA, PI-ALINEA and kNN-TD RM at a 5% level of significance in respect of the mean TISO. In respect of the maximum TISO, Q-Learning was again able to outperform ALINEA and kNN-TD RM, while its performance was found to be statistically indistinguishable from that of PI-ALINEA. In respect of the mean TISO, Q-Learning was followed in the order of relative algorithmic performance by PI-ALINEA, which was able to outperform ALINEA and kNN-TD RM at a 5% level of significance, as these implementations achieved values of 2.11 min/km, 2.81 min/km and 5.03 min/km, respectively. Similarly, PI-ALINEA was able to outperform both ALINEA and kNN-TD RM in respect of the maximum TISO, as these algorithms achieved values of 6.89 min/km, 12.37 min/km and 18.43 min/km, respectively. Finally, ALINEA was able to outperform kNN-TD RM at a 5% level of significance in respect of both the mean and maximum TISO. This ordering of the relative algorithmic performances is very clear in the box plots of Figures 10.3(g) and 10.3(h).

TABLE 10.19: *Differences in respect of the mean time spent in the system by vehicles entering the system from the N1 in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISN1 Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	7.8481×10^{-1}	7.2183×10^{-1}	9.4927×10^{-1}	1.0259×10^{-11}
ALINEA		—	9.9855×10^{-1}	9.9592×10^{-1}	2.1390×10^{-11}
PI-ALINEA			—	9.4959×10^{-1}	7.2422×10^{-9}
Q-Learning				—	1.2759×10^{-10}
Mean	1.24	1.29	1.31	1.27	0.89

TABLE 10.20: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the N1 in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISN1 Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.1059×10^{-2}	8.4514×10^{-3}	1.6416×10^{-3}	4.5598×10^{-2}
ALINEA		—	6.5037×10^{-1}	9.4738×10^{-1}	8.5456×10^{-4}
PI-ALINEA			—	3.1022×10^{-1}	6.5913×10^{-4}
Q-Learning				—	5.2876×10^{-8}
Mean	5.30	8.69	11.16	7.94	3.88

TABLE 10.21: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISR300 Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.1187×10^{-4}	2.2487×10^{-5}	5.7545×10^{-5}	$< 1 \times 10^{-17}$
ALINEA		—	4.0962×10^{-1}	5.5569×10^{-1}	$< 1 \times 10^{-17}$
PI-ALINEA			—	8.1351×10^{-1}	$< 1 \times 10^{-17}$
Q-Learning				—	$< 1 \times 10^{-17}$
Mean	8.84	7.49	7.17	7.26	14.32

TABLE 10.22: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the R300 in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISR300 Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	8.2383×10^{-2}	2.5249×10^{-1}	4.6543×10^{-2}	1.8874×10^{-15}
ALINEA		—	5.4928×10^{-1}	7.9641×10^{-1}	$< 1 \times 10^{-17}$
PI-ALINEA			—	3.9192×10^{-1}	$< 1 \times 10^{-17}$
Q-Learning				—	$< 1 \times 10^{-17}$
Mean	25.42	22.16	23.28	21.68	42.03

TABLE 10.23: *Differences in respect of the mean time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISBB Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.9781×10^{-1}	9.5141×10^{-1}	9.3889×10^{-1}	1.8497×10^{-3}
ALINEA		—	9.1052×10^{-1}	8.9741×10^{-1}	5.6620×10^{-2}
PI-ALINEA			—	9.9980×10^{-1}	8.3833×10^{-3}
Q-Learning				—	3.4199×10^{-2}
Mean	2.01	1.98	2.08	2.10	1.72

TABLE 10.24: *Differences in respect of the mean time spent in the system by vehicles entering the system from Okavango Road in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.8089×10^{-14}	2.2200×10^{-15}	2.5471×10^{-4}	2.3104×10^{-10}
ALINEA		—	1.1943×10^{-4}	$< 1 \times 10^{-17}$	7.3348×10^{-5}
PI-ALINEA			—	3.5804×10^{-11}	5.0639×10^{-7}
Q-Learning				—	1.7030×10^{-9}
Mean	0.82	2.81	2.11	1.23	5.03

10.1.4 Discussion

As was the case for the results obtained in the context of the benchmark simulation model of §5.1.2, kNN-TD RM was again able to achieve the largest reduction in respect of the TTS in

TABLE 10.25: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Okavango Road in the case of RM. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	3.0919×10^{-13}	1.1136×10^{-13}	2.4357×10^{-6}	2.5918×10^{-11}
ALINEA		—	1.2579×10^{-6}	3.5705×10^{-8}	6.5147×10^{-3}
PI-ALINEA			—	1.5303×10^{-1}	1.1145×10^{-7}
Q-Learning				—	6.3673×10^{-9}
Mean	1.50	12.37	6.89	5.42	18.43

the context of the case study model of Chapter 9. The kNN-TD RM implementation was, in fact, able to achieve the smallest value for seven of the thirteen PMIs as it was able to reduce the travel times of vehicles along the N1 and those entering the system from the Brackenfell Boulevard on-ramp. Apart from the reduction in travel times along the N1, the traffic flow along the N1 was also the most stable when applying kNN-TD RM, as may be seen from the small interquartile ranges in the box plots of Figures 10.11(a) and 10.3(b). Therefore, kNN-TD RM is considered to be the best-performing algorithm in the context of this case study. The relatively poor performance in respect of the other six PMIs may be attributed to the fact that it was the only implementation in which RM was employed at two of the three on-ramps, resulting in natural increases in travel times of vehicles entering the system from those on-ramps. Interestingly, kNN-TD RM performed significantly worse than the other RM implementations in respect of the travel times of vehicles entering the system from the Okavango Road on-ramp. The results showed that these travellers may experience substantial variances in their travel times which may make accurate time allocation and route planning difficult.

The results returned by the ALINEA, PI-ALINEA and Q-Learning implementations showed that the Okavango Road on-ramp is the best location at which to employ RM in the case study area. Of the three algorithms, Q-Learning was able to achieve the largest reduction in respect of the TTS. Apart from the larger reduction in the TTS achieved by Q-Learning, the performances of the algorithms were similar as they managed to achieve similar PMI-values in respect of the TTSN1, TTSR300 and TTSBB. Most notably, however, Q-Learning was able to achieve a significantly better trade-off between protecting the highway flow and maintaining an acceptable on-ramp queue length at the Okavango Road on-ramp than both ALINEA and PI-ALINEA. This may be favourable in a real-world implementation as it prevents too much of the traffic flow from the Okavango Road on-ramp spilling back into the arterial network. Therefore, Q-Learning is considered to be the best performing of these three algorithms.

Similarly to Q-Learning, PI-ALINEA was also able to achieve a significantly better trade-off between balancing the on-ramp queue length and protecting the highway flow at the Okavango Road on-ramp than ALINEA, while their performances in respect of all other PMIs were relatively similar, except in respect of the maximum TISN1, where PI-ALINEA performed significantly worse than ALINEA. Due to the fact, however, that PI-ALINEA was able to achieve a better balance between limiting the on-ramp queue length at the Okavango Road on-ramp and protecting the highway flow along the N1 than ALINEA, as well as achieveing a smaller TTS-value than ALINEA, PI-ALINEA is considered to be the better performing of these algorithms.

10.2 Ramp Metering with Queue Limits

As expected, RM again resulted in the formation of long on-ramp queues, especially at the Okavango Road on-ramp. This is particularly problematic due to the fact that the on-ramp at the Okavango Road interchange connects the N1 highway to the urban arterial network, and not to another highway, as is the case at the R300 on-ramp. The on-ramp at the Okavango Road is approximately 400 metres long, allowing 400 metres of space for a queue to form. Taking into account that the length of the passenger vehicles (which make up the largest percentage of the traffic flow) is set to 5 metres, and assuming headways of 1–1.5 metres between vehicles in the queue, the maximum allowable on-ramp queue length $\hat{w}$ was set to 50 vehicles in order to prevent the spill back of the on-ramp queue into the arterial network.

10.2.1 Algorithmic Implementations

Perhaps surprisingly, a closer investigation into the formation of the on-ramp queues at the Okavango Road on-ramp revealed that the limit of 50 vehicles was not exceeded in the PI-ALINEA and Q-Learning implementations, which exhibited the smallest increases in the TTSO, and mean and maximum TISO PMIs. Therefore, this queue limit was only enforced in the ALINEA and kNN-TD RM implementations. In the ALINEA implementation, the queue length restriction of Smaragdis and Papageorgiou [150] in (6.9) was again employed to ensure that the queue length does not exceed the maximum of 50 vehicles, while for the kNN-TD RM implementation, the adjusted reward function, punishing the RM agent for queue lengths longer than 50 vehicles as in (6.11), was adopted for the Okavango Road RM agent. Furthermore, due to the fact that no statistical differences were found between the performance of the kNN-TD RM implementation and the no-control case in respect of the TTSR300, it was not deemed necessary to apply a further queue limitation at the R300 on-ramp, seeing that the increase in the travel times brought about by vehicles joining the N1 from the R300 was not large enough for the algorithmic performances to be statistically distinguishable at a 5% level of significance on a system level. Furthermore, due to the fact that the R300 is also a highway, which ends at the interchange with the N1, it is expected that long on-ramp queues, when they do form, will not have the same detrimental effect on the arterial network as in the case of the Okavango Road on-ramp. The impact of the introduction of queue limitations on the performances of ALINEA and kNN-TD RM are summarised in Table 10.26.

TABLE 10.26: *The effect of employing queue limitations in the RM implementations on their overall performance in the case study.*

	ALINEA		kNN-TD	
PMI	$\hat{w} = 50$	$\hat{w} = \infty$	$\hat{w} = 50$	$\hat{w} = \infty$
TTS (veh·h)	1 928.00	1 881.50	1 750.33	1 768.29
TTSN1 (veh·h)	903.15	926.03	614.63	606.44
TTSR300 (veh·h)	924.61	838.96	1 056.11	1 014.18
TTSBB (veh·h)	67.31	67.74	60.76	59.69
TTSO (veh·h)	32.01	48.06	17.62	86.27

As may be seen in the table, the queue limit at the Okavango Road on-ramp did result in significant decreases in the total time spent in the system by vehicles joining the N1 from the Okavango Road on-ramp, as expected. In the case of ALINEA, this did, however, lead to significant increases in the TTSR300, as these vehicles travelling along the N1 did not enjoy the

same level of protected highway flow while passing the Okavango Road interchange due to the limited RM. This, in turn resulted in a significant increase in the TTS as well. From the results in the table it is evident that the queue limitation in the kNN-TD RM implementation was even more effective in reducing the TTSO than in the ALINEA implementation. This decrease in the TTSO was large enough that, although there were increases in the TTSN1, TTSR300 and TTSBB, a decrease in respect of the TTS was nevertheless recorded.

10.2.2 Algorithmic Comparison

The results from the ANOVA performed in respect of RM with queue limits, presented in Table 10.27, revealed that as for the implementations without queue limits, statistical differences again exist at a 5% level of significance between at least some pair of algorithmic output in respect of all PMIs except the maximum TISBB. Furthermore, the Levene test revealed that the variances of the algorithmic output are only statistically indistinguishable at a 5% level of significance in respect of the TTS, TTSR300 and mean TISR300, while statistical differences between the variances of at least some pair of algorithmic outputs exist in respect of all other PMIs. The Fisher LSD *post hoc* test was therefore performed in order to ascertain between which pairs of algorithmic output these differences occur in respect of the TTS, TTSR300 and mean TISR300, while the Games-Howell test was performed for this purpose in respect of all other PMIs (except the maximum TISBB, of course).

TABLE 10.27: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests associated with RM with queue limits. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

PMI	Mean value					p-value	
	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 960.01	1 928.00	1 851.21	1 822.85	1 750.33	6.5220×10^{-3}	2.4540×10^{-1}
TTSN1	884.11	903.15	930.27	904.21	614.63	3.3307×10^{-15}	4.9570×10^{-7}
TTSR300	992.19	924.61	811.55	823.43	1 056.11	8.1139×10^{-7}	9.6834×10^{-1}
TTSBB	69.71	67.31	72.71	73.56	60.76	1.5977×10^{-2}	3.7541×10^{-4}
TTSO	14.00	32.01	35.78	20.80	17.62	$< 1 \times 10^{-17}$	3.4253×10^{-8}
TISN1 Mean	1.24	1.26	1.31	1.27	0.90	2.2204×10^{-16}	6.0723×10^{-7}
TISN1 Max	5.30	8.79	11.16	7.94	4.70	2.5054×10^{-7}	4.0296×10^{-7}
TISR300 Mean	8.84	8.13	7.17	7.26	14.77	$< 1 \times 10^{-17}$	1.0094×10^{-1}
TISR300 Max	25.42	24.58	23.28	21.68	42.28	$< 1 \times 10^{-17}$	1.7470×10^{-2}
TISBB Mean	2.01	1.95	2.08	2.10	1.73	5.6960×10^{-3}	3.7072×10^{-4}
TISBB Max	5.05	4.48	4.77	5.06	4.50	1.3350×10^{-1}	2.3202×10^{-2}
TISO Mean	0.82	1.87	2.11	1.23	1.04	$< 1 \times 10^{-17}$	6.7172×10^{-9}
TISO Max	1.50	8.71	6.89	5.42	6.41	$< 1 \times 10^{-17}$	9.0261×10^{-14}

Even with the addition of a queue limit, the kNN-TD RM implementation again returned the best performance in respect of the TTS, achieving a TTS-value of 1$\dot{7}$50.33 veh·h, and outperforming ALINEA and the no-control case at a 5% level of significance, while its performance was found not to differ statistically from that of Q-Learning and PI-ALINEA. As may be seen from the p-values in Table 10.28, the kNN-TD RM implementation was followed in the order of relative algorithmic performances by Q-Learning, which achieved a TTS-value of 1 822.85 veh·h, also outperforming the no-control case, while its performance was found to be statistically on par with those of ALINEA and PI-ALINEA, which achieved TTS-values of 1 928.00 veh·h and 1 851.21 veh·h, respectively. Although ALINEA and PI-ALINEA were both able to reduce the TTS when compared with the no-control case, which returned a TTS-value of 1 960.01 veh·h, these differences were not large enough for these algorithmic performances to be classified as statistically different at a 5% level of significance. This ordering of the relative algorithmic performances is also evident in the box plots of Figure 10.4(a).

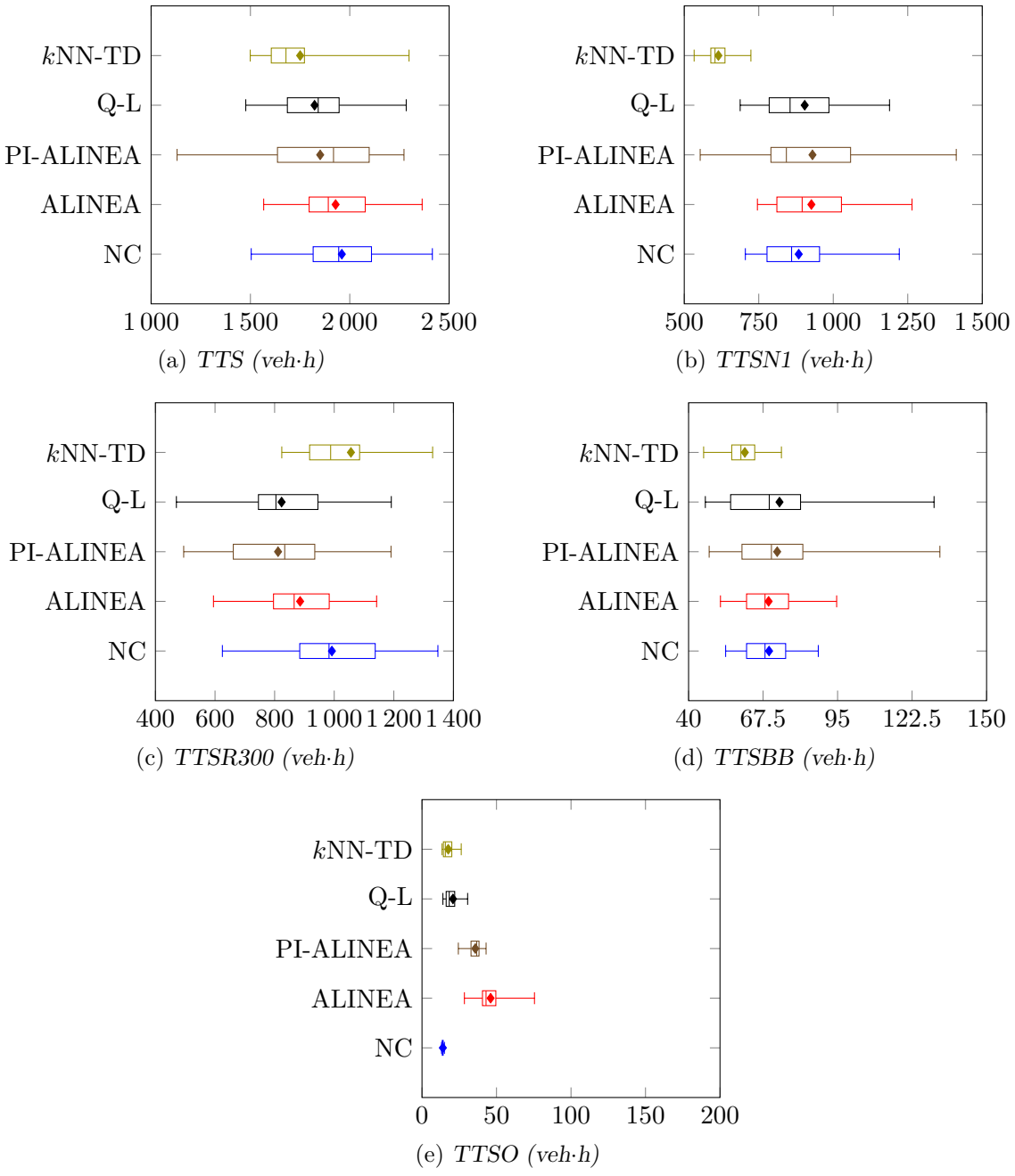


FIGURE 10.4: *Total time spent in the system PMI results for the no-control case (NC), the ALINEA control strategy, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm in the case of RM with queue limits applied to the case study model of Chapter 9.*

As expected, due to the increased protection of the highway flow resulting from the fact that kNN-TD RM is the only implementation in which RM is applied at both the R300 and Okavango Road on-ramps, kNN-TD RM achieved the smallest TTSN1-value — a 30.48% improvement over the no-control case, and outperforming all other algorithms at a 5% level of significance, as may be seen in Table 10.29. ALINEA, PI-ALINEA and Q-Learning all resulted in marginal increases of 2.15%, 5.22% and 2.27%, respectively, over the no-control case in respect of the TTSN1. These increases were, however, not large enough for the performances of these algorithms to be classified as statistically distinguishable at a 5% level of significance. This similarity in the performances of all implementations, except kNN-TD RM is also evident in the box plots of Figure 10.4(b).

Interestingly, ALINEA, PI-ALINEA and Q-Learning were all able to achieve improvements over the no-control case in respect of the TTSR300. This may be attributed to the fact that these vehicles, many of which travel along the N1 once they have joined from the R300 reap the benefits of the RM applied at the Okavango Road on-ramp. These improvements may also be seen in the box plots of Figure 10.4(c). From the results of the Fisher LSD test performed in respect of the TTSR300, presented in Table 10.30, it is evident that PI-ALINEA and Q-Learning achieved the best performance, as they outperformed the no-control case, ALINEA and kNN-TD RM at a 5% level of significance, while their performances were found to be statistically indistinguishable. ALINEA was also able to outperform kNN-TD RM, while its performance was found to be statistically on par with that of the no-control case at a 5% level of significance. Finally, the order of relative algorithmic performances is completed by kNN-TD RM, which as the only implementation in which RM is applied at the R300, resulted in an increase in the TTSR300. This increase was, however, as in the case without queue limits, not large enough for its performance to be classified as statistically different from that of the no-control case at a 5% level of significance.

The kNN-TD RM implementation again returned the best performance in respect of the TTSBB, outperforming the no-control case and PI-ALINEA at a 5% level of significance. This improvement may again be attributed to better traffic flow along the N1 as RM is applied at two on-ramps. Furthermore, as may be seen in the box plots of Figure 10.4(d), the traffic flow in the case of kNN-TD RM is more stable than in the PI-ALINEA and Q-Learning implementations, as may be deduced from the significantly smaller variances corresponding to the kNN-TD RM implementation. The means of the no-control case, ALINEA, PI-ALINEA and Q-Learning perform very similarly, as may be seen in the figure. This is corroborated by the p-values in Table 10.31, from which it is evident that the performances of these four implementations are statistically indistinguishable at a 5% level of significance.

As expected, the no-control case achieved the smallest TTSO-value of 14.00 veh·h, outperforming all of the RM implementations at a 5% level of significance, even with queue limits in place. The no-control case is followed in the order of relative algorithmic performances by Q-Learning and kNN-TD RM, which achieved TTSO-values of 20.80 veh·h and 17.62 veh·h, respectively, outperforming both ALINEA and PI-ALINEA in respect of the TTSO, while their performances were found to be statistically indistinguishable at a 5% level of significance. Finally, ALINEA and PI-ALINEA returned TTSO-values of 32.01 veh·h and 35.78 veh·h, respectively, as their performances were also found to be statistically indistinguishable from one another at a 5% level of significance, as may be seen from the p-values in Table 10.32. This order of relative algorithmic performances is also very clear in the box plots of Figure 10.4(e).

In respect of the mean and maximum TISN1 PMIs, kNN-TD RM was again the only implementation which was able to achieve smaller values than the no-control case, as may clearly be seen in the box plots of Figures 10.5(a) and 10.5(b). Furthermore, the variances in respect of both

TABLE 10.28: *Differences in respect of the total time spent in the system (TTS) by all vehicles in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	6.0322×10^{-1}	7.8739×10^{-2}	2.7141×10^{-2}	8.3498×10^{-4}
ALINEA		—	2.1345×10^{-1}	8.9184×10^{-2}	4.4257×10^{-3}
PI-ALINEA			—	6.4508×10^{-1}	1.0280×10^{-1}
Q-Learning				—	2.3984×10^{-1}
Mean	1 960.01	1 928.00	1 851.21	1 822.85	1 750.33

TABLE 10.29: *Differences in respect of the total time spent in the system by vehicles entering the system from the N1 (TTSN1) in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSN1			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.7876×10^{-1}	8.4832×10^{-1}	9.8506×10^{-1}	1.5508×10^{-11}
ALINEA		—	9.7410×10^{-1}	9.9999×10^{-1}	3.9380×10^{-13}
PI-ALINEA			—	9.8385×10^{-1}	4.4416×10^{-8}
Q-Learning				—	1.1044×10^{-9}
Mean	884.11	903.15	930.27	904.21	614.63

TABLE 10.30: *Differences in respect of the total time spent in the system by vehicles entering the system from the R300 (TTSR300) in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTSR300			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	1.6735^{-1}	2.9552×10^{-4}	6.9696×10^{-4}	1.9149×10^{-1}
ALINEA		—	2.1658×10^{-2}	3.9513×10^{-2}	7.7582×10^{-3}
PI-ALINEA			—	8.0770×10^{-1}	1.4840×10^{-6}
Q-Learning				—	4.3015×10^{-6}
Mean	992.19	924.61	811.55	823.42	1 056.11

TABLE 10.31: *Differences in respect of the total time spent in the system by vehicles entering the system from Brackenfell Boulevard (TTSBB) in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSBB			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.5267×10^{-1}	9.2787×10^{-1}	9.1740×10^{-1}	2.7231×10^{-3}
ALINEA		—	7.2085×10^{-1}	7.3464×10^{-1}	2.5169×10^{-1}
PI-ALINEA			—	9.9985×10^{-1}	1.4125×10^{-2}
Q-Learning				—	5.0666×10^{-2}
Mean	69.71	67.31	72.71	73.56	60.76

the mean and maximum TISN1-values were significantly smaller when applying kNN-TD RM than in any of the other implementations, as indicated by the smaller interquartile ranges of the corresponding box plots. These improvements are corroborated by the p-values in Tables 10.33 and 10.34, as kNN-TD RM was the only implementation able to outperform the no-control case

TABLE 10.32: *Differences in respect of the total time spent in the system by vehicles entering the system from Okavango Road (TTSO) in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

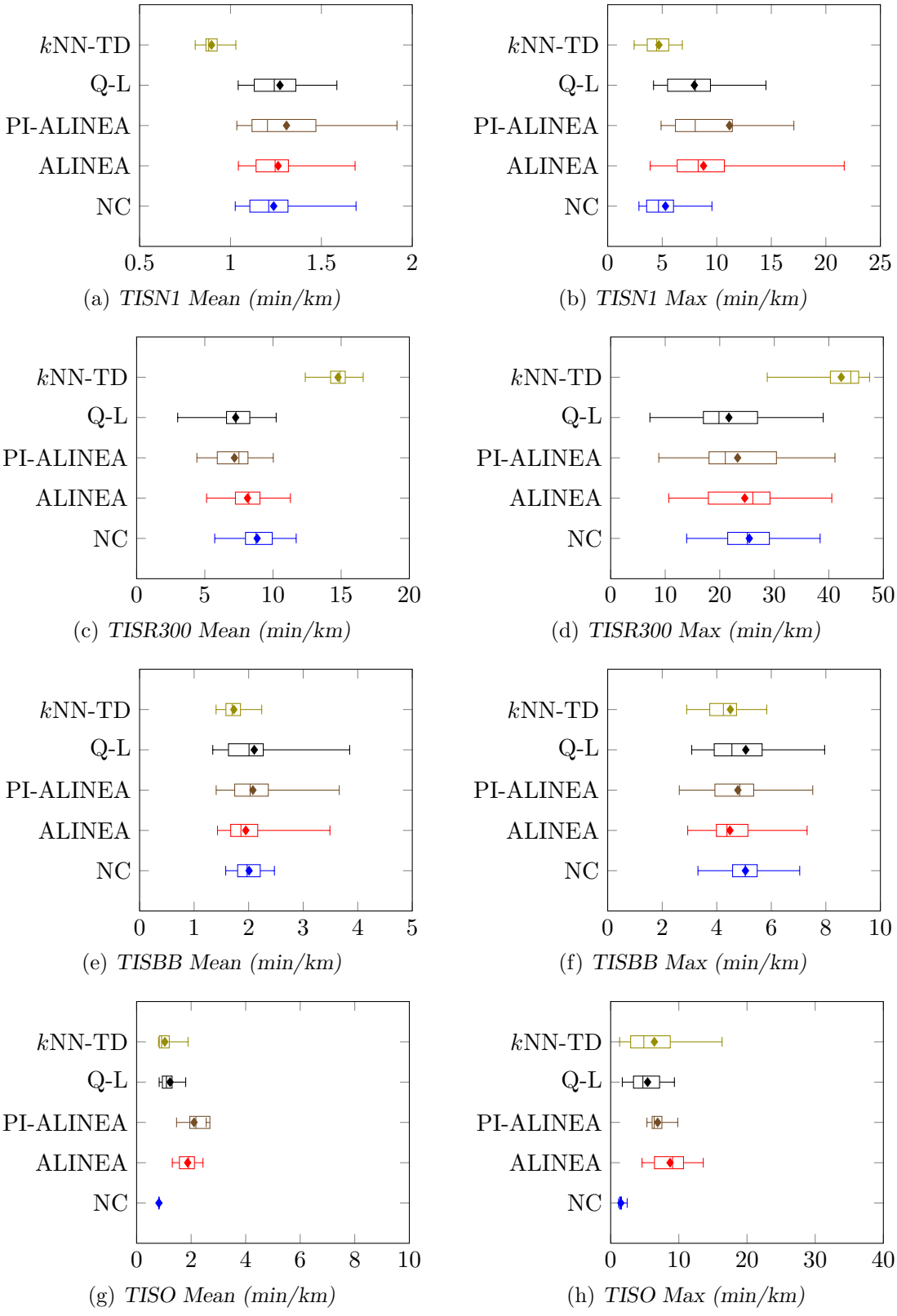
		Games-Howell test p-values: TTSO			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.6423×10^{-15}	4.9294×10^{-15}	4.1273×10^{-4}	9.4946×10^{-4}
ALINEA		—	6.8445×10^{-2}	1.1232×10^{-6}	$< 1 \times 10^{-17}$
PI-ALINEA			—	5.4877×10^{-11}	1.3703×10^{-11}
Q-Learning				—	3.0687×10^{-1}
Mean	14.00	32.01	35.78	20.80	17.62

in respect of the mean TISN1, while it was also the only implementation not outperformed by the no-control case in respect of the maximum TISN1. The performances of ALINEA, PI-ALINEA and Q-Learning, on the other hand, were found to be statistically indistinguishable from one another at a 5% level of significance in respect of both of these PMIs, while they performed statistically on-par with the no-control case in respect of the mean TISN1 and were outperformed by the no-control case in respect of the maximum TISN1.

As expected, the kNN-TD RM implementation yielded the largest mean and maximum TISR300-values due to the fact that RM is applied at the R300, and was thus outperformed in respect of both these PMIs at a 5% level of significance by all other algorithms as well as the no-control case. ALINEA, PI-ALINEA and Q-Learning, on the other hand, were all able to reduce the mean TISR300 when compared with the no-control case. PI-ALINEA and Q-Learning returned the best performance, outperforming the no-control case and ALINEA at a 5% level of significance in respect of the mean TISR300, while the performances of ALINEA and the no-control case were found to be statistically indistinguishable from one another at a 5% level of significance, as is evident from the p-values in Table 10.35. In respect of the maximum TISR300, ALINEA, PI-ALINEA and Q-Learning were again able to achieve improvements over the no-control case, but these improvements were not large enough to outperform the no-control case or each other at a 5% level of significance, as may be deduced from the p-values in Table 10.36. These trends in respect of the relative algorithmic performances are also evident in the box plots of Figures 10.5(c) and 10.5(d).

As may be inferred from the p-values in Table 10.37, kNN-TD RM achieved the smallest mean TISBB-value, outperforming all other implementations, except for ALINEA, at a 5% level of significance. This improvement by kNN-TD RM may largely be attributed to a reduction in the variance of the mean TISBB, as may be seen from the small interquartile range of the corresponding box plot in Figure 10.5(e). A similar reduction in the variance may be seen in respect of the maximum TISBB in Figure 10.5(f), although the performances of all algorithms were found to be statistically indistinguishable at a 5% level of significance in respect of the maximum TISBB, as is evident from the results of the ANOVA in Table 10.27. Similarly, the performances of ALINEA, PI-ALINEA and Q-Learning were also found to be statistically indistinguishable at a 5% level of significance from one another and the no-control case in respect of the mean TISBB, as may be deduced from the p-values in Table 10.37.

Naturally, the no-control case returned the smallest mean and maximum TISO-values of 0.82 min/km and 1.50 min/km, respectively, outperforming all four RM implementations at a 5% level of significance, as may be seen from the p-values in Tables 10.38 and 10.39. From the box plots in Figure 10.5(g), it is evident that the no-control case is followed in the order of relative algorithmic performacnes by Q-Learning and kNN-TD RM, which returned mean TISO-



(a) *TISN1 Mean (min/km)* (b) *TISN1 Max (min/km)*

(c) *TISR300 Mean (min/km)* (d) *TISR300 Max (min/km)*

(e) *TISBB Mean (min/km)* (f) *TISBB Max (min/km)*

(g) *TISO Mean (min/km)* (h) *TISO Max (min/km)*

FIGURE 10.5: *Mean and maximum time spent in the system PMI results for the no-control case (NC), the ALINEA control strategy, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm in the case of RM with queue limits applied to the case study model of Chapter 9.*

values of 1.23 min/km and 1.04 min/km respectively, outperforming both ALINEA and PI-ALINEA, while their performances were statistically indistinguishable. ALINEA was finally able to outperform PI-ALINEA at a 5% level of significance in respect of the mean TISO. As may be seen in Figure 10.5(h), a similar trend emerged in respect of the maximum TISO. Q-Learning again returned the best performance of the RM implementations, achieving a maximum TISO-value of 5.42 min/km, outperforming ALINEA while its performance was found to be statistically on par with that of PI-ALINEA and kNN-TD RM, which returned maximum TISO-values of 6.89 min/km and 6.41 min/km, respectively. As may be seen in Table 10.39, PI-ALINEA and kNN-TD RM were also able to outperform ALINEA, which achieved the largest maximum TISO-value of 8.71 min/km, at a 5% level of significance.

TABLE 10.33: *Differences in respect of the mean time spent in the system by vehicles entering the system from the N1 in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISN1 Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.7727×10^{-1}	7.2183×10^{-1}	9.4927×10^{-1}	1.3704×10^{-11}
ALINEA		—	9.2014×10^{-1}	9.9947×10^{-1}	1.7208×10^{-13}
PI-ALINEA			—	9.7496×10^{-1}	1.0364×10^{-8}
Q-Learning				—	1.7814×10^{-10}
Mean	1.24	1.26	1.31	1.27	0.90

TABLE 10.34: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the N1 in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISN1 Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	3.6448×10^{-4}	8.4514×10^{-3}	1.6416×10^{-3}	7.8569×10^{-1}
ALINEA		—	6.3082×10^{-1}	8.3973×10^{-1}	9.3920×10^{-6}
PI-ALINEA			—	3.1022×10^{-1}	2.2798×10^{-3}
Q-Learning				—	1.2496×10^{-5}
Mean	5.30	8.78	11.16	7.94	4.70

TABLE 10.35: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISR300 Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	5.8756×10^{-2}	1.4236×10^{-5}	3.7948×10^{-5}	$< 1 \times 10^{-17}$
ALINEA		—	1.0630×10^{-2}	2.0340×10^{-2}	$< 1 \times 10^{-17}$
PI-ALINEA			—	8.0885×10^{-1}	$< 1 \times 10^{-17}$
Q-Learning				—	$< 1 \times 10^{-17}$
Mean	8.84	8.13	7.17	7.26	14.77

10.2.3 Discussion

Unlike in the benchmark model of §5.1.2, the best-performing RM implementation with queue limits in the case study model of Chapter 9 was the kNN-TD RM implementation. The kNN-TD

TABLE 10.36: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the R300 in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISR300 Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.9158×10^{-1}	7.8870×10^{-1}	2.6388×10^{-1}	$< 1 \times 10^{-17}$
ALINEA		—	9.6833×10^{-1}	5.9770×10^{-1}	$< 1 \times 10^{-17}$
PI-ALINEA			—	9.3156×10^{-1}	$< 1 \times 10^{-17}$
Q-Learning				—	$< 1 \times 10^{-17}$
Mean	25.42	24.58	23.28	21.68	42.28

TABLE 10.37: *Differences in respect of the mean time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISBB Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	9.5829×10^{-1}	9.5141×10^{-1}	9.3889×10^{-1}	1.6289×10^{-4}
ALINEA		—	7.7856×10^{-1}	7.8580×10^{-1}	7.8221×10^{-2}
PI-ALINEA			—	9.9980×10^{-1}	4.8226×10^{-3}
Q-Learning				—	2.7735×10^{-2}
Mean	2.01	1.95	2.08	2.10	1.73

TABLE 10.38: *Differences in respect of the mean time spent in the system by vehicles entering the system from Okavango Road in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Mean			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	6.1060×10^{-15}	2.2200×10^{-15}	2.5471×10^{-4}	8.3369×10^{-4}
ALINEA		—	2.6476×10^{-2}	7.8243×10^{-7}	$< 1 \times 10^{-17}$
PI-ALINEA			—	3.5804×10^{-11}	9.4494×10^{-12}
Q-Learning				—	2.7481×10^{-1}
Mean	0.82	1.87	2.11	1.23	1.04

TABLE 10.39: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Okavango Road in the case of RM with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Max			
Algorithm	No Control	ALINEA	PI-ALINEA	Q-Learning	kNN-TD
No Control	—	2.5868×10^{-14}	1.1135×10^{-13}	2.3568×10^{-6}	4.5555×10^{-6}
ALINEA		—	8.2343×10^{-3}	4.6256×10^{-4}	9.1453×10^{-3}
PI-ALINEA			—	1.5303×10^{-9}	9.7359×10^{-1}
Q-Learning				—	8.3927×10^{-1}
Mean	1.50	8.71	6.89	5.42	6.41

RM implementation was, in fact, able to achieve the smallest value for nine of the thirteen PMIs when queue limits are implemented. As stated above, the largest effect of the addition of queue limits on the kNN-TD RM implementation was that the queue length at the Okavango Road on-ramp was controlled very effectively, reducing the mean travel times of the vehicles joining the

N1 from the Okavango Road on-ramp from the largest value in the case where queue limits are not applied, to the smallest value when queue limits are, in fact, implemented, while maintaining the good performances achieved for vehicles travelling along the N1 only and vehicles joining the N1 from the Brackenfell Boulevard on-ramp.

Q-Learning maintained its second place in the overall order of algorithmic performances. The performance of Q-Learning remained unchanged due to the fact that the implementation of an additional queue restriction was not required. Similarly, PI-ALINEA retained its third place in the order of relative algorithmic performances, as the performance of ALINEA worsened with the addition of the queue limitation. Although the implementation of the queue limit was effective in reducing the travel times of vehicles joining the N1 from the Okavango Road on-ramp in the ALINEA implementation, in respect of which ALINEA now achieved smaller values than PI-ALINEA, this improvement had a negative effect, especially on the travel times of vehicles joining the N1 from the R300, as these vehicles no longer experienced the same level of protected traffic flow along the highway. As a result, an increase in the TTS was also observed. ALINEA was therefore the worst-performing RM implementation, while still achieving improvements over the no-control case in respect of the TTS, TTSR300 and TTSBB PMIs.

10.3 Variable Speed Limits

This section is devoted to a description of the parameter evaluation and algorithmic performance comparison performed in respect of the VSL implementations within the case study simulation model of Chapter 9. A parameter evaluation is first performed in respect of the MTFC, Q-Learning and kNN-TD VSL implementations with the aim of finding the best-performing target density in respect of the MTFC implementations and the best update rule for adjusting the upstream speed limits in respect of the RL implementations, as well as the best-performing combination of VSL agents in the case study area. Once these parameter combinations have been determined, the relative algorithmic performances are compared. The results of this comparison are presented and interpreted by means of box plots in which the means, medians and interquartile ranges of the PMIs of §9.3 are indicated, as well as tables indicating whether or not statistical differences exist between the PMI-values returned by each pair of algorithms at a 5% level of significance.

10.3.1 Algorithmic Implementations

MTFC for VSLs may be implemented on the sections of highway directly upstream of the three expected bottlenecks at the R300, Brackenfell Boulevard and Okavango Road on-ramps. The implementations at each of these on-ramps take the same form as that in Chapter 7. The lengths of the application and acceleration areas are again set to 100 metres and 175 metres, respectively, while the downstream density is measured at the bottleneck on the section where the highway and on-ramp traffic flows merge. Similarly to the implementation in the benchmark model of §5.1.2, VSLs are again employed every 100 metres upstream of the application area in order to smoothe the transition from the nominal speed limit of 120 km/h down to the VSL determined according to (3.32) and (3.33), each indicating a speed limit that is 10 km/h faster than the next VSL displayed directly downstream. Finally, the value of the nonnegative controller parameter K_I is retained at 0.005, which was found to yield the best performance in the parameter evaluation in §7.5.1.

As in the MTFC implementations, RL VSL agents may be implemented at each of the three interchanges in the case study area, as shown in Figure 10.6. As may be seen in the figure, two VSLs, namely VSL_{R_1} and VSL_R, are applied before the bottleneck at the R300 on-ramp. VSL_{R_1} is applied from the start of the simulated area at O_1 up to directly after the R300 off-ramp, which leads to D_1. Thereafter, VSL_R is applied until the R300 on-ramp. As there is only a single section of highway between the R300 on-ramp and the Brackenfell Boulevard on-ramp, only a single VSL, namely VSL_B, is applied on this section, ahead of the expected bottleneck at the Brackenfell Boulevard on-ramp. After the Brackenfell Boulevard on-ramp, the first of the VSLs corresponding to the agent located at the Okavango Road on-ramp, namely VSL_{O_1}, is applied. This speed limit is enforced until directly after the Okavango Road off-ramp which leads to D_2. After the off-ramp at the Okavango Road interchange, VSL_O is applied up to the section directly after the Okavango Road on-ramp, at which point the normal speed limit of 120 km/h is restored.

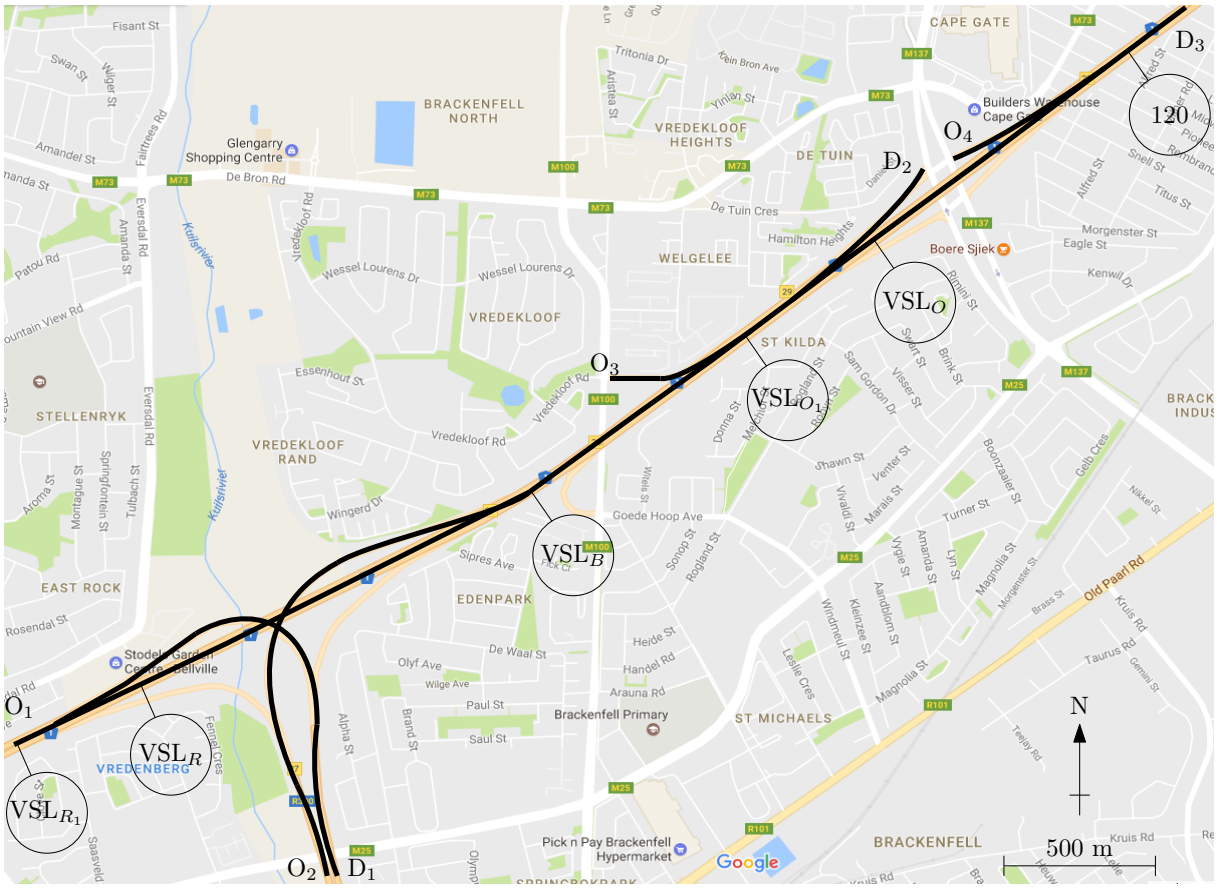


FIGURE 10.6: *The locations at which VSLs (indicated by the speed limit signs) are applied in the case study area.*

As was the case for the RM agents, the state spaces for the VSL agents remain unchanged for the case study implementation. The first VSL agent is focused on the bottleneck created at the R300 on-ramp. The downstream density is therefore taken as the density at the section where the on-ramp and highway traffic flows merge. The application density is the density on the section between the R300 off-ramp and the R300 on-ramp, where VSL_R is applied. Finally, the upstream density is measured on the section of the N1 before the R300 off-ramp, where VSL_{R_1} is applied. Employing the same action space as in the VSL implementation of Chapter 7, VSL_R is adjusted according to (7.1), while VSL_{R_1} is adjusted according to either (7.2) or (7.4). Finally,

the reward function in (7.3) is employed, where q denotes the outflow out of the bottleneck location (*i.e.* q is the flow measured directly downstream of the lane merge at the R300 on-ramp and the N1 highway).

The downstream density for the VSL agent corresponding to the Brackenfell Boulevard on-ramp is, as for the R300 VSL agent, measured at the section where the traffic flows from the Brackenfell Boulevard on-ramp and the N1 merge. The application density is measured on the section of the N1 between the R300 on-ramp and the Brackenfell Boulevard on-ramp, where VSL_B is applied. Finally, although the Brackenfell Boulevard VSL agent does not alter the speed limit on the section of the N1 between the R300 off-ramp and the R300 on-ramp, the density of this section is included in the agents' state space as the upstream density. The agent adjusts the speed limit VSL_B according to (7.1). Finally, the agent is rewarded according to (7.3), where q again denotes the flow of vehicles on the N1 directly after the lane merge of the Brackenfell Boulevard on-ramp and the N1 highway.

For the VSL agent located at the Okavango Road interchange, the downstream density is again measured on the section of highway where the on-ramp and highway traffic flows merge. The application density is measured on the section of the N1 between the Okavango Road off-ramp and the Okavango Road on-ramp, while the upstream density is measured on the section of the N1 between the Brackenfell Boulevard on-ramp and the Okavango Road off-ramp. VSL_O is adjusted according to (7.1), while VSL_{O_1} is again adjusted according to either (7.2), or (7.4). As was the case for both the R300 and Brackenfell Boulevard VSL agents, the Okavango Road agent is rewarded according to (7.3), where q denotes the flow along the N1 after the traffic flows from the Okavango Road on-ramp and the N1 highway have merged.

10.3.2 Parameter Evaluations

This section is devoted to a parameter evaluation with the aim of finding the best-performing target densities for the MTFC implementations as well as the best-performing update rule for both VSL_{R_1} and VSL_{O_1} in respect of both the Q-Learning and kNN-TD VSL implementations, as measured by the total time spent in the system by all vehicles. Another aim in this section is to find the best-performing combinations of VSL agents in the case study area.

MTFC parameter evaluation

Due to the fact that the MTFC feedback controller by Müller *et al.* [105] is density-based, the same step-wise approach as employed for determining the best-performing target densities in the RM implementations was employed for the MTFC controller in the case study. The results of the initial parameter evaluation of target densities between 24 veh/km and 34 veh/km indicated that if an MTFC controller is employed only at the expected bottleneck at the R300 on-ramp, the best-performing target density is 33 veh/km. The results of the finer investigation of the unit interval around 33 veh/km are shown in Table 10.40. As may be seen in the table, setting the target density to 33.2 veh/km yielded the smallest TTS-value. The target density is therefore set to 33.2 veh/km in all further comparisons conducted in this chapter where an MTFC feedback controller is employed at the expected bottleneck at the R300 on-ramp.

In respect of the feedback controller implemented to address the expected bottleneck at the Brackenfell Boulevard on-ramp, two cases were again considered. In the first case, MTFC was only employed at the Brackenfell Boulevard on-ramp merge, while in the second case, MTFC was employed at both the Brackenfell Boulevard and R300 on-ramps. As may be seen from the

TABLE 10.40: *Parameter evaluation results for the MTFC VSL implementation at the R300 on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	32.0	32.5	32.6	32.7	32.8	32.9	33.0
—	3 082.14	3 062.83	3 045.29	3 135.07	3 106.26	3 094.67	3 021.51
	Target density $\hat{\rho}$						
Combination	33.1	33.2	33.3	33.4	33.5	34.0	
—	3 055.49	2 961.65	3 076.66	3 034.40	3 042.10	3 038.58	

results in Table 10.41, employing MTFC only at the Brackenfell Boulevard on-ramp consistently resulted in smaller TTS-values than the combined case. As may be seen in the table, the finer investigation around the target density of 34 veh/km indicated that setting the target density to 34.4 veh/km yielded the best performance.

TABLE 10.41: *Parameter evaluation results for the MTFC VSL implementation at the Brackenfell Boulevard on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	33	33.5	33.6	33.7	33.8	33.9	34.0
Alone	4 346.74	4 313.02	4 342.66	4 369.54	4 385.65	4 418.21	4 274.09
R300	4 437.18	—	—	—	—	—	4 382.55
	Target density $\hat{\rho}$						
Combination	34.1	34.2	34.3	34.4	34.5	35	
Alone	4 327.29	4 255.91	4 248.24	4 184.24	4 254.01	4 378.89	
R300	—	—	—	—	—	4 375.24	

Due to the fact that MTFC at the Brackenfell Boulevard on-ramp consistently performed worse than MTFC at the R300 on-ramp, the two combinations considered in respect of MTFC at the Okavango Road on-ramp entail either employing MTFC only at the expected bottleneck corresponding to the Okavango Road on-ramp merge, or employing MTFC at both the R300 on-ramp and the Okavango Road on-ramp merges. From the results of the initial rough parameter evaluation it is evident that employing MTFC only at the bottleneck corresponding to the Okavango Road on-ramp merge consistently yields smaller TTS-values. As a result, the finer parameter evaluation around the previously determined best-performing target density of 37 veh/km was performed for the case where MTFC is only applied before the Okavango Road on-ramp. The results of this parameter evaluation are presented in Table 10.42. As may be seen in the table, the smallest TTS-value is achieved when setting the target density to 37 veh/km. As a result, a target density of 37 veh/km for MTFC at the bottleneck before the Okavango Road on-ramp is employed for all further comparisons involving MTFC in this chapter.

Q-Learning parameter evaluation

The parameter evaluations performed to determine whether a suitable value for the variable δ in (7.2), or the expression in (7.4) should be used, were again conducted adopting a stepwise approach. The parameter evaluation aimed at determining the speed limit VSL_{R_1} (applied before the R300 off-ramp) was conducted first. As was the case in the parameter evaluation conducted in Chapter 7, three cases were considered. In the first case, the variable δ in (7.2)

TABLE 10.42: *Parameter evaluation results for the MTFC VSL implementation at the Okavango Road on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	36	36.5	36.6	36.7	36.8	36.9	37.0
Alone	1 953.80	1 924.11	1 924.89	1 930.41	1 899.84	1 916.05	1 863.78
R300	2 965.60	—	—	—	—	—	3 118.94
	Target density $\hat{\rho}$						
Combination	37.1	37.2	37.3	37.4	37.5	38	
Alone	1 879.69	1 878.31	1 922.56	1 886.96	1 880.18	1 938.98	
R300	—	—	—	—	—	2 960.40	

is assigned a value of 10, while in the second case $\delta = 20$. In the third case, the expression in (7.4) is employed. Once the best-performing expression had been determined for VSL_{R_1}, the focus shifted to VSL_B. Due to the fact that the VSL agent at the Brackenfell Boulevard on-ramp only adjusts a single speed limit value, the parameter evaluation conducted in respect of this agent was only aimed at determining whether the agent would work more effectively by itself, or in combination with the VSL agent at the R300 on-ramp (for which the parameter evaluation had already been completed). Finally, in the case of the VSL agent at the Okavango Road intersection, the various expressions for adjusting VSL_{O_1} were again investigated. This parameter evaluation was performed for two different scenarios — one in which only the VSL agent at the Okavango Road intersection is employed, and one where this agent is employed together with the VSL agents at the R300 and Brackenfell Boulevard on-ramps, respectively. The results of this parameter evaluation are summarised in Table 10.43.

TABLE 10.43: *Parameter evaluation results for VSLs using the Q-Learning algorithm, measured as the TTS by the vehicles (in veh·h).*

	R300	Brackenfell		Okavango	
δ		Alone	Combined	Alone	Combined
Case 1	1 845.00	1 973.64	1 833.53	1 924.49	1 925.68
Case 2	1 893.61	—	—	1 895.81	1 851.46
Case 3	1 866.05	—	—	1 846.61	1 917.65

As may be seen in the table, the parameter evaluation performed in respect of the VSL agent at the R300 interchange revealed that setting $\delta = 10$ yielded the best performance. As a result, a value of $\delta = 10$ was henceforth employed when determining VSL_{R_1}. It was furthermore found that employing a VSL agent at both the R300 interchange and the Brackenfell Boulevard on-ramp resulted in additional performance improvements. As may be seen from the results presented in Table 10.43, this was also the best-performing combination of Q-Learning VSL agents. Therefore, this is the combination of Q-Learning VSL agents employed in all further comparisons conducted in this chapter.

*k*NN-TD parameter evaluation

The parameter evaluation for the *k*NN-TD VSL agents was conducted in the same manner as for the Q-Learning VSL agents. The results of this parameter evaluation are presented in

Table 10.44. As may be seen in the table, for the parameter evaluation performed in the case of the VSL agent at the R300 interchange, setting $\delta = 20$ when determining VSL_{R_1} yielded the best performance. Furthermore, incorporating a kNN-TD VSL agent at the Brackenfell Boulevard did not yield additional improvements. Therefore, when the parameter evaluation was performed for the Okavango Road VSL agent, the combination only included the R300 interchange VSL agent. As may be seen in Table 10.44, the combination of the R300 and Okavango Road interchanges with δ-values of 20 and 10, respectively, resulted in the smallest TTS-values. As a result, this parameter value combination is used in all further comparisons conducted in this chapter.

TABLE 10.44: *Parameter evaluation results for VSLs using the kNN-TD algorithm, measured as the TTS by the vehicles (in veh·h).*

	R300	Brackenfell		Okavango	
δ		Alone	Combined	Alone	Combined
Case 1	1 965.70	1 901.08	1 854.25	1 937.58	1 840.83
Case 2	1 846.14	—	—	1 858.56	1 954.29
Case 3	1 874.76	—	—	1 925.21	1 852.09

10.3.3 Algorithmic Comparison

As may be deduced from the p-values of the ANOVA and Levene statistical tests conducted on the PMI-values returned by the VSL implementations, presented in Table 10.45, the ANOVA revealed that there are differences between at least some pair of algorithmic output data in respect of the TTSN1, TTSR300, TTSO, mean and maximum TISN1, mean TISR300 and mean TISO PMIs at a 5% level of significance, while the algorithms were found to perform statistically similarly in respect of the other PMIs. Furthermore, Levene's test revealed that the variances of the algorithmic output data sets are statistically different at a 5% level of significance in respect of the TTSN1, TTSBB, and mean and maximum TISN1 PMIs. Hence the Games-Howell test was employed in order to determine between which pairs of algorithmic output the differences in respect of the TTSN1, mean and maximum TISN1 PMIs occur, while the Fisher LSD test was employed for this purpose in respect of the TTSR300, TTSO, mean TISR300 and mean TISO PMIs.

As may be seen from the results of the ANOVA in Table 10.45, none of the RL VSL implementations were able to outperform the no-control case in respect of the TTS at a 5% level of significance. Although no statistically significant differences between any of the algorithimc implementations were found at a 5% level of significance, Q-Learning for VSLs achieved an improvement of 6.45% over the no-control case, while kNN-TD was able to achieve a 6.08% improvement over the no-control case in respect of the TTS. MTFC, on the other hand, was able to achieve an improvement of only 4.91% over the no-control case. Based on the smaller interquartile ranges of the box plots corresponding to the VSL implementations in Figure 10.7(a), one may argue that the reduction in the TTS-values is due to reduced variances, which may provide evidence of the homogenisation effect of VSLs on traffic flow, although the variances were shown not to be statistically different at a 5% level of significance by the Levene test.

In respect of the TTSN1, Q-Learning achieved a 7.35% improvement over the no-control case, while the kNN-TD implementation achieved an improvement of only 0.79%. For the MTFC implementations, however, an increase in the TTSN1 of 6.53% was recorded. As may be seen from Table 10.46, all three VSL implementations performed statistically indistinguishably at a

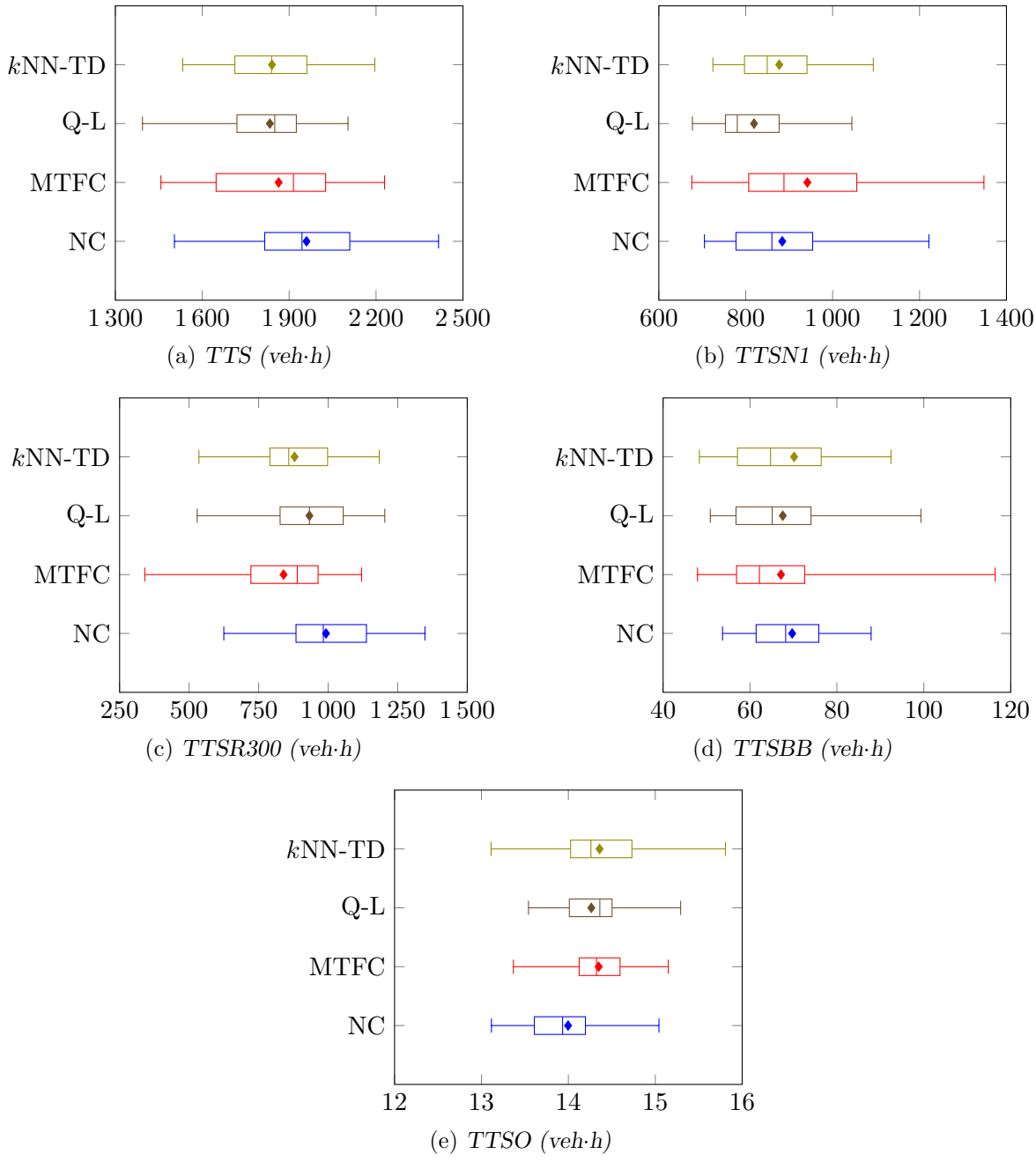


FIGURE 10.7: *Total time spent in the system PMI results for the no-control case (NC), the Q-Learning algorithm (Q-L) and the kNN-TD algorithm in the case of VSLs applied to the case study model of Chapter 9.*

TABLE 10.45: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests associated with VSLs. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value				*p*-value	
PMI	No Control	MTFC	Q-Learning	*k*NN-TD	ANOVA	Levene's Test
TTS	1 960.01	1 863.78	1 833.53	1 840.825	6.9075×10^{-2}	2.3765×10^{-1}
TTSN1	884.11	941.84	819.15	877.13	7.9256×10^{-3}	1.2113×10^{-2}
TTSR300	992.19	839.63	932.57	879.18	3.4349×10^{-3}	7.6323×10^{-1}
TTSBB	69.71	67.13	67.55	70.15	8.1634×10^{-1}	1.5800×10^{-2}
TTSO	14.00	14.35	14.26	14.36	3.0718×10^{-2}	5.5998×10^{-1}
TISN1 Mean	1.24	1.31	1.15	1.22	5.7799×10^{-3}	7.0542×10^{-3}
TISN1 Max	5.30	9.76	6.16	5.41	4.5454×10^{-6}	2.0942×10^{-4}
TISR300 Mean	8.84	7.46	8.27	7.82	1.3916×10^{-3}	8.2245×10^{-1}
TISR300 Max	25.42	21.93	24.49	23.79	3.3313×10^{-1}	5.4789×10^{-1}
TISBB Mean	2.01	1.94	1.96	2.02	8.4542×10^{-1}	1.5605×10^{-1}
TISBB Max	5.05	4.70	4.78	5.08	6.6221×10^{-1}	1.3972×10^{-1}
TISO Mean	0.82	0.84	0.82	0.82	$< 1 \times 10^{-17}$	2.5264×10^{-1}
TISO Max	1.50	1.53	1.45	1.45	4.3896×10^{-1}	3.5244×10^{-1}

5% level of significance from the no-control case in respect of the TTSN1. Q-Learning, which achieved the smallest TTSN1-value, was, however, able to outperform MTFC at a 5% level of significance, which returned the largest TTSN1-value, while the performances of all other algorithmic implementations were also found to be statistically indistinguishable. Based on the box plots in Figure 10.7(b), one may again argue that there was homogenisation of traffic flow on the N1 in the case of the RL VSL implementations, as the RL VSL implementations again achieved smaller interquartile ranges than the no-control case and the MTFC implementation. These decreases in variances by the RL VSL implementations were confirmed by the Levene test.

In respect of the TTSR300, MTFC was able to outperform the no-control case and Q-Learning at a 5% level of significance, while its performance was found to be statistically indistinguishable from that of *k*NN-TD for VSLs, as may be deduced from the *p*-values in Table 10.47. The *k*NN-TD implementation was also able to outperform the no-control case, while its performance was found not to differ statistically from that of Q-Learning. This similarity between the performances of *k*NN-TD and Q-Learning for VSLs is clearly visible in the box plots of Figure 10.7(c). Interestingly, as may be seen in Figure 10.7(c), there seems to have been a definitive improvement in respect of the TTSR300 by all three VSL implementations, instead of an improvement that may be the result of homogenisation of traffic flow only. A possible explanation for this phenomenon in respect of the RL implementations is that due to the VSLs enforced, the merging of traffic flows from the R300 occurs more smoothly, resulting in the corresponding reduction in the TTSR300.

As may have been expected, the VSLs had little effect on the total time spent in the system by vehicles joining the N1 traffic flow from the Brackenfell Boulevard on-ramp. This expectation is confirmed by the results of the ANOVA, which revealed that the algorithmic performances of all implementations were statistically indistinguishable at a 5% level of significance in respect of the TTSBB. Interestingly, as may be seen in Figure 10.7(d), an increase was observed in the variance of the output data generated by all three of the VSL implementations in respect of the TSBB. This increase in the variances was confirmed by the Levene test, which proved that statistically significant differences exist between at least some pair of algorithms' output at a 5% level of significance.

Interestingly, the VSLs did have an effect on the total time spent in the system by vehicles joining the N1 traffic flow from the Okavango Road on-ramp, as may be deduced from the p-values in Table 10.48, which reveal that the no-control case was able to outperform the MTFC and kNN-TD VSL implementations at a 5% level of significance. The performance of Q-Learning was, however, found to be statistically indistinguishable from those of the no-control case, MTFC, and kNN-TD. Finally, kNN-TD for VSLs and MTFC were also found to perform statistically similarly at a 5% level of significance. This increase by the VSL implementations in respect of the TTSO is clearly visible in the box plots of Figure 10.7(e).

TABLE 10.46: *Differences in respect of the total time spent in the system entering the highway from the N1 (TTSN1) by all vehicles in the case of VSLs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTSN1			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	4.9224×10^{-1}	1.5987×10^{-1}	9.9619×10^{-2}
MTFC		—	1.1303×10^{-2}	3.4781×10^{-1}
Q-Learning			—	1.7358×10^{-1}
Mean	884.11	941.84	819.15	877.13

TABLE 10.47: *Differences in respect of the total time spent in the system by vehicles entering the system from the R300 (TTSR300) in the case of VSLs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TTSR300			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	5.2150×10^{-4}	1.6570×10^{-1}	9.3233×10^{-3}
MTFC		—	3.1695×10^{-2}	3.5672×10^{-1}
Q-Learning			—	2.1410×10^{-1}
Mean	992.19	839.63	932.57	879.18

TABLE 10.48: *Differences in respect of the total time spent in the system by vehicles entering the system from Okavango Road (TTSO) in the case of VSL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TTSO			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	1.1174×10^{-2}	5.2603×10^{-2}	9.0449×10^{-3}
MTFC		—	5.3630×10^{-1}	9.3914×10^{-1}
Q-Learning			—	4.8733×10^{-1}
Mean	14.00	14.35	14.26	14.36

As for the total time spent in the system by vehicles entering the network along the N1, statistical differences could be identified only between the performances of MTFC and Q-Learning in respect of the mean TISN1 at a 5% level of significance. Q-Learning achieved the smallest TISN1-value of 1.15 min/km, while kNN-TD and the no-control case achieved mean TISN1-values of 1.22 min/km and 1.24 min/km, respectively. Finally, MTFC achieved the largest mean TISN1-value of 1.31 min/km. This improvement in the mean TISN1 by Q-Learning may

be seen in the box plots of Figure 10.8(a), in which the similarity in performance between kNN-TD and the no-control case, and the increase observed for MTFC are also clear. Interestingly, the ordering in respect of the maximum TISN1 differs from that for the mean TISN1, as the no-control achieved the smallest value of 5.30 min/km, followed by kNN-TD with a maximum TISN1-value of 5.41 min/km, Q-Learning achieved a maximum TISN1-value of 6.16 min/km. Due to the small differences in these values, these three implementations were found to perform statistically similarly at a 5% level of significance. All three of these implementations were, however, able to outperform MTFC, for which a maximum TISN1-value of 9.76 min/km was recorded. This considerable increase may be due to the fact that a so-called artificial bottleneck is created by the MTFC controller, which may increase the travel times along the highway. This change in the ordering of the relative algorithmic performances is clearly visible in the box plots in Figure 10.8(b).

In respect of the mean TISR300, MTFC achieved the best performance as it outperformed the no-control case and Q-Learning at a 5% level of significance, as may be deduced from the p-values in Table 10.51. The kNN-TD implementation, which was found to perform statistically indistinguishably from both MTFC and Q-Learning, was also able to outperform the no-control case at a 5% level of significance. The MTFC and kNN-TD VSL implementations were able to achieve improvements of 15.61% and 11.54% over the no-control case, respectively, while Q-Learning was able to achieve a reduction of 6.45% over the no-control case. These marginal improvements may also be seen in the box plots in Figure 10.8(c). The algorithmic ordering in respect of the maximum TISR300 is the same as that for the mean TISR300, as MTFC, kNN-TD, Q-Learning and the no-control case achieved values of 21.93 min/km, 23.79 min/km, 24.49 min/km and 25.42 min/km, respectively. These improvements were, however, not large enough for the algorithmic performances to be classified as being statistically different at a 5% level of significance. This closeness of the algorithmic performances in respect of the maximum TISR300 may be seen in the box plots of Figure 10.8(d).

TABLE 10.49: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of VSLs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISN1 Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	4.9289×10^{-1}	1.2715×10^{-1}	9.6955×10^{-1}
MTFC		—	8.8995×10^{-3}	2.5918×10^{-1}
Q-Learning			—	2.2515×10^{-1}
Mean	1.24	1.31	1.15	1.22

TABLE 10.50: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of VSLs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TISN1 Max			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	2.1200×10^{-3}	5.4365×10^{-1}	9.9792×10^{-1}
MTFC		—	1.8095×10^{-2}	2.7067×10^{-3}
Q-Learning			—	6.4138×10^{-1}
Mean	5.30	9.76	6.16	5.41

TABLE 10.51: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of VSLs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

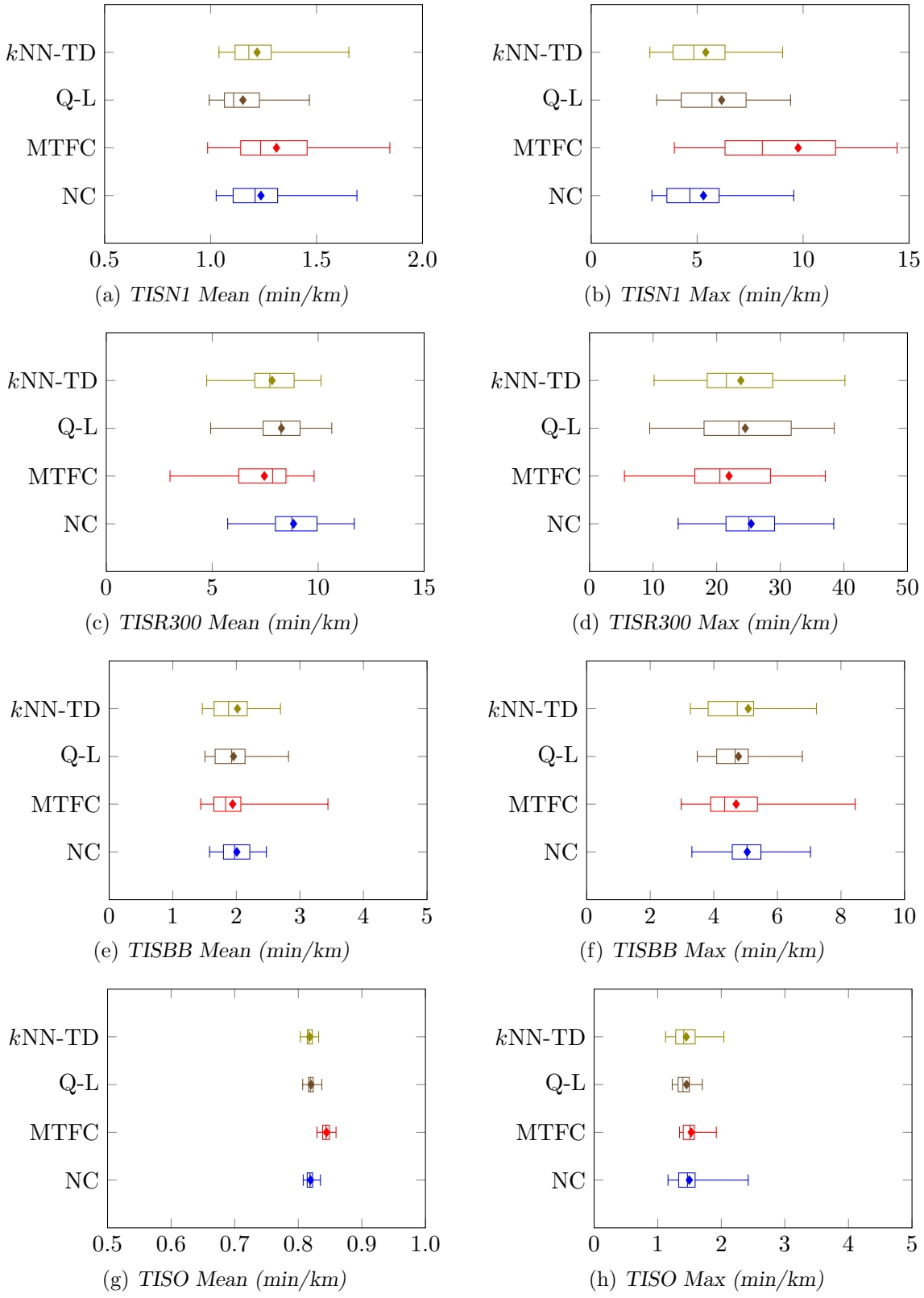
	Fisher LSD test p-values: TISR300 Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	1.8809×10^{-4}	1.1112×10^{-1}	5.3283×10^{-3}
MTFC		—	2.6144×10^{-2}	3.1066×10^{-1}
Q-Learning			—	2.1946×10^{-1}
Mean	8.84	7.46	8.27	7.82

TABLE 10.52: *Differences in respect of the mean time spent in the system by vehicles entering the system from Okavango Road in the case of VSLs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TISO Mean			
Algorithm	No Control	MTFC	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	6.0626×10^{-1}	5.0916×10^{-1}
MTFC		—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Q-Learning			—	2.4080×10^{-1}
Mean	0.82	0.84	0.82	0.82

Interestingly, there seems to be an increase in the variances of the results returned by the VSL implementations in respect of the mean TISBB, when compared to that of the no-control case, as may be seen in Figure 10.8(e). The Levene test, however, revealed that the variances are not statistically different at a 5% level of significance. As may be seen in the figure, the algorithmic means in respect of the mean TISBB are also very similar. This finding was corroborated by the results of the ANOVA, which showed that the performances of the VSL implementations were statistically indistinguishable from the no-control case in respect of the mean TISBB at a 5% level of significance. The situation in respect of the maximum TISBB is very similar, as again no statistical differences could be identified between any of the algorithmic performances at a 5% level of significance. The similarity of these performances in respect of the maximum TISBB is also evident in the box plots in Figure 10.8(f).

As was already indicated by the TTSO PMI, the VSL implementations had an effect on the travel time of vehicles entering the network from the Okavango Road on-ramp at system level. This increase is also evident in the mean TISO-values presented in Table 10.52. As may be seen in the table, MTFC was outperformed by all three other implementations at a 5% level of significance, while the latter three implementations were found to perform statistically indistinguishably. This similarity of the performances of the no-control case, Q-Learning and kNN-TD for VSLs is very clear in the box plots of Figure 10.8(g). In respect of the maximum TISO, no statistical differences were identified between the algorithmic performances at a 5% level of significance, as may be seen from the results of the ANOVA in Table 10.45. One may, however, argue that the VSL implementations resulted in reduced variances in respect of the maximum TISO, as may be seen from the smaller interquartile ranges of the box plots corresponding to the VSL implementations in Figure 10.8(h), although the Levene test revealed that these differences in the variances are not statistically significant at a 5% level of significance.



(a) *TISN1 Mean (min/km)* (b) *TISN1 Max (min/km)*

(c) *TISR300 Mean (min/km)* (d) *TISR300 Max (min/km)*

(e) *TISBB Mean (min/km)* (f) *TISBB Max (min/km)*

(g) *TISO Mean (min/km)* (h) *TISO Max (min/km)*

FIGURE 10.8: *Mean and maximum time in spent in the system PMI results for the no-control case (NC), the Q-Learning algorithm (Q-L) and the kNN-TD algorithm in the case of VSLs applied to the case study model of Chapter 9.*

10.3.4 Discussion

Taking into consideration the heavy traffic conditions prevailing in the case study area, as well as the fact that in Scenario 1 of the benchmark simulation model (which represented the heaviest traffic demand) the VSL implementations were unable to improve upon the no-control case, the VSL implementations performed comparatively well in the case study context, as all three implementations were able to achieve improvements over the no-control case in respect of the TTS. The RL implementations were furthermore able to achieve reductions in respect of both the TTSN1 and TTSR300 implementations, indicating that the improvements were achieved by the vehicles spending the longest time on the highway, as may have been expected. These improvements by the RL implementations were again largely due to reduced variances in the travel times of vehicles entering the simulated area on the N1, while an absolute improvement was observed in respect of the TTSR300, which may be due to a combination of an improved process whereby the vehicles from the R300 merge with the vehicles on the N1, as well as improved traffic flow along the highway at the bottlenecks of the Brackenfell Boulevard and Okavango Road on-ramps. Although the improvements in respect of the individual vehicle travel times were generally not large enough to be of statistical significance, except in the case of the mean TISR300 values, where kNN-TD outperformed the no-control case, these improvements compounded so as to have a significant effect on system level, as reflected by the TTSN1 and TTSR300 PMIs.

Although not quite as effective in reducing the TTS as the RL implementations, the MTFC implementation was also able to improve upon the no-control case in respect of the TTS, and TTSR300 PMIs. As may have been expected, the improvements achieved by the MTFC implementation are generally due to an absolute improvement of travel times, rather than reduced variances as in the case of the RL implementations, as homogenisation of traffic flow does not result from the relatively short application areas employed in the MTFC implementation. Increases in the variances compared with the no-control case were, in fact, recorded in respect of the PMIs corresponding to vehicles entering the simulated area on the N1 and the vehicles joining the N1 from the Brackenfell Boulevard on-ramp. These increase may be due to varying levels of mainline metering being applied in the various simulation runs, as the level of congestion may vary slightly. Due to the increased stability of the traffic flow when the RL implementations are employed (as indicated by the smaller interquartile ranges in the majority of the box plots) as well as the smaller TTS-values achieved by the RL implementations, the RL implementations were deemed to perform better than the MTFC implementation.

10.4 Multi-Agent Reinforcement Learning

This section is devoted to a description of the reward function evaluation and algorithmic performance comparison performed in respect of the MARL implementations within the context of the case study simulation model of Chapter 9. A reward function evaluation is performed first in respect of the independent MARL, hierarchical MARL and maximax MARL implementations with the aim of finding the best-performing combination of reward functions for the RM and VSL agents at each of the on-ramps considered. Once these combinations have been determined, the relative algorithmic performances are compared. Similarly to the comparison conducted in respect of the MARL algorithms for the benchmark simulation model of Chapter 8, the MARL implementations are compared with one another as well as with the kNN-TD RM implementation of §10.1, which yielded the best performance thus far. The results of this comparison are presented and interpreted by means of box plots in which the means, medians and interquartile

ranges of the PMIs of §9.3 are indicated, as well as tables indicating whether or not statistical differences exist between the PMI-values for each pair of algorithms at a 5% level of significance.

10.4.1 Algorithmic Implementations

Due to the fact that the kNN-TD learning algorithm was again considered to be the best-performing algorithm in respect of both RM and VSLs in the single agent paradigms within the context of this case study (as it was the case in the benchmark simulation model), only the kNN-TD algorithm is implemented in the three MARL approaches. For both the RM and VSL implementations, the best results were achieved by employing two RM or VSL kNN-TD RL agents in the case study area. The first of these is at the R300 interchange, while the second is at the Okavango Road interchange. As a result, only these two locations are considered for the MARL implementations. Therefore, there are two MARL implementations in the case study area, as may be seen in Figure 10.9. The first MARL implementation corresponds to the R300 interchange, and consists of the ramp meter placed at the R300 on-ramp, denoted by O_2, and the speed limits VSL_R and VSL_{R_1}. The target density of the agents in this MARL implementation is set to 28 veh/km, which was determined to be the best-performing target density in the RM parameter evaluation at the R300 on-ramp in §10.1.3. VSL_{R_1} is updated according to (7.2) with $\delta = 20$, which was found to yield the best results in the VSL parameter evaluation conducted for VSLs at the R300 interchange in §10.2.3. The second MARL implementation controls the ramp meter placed at the Okavango Road on-ramp, denoted by O_4, and the speed limits VSL_O and VSL_{O_1}. The target density of the agents in the second MARL implementation is set to 35.5 veh/km, which was determined to be the best-performing target density in the RM parameter evaluation of §10.1.2. Finally, VSL_{O_1} is updated according to (7.2) with $\delta = 10$, which was found to yield the best performance in the VSL parameter evaluation conducted in §10.2.2.

10.4.2 Reward Function Evaluations

This section is devoted to determining which combinations of reward functions yield the best performance when implemented in each of the MARL approaches at the two locations where MARL is employed within the case study model. As in the parameter evaluations of §10.1.2 and §10.2.2, the reward function evaluation was carried out following a step-wise approach. The best-performing combination of reward functions was first determined for the MARL implementation at the R300 interchange, followed by the reward function evaluation for the MARL implementation at the Okavango Road interchange. For each of the MARL implementations, three cases with different combinations of reward functions were considered. In the first of these cases, the reward function of the RM agent is based on the downstream density as in (6.2) and the reward function of the VSL agent is based on the outflow out of the bottleneck location at the respective on-ramp as in (8.6). In the second case, both agents are rewarded based on density, according to (6.2), while in the third case, both agents are rewarded based on the flow of vehicles out of the bottleneck locations at the respective on-ramp according to (8.6).

As may be seen in Table 10.53, employing MARL approaches at both the R300 interchange and the Okavango Road interchange leads to lower TTS-values for all three MARL approaches. For independent MARL, it was found that the best performance, measured in terms of the TTS, was achieved for the reward function combination of Case 1, where the RM and VSL agent is rewarded based on the downstream density and the outflow out of the bottleneck at the R300 interchange, respectively, and for the reward function combination of Case 2, where both agents are rewarded based on the downstream density at the Okavango Road interchange.

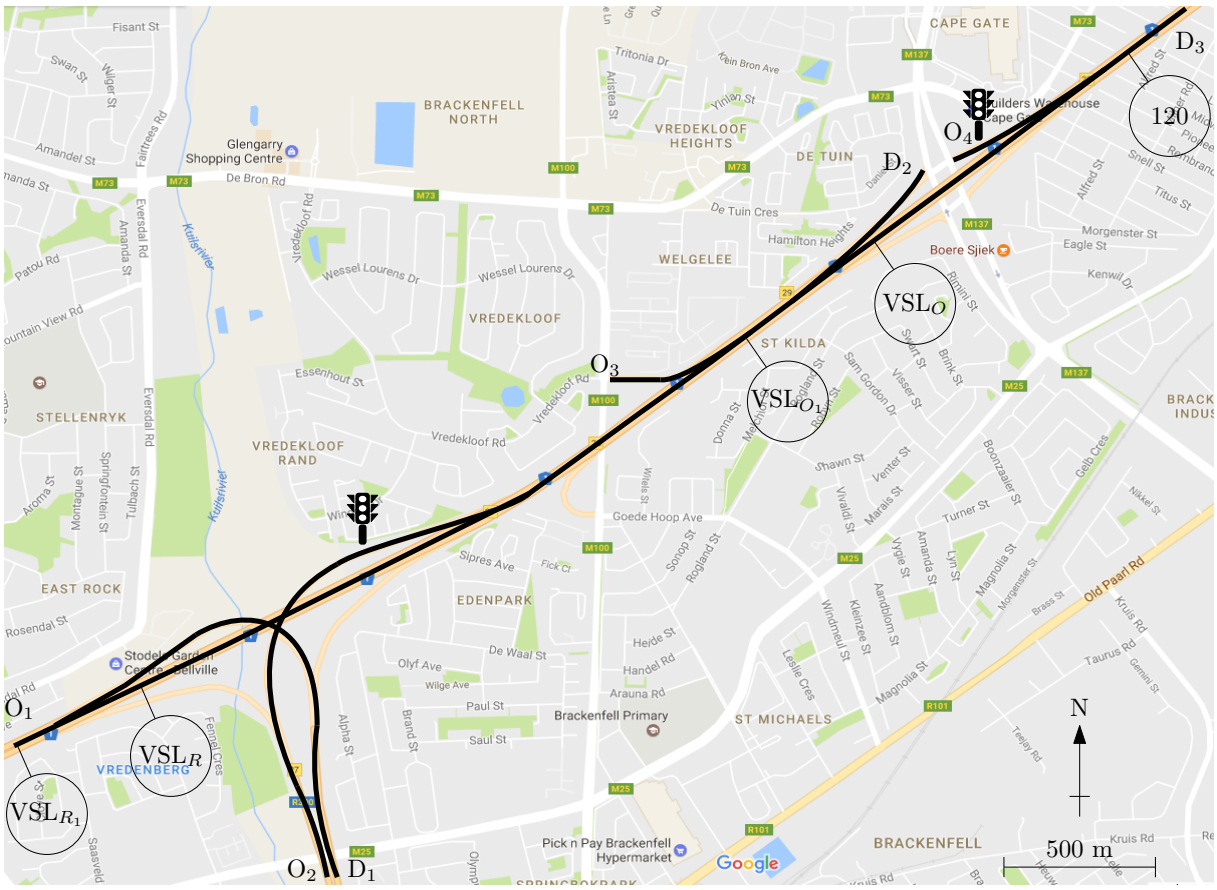


FIGURE 10.9: *The locations at which RM (indicated by the traffic lights) and VSLs (indicated by the speed limit signs) are applied in the context of the MARL approaches in the case study area.*

For the hierarchical MARL approach, rewarding both the RM agents, and both the VSL agents based on the respective downstream densities yielded the best performance in respect of the TTS. Finally, the results of the reward function evaluation for the maximax MARL approach indicated that employing the reward function combination of Case 1, where the RM and VSL agent is rewarded based on the downstream density and the outflow out of the bottleneck location at the R300 interchange, respectively, and the reward function combination of Case 3, where the RM and VSL agent is rewarded based on the outflow of traffic out of the bottleneck location at the Okavango Road interchange, resulted in the smallest TTS-values.

10.4.3 Algorithmic Comparison

The p-values of the ANOVA and Levene statistical tests conducted in respect of the PMI-values returned by the MARL approaches are presented in Table 10.54. The ANOVA revealed that there are, in fact, statistical differences at a 5% level of significance between the means returned by at least some pair of algorithms in respect of all thirteen PMIs. Furthermore, Levene's test revealed that the variances of the PMI-values returned by the algorithms were statistically indistinguishable for the TTSBB, mean TISR300, mean TSIBB and maximum TISBB PMIs.

TABLE 10.53: *Reward function evaluation results for MARL within the context of the case study model, measured as the TTS by the vehicles in veh·h.*

	R300 MARL Approach			Okavango Road MARL Approach		
Reward	Independent	Hierarchical	Maximax	Independent	Hierarchical	Maximax
Case 1	1 766.43	1 880.65	1 895.13	1 889.80	1 889.26	2 150.61
Case 2	1 798.48	1 732.20	2 125.30	1 754.67	1 711.08	2 012.08
Case 3	2 161.96	2 066.23	1 998.60	2 310.99	2 139.40	1 756.36

Therefore, the Fisher LSD test was performed in order to determine between which pairs of algorithms significant differences occur in respect of these PMIs. The Games-Howell test was performed for this purpose in respect of all other PMIs.

TABLE 10.54: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests associated with MARL. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					*p*-value	
PMI	No Control	*k*NN-TD	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	1 960.01	1 768.29	1 754.67	1 711.08	1 756.36	1.1019×10^{-7}	6.4387×10^{-8}
TTSN1	884.11	606.44	618.31	622.46	866.80	$< 1 \times 10^{-17}$	3.9214×10^{-10}
TTSR300	992.19	1 014.18	997.98	1 015.65	809.79	6.2046×10^{-8}	2.2653×10^{-5}
TTSBB	69.71	59.69	56.98	57.77	63.75	3.6898×10^{-5}	7.5751×10^{-1}
TTSO	14.00	86.27	80.92	13.62	15.36	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISN1 Mean	1.24	0.89	0.90	0.90	1.21	$< 1 \times 10^{-17}$	3.3968×10^{-10}
TISN1 Max	5.30	3.88	3.13	3.55	7.22	2.8977×10^{-14}	1.3144×10^{-4}
TISR300 Mean	8.84	14.32	14.68	15.04	7.21	$< 1 \times 10^{-17}$	5.8136×10^{-1}
TISR300 Max	25.42	42.03	41.99	43.05	23.05	$< 1 \times 10^{-17}$	1.8464×10^{-2}
TISBB Mean	2.01	1.72	1.64	1.66	1.86	2.2851×10^{-6}	5.8136×10^{-1}
TISBB Max	5.05	4.46	4.10	4.14	4.25	1.8617×10^{-3}	6.3882×10^{-1}
TISO Mean	0.82	5.03	4.71	0.79	0.81	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISO Max	1.50	18.43	18.92	1.47	1.39	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$

As may be seen in Figure 10.10(a), all of the MARL approaches were able to improve on the no-control case in respect of the TTS. This finding is corroborated by the *p*-values in Table 10.55, as all of the MARL approaches outperformed the no-control case at a 5% level of significance. Interestingly, all of the MARL approaches were found to perform statistically on par with one another and the *k*NN-TD RM implementation at a 5% level of significance, although all the MARL approaches achieved smaller TTS-values than *k*NN-TD RM. The MARL approaches did, however, return smaller variances in respect of the TTS than both the no-control case and *k*NN-TD RM, as may be deduced from the smaller interquartile ranges corresponding to the MARL approaches in the box plots of Figure 10.10(a). These differences in the variances were confirmed statistically by Levene's test. This may be an indication that the VSLs employed in the MARL approaches are able to achieve homogenisation of traffic flow, which results in reduced variances in the output data.

In respect of the TTSN1, *k*NN-TD RM, independent MARL and hierarchichal MARL achieved the best performance as they returned TTSN1-values of 606.44 veh·h, 618.31 veh·h and 622.46 veh·h, respectively, thereby outperforming both the no-control case and maximax MARL at a 5% level of significance, as may be deduced from the *p*-values in Table 10.56. The similarity in performance of these three implementations is evident in the box plots of Figure 10.10(b), as

their performances were found to be statistically indistinguishable at a 5% level of significance. Although maximax MARL achieved a slightly smaller TTSN1-value than the no-control case, their performances were also found to be statistically on par at a 5% level of significance.

Interestingly, in respect of the TTSR300, maximax MARL was able to outperform all other algorithms and the no-control case at a 5% level of significance, as may be inferred from the p-values in Table 10.57. The performances of kNN-TD RM, independent MARL and hierarchical MARL, on the other hand, were all found to be statistically indistinguishable from that of the no-control case at a 5% level of significance. The MARL approaches and kNN-TD RM were, however, again able to achieve significantly smaller variances in the output data in respect of the TTSR300, as may be seen in the box plots in Figure 10.10(c).

All of the MARL approaches, as well as the kNN-TD RM implementation were able to outperform the no-control case in respect of the TTSBB, as may be inferred from the p-values in Table 10.58. Independent MARL and hierarchical MARL achieved the smallest TTSBB-values of 56.98 veh·h and 57.77 veh·h, respectively, thereby outperforming maximax MARL at a 5% level of significance. The kNN-TD RM implementation returned a TTSBB-value of 59.69 veh·h, and was found to perform statistically on par with all of the MARL implementations. Finally, maximax MARL achieved a TTSBB-value of 63.75 veh·h, while the no-control case returned a value of 69.71 veh·h. Interestingly, kNN-TD RM and all of the MARL approaches resulted in an increase in the variances of the output data, indicated by the larger interquartile ranges in the box plots corresponding to the MARL approaches in respect of the TTSBB, as may be seen in Figure 10.10(d). The results of Levene's test, however, revealed that these increases in the variances were statistically indistinguishable at a 5% level of significance.

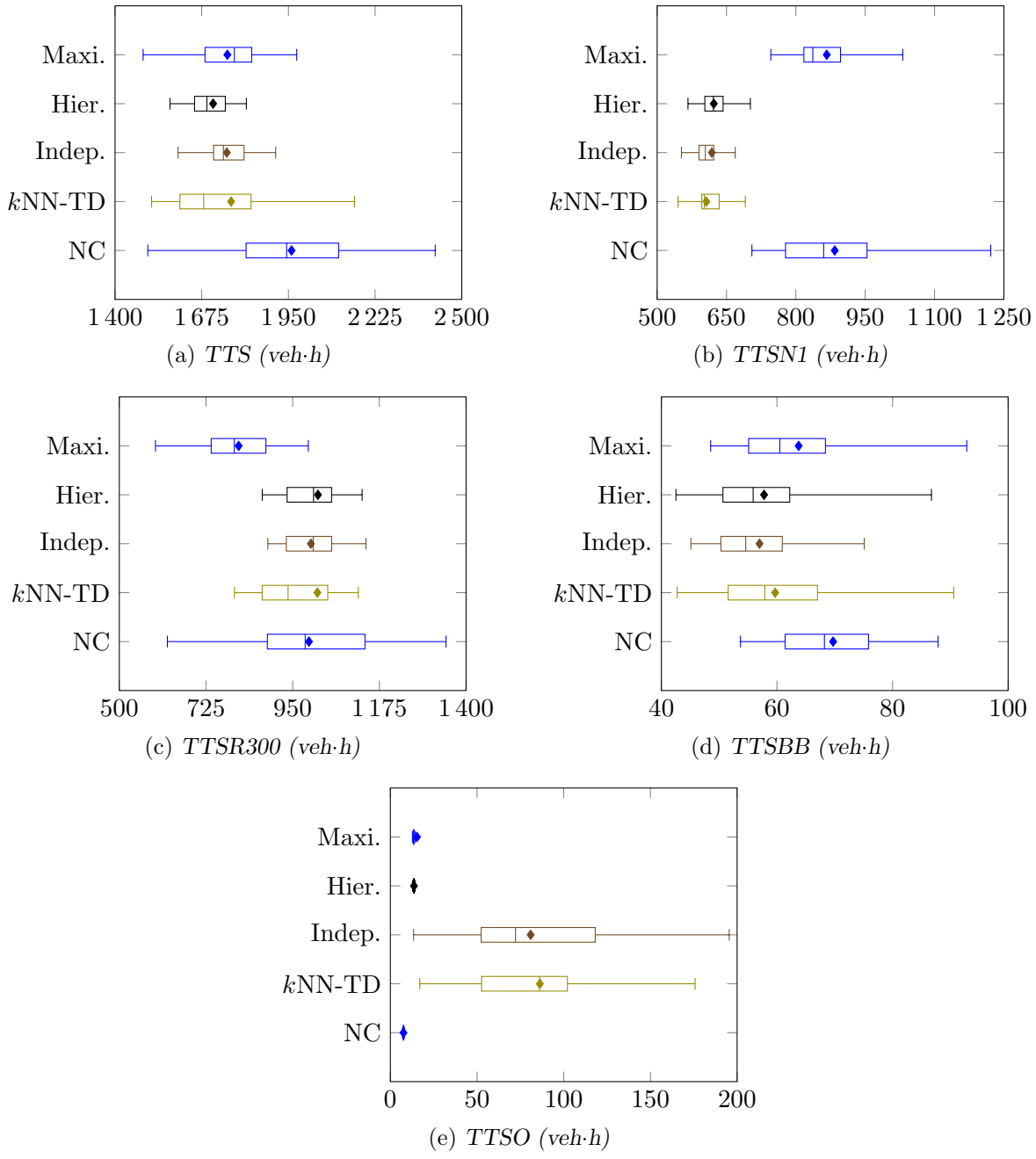
TABLE 10.55: *Differences in respect of the total time spent in the system (TTS) by all vehicles in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTS				
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.4035×10^{-2}	2.3054×10^{-4}	1.5049×10^{-5}	5.1973×10^{-4}
kNN-TD		—	9.9793×10^{-1}	7.2819×10^{-1}	9.9908×10^{-1}
Independent			—	3.3870×10^{-1}	9.9999×10^{-1}
Hierarchical				—	5.2461×10^{-1}
Mean	1 960.01	1 768.29	1 754.67	1 711.08	1 756.36

TABLE 10.56: *Differences in respect of the total time spent in the system by vehicles entering the system from the N1 (TTSN1) in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTSN1				
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	9.1958×10^{-12}	1.6270×10^{-11}	4.9141×10^{-11}	9.7620×10^{-1}
kNN-TD		—	8.6501×10^{-1}	4.0191×10^{-1}	$< 1 \times 10^{-17}$
Independent			—	9.9662×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	$< 1 \times 10^{-17}$
Mean	884.11	606.44	618.31	622.46	866.80

Perhaps unexpectedly, the performances of both hierarchical MARL and maximax MARL were found to be statistically on par with the no-control case at a 5% level of significance in respect



(a) *TTS (veh·h)*

(b) *TTSN1 (veh·h)*

(c) *TTSR300 (veh·h)*

(d) *TTSBB (veh·h)*

(e) *TTSO (veh·h)*

FIGURE 10.10: *Total time spent in the system PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) applied to the case study model of Chapter 9.*

TABLE 10.57: *Differences in respect of the total time spent in the system by vehicles entering the system from the R300 (TTSR300) in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSR300			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	9.9225×10^{-1}	9.9988×10^{-1}	9.6979×10^{-1}	9.7129×10^{-5}
kNN-TD		—	9.9376×10^{-1}	9.9999×10^{-1}	2.6748×10^{-4}
Independent			—	9.3406×10^{-1}	7.6583×10^{-10}
Hierarchical				—	4.5142×10^{-9}
Mean	992.19	1 014.18	997.98	1 015.65	809.79

TABLE 10.58: *Differences in respect of the total time spent in the system by vehicles entering the system from Brackenfell Boulevard (TTSBB) in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTSBB			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	4.9033×10^{-4}	1.2267×10^{-5}	3.8182×10^{-5}	3.5715×10^{-2}
kNN-TD		—	3.3738×10^{-1}	4.9593×10^{-1}	1.5013×10^{-1}
Independent			—	7.7994×10^{-1}	1.7238×10^{-2}
Hierarchical				—	3.4913×10^{-2}
Mean	69.71	59.69	56.98	57.77	63.75

TABLE 10.59: *Differences in respect of the total time spent in the system by vehicles entering the system from Okavango Road (TTSO) in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSO			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	5.7257×10^{-10}	2.4168×10^{-7}	5.8702×10^{-2}	9.2744×10^{-1}
kNN-TD		—	9.8984×10^{-1}	5.0931×10^{-10}	5.6237×10^{-10}
Independent			—	2.1663×10^{-7}	3.3444×10^{-7}
Hierarchical				—	8.4033×10^{-1}
Mean	14.00	86.27	80.92	13.62	15.36

of the TTSO, although both these approaches employ RM at the Okavango Road on-ramp. As may be seen in Figure 10.10(e), kNN-TD RM and independent MARL exhibited an expected increase in travel times for vehicles joining the N1 from the Okavango Road on-ramp due to RM. This finding is confirmed by the p-values presented in Table 10.59, which show that kNN-TD RM and independent MARL were outperformed by the no-control case, hierarchical MARL and maximax MARL at a 5% level of significance, while they were found to be statistically indistinguishable from one another.

In a trend similar to that for the TTSN1, kNN-TD RM, independent MARL and hierarchical MARL again achieved the best performance in respect of both the mean and maximum TISN1, outperforming the no-control case and maximax MARL at a 5% level of significance, as may be inferred from the p-values in Tables 10.60 and 10.61. In respect of the mean TISN1, the no-control case and maximax MARL were found to perform statistically on par with one another at a 5% level of significance, while the no-control case was able to outperform maximax MARL at a 5% level of significance with respect to the maximum TISN1. These trends in the relative

algorithmic performances of the algorithms are also visible in the box plots of Figures 10.11(a) and 10.11(b).

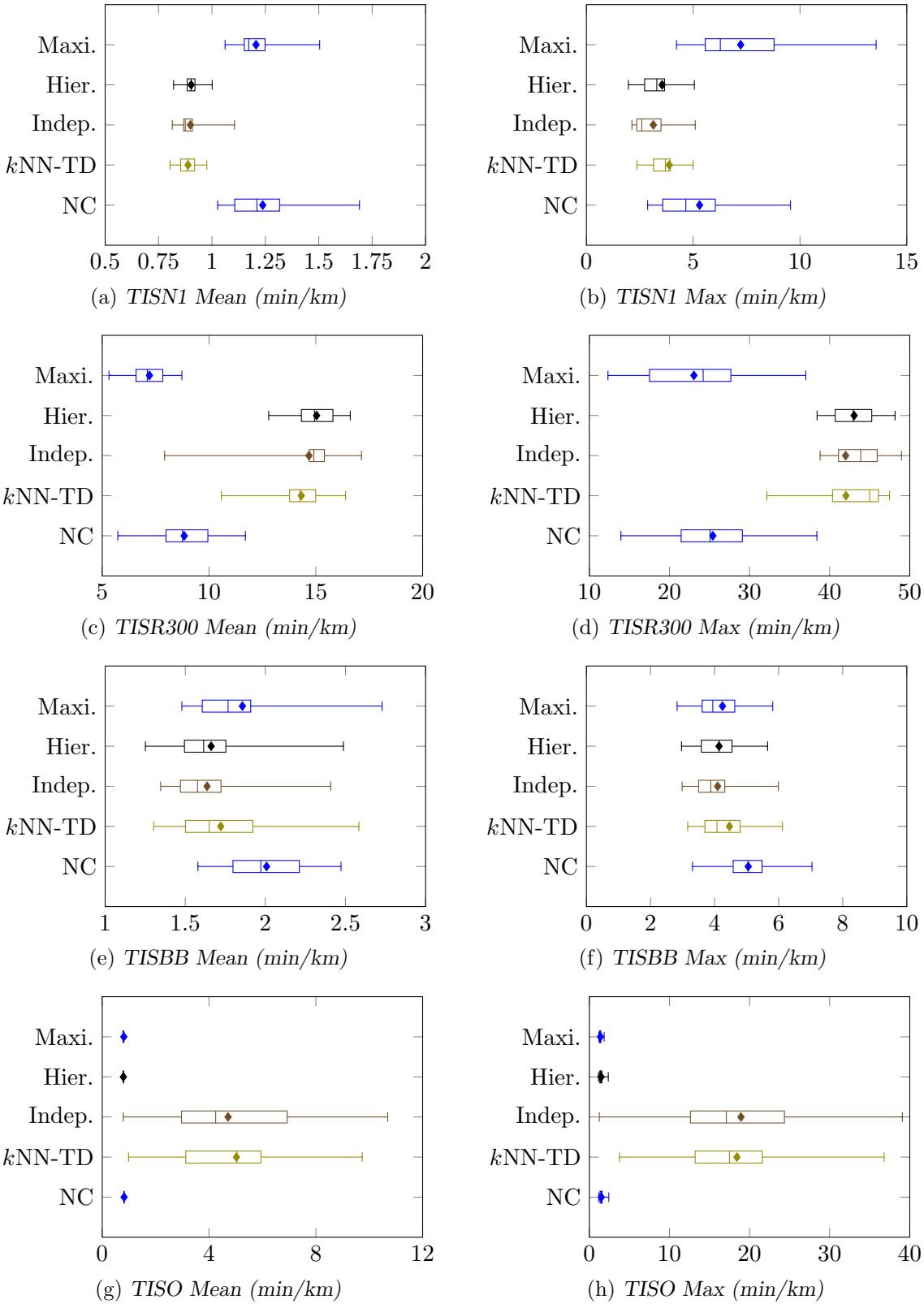
Maximax MARL was able to outperform all other algorithms and the no-control case in respect of the mean TISR300 at a 5% level of significance, as may be deduced from the p-values in Table 10.62. Maximax MARL was followed in the order of relative algorithmic performances by the no-control case, which was able to outperform kNN-TD RM, independent MARL and hierarchical MARL at a 5% level of significance. The kNN-TD RM implementation achieved the third-smallest mean TISR300-value, outperforming hierarchical MARL at a 5% level of significance, while it was found to perform statistically on par with independent MARL. Independent MARL and hierarchical MARL were also found to perform statistically indistinguishably at a 5% level of significance. This order of relative algorithmic performances is evident in the box plots of Figure 10.11(c). As may be seen in Figure 10.11(d), the order of relative algorithmic performances in respect of the maximum TISR300 is similar to that for the mean TISR300. This observation is confirmed by the p-values in Table 10.63, apart from the fact that, in respect of the maximum TISR300, maximax MARL and the no-control case were found to perform statistically on par, while kNN-TD RM, independent MARL and hierarchical MARL were found to be statistically indistinguishable at a 5% level of significance.

As for the TTSBB, all of the MARL implementations, as well as kNN-TD RM, were able to outperform the no-control case at a 5% level of significance in respect of the mean TISBB, as may be inferred from the p-values in Table 10.64. Independent MARL and hierarchical MARL returned the smallest mean TISBB-values of 1.64 min/km and 1.66 min/km, respectively, thereby outperforming maximax MARL, which achieved a mean TISBB-value of 1.86 min/km, at a 5% level of significance. Although maximax MARL achieved a larger mean TISBB-value than kNN-TD RM, which returned a value of 1.72 min/km, the two algortihms were found to perform statistically on par at a 5% level of significance. This order of relative algorithmic performances is evident in Figure 10.11(e). In respect of the maximum TISBB, all of the algorithms were again able to outperform the no-control case at a 5% level of significance, as may be inferred from the p-values in Table 10.65. Unlike for the mean TISBB, however, the performances of kNN-TD RM and all of the MARL implementations were found to perform statistically indistinguishably at a 5% level of significance in respect of the maximum TISBB. This similarity is clearly visible in the box plots of Figure 10.11(f).

As was the case for the TTSO, hierarchical MARL and maximax MARL performed statistically on par with the no-control case in respect of both the mean and maximum TISO at a 5% level of significance, although both these implementations employ RM at the Okavango Road on-ramp. This similarity is evident in Figures 10.11(g) and 10.11(h). The expected increase in the mean and maximum TISO is again reflected by kNN-TD RM and independent MARL, which were outperformed by the no-control case, hierarchical MARL and maximax MARL at a 5% level of significance in respect of both these PMIs, as may be deduced from the p-values in Tables 10.66 and 10.67.

10.4.4 Discussion

As was the case in the context of the benchmark simulation model of §5.1.2, the MARL implementations were again able to achieve improvements over and above those achieved by single-agent RM or VSL implementation in the context of the case study simulation model of Chapter 9. Although these improvements were statistically indistinguishable from that of kNN-TD RM (the best-performing single-agent implementation) at a 5% level of significance in respect of the TTS, the MARL approaches did result in a number of interesting improvements over single-agent



(a) *TISN1 Mean (min/km)*
(b) *TISN1 Max (min/km)*
(c) *TISR300 Mean (min/km)*
(d) *TISR300 Max (min/km)*
(e) *TISBB Mean (min/km)*
(f) *TISBB Max (min/km)*
(g) *TISO Mean (min/km)*
(h) *TISO Max (min/km)*

FIGURE 10.11: *Mean and maximum time spent in the system PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) applied to the case study model of Chapter 9.*

TABLE 10.60: *Differences in respect of the mean time spent in the system by vehicles entering the system from the N1 in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISN1 Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.0260×10^{-11}	1.2521×10^{-11}	3.7440×10^{-11}	9.1094×10^{-1}
kNN-TD		—	9.4657×10^{-1}	6.4012×10^{-1}	$< 1 \times 10^{-17}$
Independent			—	9.9869×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	$< 1 \times 10^{-17}$
Mean	1.24	0.89	0.90	0.90	1.21

TABLE 10.61: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the N1 in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISN1 Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	4.5598×10^{-2}	1.0482×10^{-3}	1.1921×10^{-2}	2.9182×10^{-2}
kNN-TD		—	2.8773×10^{-1}	9.0297×10^{-1}	1.0359×10^{-6}
Independent			—	8.5234×10^{-1}	1.4565×10^{-8}
Hierarchical				—	2.1628×10^{-2}
Mean	5.30	3.88	3.13	3.55	7.22

TABLE 10.62: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISR300 Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	6.3408×10^{-6}
kNN-TD		—	3.0126×10^{-1}	4.0671×10^{-2}	$< 1 \times 10^{-17}$
Independent			—	3.0570×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	$< 1 \times 10^{-17}$
Mean	8.84	14.32	14.68	15.04	7.21

TABLE 10.63: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the R300 in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISR300 Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.4017×10^{-11}	1.9473×10^{-11}	4.8428×10^{-13}	6.5593×10^{-1}
kNN-TD		—	9.9999×10^{-1}	9.2044×10^{-1}	1.1919×10^{-11}
Independent			—	9.4738×10^{-1}	1.4159×10^{-11}
Hierarchical				—	1.7319×10^{-13}
Mean	25.42	42.03	41.99	43.05	23.05

TABLE 10.64: *Differences in respect of the mean time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISBB Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	1.645×10^{-4}	1.4478×10^{-6}	6.4462×10^{-6}	4.2816×10^{-2}
kNN-TD		—	2.4954×10^{-1}	4.1821×10^{-1}	6.9689×10^{-2}
Independent			—	7.3115×10^{-1}	3.3443×10^{-3}
Hierarchical				—	9.2165×10^{-3}
Mean	2.01	1.72	1.64	1.66	1.86

TABLE 10.65: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISBB Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	2.5565×10^{-2}	3.4552×10^{-4}	6.2982×10^{-4}	2.4329×10^{-3}
kNN-TD		—	1.6082×10^{-1}	2.1746×10^{-1}	4.0809×10^{-1}
Independent			—	8.6457×10^{-1}	5.6290×10^{-1}
Hierarchical				—	6.8314×10^{-1}
Mean	5.05	4.46	4.10	4.14	4.25

TABLE 10.66: *Differences in respect of the mean time spent in the system by vehicles entering the system from Okavango Road in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Mean			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	2.3104×10^{-10}	1.4448×10^{-7}	6.7536×10^{-1}	9.7001×10^{-1}
kNN-TD		—	9.8853×10^{-1}	1.9940×10^{-10}	2.1522×10^{-10}
Independent			—	1.2605×10^{-7}	1.3670×10^{-7}
Hierarchical				—	8.0909×10^{-1}
Mean	0.82	5.03	4.71	0.79	0.81

TABLE 10.67: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Okavango Road in the case of MARL. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Max			
Algorithm	No Control	kNN-TD	Independent	Hierarchical	Maximax
No Control	—	2.5918×10^{-11}	1.8012×10^{-8}	9.9476×10^{-1}	3.1228×10^{-1}
kNN-TD		—	9.9965×10^{-1}	2.4899×10^{-11}	2.2457×10^{-11}
Independent			—	1.7431×10^{-8}	1.5859×10^{-8}
Hierarchical				—	6.5165×10^{-1}
Mean	1.50	18.43	18.92	1.47	1.39

kNN-TD RM. Notably, none of the MARL implementations resulted in a statistically significant increase in the TTSR300, although RM is applied at the R300 on-ramp. The expected increases in the travel times for vehicles joining the N1 from the R300 were, however, reflected in both the mean and maximum TISR300 PMIs. This finding suggests that although there were statisti-

cally significant increases in respect of the mean and maximum TISR300-values, these increases are not large enough to result in statistically significant increases at system level. Remarkably, the hierarchical and maximax MARL implementations did not result in statistically significant increases in the TTSO, or mean and maximum TISO-values. This suggests that, in the context of this case study, these MARL approaches may not result in significant increases in the queue lengths at the Okavango Road on-ramp due to RM if it is applied effectively in conjunction with VSLs, while statistically significant reductions in respect of the travel times along the highway may still be achieved. Furthermore, apart from the PMIs related to the vehicles entering the system from the Brackenfell Boulevard on-ramp, there generally seems to be a reduction in the variances (indicated by smaller interquartile ranges in the box plots corresponding to hierarchical MARL and maximax MARL) of the PMI output data when hierarchical MARL and maximax MARL are employed. This suggests that effective homogenisation of traffic flow may result due to the presence of VSLs. As stated above, hierarchical MARL did not result in statistically significant increases in the TTSR300 or TTSO-values. Furthermore, hierarchical MARL achieved a smaller TTS-value than maximax MARL, which was the only other MARL implementation that returned TTSR300 and TTSO-values which are statistically indistinguishable from the no-control case. Therefore, hierarchical MARL is considered to be the best-performing implementation in the context of this case study.

10.5 Multi-Agent Reinforcement Learning with Queue Limits

Due to the RM component in the MARL implementations there still exists the potential for the build up of undesirably long on-ramp queues at the R300 and Okavango Road on-ramps. Therefore, an on-ramp queue consideration is implemented, as was the case in the RM implementations, ensuring that the on-ramp queue length at the Okavango Road on-ramp does not exceed 50 vehicles which could cause severe congestion problems in the arterial road network connected to the N1 highway.

10.5.1 Algorithmic Implementations

As pointed out above, both the hierarchical and maximax MARL implementations in their original form were able to limit the formation of on-ramp queues at the Okavango Road on-ramp, while maximax MARL was also successful in this regard at the R300 on-ramp. Therefore, the queue limitation was introduced only in the independent MARL implementation for which excessively long travel times of vehicles joining the highway from the Okavango Road on-ramp were recorded. The effect of the implementation of the queue limitation on the overall performance of the independent MARL implementation is summarised in Table 10.68. The queue limitation was again implemented according to (6.11), by punishing the RM agent for queue lengths which exceed the maximum allowable queue length of 50 vehicles.

As may be seen in the table, the addition of the queue limitation did result in an increase in respect of the TTS, although this increase was not large enough to classify the performances as statistically significantly different at a 5% level of significance. The travel times of those vehicles entering the simulated area on the N1 remained largely unchanged, while a significant increase in the travel times of vehicles joining the N1 from the R300 was recorded. This may again be attributed to the lesser level of RM applied at the Okavango Road on-ramp, resulting in the observation that those vehicles travelling along the N1 which joined from the R300 require more time to traverse the simulated length of the N1. In respect of the TTSBB, a small increase in the travel times was again observed, possibly due to the same reason as that for the vehicles

TABLE 10.68: *The effect of employing queue limitations in the RM implementations on their overall performance in the case study.*

	Independent MARL	
PMI	$\hat{w} = 50$	$\hat{w} = \infty$
TTS (veh·h)	1 761.45	1 754.67
TTSN1 (veh·h)	614.73	618.31
TTSR300 (veh·h)	1 046.13	997.29
TTSBB (veh·h)	58.79	56.98
TTSO (veh·h)	40.60	56.98

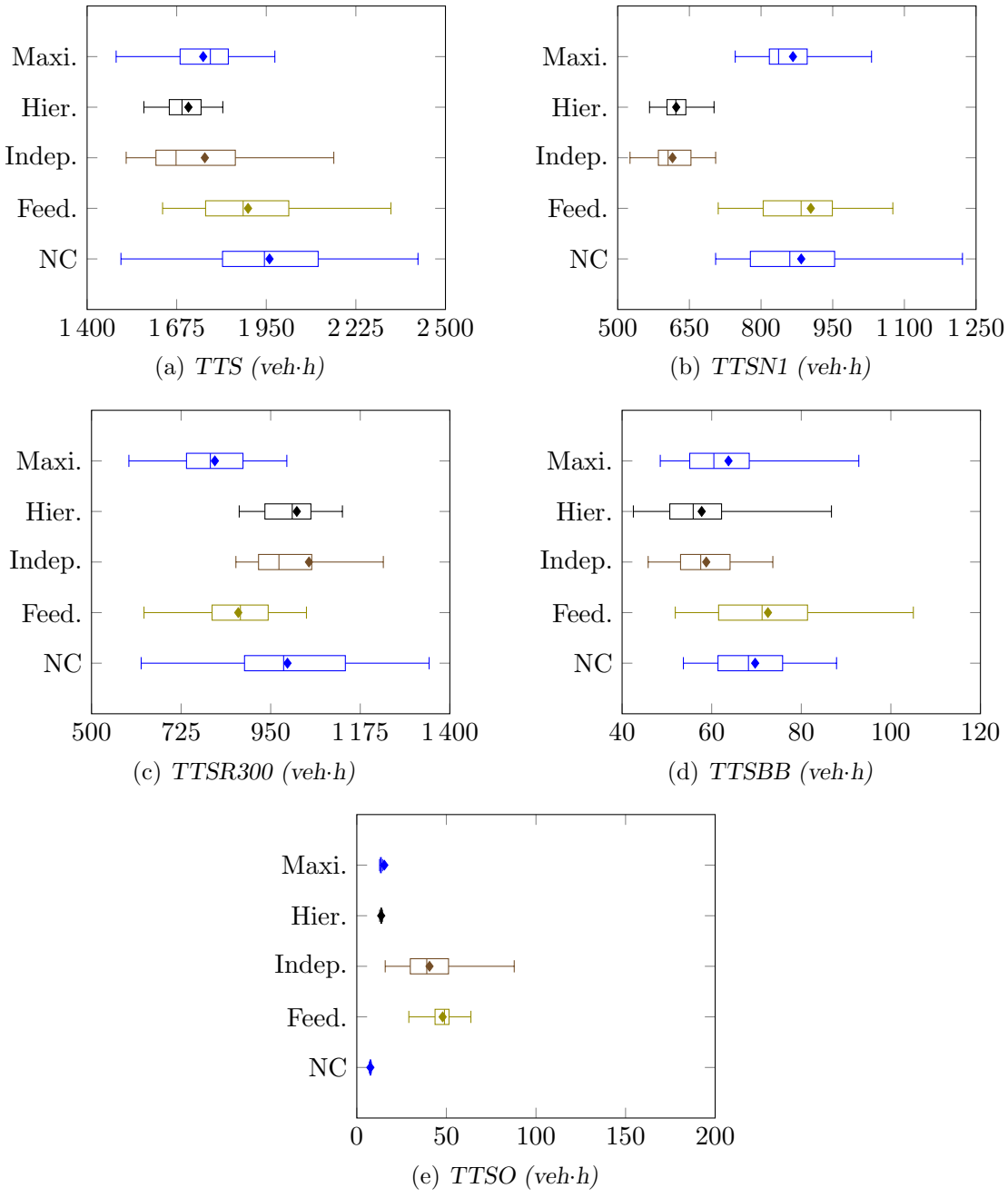
joining the N1 from the R300. As expected, a significant decrease in the TTSO was observed due to larger metering rates being applied in order to prevent the formation of excessively long on-ramp queues.

For the purpose of comparison, the integrated feedback controller of Carlson *et al.* [24] was also implemented. Due to the fact that both PI-ALINEA and the MTFC controller of Müller *et al.* [105] were both able to achieve the largest improvements over the no-control case in respect of the TTS when implemented only at the Okavango Road on-ramp, only a single integrated controller was implemented at the Okavango Road on-ramp. As for the integrated feedback controller implementation in Chapter 8, the best-performing target densities and controller parameters found for the individual controllers in §10.1.2 and §10.3.2 were retained for the integrated controller.

10.5.2 Algorithmic Comparison

As may be seen from the results of the ANOVA performed for the MARL implementations with the addition of a queue limitation, presented in Table 10.69, statistical differences were observed between at least some pair of algorithmic outputs at a 5% level of significance in respect of all PMIs. The Levene test furthermore revealed that the variances of the algorithmic output data are only statistically indistinguishable at a 5% level of significance in respect of the TTSBB, mean TISR300, and mean and maximum TISBB PMIs, while the variances of the algorithmic output differ for at least some pair of algorithms in respect of all other PMIs. As a result, the Games-Howell *post hoc* test was performed in order to ascertain between which pairs of algorithmic output these differences occur in respect of all PMIs, except for the TTSBB, mean TISR300, and mean and maximum TISBB, while the Fisher LSD test was performed for this purpose in respect of the TTSBB, mean TISR300, and mean and maximum TISBB.

Hierarchical MARL and maximax MARL again achieved the best performances in respect of the TTS, achieving improvements of 12.70% and 10.39% over the no-control case, outperforming the no-control case, the feedback controller and independent MARL at a 5% level of significance, as may be deduced from the p-values in Table 10.70. Hierarchical and maximax MARL were followed in the order of relative algorithmic performances by independent MARL, while its performance was found not to differ statistically from that of the feedback controller, but was able to outperform the no-control case (achieving an improvement of 10.13%). Although the feedback controller was able to improve upon the no-control case by 3.36%, this improvement was not large enough for the performance of the feedback controller to be classified as statistically different from that of the no-control case at a 5% level of significance. This order of relative algorithmic performances is also evident in the box plots of Figure 10.12(a). As may be seen in the figure, the reduced variances indicated by the smaller interquartile ranges corresponding to



(a) *TTS (veh·h)*

(b) *TTSN1 (veh·h)*

(c) *TTSR300 (veh·h)*

(d) *TTSBB (veh·h)*

(e) *TTSO (veh·h)*

FIGURE 10.12: *Total time spent in the system PMI results for the no-control case (NC), the integrated feedback controller (Feed.), independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) with queue limits applied to the case study model of Chapter 9.*

TABLE 10.69: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests associated with MARL with queue limits. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	Feed.	Indep.	Hier.	Maxi.	ANOVA	Levene's Test
TTS	1 960.01	1 894.11	1 761.45	1 711.08	1 756.36	8.6625×10^{-8}	2.5877×10^{-4}
TTSN1	884.11	904.14	614.73	622.46	866.80	$< 1 \times 10^{-17}$	1.5207×10^{-7}
TTSR300	992.19	868.62	1 046.13	1 015.65	809.79	6.0219×10^{-10}	9.9230×10^{-3}
TTSBB	69.71	72.56	58.79	57.77	63.75	1.2036×10^{-6}	2.2861×10^{-1}
TTSO	14.00	47.90	40.58	13.62	15.36	$< 1 \times 10^{-17}$	2.3814×10^{-13}
TISN1 Mean	1.24	1.26	0.89	0.90	1.21	$< 1 \times 10^{-17}$	3.1680×10^{-3}
TISN1 Max	5.30	9.46	4.47	3.55	7.22	4.5841×10^{-13}	1.7544×10^{-6}
TISR300 Mean	8.84	7.72	14.62	15.04	7.21	$< 1 \times 10^{-17}$	5.6218×10^{-2}
TISR300 Max	25.42	23.03	42.30	43.05	23.05	$< 1 \times 10^{-17}$	5.0009×10^{-5}
TISBB Mean	2.01	2.08	1.68	1.66	1.86	9.2702×10^{-8}	1.6794×10^{-1}
TISBB Max	5.05	5.17	4.12	4.14	4.25	6.3935×10^{-7}	7.6698×10^{-1}
TISO Mean	0.82	2.82	2.36	0.79	0.81	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISO Max	1.50	11.23	10.92	1.47	1.39	$< 1 \times 10^{-17}$	4.2697×10^{-10}

the MARL implementations indicate that, again, there may be homogenisation of traffic flow due to the VSLs.

As may clearly be seen in the box plots in Figure 10.12(b), hierarchical MARL and independent MARL returned the best performance in respect of the TTSN1, outperforming all other implementations at a 5% level of significance. This is corroborated by the p-values in Table 10.71. The performances of the feedback controller, maximax MARL and the no-control case, on the other hand, were found to be statistically indistinguishable from one another at a 5% level of significance. Although the implementations performed statistically similarly, maximax MARL was able to improve upon the no-control case, while the feedback controller resulted in an increase in the TTN1 when compared with the no-control case.

The good performances of independent MARL and hierarchical MARL in respect of the TTSN1 are, perhaps as expected, offset by relatively poor performances in respect of the TTSR300, as they were outperformed by both the feedback controller and maximax MARL at a 5% level of significance, as may be deduced from the p-values in Table 10.72. Maximax MARL did, in fact, achieve the smallest TTSR300-value, outperforming all other algorithms at a 5% level of significance. The feedback controller was also able to outperform the no-control case at a 5% level of significance, while the performances of the no-control case, independent MARL and hierarchical MARL were found not to differ statistically. These trends are again clearly visible in the box plots of Figure 10.12(c).

The performances of independent MARL and hierarchical MARL were again very similar in respect of the TTSBB, as these algorithms returned the smallest TTSBB-values of 58.79 veh·h and 57.77 veh·h, respectively, outperforming the no-control case and the feedback controller at a 5% level of significance, while their performances were statistically indistinguishable from that of maximax MARL. The TTS-value of 63.75 veh·h achieved by maximax MARL was small enough to outperform the feedback controller, while its performance was found to be statistically indistinguishable from that of the no-control case at a 5% level of significance, as may be seen in Table 10.73. Finally, the feedback controller returned the largest TTSBB-value of 75.25 veh·h placing its performance statistically on par with that of the no-control case, which returned a mean TTSBB-value of 69.71 veh·h. This ordering of relative algorithmic performances is also evident in the box plots of Figure 10.12(d).

In respect of the TTSO, hierarchical MARL and maximax MARL were again found to perform statistically on par with the no-control case at a 5% level of significance, although RM is applied at the Okavango Road on-ramp. These three implementations, however, outperformed both the feedback controller and independent MARL at a 5% level of significance in respect of the TTSO. The expected increases due to the RM in the TTSO were reflected by both the feedback controller and independent MARL, as may be seen in the box plots of Figure 10.12(e). Interestingly, the feedback controller returned a significantly smaller variance than independent MARL, indicating a more stable RM policy. As may be seen from the p-values in Table 10.74, the large variance of the independent controller resulted in the fact that, although its mean TTSO-value is smaller than that of the feedback controller, their performances were found to be statistically similar at a 5% level of significance.

TABLE 10.70: *Differences in respect of the total time spent in the system (TTS) by all vehicles in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	7.0236×10^{-1}	7.6271×10^{-3}	1.5049×10^{-5}	5.1973×10^{-4}
Feedback		—	8.5021×10^{-2}	9.9523×10^{-5}	6.9202×10^{-3}
Independent			—	7.8424×10^{-1}	9.9996×10^{-1}
Hierarchical				—	5.2461×10^{-1}
Mean	1 960.01	1 894.11	1 761.45	1 711.08	1 756.36

TABLE 10.71: *Differences in respect of the total time spent in the system by vehicles entering the system from the N1 (TTSN1) in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSN1			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	9.7751×10^{-1}	1.3879×10^{-11}	4.9141×10^{-11}	9.7620×10^{-1}
Feedback		—	3.9163×10^{-12}	1.4928×10^{-11}	7.2358×10^{-1}
Independent			—	9.4677×10^{-1}	1.0900×10^{-12}
Hierarchical				—	$< 1 \times 10^{-17}$
Mean	884.11	904.14	614.73	622.46	866.80

TABLE 10.72: *Differences in respect of the total time spent in the system by vehicles entering the system from the R300 (TTSR300) in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSR300			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	2.6570×10^{-2}	7.8219×10^{-1}	9.6979×10^{-1}	9.7129×10^{-5}
Feedback		—	1.1522×10^{-3}	2.1962×10^{-4}	3.4325×10^{-1}
Independent			—	9.4075×10^{-1}	3.7281×10^{-6}
Hierarchical				—	4.5142×10^{-9}
Mean	992.19	868.62	1 046.13	1 015.65	809.79

TABLE 10.73: *Differences in respect of the total time spent in the system by vehicles entering the system from Brackenfell Boulevard (TTSBB) in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

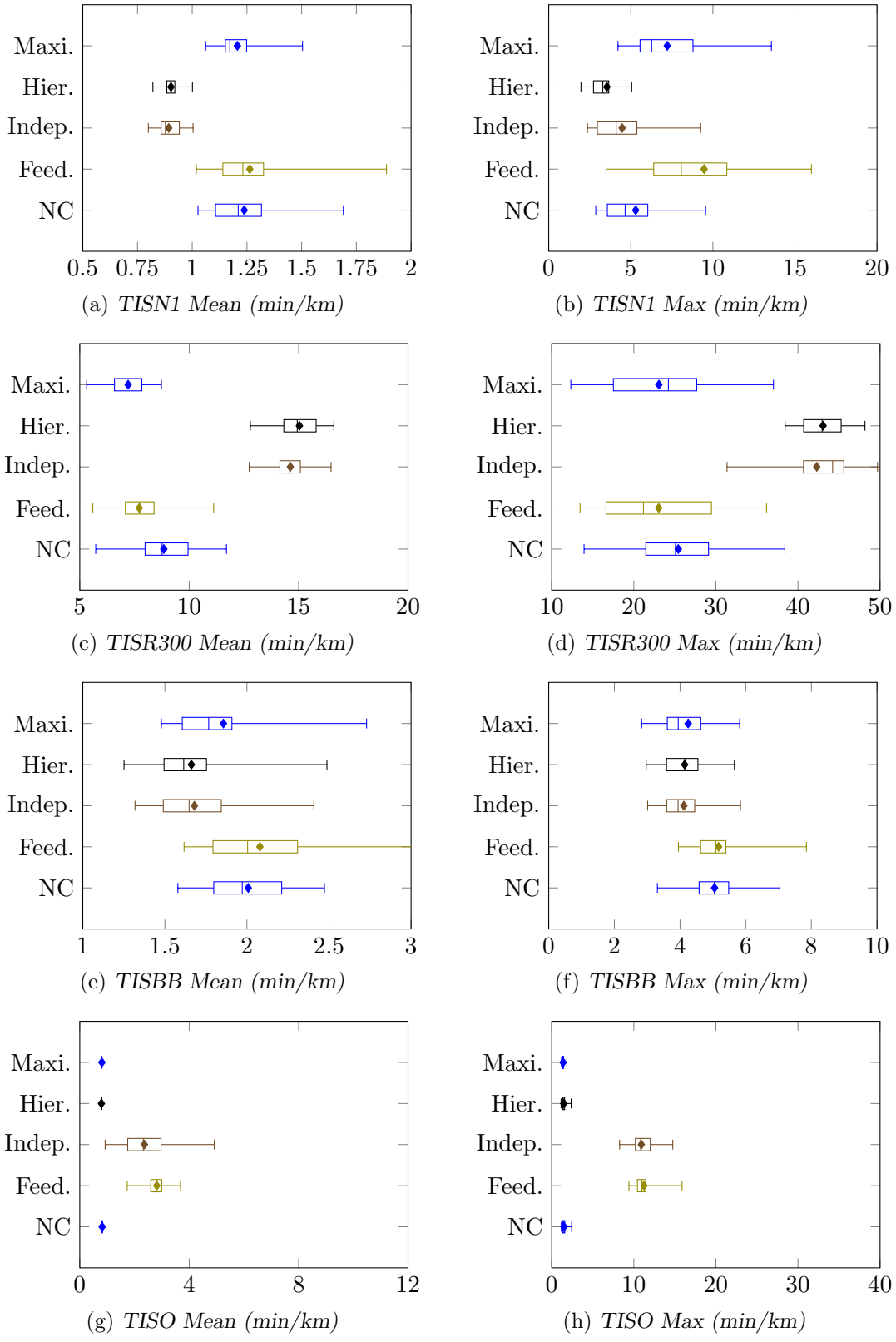
		Fisher LSD test p-values: TTSBB			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	3.5020×10^{-1}	4.6302×10^{-4}	1.3658×10^{-4}	5.2487×10^{-2}
Feedback		—	1.2733×10^{-5}	3.0632×10^{-6}	4.4129×10^{-3}
Independent			—	7.3734×10^{-1}	1.0573×10^{-1}
Hierarchical				—	5.1458×10^{-2}
Mean	69.71	75.25	58.79	57.77	63.75

TABLE 10.74: *Differences in respect of the total time spent in the system by vehicles entering the system from Okavango Road (TTSO) in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSO			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.9980×10^{-15}	2.3059×10^{-9}	5.8702×10^{-2}	9.2744×10^{-1}
Feedback		—	1.6378×10^{-1}	$< 1 \times 10^{-17}$	1.1432×10^{-11}
Independent			—	1.6998×10^{-9}	7.0734×10^{-9}
Hierarchical				—	8.4033×10^{-1}
Mean	14.00	47.90	40.58	13.62	15.36

As may have been expected, the order of relative algorithmic performances in respect of the mean and maximum TISN1 PMIs is the same as that in respect of the TTSN1. These trends are clearly visible in the box plots of Figures 10.13(a) and 10.13(b). Independent and hierarchical MARL returned the smallest mean TISN1-values of 0.89 min/km and 0.90 min/km, respectively, outperforming all other algorithms at a 5% level of significance, as may be seen in Table 10.75. The performances of the feedback controller, maximax MARL and the no-control case, on the other hand, were found to be statistically indistinguishable at a 5% level of significance, as they returned mean TISN1-values of 1.26 min/km, 1.21 min/km and 1.24 min/km, respectively. Similarly, in respect of the maximum TISN1, hierarchical MARL, which returned a value of 3.55 min/km, outperformed all other algorithms, except for independent MARL, from which it was found to be statistically indistinguishable at a 5% level of significance, as may be deduced from the p-values in Table 10.76. Independent MARL, which returned a maximum TISN1-value of 4.47 min/km, was also able to outperform both the feedback controller and maximax MARL at a 5% level of significance, while its performance was found to be statistically indistinguishable from that of the no-control case, which returned a maximum TISN1-value of 5.30 min/km. Interestingly, the no-control case was able to outperform both the feedback controller and maximax MARL, which returned maximum TISN1-values of 9.46 min/km and 7.22 min/km, respectively, at a 5% level of significance, while the performances of the latter two were found to be statistically indistinguishable.

In respect of the mean TISR300, the feedback controller and maximax MARL, achieved the best performances, outperforming the no-control case and both independent and hierarchical MARL at a 5% level of significance, while their performances were statistically similar, as is evident from the p-values in Table 10.77. Independent and hierarchical MARL again exhibited the typical increases in the travel times of the vehicles joining the highway from an on-ramp if RM is applied, as they were outperformed by the no-control case in respect of the mean TISR300, while their performances were statistically indistinguishable at a 5% level of significance. As may



(a) *TISN1 Mean (min/km)*

(b) *TISN1 Max (min/km)*

(c) *TISR300 Mean (min/km)*

(d) *TISR300 Max (min/km)*

(e) *TISBB Mean (min/km)*

(f) *TISBB Max (min/km)*

(g) *TISO Mean (min/km)*

(h) *TISO Max (min/km)*

FIGURE 10.13: *Mean and maximum time spent in the system PMI results for the no-control case (NC), the kNN-TD RM algorithm, independent MARL (Indep.), hierarchical MARL (Hier.) and maximax MARL (Maxi.) with queue limits applied to the case study model of Chapter 9.*

be seen from Figures 10.13(c) and 10.13(d), the order of the relative algorithmic performances in respect of the maximum TISR300 is the same as that for the mean TISR300. A closer inspection of the p-values in Table 10.78, however, revealed that in respect of the maximum TISR300, the performances of the no-control case, the feedback controller and maximax MARL were all found to be statistically indistinguishable at a 5% level of significance, while these three implementations were all able to outperform both independent and hierarchical MARL.

Interestingly, in respect of both the mean and maximum travel times of vehicles joining the N1 from the Brackenfell Boulevard on-ramp, only the MARL implementations were able to achieve improvements over the no-control case, as is evident from Tables 10.79 and 10.80. The performances of independent and hierarchical MARL were once again found to be statistically indistinguishable at a 5% level of significance in respect of both of these PMIs. Furthermore, they were able to outperform all other algorithms in respect of the mean TISBB, while outperforming all algorithms, except for maximax MARL, in respect of the maximum TISBB. Maximax MARL achieved the next best performance in respect of both these PMIs, outperforming the feedback controller in respect of the mean TISBB, and outperforming both the feedback controller and the no-control case in respect of the maximum TISBB. The feedback controller was the worst-performing implementation, resulting in increases over the no-control case in both the mean and maximum TISBB PMIs, although these increases were not large enough for their performances to be classified as statistically different at a 5% level of significance. These trends are also evident from the box plots in Figures 10.13(e) and 10.13(g).

As for the TTSO, the hierarchical and maximax MARL implementations were found to perform statistically indistinguishably from the no-control case in respect of both the mean and maximum TISO, as may be seen from the p-values in Tables 10.81 and 10.82. The similarity in the performances of hierarchical MARL, maximax MARL and the no-control case in respect of the mean and maximum TISO is also very clear in Figures 10.13(g) and 10.13(h). The no-control case, hierarchical MARL and maximax MARL were, however, all able to outperform both the feedback controller and independent MARL at a 5% level of significance in respect of both these PMIs, while the performances of the latter two algorithms were found to perform statistically indistinguishable at a 5% level of significance in respect of both these PMIs.

TABLE 10.75: *Differences in respect of the mean time spent in the system by vehicles entering the system from the N1 in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

Algorithm	No Control	Games-Howell test p-values: TISN1 Mean Feedback	Independent	Hierarchical	Maximax
No Control	—	9.7843×10^{-1}	1.0144×10^{-11}	3.7440×10^{-11}	9.1094×10^{-1}
Feedback		—	7.8394×10^{-12}	2.4015×10^{-11}	5.7126×10^{-1}
Independent			—	9.2340×10^{-1}	1.1153×10^{-12}
Hierarchical				—	$< 1 \times 10^{-17}$
Mean	1.24	1.26	0.89	0.90	1.21

10.5.3 Discussion

As may have been expected, due to the fact that additional queue restrictions were not required in the hierarchical MARL and maximax AMRL implementations, these two implementations again achieved the best and second-best performances, respectively, when compared with the independent MARL implementation (with the addition of a queue limit) and the integrated feedback controller. The introduction of a queue limitation in the independent MARL imple-

TABLE 10.76: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the N1 in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISN1 Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.3156×10^{-3}	5.5236×10^{-1}	1.1921×10^{-2}	2.9182×10^{-2}
Feedback		—	7.0162×10^{-5}	3.3300×10^{-6}	1.9395×10^{-1}
Independent			—	2.2869×10^{-1}	1.2634×10^{-4}
Hierarchical				—	2.1628×10^{-2}
Mean	5.30	9.46	4.47	3.55	7.22

TABLE 10.77: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISR300 Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	1.0999×10^{-4}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	4.3726×10^{-8}
Feedback		—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	7.3149×10^{-2}
Independent			—	1.4461×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	$< 1 \times 10^{-17}$
Mean	8.84	7.72	14.62	15.04	7.21

TABLE 10.78: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the R300 in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISR300 Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	6.6832×10^{-1}	9.7866×10^{-13}	4.8428×10^{-13}	6.5593×10^{-1}
Feedback		—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	9.9999×10^{-1}
Independent			—	9.5246×10^{-1}	$< 1 \times 10^{-17}$
Hierarchical				—	1.7319×10^{-13}
Mean	25.42	23.03	42.30	43.05	23.05

TABLE 10.79: *Differences in respect of the mean time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISBB Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	3.7792×10^{-1}	7.7486×10^{-5}	3.1355×10^{-5}	6.2680×10^{-2}
Feedback		—	2.0104×10^{-6}	7.1908×10^{-7}	6.5202×10^{-3}
Independent			—	8.1800×10^{-1}	2.9951×10^{-2}
Hierarchical				—	1.6632×10^{-2}
Mean	2.01	2.08	1.68	1.66	1.86

TABLE 10.80: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISBB Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	6.0806×10^{-1}	1.3213×10^{-4}	1.8996×10^{-4}	9.2513×10^{-4}
Feedback		—	1.7579×10^{-5}	2.6119×10^{-5}	1.4887×10^{-4}
Independent			—	9.2243×10^{-1}	5.8602×10^{-1}
Hierarchical				—	6.5462×10^{-1}
Mean	5.05	5.17	4.12	4.14	4.25

TABLE 10.81: *Differences in respect of the mean time spent in the system by vehicles entering the system from Okavango Road in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Mean			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	3.9970×10^{-15}	8.8718×10^{-10}	6.7536×10^{-1}	9.7001×10^{-1}
Feedback		—	9.3545×10^{-2}	$< 1 \times 10^{-17}$	9.6811×10^{-14}
Independent			—	6.0068×10^{-10}	6.9977×10^{-10}
Hierarchical				—	8.0909×10^{-1}
Mean	0.82	2.82	2.36	0.79	0.81

TABLE 10.82: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Okavango Road in the case of MARL with queue limits. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Max			
Algorithm	No Control	Feedback	Independent	Hierarchical	Maximax
No Control	—	7.7827×10^{-14}	1.4322×10^{-14}	9.9476×10^{-1}	3.1228×10^{-1}
Feedback		—	9.6858×10^{-1}	9.1259×10^{-14}	1.9651×10^{-14}
Independent			—	1.2434×10^{-14}	7.9940×10^{-15}
Hierarchical				—	6.5165×10^{-1}
Mean	1.50	11.23	10.92	1.47	1.39

mentation was again effective in limiting the length of the on-ramp queue at the Okavango Road on-ramp. Interestingly, the independent MARL implementation performed very similarly to the hierarchical MARL implementation in respect of all PMIs, except the TTSO and the mean and maximum TISO PMIs, in respect of which independent MARL was outperformed by the hierarchical MARL implementation, thus providing further evidence for more effective cooperation of the RM and VSL agents if communication between the agents is employed. Although the feedback controller was able to achieve improvements over the no-control case in respect of the TTS and TTSR300, while performing statistically on par with the no-control case in respect of the TTSN1, it was generally considered to be the worst-performing integrated control strategy, as it was consistently outperformed in respect of the TTSN1, TTSBB and TTSO PMIs by all three MARL implementations.

10.6 Chapter Summary

This chapter opened in §10.1 with a description of the various RM implementations within the context of the case study simulation model of Chapter 9. The implementations of the RM agents at the various on-ramps were detailed in §10.1.1, while the focus shifted in §10.1.2 to a parameter evaluation with the aim of determining the best-performing target density values for each of the respective RM agents. This was followed by a statistical algorithmic performance comparison in §10.1.3, while a discussion on some of the key findings of the section was presented in §10.1.4. Thereafter, a queue limitation was introduced in §10.2 aimed at preventing the build-up of excessively long on-ramp queues, and the customary algorithmic performance comparison followed, while the section again closed with a discussion on some of the key findings.

A similar presentation followed in §10.3 for the VSL implementations within the context of the case study model. The algorithmic implementations were outlined in §10.3.1, and this was followed by a parameter evaluation aimed at determining the best-performing target densities in the case of MTFC and the best-performing VSL update rules in the case of the RL implementations in §10.3.2. An algorithmic performance comparison was finally performed in §10.3.3, and a discussion followed, highlighting the key findings in §10.3.4.

The focus shifted in §10.4 to the MARL approaches implemented in the case study model, with a description of the various MARL implementations in §10.4.1. This was followed by a reward function evaluation in §10.4.2 aimed at determining the best-performing combination of reward functions for each of the MARL approaches. The customary algorithmic performance comparison followed in §10.4.3, and the section finally closed in §10.4.4 with a discussion on some of the key findings related to the MARL implementations. Queue limitations were introduced in the context of MARL in §10.5, and this was followed by a thorough algorithmic performance comparison, before the chapter finally closed with a discussion on some of the key findings in respect of MARL with queue limits in §10.5.3.

Part III

Future Technologies

CHAPTER 11

Ramp Metering by Autonomous Vehicles

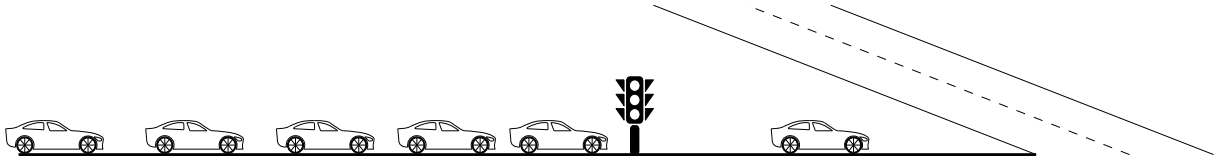
Contents

The purpose of this chapter is to provide a detailed description of the implementation of autonomous vehicles for RM in the context of the benchmark simulation model of §5.1.2. The chapter opens in §11.1 with a description of how varying percentages of *autonomous vehicles* (AVs) are incorporated in the simulation model, as well as an explanation of the novel concept of employing AVs for the purpose of RM. This is followed in §11.2 with a thorough description of this RM problem in the context of RL, which serves as the blueprint for the implementation of the Q-Learning and kNN-TD RL algorithms. An algorithmic parameter evaluation follows in §11.5 after which the performance of RM by AVs is compared to that of conventional RM in §11.6. The chapter finally closes in §11.7 with a brief summary of the work included in the chapter.

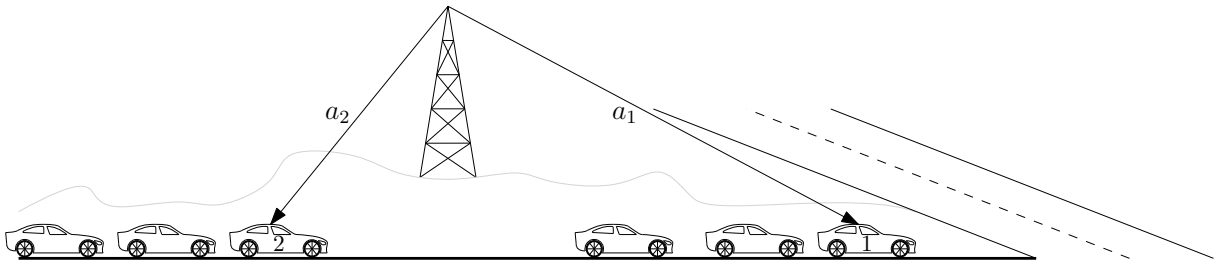
11.1 Autonomous Vehicles for Ramp Metering

Conventionally, RM is enforced by placing a traffic light at an on-ramp which allows one vehicle to enter the highway stream during each green phase. This form of RM was adopted in Chapters 6, 8 and 10. From the literature reviewed in §3.3 it is evident that AVs, once they are commercially available, may effectively be employed in order to improve the traffic flow along the highway. The literature review, however, revealed that in most of the work conducted with a focus on traffic flow improvement by means of detailed instructions to AVs, the focus was on providing instructions only to those vehicles travelling along the highway, while RM, when employed in these studies was implemented in the conventional method. Due to the fact that RM is generally the most effective highway traffic control measure, as is evident from the results presented in Part II of this dissertation, the aim in Part III is to assess the possibilities of the novel notion of employing AVs towards achieving effective RM.

The concept behind the use of AVs for RM is based on vehicle-to-infrastructure communication, where instructions may be given to a vehicle from a local TMC. More specifically, these instructions are speed values at which the AVs should travel while on the on-ramp. It is envisioned that, if these speed values are small enough, the AVs will collectively regulate, to a certain extent, the traffic flow allowed onto the highway in a manner akin to that achieved by the traffic light in conventional RM. A graphical comparison between conventional RM and RM by AVs is shown in Figure 11.1.



(a) Conventional ramp metering



(b) Ramp metering by autonomous vehicles

FIGURE 11.1: *A graphical comparison between (a) conventional RM and (b) RM achieved by two autonomous vehicles (the vehicles numbered 1 and 2).*

As may be seen in the figure, specific actions a_1 and a_2 are communicated to two AVs labelled 1 and 2. According to these actions, the AVs reduce their speeds, while the vehicles behind the AVs are forced to travel at the same lower speed, thereby creating large headways between the last vehicle trailing an AV and the following AV and thus regulating the flow of traffic allowed to enter the highway. Although this approach is not as rigid as conventional RM in the sense that fixed metering rates may be imposed (due to the fact that the specific arrival times of AVs are not known), it is expected that this method may be effective in improving the flow of traffic along the highway, especially as larger and larger numbers of AVs are found on roads, avoiding the long on-ramp queues for which conventional RM is notorious.

11.2 Formulation as a Reinforcement Learning Problem

Due to the fact that RM by AVs is based on the same concept as conventional RM, controlling the highway density by reducing the flow of traffic entering the highway from an on-ramp, the formulation as an RL problem of such an RM by AVs implementation takes a very similar form as that of the conventional RM formulation. As shown in Figure 11.2, each autonomous vehicle performs two actions, a_1 and a_2 while travelling along the on-ramp. The first of these $a_1 = V_{\text{RM}}$, dictates the speed at which the AV should travel along the on-ramp for a distance ℓ_{OR}, until it receives an instruction to accelerate to the nominal speed limit $a_2 = V_{\text{HW}}$ applied along the highway. The reason for this acceleration is that the vehicles joining the highway from the on-ramp should be travelling at approximately the same speed as those vehicles already travelling along the highway in order to facilitate a smooth merge of the two traffic flows.

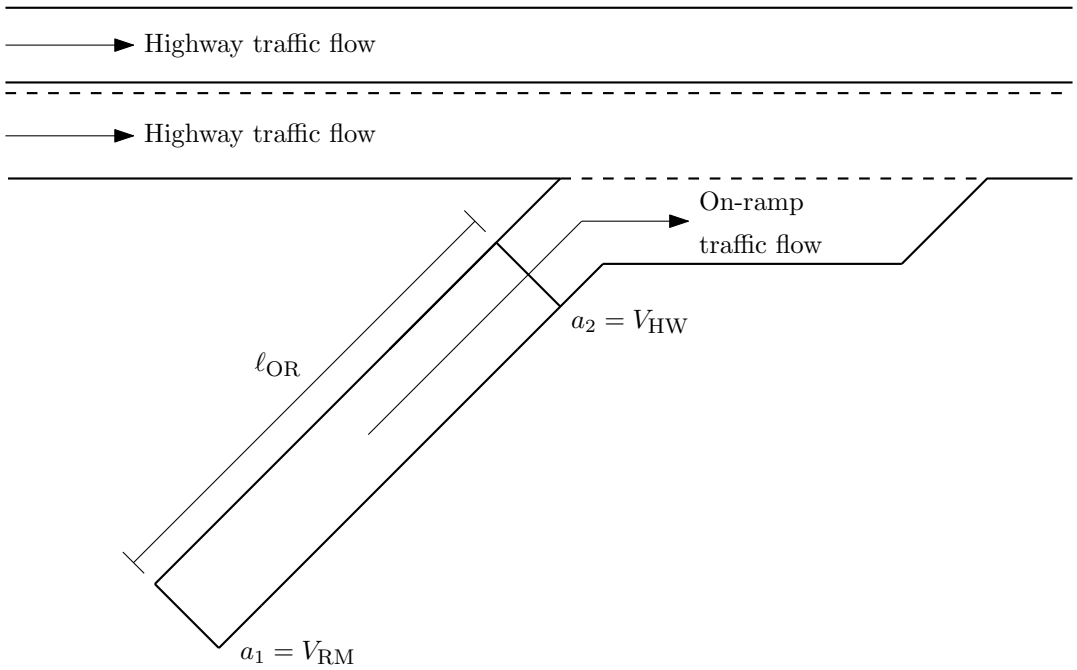


FIGURE 11.2: *The RM by AVs implementation adopted within the benchmark model of §5.1.2.*

11.2.1 The State Space

Due to the success of the RL implementations for conventional RM, the same state space as for the conventional RM implementation is adopted for the implementation with AVs, as may be seen in Figure 11.3. The first state is again the density ρ_{ds} directly downstream of the on-ramp, which was, as for the conventional RM implementation, selected as it provides the learning agent with direct feedback in terms of the quality of the previous action (this density provides information about the state of traffic flow at the bottleneck, and subsequently is the earliest indicator of impending congestion).

The second state variable is the density ρ_{us} upstream of the on-ramp. This state was again included to provide the learning agent with an indication of the impending traffic demand, as well as an indication of the severity of the congestion, if any, and how far the congestion may have propagated backwards along the highway.

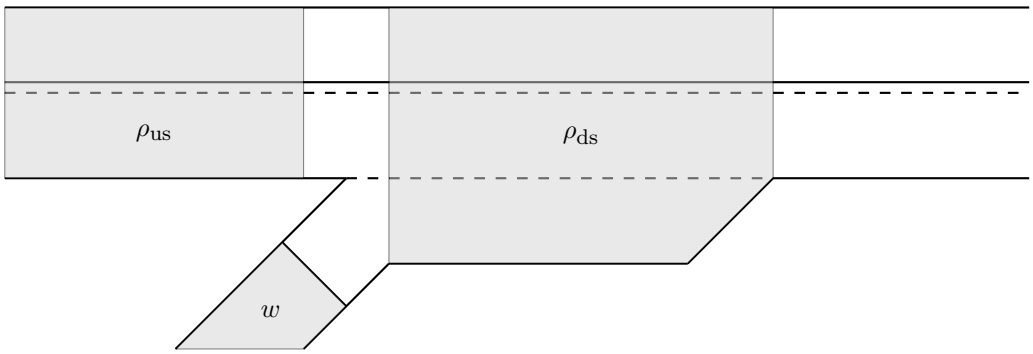


FIGURE 11.3: *A representation of the state space for the RM by AVs problem in the context of the benchmark model of §5.1.2.*

The third and final state variable is again the on-ramp density and queue length, w. This variable was again selected so as to provide the learning agent with an indication of the traffic situation on the on-ramp, where the actions specified by the agent are performed. It is therefore expected that the on-ramp density may also be an important descriptor of the state space, similarly to the downstream density providing an indication of the quality of the action chosen.

11.2.2 The Action Space

As in both the RM and VSL implementations in Chapters 6 and 7, a direct action selection policy is applied in the context of RM by AVs in this dissertation, in pursuit of a faster learning rate. The RM agent may choose an action a_1 from the set of actions $\mathcal{A} = \{10, 20, 30, 40, 50, 60, 70, 80\}$, where each action denotes a discrete speed value in km/h at which the AV should travel along the length ℓ_{OR} of the specified section of the on-ramp, as illustrated in Figure 11.2. Note that in Figure 11.2, a second action, a_2 is communicated to the AV. This action is fixed, as AVs are instructed to speed up to the nominal speed limit applied on the highway network at the on-ramp merge. The learning agent therefore only chooses a single action a_1 which is then communicated to the AVs.

11.2.3 The Reward Function

Due to the success of the density-based method of rewarding the RL agent in the context of the conventional RM implementation of Chapter 6, the same approach towards providing feedback to the RL agent is adopted for the case where the RM is enforced by AVs. The goal of the agent remains to minimise the total time spent in the system by all vehicles, which may be achieved by maintaining a density close to the critical density at the bottleneck which, in turn, results in maximum throughput being achieved at the bottleneck. Therefore the reward function defined for the conventional RM RL agent in (6.2), which punishes the RL agent for deviations from the critical density, is also adopted for the RM by AVs implementation.

11.3 Q-Learning for Ramp Metering by AVs

As in the RM and VSL implementations, the state space for the Q-Learning implementation is again discretised so as to facilitate a tabular representation of the state space and the resulting action-value function $Q(s, a)$. The downstream density is discretised into $n_{\rho_{\text{ds}}} = 10$ equi-spaced

intervals. The upstream density is similarly discretised into $n_{\rho\text{us}} = 10$ equi-spaced intervals. The on-ramp queue length is finally discretised into nine intervals according to

$$n_w = \begin{cases} 0.9 & \text{if } \frac{w}{100} > 0.8, \\ 0.8 & \text{if } \frac{w}{100} > 0.7, \\ 0.7 & \text{if } \frac{w}{100} > 0.6, \\ 0.6 & \text{if } \frac{w}{100} > 0.5, \\ 0.5 & \text{if } \frac{w}{100} > 0.4, \\ 0.4 & \text{if } \frac{w}{100} > 0.3, \\ 0.3 & \text{if } \frac{w}{100} > 0.2, \\ 0.2 & \text{if } \frac{w}{100} > 0.1, \text{ and} \\ 0.1 & \text{if } \frac{w}{100} \geq 0. \end{cases} \tag{11.1}$$

This discretisation differs from the one implemented for conventional RM and is clustered around smaller queue length values. This discretisation was adopted as it is expected that the on-ramp queues will not reach the same lengths in the RM by AVs implementation as they did for the conventional RM implementation, due to the fact that vehicles travelling along the on-ramp will never come to an absolute stop. This discretisation results in a state space consisting of $|n_{\rho_{\text{ds}}}| \times |n_{\rho\text{us}}| \times |n_w| = 900$ states. A table-based approach to Q-value approximation is again adopted, as was the case for both conventional RM and VSLs, employing AnyLogic's built-in Microsoft SQL Server functionality. Q-Learning is implemented within the benchmark model of §5.1.2 as outlined in Algorithm 2.3. In order to find an effective trade-off between exploration of the state-action space and exploitation of that which has already been learnt by the agent, the same rules for determining an adaptive α-value and adaptive ϵ-value as given in (6.5) and (6.6), respectively, are employed in the Q-Learning implementation for RM by AVs.

11.4 kNN-TD learning for Ramp Metering by AVs

Due to the fact that maximum vehicle throughput is achieved at the critical traffic density, the centres chosen for both the downstream density and the upstream density should be clustered around the critical density value so as to be able to provide more accurate approximations of the action value when the measured density is close to the critical density. The critical density of highway segments is typically around 28 vehicles/km [130]. As a result, the downstream centres were chosen as $\{15, 22, 25, 27, 29, 33, 38, 45, 55, 70\}$, while the centres for the upstream density are placed at $\{12, 20, 25, 30, 70, 75, 80\}$. Note that these centre-values are the same as those employed in the conventional RM implementation. The centres for the on-ramp queue length, however, differ from those in the conventional RM implementation due to the expectation that the queue build-up on the on-ramp will not be as severe as it was in the conventional RM implementation. The on-ramp centres were therefore chosen as $\{3, 5, 7, 9, 11, 15, 20, 30, 50\}$. The lookup table used for storing and updating the centre-action values was, as in the case of all previous RL implementations, created using AnyLogic's built-in database functionality. The learning rate α is again determined as in (6.5), while the state-dependent ϵ-value is calculated according to (6.8), as was the case in the kNN-TD learning implementation of conventional RM.

11.5 Parameter Evaluation

A thorough performance evaluation of the novel RM technique enforced by AVs is performed in this section. The section opens in §11.5 with a parameter evaluation, aimed at determining the

best-performing target density values for both the Q-Learning and kNN-TD implementations. Once this target density value has been found, the focus shifts to the effect that various other parameters, such as the length of the on-ramp, the percentage of AVs present in the traffic flow, and the traffic demand have on the performance of RM by AVs.

11.5.1 Target Density Parameter Evaluation

The focus in this section is on determining the best-performing target densities for the Q-Learning and kNN-TD learning algorithms in respect of the RM by AV implementations. RM by AVs may be employed in two different ways. In the first of these, the speed at which the AV should travel is determined at fixed time intervals (such as the red phase time in conventional RM). This fixed time is again set to two minutes, as was the case in the conventional RM implementation. In the second case, the speed at which the AV should travel along the on-ramp is determined at the specific point in time when an AV enters the on-ramp. Every AV thus receives an individual action indicating the speed at which it should travel along the on-ramp. The best-performing target densities for both of these approaches are determined in this section for the Q-Learning and kNN-TD learning implementations. For the purposes of this evaluation, the traffic flow comprises 10% autonomous vehicles and 90% human-driven vehicles. Furthermore, the underlying geometry of the benchmark simulation model of §5.1.2 remains unchanged, and as such, the simulated on-ramp has a length ℓ_{OR} of 250 metres.

Q-Learning

For the sake of consistency, the parameter evaluations for determining the target densities in respect of RM by AVs were, again, performed in the context of Scenario 2 of §5.3.2. As in the conventional RM implementations in Chapter 6, an initial rough parameter evaluation of target densities between 24 veh/km and 34 veh/km was performed in order to determine the best-performing target densities in the case of RM enforced by AVs. Following this initial investigation, the target densities 22 veh/km and 23 veh/km were also considered. In the case where the speed limits are determined at fixed two-minute intervals, the best performing target density, as determined during this initial rough parameter evaluation, was 23 veh/km. Therefore, the surrounding unit interval was considered in increments of 0.1 veh/km. The results of this finer parameter evaluation are shown in Table 11.1. As may be seen in the table, the smallest TTS-value was achieved when setting the target density to 23.1 veh/km.

TABLE 11.1: *Parameter evaluation results for the time-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

Target density $\hat{\rho}$						
22.0	22.5	22.6	22.7	22.8	22.9	23.0
873.91	865.01	933.79	866.70	864.97	848.46	867.47
Target density $\hat{\rho}$						
23.1	23.2	23.3	23.4	23.5	24.0	
845.91	880.60	881.92	869.79	892.10	908.29	

This process was repeated for the Q-Learning implementation where AVs entering the on-ramp in the simulated area trigger the RL algorithm and receive individually determined on-ramp speed assignments. The initial rough parameter evaluation of target densities revealed that in

the vehicle-triggered implementation, the smallest TTS-value was achieved when setting the target density to 24 veh/km. Therefore, the target density of 23 veh/km as well the target densities in the unit intervals around 24 veh/km were investigated in intervals of 0.1 veh/km, as may be seen in Table 11.2. From the results in the table it is evident that the best-performing target density was 23.7 veh/km, achieving a TTS-value of 815.05 veh·h, which is a 28.62% improvement over the no-control case.

TABLE 11.2: *Parameter evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

Target density $\hat{\rho}$						
23.0	23.5	23.6	23.7	23.8	23.9	24.0
856.60	828.35	834.44	815.05	854.40	867.80	820.04
Target density $\hat{\rho}$						
24.1	24.2	24.3	24.4	24.5	25.0	
822.91	835.79	852.90	854.19	836.94	881.48	

From the results in Tables 11.1 and 11.2, it is evident that in the Q-Learning implementation of RM enforced by AVs, employing the strategy where vehicles entering the on-ramp trigger the Q-Learning algorithm consistently achieved smaller TTS-values than the case where Q-Learning is triggered at distinct points in time, and all AVs entering the on-ramp during that interval receive the same on-ramp speed instruction.

*k*NN-TD Learning

As with the Q-Learning implementations, the effectiveness of the *k*NN-TD algorithm for RM by AVs, triggered at pre-specified time intervals was investigated in unit intervals for target densities ranging from 24 veh/km to 34 veh/km. This initial investigation indicated that the smallest TTS-value was achieved at a target density of 25 veh/km. Therefore, the surrounding unit intervals were again investigated more closely in intervals of 0.1 veh/km. The results of this investigation are presented in Table 11.3. As may be seen in the table, the target density corresponding to the smallest TTS-value remained 25 veh/km.

TABLE 11.3: *Parameter evaluation results for the time-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

Target density $\hat{\rho}$						
24.0	24.5	24.6	24.7	24.8	24.9	25.0
864.76	864.10	866.25	888.74	863.17	879.96	844.55
Target density $\hat{\rho}$						
25.1	25.2	25.3	25.4	25.5	26.0	
850.65	851.77	859.21	877.56	895.86	872.96	

The effectiveness of the case where the *k*NN-TD learning algorithm is triggered by AVs entering the simulated on-ramp area was again investigated in target density intervals of 1 veh/km for the densities ranging from 24 veh/km to 34 veh/km. In the vehicle-triggered case, this initial investigation indicated that the smallest TTS-value could be achieved when setting the target density to 24 veh/km, and subsequently the unit intervals around 24 veh/km were, once again,

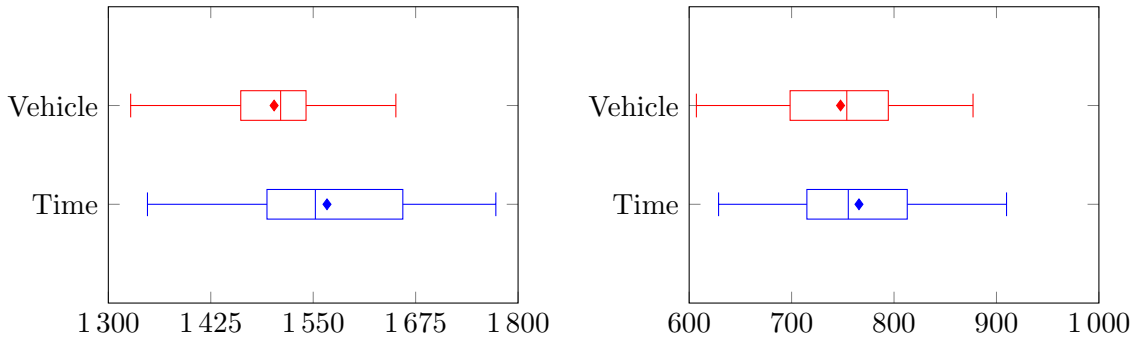
investigated in intervals of 0.1 veh/km. This finer investigation revealed that setting the target density to 24 veh/km did, indeed, result in the overall smallest TTS-value, as may be seen in Table 11.4.

TABLE 11.4: *Parameter evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

Target density $\hat{\rho}$						
23.0	23.5	23.6	23.7	23.8	23.9	24.0
845.02	843.78	857.82	848.08	840.88	814.74	807.09
Target density $\hat{\rho}$						
24.1	24.2	24.3	24.4	24.5	25.0	
847.30	865.83	853.20	857.15	845.57	841.22	

As was the case in the Q-Learning implementations, the implementation where the kNN-TD algorithm is triggered by AVs entering the on-ramp again consistently achieved smaller TTS-values than the case where the learning algorithm is triggered at fixed time intervals. An explanation for this observation may be that, due to the fact that an updated state estimate is employed every time when determining a speed limit for an AV in the vehicle-triggered case, the resulting actions may be chosen more accurately for the current traffic situation than in the time-triggered cases.

Due to the fact that, in cases of large on-ramp traffic demand, as well as increased percentages of AVs in the traffic flow, the inter-arrival times of AVs may become relatively short, there exists a danger that the state estimation accuracy may decrease, as the on-ramp demand and AV percentage increase. This may be the case because the density estimation is calculated as the average densities measured on these highway sections since the previous learning iteration. In order to assess whether increased traffic demand at the on-ramp and increased penetration of AVs in the traffic flow have an impact on the finding that vehicle-triggered RM by AVs performs better than time-triggered RM by AVs, the same comparison was performed, considering the kNN-TD RM implementation (in which the state estimation is performed in smaller intervals, resulting in a greater probability of inaccurate state estimations) in respect of Scenarios 1 and 3 of §5.3.2 (which present the largest on-ramp traffic demand) with an AV penetration of 30%. The results of this comparison are presented in Figure 11.4.



(a) *Scenario 1* (b) *Scenario 3*

FIGURE 11.4: *Total time spent in the system by vehicles (in veh·h) in the vehicle and time-triggered RM by AV implementations in the context of (a) Scenario 1 and (b) Scenario 3, with a traffic flow comprising 30% AVs and 70% human-driven vehicles.*

As may be seen in the figure, even in the cases of large on-ramp demands, and an increased AV penetration rate, the vehicle-triggered implementations were able to achieve smaller mean TTS-values than the time-triggered implementations. Furthermore, an overall improvement in performance in respect of the TTS was observed for the vehicle-triggered implementation when compared with the time-triggered implementation. In Scenario 1, the vehicle-triggered RM implementation also resulted in a reduced variance when compared with the time-triggered implementation, indicating a more stable traffic flow when the vehicle-triggered RM is employed. Due to the fact that the vehicle-triggered RM implementations consistently performed better than the time-triggered implementations, all further comparisons and parameter evaluations conducted in this chapter are performed in respect of the vehicle triggered kNN-TD and Q-Learning implementations. Finally, target densities of 23.7 veh/km and 24.0 veh/km are employed in all further comparisons involving the Q-Learning and kNN-TD implementations, respectively, conducted in this chapter.

11.5.2 On-ramp Length Parameter Evaluation

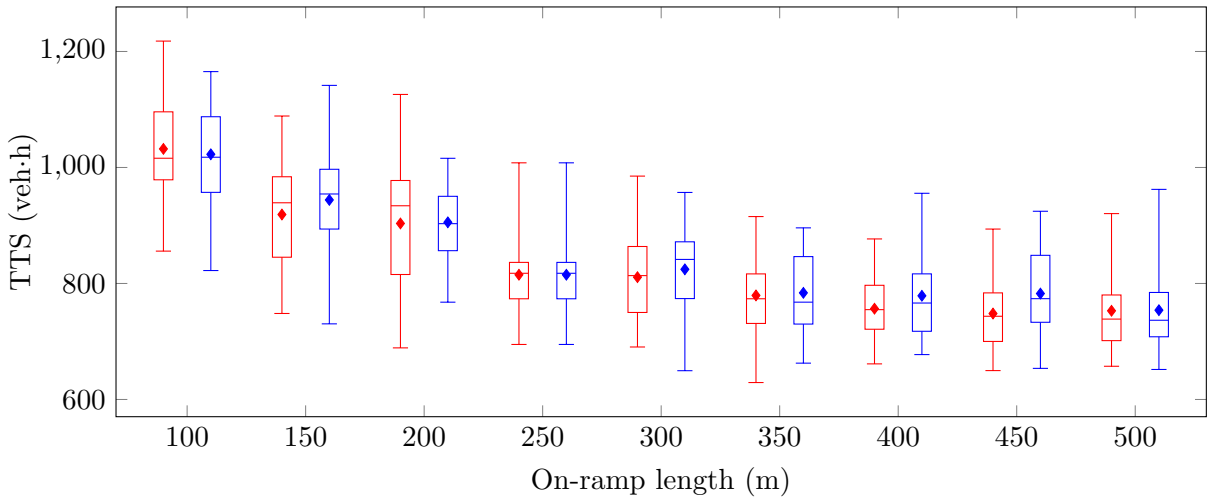
Due to the fact that the RM is now applied by AVs travelling slowly along the on-ramp, the on-ramp length is expected to have a significant effect on the effectiveness of the RM applied. The aim in this section is therefore to assess how sensitive RM by AVs is to changes in the on-ramp length, as well as the type of relationship that relates the on-ramp length to the performance of RM by AVs. This evaluation was again performed for both the Q-Learning and kNN-TD implementations. Similarly to all prior parameter evaluation experiments performed in respect of the benchmark model of §5.1.2, this evaluation is performed in Scenario 2 of §5.3.2. As for the target density parameter evaluation, the proportion of AVs present in the traffic flow was taken as 10%.

Q-Learning

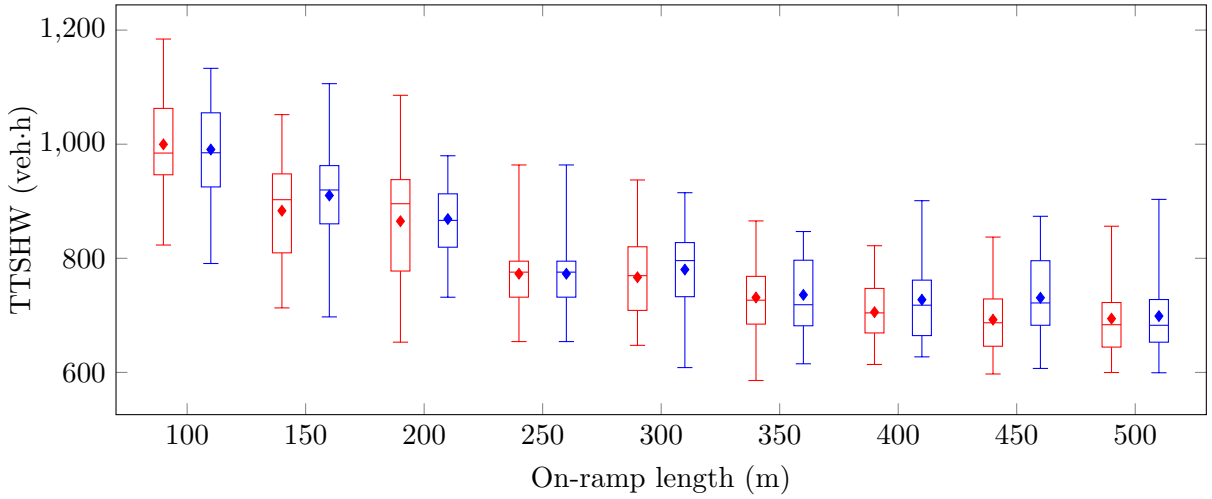
In order to assess the effect that varying on-ramp lengths have on the effectiveness of RM by AVs, performance of RM by AVs in Scenario 2 of §5.3.2 was evaluated in 50 metre intervals for on-ramp lengths ℓ_{OR} ranging from 100 metres to 500 metres. Furthermore, in order to assess the rigidity of policies generated by the Q-Learning algorithm, the performances of individually trained policies (*i.e.* policies that were obtained by training the algorithm at each of the on-ramp length intervals) are compared with the performance of extrapolated policies (*i.e.* policies that were obtained by training Q-Learning with an on-ramp length of 250 metres, and subsequently applying that policy in all scenarios with differing on-ramp lengths). A summary of these algorithmic performances may be seen in Figure 11.5.

As may be seen in Figure 11.5(a), a definite improvement in the TTS-values was observed as the on-ramp length increased. This improvement was, however, expected due to the fact that in the case of a longer on-ramp, an AV naturally travels along the on-ramp for a longer period of time, providing greater opportunity to slow down the traffic flow, thereby reducing the traffic flow onto the highway and effectively resulting in a larger metering rate.

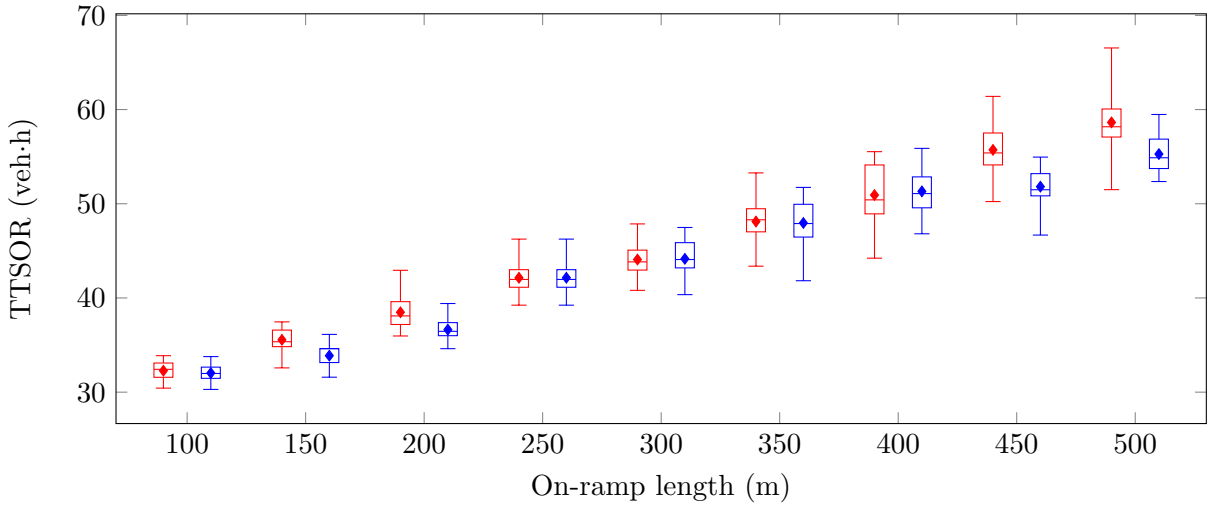
As expected from an RM implementation, the improvements in respect of the TTS were achieved by those vehicles travelling along the highway, because the flow of vehicles onto the highway from the on-ramp is now metered. This trend is clearly visible in the box plots in Figure 11.5(b). Interestingly, however, the largest rate of improvement was observed between on-ramp lengths of 100 and 250 metres, after which, although improvements were still observed, the rate of improvement clearly slowed. A possible reason for this levelling in the performance in respect



(a)



(b)



(c)

FIGURE 11.5: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying on-ramp lengths in Scenario 2.*

of the TTSHW may be that due to the increased on-ramp length, two or more AVs may be present on the on-ramp at any given point in time, which implies that the maximum possible metering by each of these vehicles is applied (*i.e.* no AV can hold up any more human-driven vehicles on the on-ramp, because all human-driven vehicles are already effectively in a platoon behind an AV), while the small improvements that are observed may be due to the fact that the vehicles now spend longer times on the on-ramp before entering the highway traffic flow due to the increased on-ramp length, making the RM marginally more effective.

In respect of the travel times by vehicles entering the highway from the on-ramp, an approximately linear increase in travel times is observed as the on-ramp length increases, as may be seen from the box plots in Figure 11.5(c). Again, this increase may have been expected, as the travel times for vehicles on the on-ramp are expected to increase as a result of being held up by AVs on longer stretches of the on-ramp as the on-ramp length increases. Due to the fact that the on-ramp length increases linearly, the increase in the associated travel times may also have been expected to be linear. Furthermore, an increase in the variances of the TTSOR was observed for longer on-ramp lengths. This was again expected, as a longer on-ramp implies that more vehicles are influenced by the AVs, while the speeds of the AVs depend on the prevailing traffic conditions which are naturally stochastic in nature.

Finally, as is clearly evident from the box plots in Figure 11.5, the performances of the individually trained and extrapolated policies are very similar in respect of the TTS, TTSHW and TTSOR PMIs. This finding is corroborated by the mean values for each of these PMIs in Table 11.5, although the individually trained policies were generally able to achieve marginally smaller TTS and TTSHW-values than the extrapolated policies.

TABLE 11.5: *Parameter evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

PMI	Policy	On-ramp length (m) 100	150	200	250	300	350	400	450	500
TTS	Extra.	1 022.72	944.01	905.41	815.05	824.39	783.81	778.73	782.61	753.97
	Indiv.	1 032.00	918.89	903.46	815.05	810.88	779.40	756.50	748.14	752.87
TTSHW	Extra.	990.69	910.14	868.77	772.90	780.23	735.85	727.41	730.80	698.69
	Indiv.	999.72	883.33	864.98	772.90	766.80	731.28	705.57	692.42	694.22
TTSOR	Extra.	32.03	33.87	36.64	42.14	44.15	47.96	51.31	51.81	55.28
	Indiv.	32.28	35.56	38.48	42.14	44.08	48.11	50.93	55.72	58.64

*k*NN-TD Learning

As for the Q-Learning implementation, the effectiveness of the policies learnt by the *k*NN-TD algorithm were evaluated in 50 metre intervals ranging from the shortest on-ramp length of 100 metres to the longest on-ramp length of 500 metres. This comparison was again performed within the context of the benchmark simulation model of §5.1.2 with traffic demand as in Scenario 2 of §5.3.2. The rigidity of the policies was again evaluated as the performances of individually trained policies were compared with the performance of the policy learnt with an on-ramp length ℓ_{OR} of 250 metres for all the various on-ramp lengths, as may be seen from the box plots in Figure 11.6.

From the box plots in Figure 11.6(a) it is evident that a definite improvement was recorded in respect of the TTS as the length of the on-ramp increased (as was the case in the Q-Learning implementation), confirming the finding that longer on-ramps lead to more effective RM by AVs, as determined in the Q-Learning implementation. Furthermore, as the traffic conditions along

the highway improve, the traffic flow becomes more stable, as indicated by the smaller variances of the box plots corresponding to the results in the cases of longer on-ramps.

As expected, the improvements observed in the TTS are again due to improvements observed in respect of the TTSHW as the traffic flow along the highway improves as a result of RM at the on-ramp. This trend is again clearly evident from the box plots in Figure 11.6(b). Note again that along with the absolute improvements in respect of the TTSHW, there is also a reduction of variances as the on-ramp length increases, as observed in respect of the TTS as well. Furthermore, the largest rate of improvement was again achieved between the on-ramp lengths of 100 and 250 metres, after which the rate of improvement is visibly reduced. The suspected reason for this is again that multiple AVs are present on the on-ramp at any point in time in the case of longer on-ramps, and as a result, the number of human-driven vehicles caught behind these AVs remains constant.

The trend in respect of the TTSOR is again similar to that observed in the Q-Learning implementation, as there appears to be an approximately linear growth in the TTSOR as the length of the on-ramp increases. Together with the absolute increases in the TTSOR, increases in the variances are also again observed, as the traffic flow on the on-ramp becomes more variable due to the RM by the AVs.

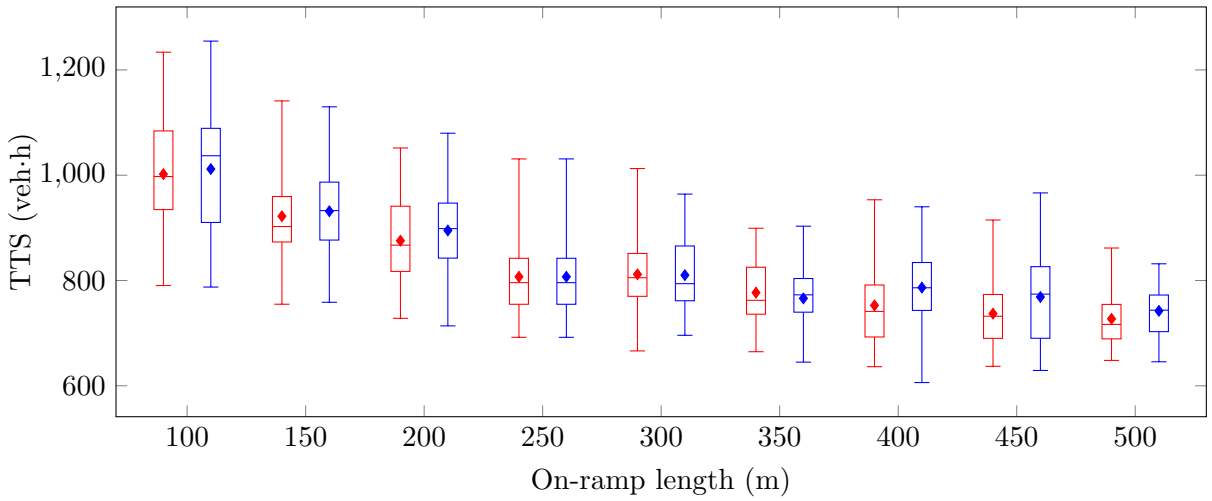
As was the case in the Q-Learning implementation, the performances of the individually trained and extrapolated policies were again very similar in respect of the kNN-TD implementation, as is evident from the box plots in Figure 11.6. The individually trained policies did, however, typically yield smaller mean TTS and TTSHW-values, as may be seen from the mean values presented in Table 11.6, indicating that, as expected, marginal improvements may be achieved by training the algorithms individually for each of the on-ramp length scenarios, while the extrapolation did yield results that were generally very similar to those of the individually trained policies.

TABLE 11.6: *Parameter evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

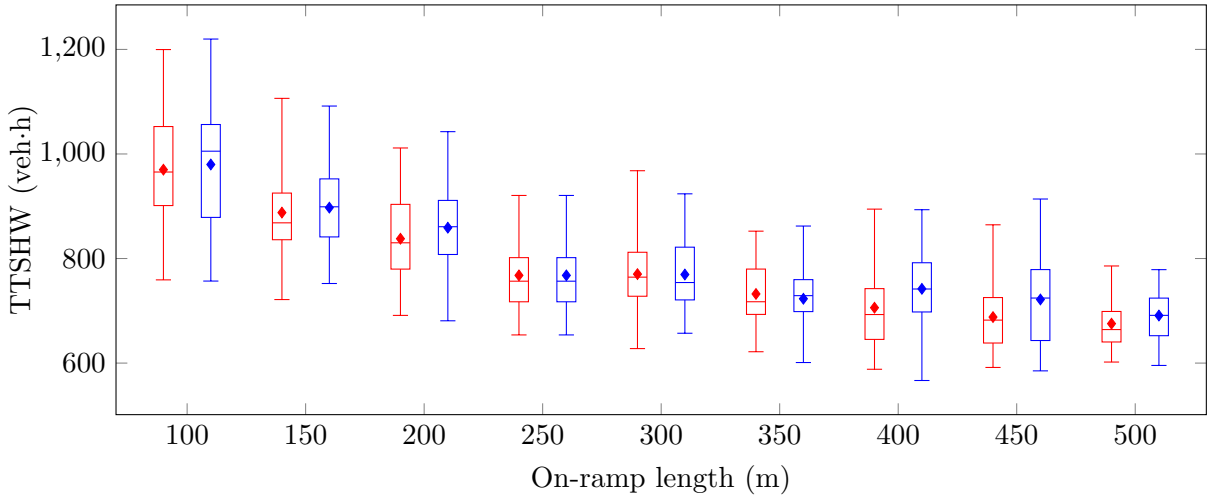
PMI	Policy	On-ramp length (m) 100	150	200	250	300	350	400	450	500
TTS	Extra.	1 011.72	931.55	895.04	807.09	810.21	765.98	786.58	768.63	742.29
	Indiv.	1 002.07	922.03	875.61	807.09	811.73	776.90	752.72	737.07	727.19
TTSHW	Extra.	979.85	897.36	858.97	767.77	769.39	723.02	742.00	721.70	690.98
	Indiv.	969.96	887.85	837.88	767.77	770.21	732.25	705.95	688.03	675.42
TTSOR	Extra.	31.86	34.19	36.07	39.32	40.82	42.96	44.57	46.93	51.31
	Indiv.	32.10	34.19	37.73	39.32	41.51	44.65	46.77	49.04	51.76

11.5.3 AV Percentage Parameter Evaluation

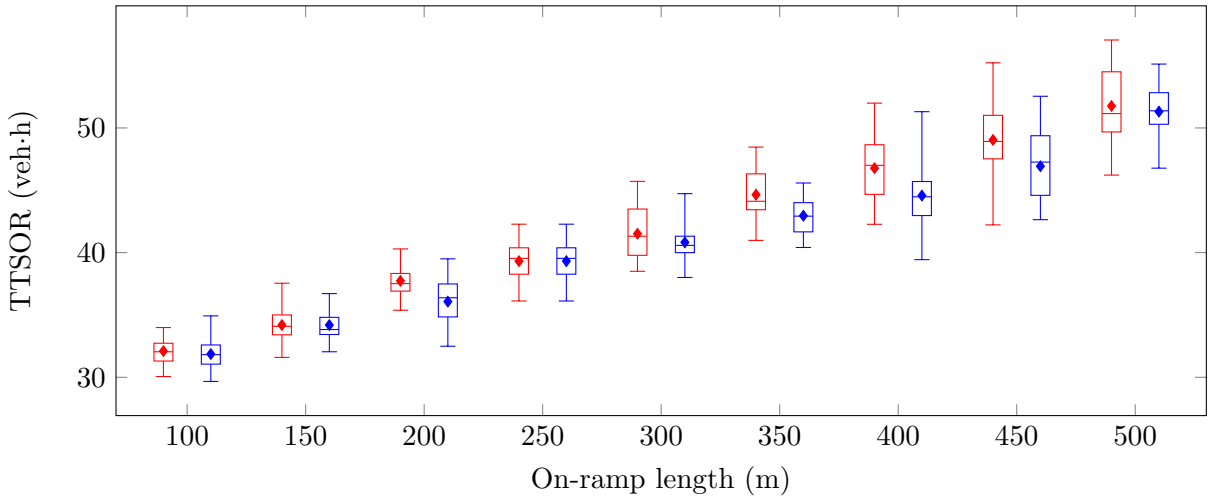
Another parameter that is expected to have a significant effect on the performance of RM by AVs is the proportion of AVs that are present in the traffic flow at any given point in time. In this section the focus shifts to evaluating the effect that increases in the proportion of AVs have on the the performance of RM by AVs, as well as measuring the rigidity of policies in respect of their performances when different percentages of AVs are present. This performance comparison may be valuable in assessing whether the RM agents should be re-trained at specific intervals as the traffic flow composition changes and more AVs are present in the traffic flow, or whether once-off trained policies are robust enough to withstand relatively large variations in the traffic flow composition. This evaluation is again performed in the context of the benchmark



(a)



(b)



(c)

FIGURE 11.6: *A comparison of the performance of kNN-TD learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying on-ramp lengths in Scenario 2.*

model of §5.1.2 and, due to the significant variations in demand between the different scenarios, resulting in large variations in the number of AVs present in the simulation model at any point in time, this evaluation is not restricted to Scenario 2, but is performed in each of the four scenarios of traffic flow of §5.3.2. The on-ramp length ℓ_{OR} was set to 250 metres, as in the original implementation of §5.1.2, for the purposes of this comparison. The choice of 250 metres was also deemed reasonable as this was the point at which the rate of improvement slowed in the parameter evaluations performed in respect of the various on-ramp lengths in the previous section.

Q-Learning

In order to assess the effect of varying the proportion of AVs present in the traffic flow on the performance of RM by AVs, an initial investigation of AV percentages between 2.5% and 20% was performed in intervals of 2.5%. Once this initial investigation had been completed, AV percentages of 25% and 30% were also considered in order to assess whether the trends observed in the initial investigation may be extrapolated accurately in cases where larger proportions of AVs are present in the traffic flow. In order to assess the rigidity of the policies generated, the performance of the policy obtained by training the Q-Learning algorithm with an AV percentage of 10% was compared with the individually trained policies for each of the various AV percentages. The results of this comparison are presented in the form of box plots in Figures 11.7–11.10.

As may be seen in Figures 11.7(a), 11.8(a), 11.9(a) and 11.10(a), clear improvements in respect of the TTS over the no-control case (indicated by the box plots in black) were observed for all AV percentages evaluated in all four scenarios. Interestingly, in Scenarios 2, 3 and 4 the improvement in respect of the TTS represents an approximately exponential decay, as a large rate of improvement is observed for AV percentages ranging from 2.5% to 7.5% after which the performances in respect of all larger AV percentages remain relatively similar. These results are similar to those reported by Schakel *et al.* [141] in their implementation of providing in-car advice in respect of speed, lane and headway to vehicles travelling along a highway, in the sense that in their study significant improvements were also reported with an AV penetration of only 2.5%, while the rate of improvement decayed similarly as the proportion of AVs present in the traffic flow increased. This decay in the rate of improvement may again be explained by the fact that as the proportion of AVs on the on-ramp increases, the number of vehicles affected by each AV decreases, until such time that, at a specific AV percentage, approximately all human-driven vehicles are affected by AVs, and the gains in effectiveness of the RM remain approximately similar.

Although significant improvements were again observed with an AV percentage of only 2.5% in Scenario 1, the improvement in respect of the TTS appears to follow a step function, as the improvements achieved by all different levels of AV percentages are relatively similar. A possible explanation for this observation is that the RM by AVs is already efficient at low penetration rates due to the large on-ramp demands, while the large highway demand limits the overall effectiveness of RM by AVs, because in the case of RM by AVs smaller metering rates than in conventional RM are achievable, and so the desired downstream density cannot be achieved, unavoidably resulting in congestion.

From the box plots in Figures 11.7(b), 11.8(b), 11.9(b) and 11.10(b), it is evident that, as expected, the improvements in respect of the TTS are achieved by the TTSHW, as the trends observed in these box plots are very similar to the trends observed in the corresponding box plots of the TTS. In Scenarios 2–4, the improvements in respect of the TTSHW again take the approximate shape of an exponential decay, as the TTSHW decreases with a corresponding

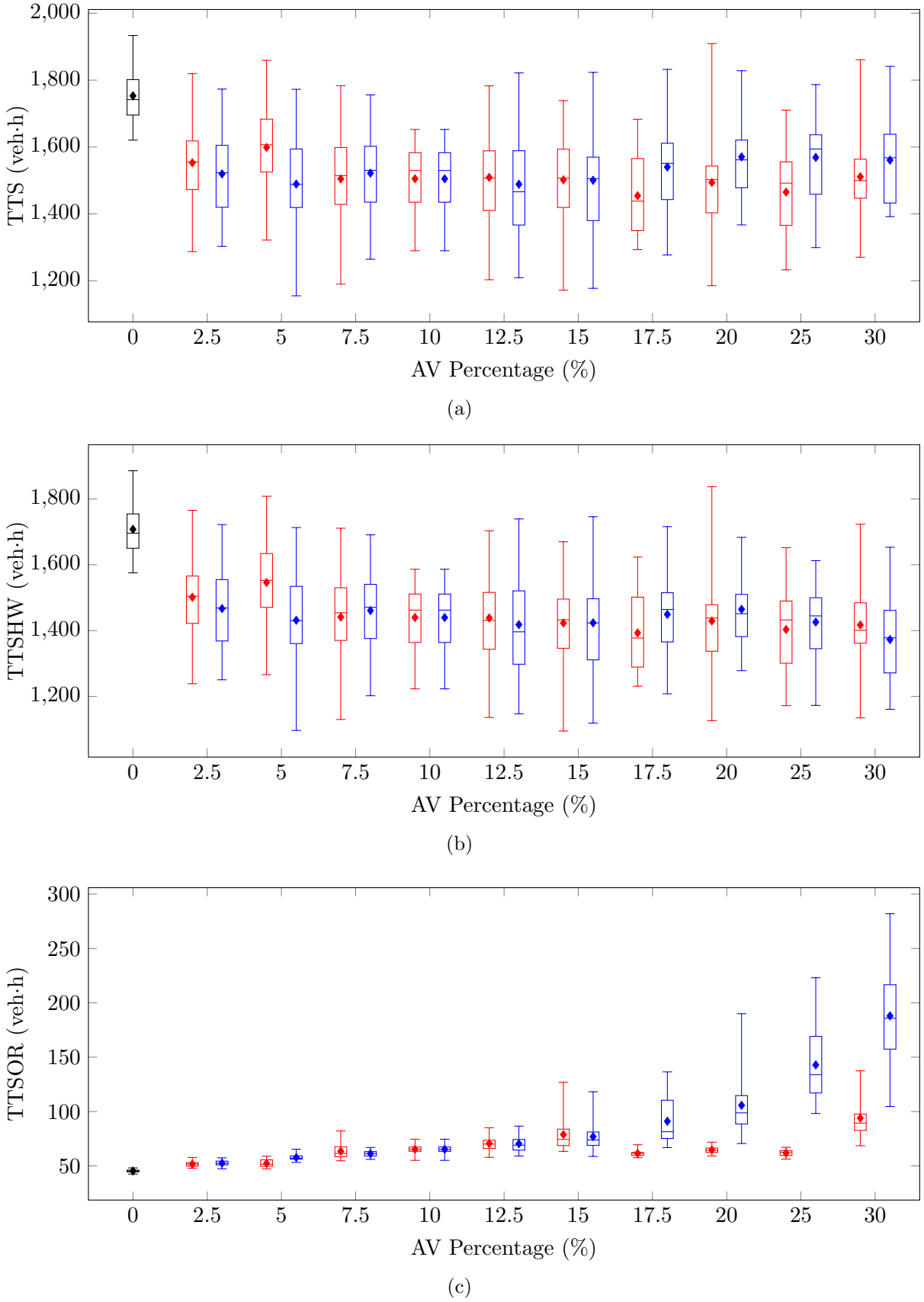
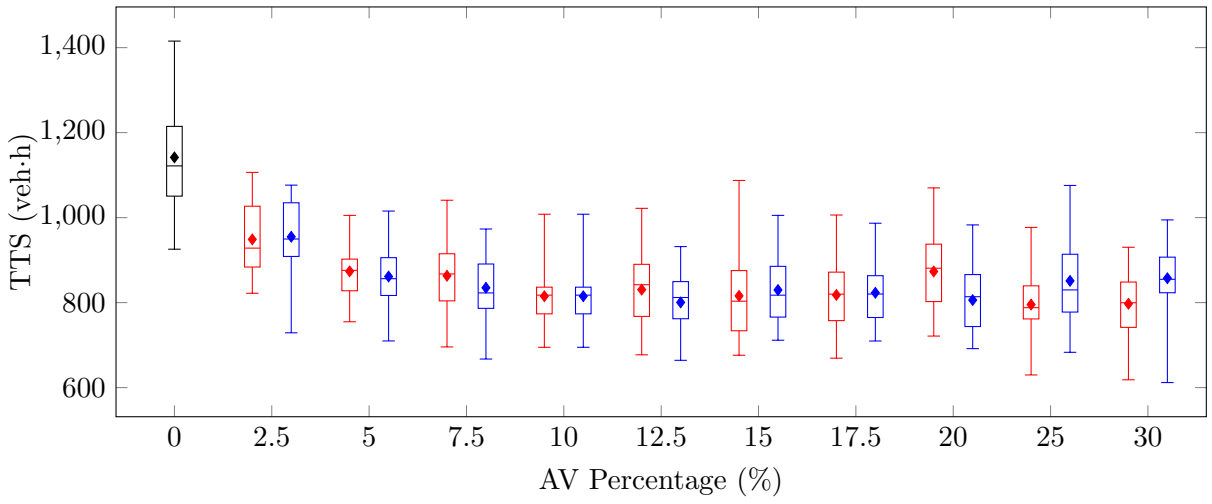
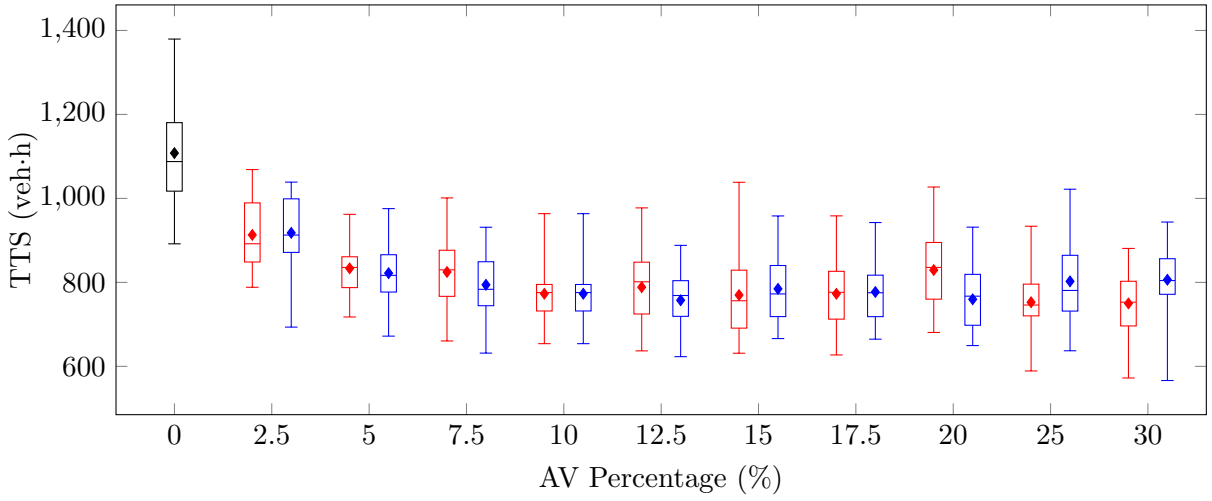


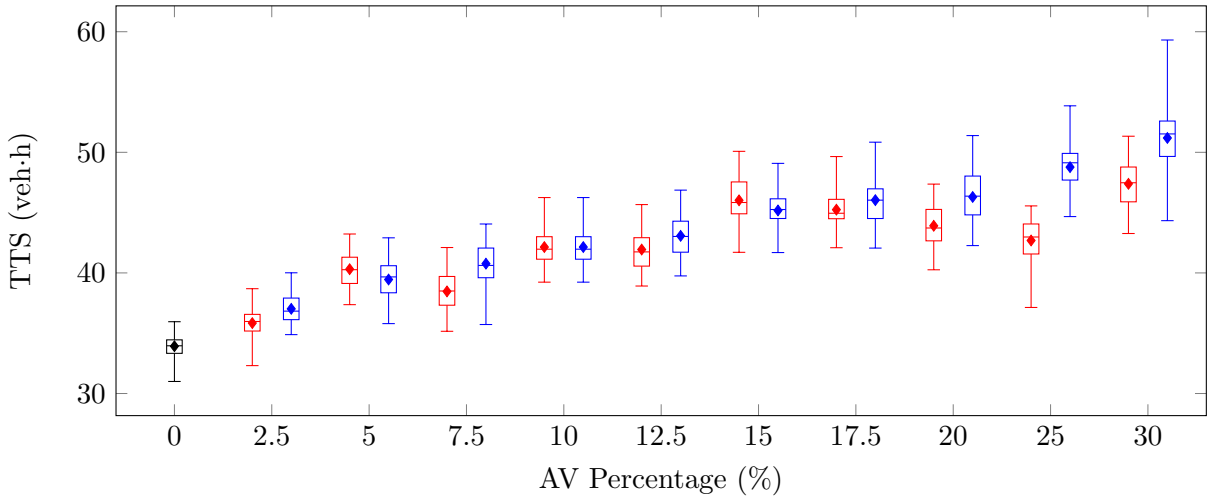
FIGURE 11.7: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 1.*



(a)



(b)



(c)

FIGURE 11.8: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 2.*

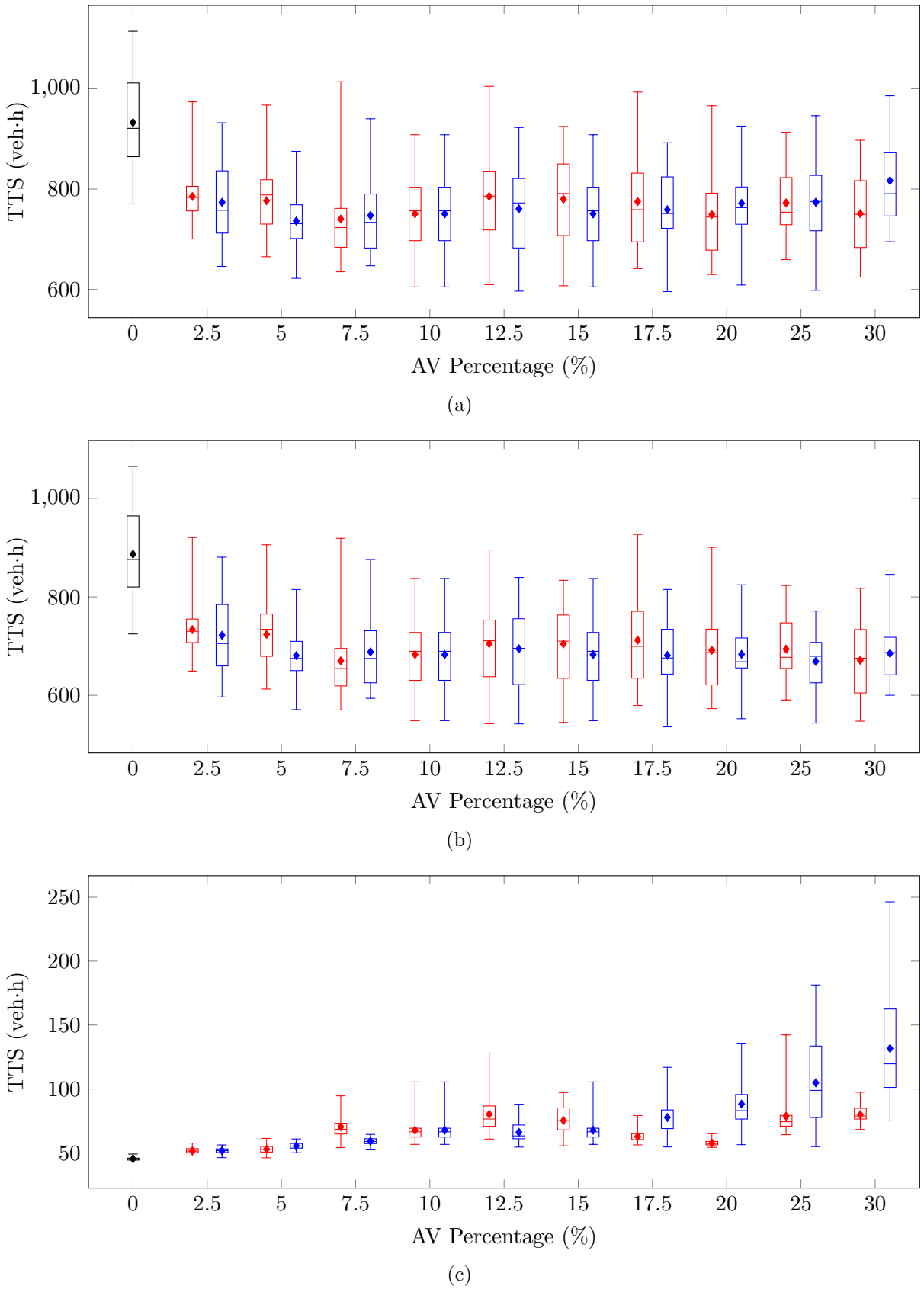
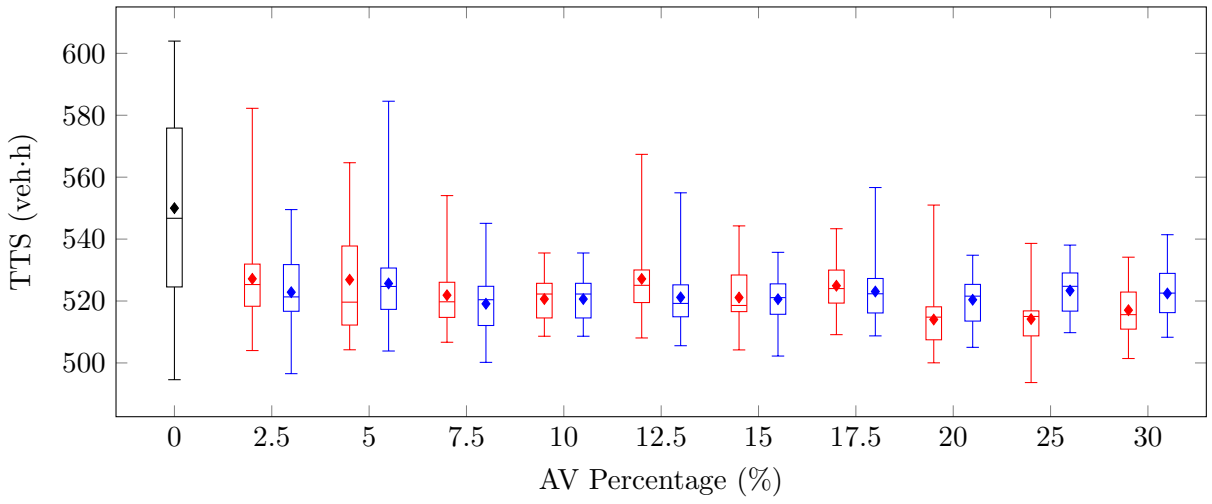
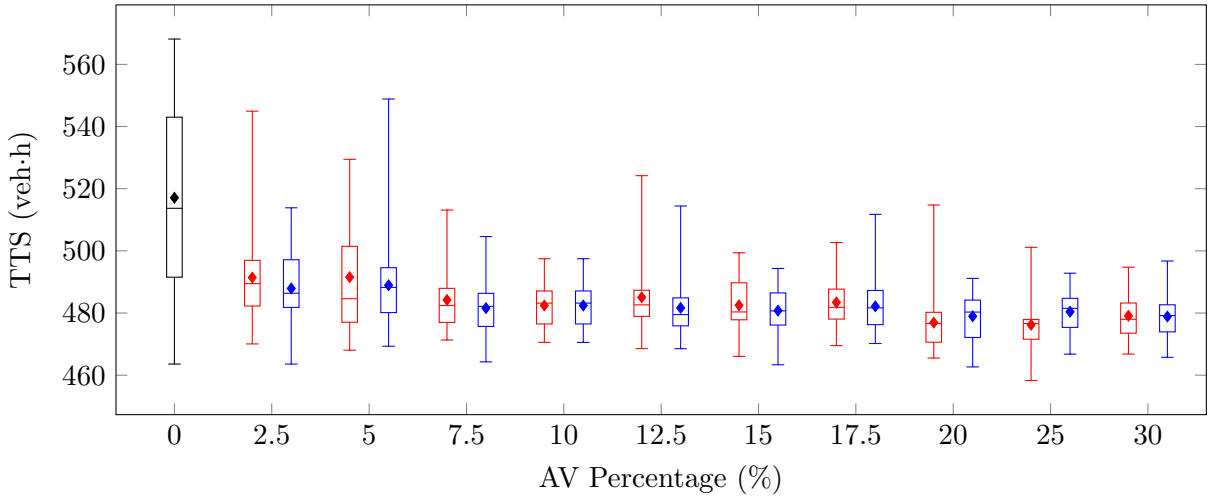


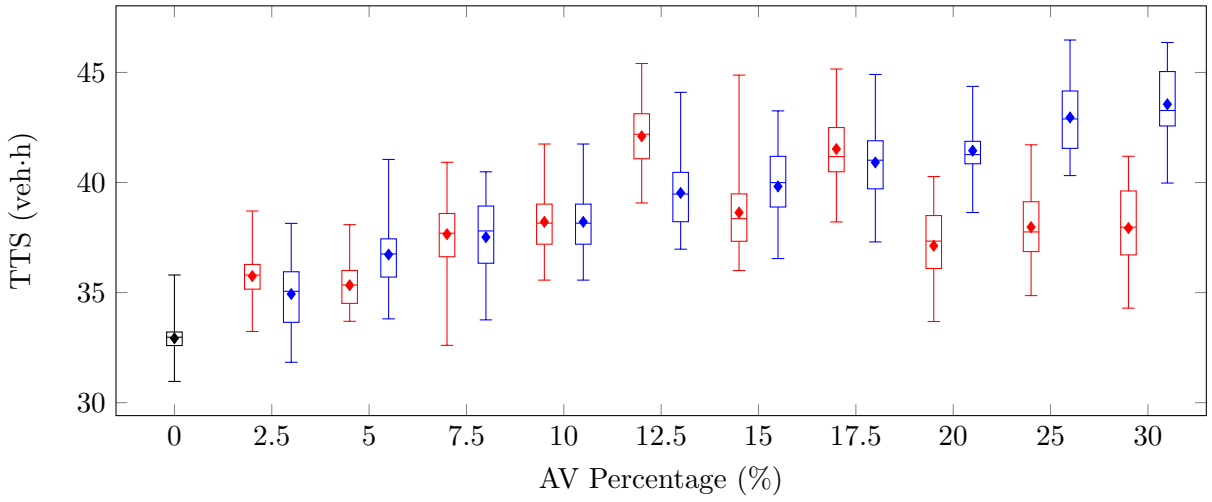
FIGURE 11.9: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 3.*



(a)



(b)



(c)

FIGURE 11.10: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 4.*

increase in the AV percentage, while in the case of Scenario 1, the step decrease is again observed in the TTSHW from the no-control case to the case of 2.5% AVs, and the performances of all the different AV percentages remain relatively similar.

As expected, the improvements in respect of the TTS and TTSHW are achieved at the expense of the travel times of the vehicles joining the highway from the on-ramp. These increases in respect of the TTSOR are clearly visible in the box plots of Figures 11.7(c), 11.8(c), 11.9(c) and 11.10(c). As may be seen in the figures, the common trend in all four scenarios is that there is an increase in the TTSOR as the proportion of AVs in the traffic flow increases. This increase was expected, because as the number of AVs on the on-ramp increases, the number of vehicles influenced by the AVs increases. As a result of the AVs being instructed to travel slowly, this results in an overall increase in the TTSOR as all vehicles joining the highway from the on-ramp travel at lower speeds along the on-ramp.

From the box plots in figures Figures 11.7–11.10 it is evident that the performance of the extrapolated policy is again very similar to the performances of the individually trained policies, especially in respect of the TTS and TTSHW PMIs. This finding is also corroborated by the mean values of these PMIs presented in Tables 11.7–11.10. From the tables it is evident that, generally, the individually trained policies achieved marginally smaller TTS values than extrapolated policies, especially in cases where the AV percentage was relatively high. There is, however, no clear trend that emerges in respect of the individually trained policies always outperforming the extrapolated policies, or *vice versa*, in respect of these PMIs. A possible explanation for this observation is that every state has an intrinsic value that the RL agent learns over time, regardless of the traffic situation, while certain actions are effective in achieving those states, which results in a relatively high robustness in respect of the performances, specifically when considering the TTS and TTSHW PMIs.

From the box plots in respect of the TTSOR, presented in Figures 11.7(c), 11.8(c), 11.9(c) and 11.10(c), however, it is evident that this robustness is not as strong in respect of the vehicles joining the highway from the on-ramp, as the individually trained policies consistently achieved smaller TTSOR-values than the extrapolated policy when the AV percentage exceeded 15%. This observation may be explained by the fact that the extrapolated policy was trained with 10% of the traffic flow being AVs. In order to achieve effective RM when only 10% of the vehicles on the on-ramp are autonomous, the speed limits assigned to these vehicles have to be very small, so that each vehicle can hold up as many vehicles as possible. As the number of AVs on the on-ramp increases, however, larger speeds may be assigned to the AVs while maintaining a similar metering rate, because more vehicles are naturally affected by these AVs as a result of the higher rate of AVs entering the on-ramp. When the RL agent is trained with few AVs on the road, the action resulting in the most desirable next state requires the AV to travel very slowly on the on-ramp according to the policy learnt. This policy is not adapted when compared with the individual policies trained for higher percentages of AVs in the traffic flow, resulting in the case where every AV entering the on-ramp is still assigned a very small speed to travel at, which results in the significantly larger TTSOR-values than those recorded for the individually trained policies due to the increased number of AVs on the road. In the case of the individually trained policies, however, the RL agent learns that the best-performing metering rate may be achieved by assigning larger speeds to the larger number of AVs, thus controlling the on-ramp traffic more effectively, which resulted in the smaller TTSOR-values achieved by the individually trained policies when compared with the extrapolated policies. This observation is especially clear in Scenarios 1 and 3, which represent the highest on-ramp demand, therefore amplifying this effect of the slow-travelling AVs in respect of the TTSOR. These significant increases in respect of the TTSOR brought about by the extrapolated policies are also clearly evident from

TABLE 11.7: *AV percentage parameter evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 1 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	1 519.63	1 489.20	1 522.05	1 505.07	1 488.65	1 500.59	1 540.24	1 570.44	1 568.88	1 560.79
	Indiv.	1 522.63	1 598.59	1 504.65	1 505.07	1 509.04	1 501.88	1 454.39	1 493.93	1 465.15	1 511.05
TTSHW	Extra.	1 467.05	1 431.44	1 460.76	1 439.77	1 418.22	1 423.65	1 449.16	1 464.71	1 425.97	1 372.84
	Indiv.	1 501.25	1 546.23	1 441.17	1 439.77	1 438.45	1 423.04	1 392.85	1 429.21	1 403.30	1 417.04
TTSOR	Extra.	52.58	57.77	61.28	65.30	70.59	76.94	91.07	105.73	142.91	187.95
	Indiv.	51.68	52.35	63.48	65.30	70.59	78.84	61.54	64.73	61.85	94.02

TABLE 11.8: *AV percentage parameter evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 2 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	954.97	861.41	834.94	815.05	800.49	829.73	823.08	805.97	851.17	857.42
	Indiv.	948.81	873.67	863.19	815.05	830.49	815.88	818.04	873.29	795.41	797.25
TTSHW	Extra.	917.95	821.96	794.18	772.90	757.42	784.55	777.04	759.67	802.39	806.23
	Indiv.	912.66	833.36	824.72	772.90	788.55	769.86	772.79	829.39	752.72	749.87
TTSOR	Extra.	37.02	39.45	40.77	42.14	43.07	45.18	46.04	46.30	48.77	51.19
	Indiv.	35.84	40.31	38.47	42.14	41.94	46.02	45.24	43.91	42.69	47.39

TABLE 11.9: *AV percentage parameter evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 3 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	773.61	736.29	747.30	750.67	760.65	750.57	758.85	771.58	773.71	816.79
	Indiv.	785.29	776.80	740.39	750.67	785.31	779.91	775.07	749.25	772.53	751.03
TTSHW	Extra.	721.96	680.75	688.12	682.85	694.59	682.85	681.11	683.37	668.87	685.13
	Indiv.	733.62	723.89	669.99	682.85	705.09	704.60	712.17	691.54	693.86	671.18
TTSOR	Extra.	51.65	55.54	59.18	67.72	66.06	67.72	77.74	88.21	104.85	131.66
	Indiv.	51.67	52.91	70.40	67.72	80.22	75.31	62.89	57.71	78.67	79.85

TABLE 11.10: *AV percentage parameter evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 4 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	522.85	525.67	519.10	520.63	521.18	520.59	523.06	520.40	523.37	522.44
	Indiv.	527.17	526.88	521.88	520.60	527.14	521.13	524.94	514.02	514.17	517.05
TTSHW	Extra.	487.92	488.94	481.58	482.42	481.66	480.77	482.15	478.95	480.42	478.88
	Indiv.	491.42	491.55	484.22	482.42	485.04	482.49	483.41	476.89	476.20	479.11
TTSOR	Extra.	34.94	36.73	37.52	38.21	39.53	39.82	40.91	41.45	42.96	43.55
	Indiv.	35.75	35.34	37.65	38.21	42.10	38.63	41.52	37.13	37.98	37.93

the mean TTSOR-values presented in Tables 11.7–11.10. From the box plots in the figures, it is evident that the performances of the individually trained and extrapolated policies are very similar in cases where the AV percentages are small. The reason for this observation may be that, in order to achieve effective metering rates with the small number of AVs present on the on-ramp, these AVs have to travel very slowly. This behaviour is learnt in both the individually trained policies and the extrapolated policies, due to the fact that in the case of 10% AVs, these AVs have to travel very slowly in order to limit the flow of vehicles from the on-ramp onto the highway.

kNN-TD Learning

The same procedure for assessing the effect of increasing percentages of AVs in the traffic flow on the performance of RM by AVs as employed for Q-Learning was again employed in the case of kNN-TD learning. The performances of the individually trained policies and the extrapolated policies (trained with an AV percentage of 10% in Scenarios 1, 2 and 4, and an AV percentage of 20% in Scenario 3) were again compared in respect of AV percentages ranging from 2.5% to 20% in 2.5% intervals, while AV percentages of 25% and 30% were also again considered. This performance comparison was performed in all four scenarios of traffic demand of §5.3.2 within the context of the benchmark simulation model of §5.1.2. The results of this comparison are presented in Figures 11.11–11.14.

As in the case of the Q-Learning algorithm, the kNN-TD learning implementation of RM by AVs was able to achieve improvements over the no-control case (indicated in the box plots in black) in respect of the TTS in all four scenarios, in all of the evaluations performed in respect of the percentage of AVs present in the traffic flow, as may be seen in the box plots in Figures 11.11(a), 11.12(a), 11.13(a) and 11.14(a). In Scenarios 2–4, an approximately exponential decay is again observed in respect of the improvements in respect of the TTS, while in Scenario 1, a similar step in performance as that recorded for the Q-Learning implementation is again observed. The exponential decay in the TTS corresponding to the increase in AV percentage may again be explained by the fact that the RM becomes increasingly effective as more AVs are present on the on-ramp which may be employed for metering purposes, while the decrease in the rate of improvement may, as in the Q-Learning implementation, be attributed to the fact that as the number of AVs on the on-ramp increases, all human-driven vehicles are affected by an AV at some point, which reduces the impact of more AVs being present on the on-ramp. Similarly, the step in respect of the TTS in Scenario 1 is again attributed to the large traffic demand on both the highway and the on-ramp, which implies that the point at which all human-driven vehicles on the on-ramp are affected by AVs is reached sooner, due to the larger on-ramp demand, while the high traffic volumes on the highway combined with the large on-ramp demand and the smaller metering rates achievable by RM by AVs result in congestion regardless of the metering rate employed.

As expected, the improvements in the TTS were again due to improvements in respect of the TTSHW, as may clearly be seen in the box plots corresponding to the TTSHW in Figures 11.11(b), 11.12(b), 11.13(b) and 11.14(b), as the trends observed in respect of the TTSHW are very similar to those observed for the TTS. An approximately exponential decay is again observed in the TTSHW values in Scenarios 2–4, while in Scenario 1 the same step decrease as that in respect of the TTS was recorded from the no-control case to the 2.5% AV implementation. The performances of all other AV percentage implementations were relatively similar to the case of 2.5% AVs in Scenario 1.

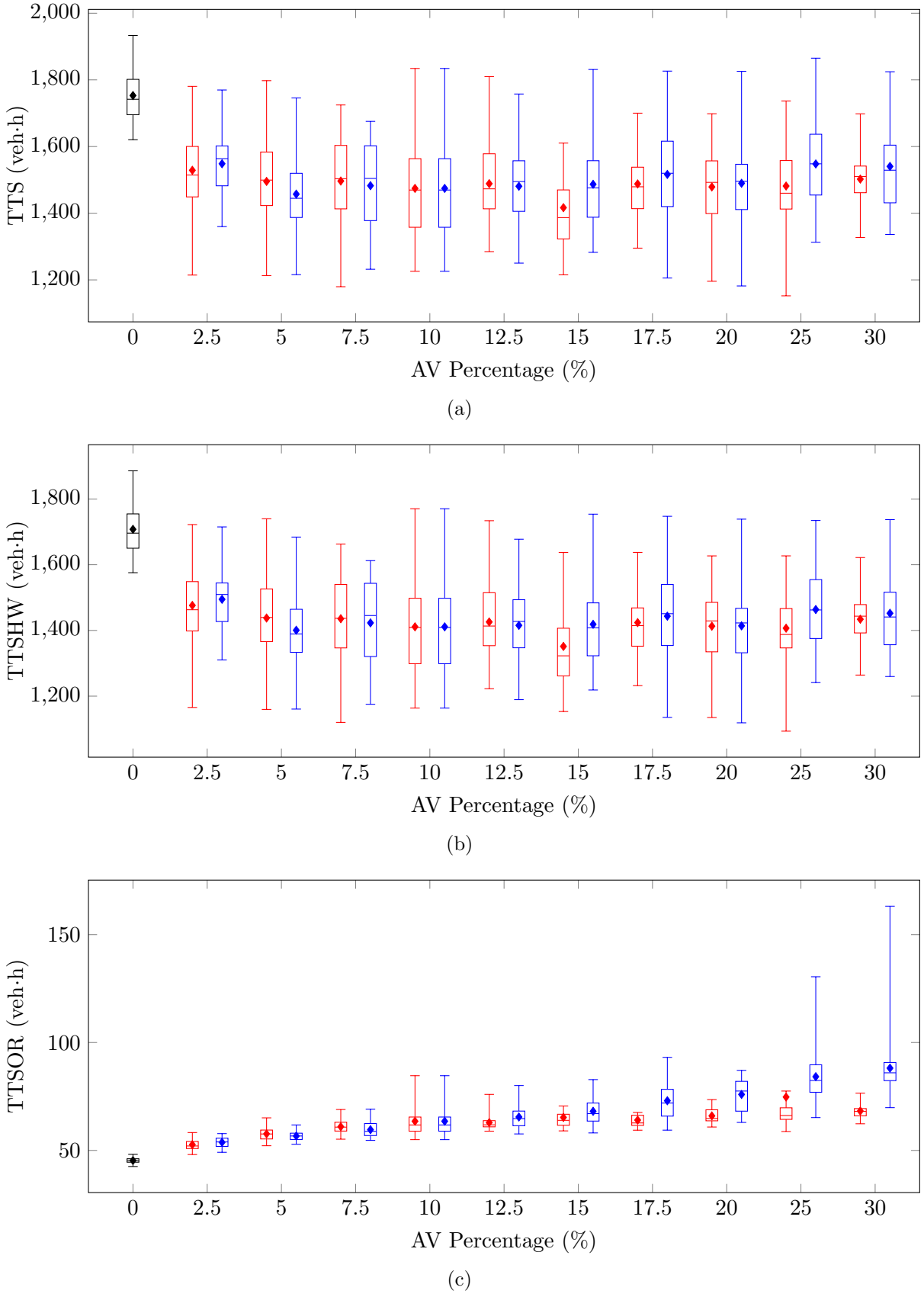
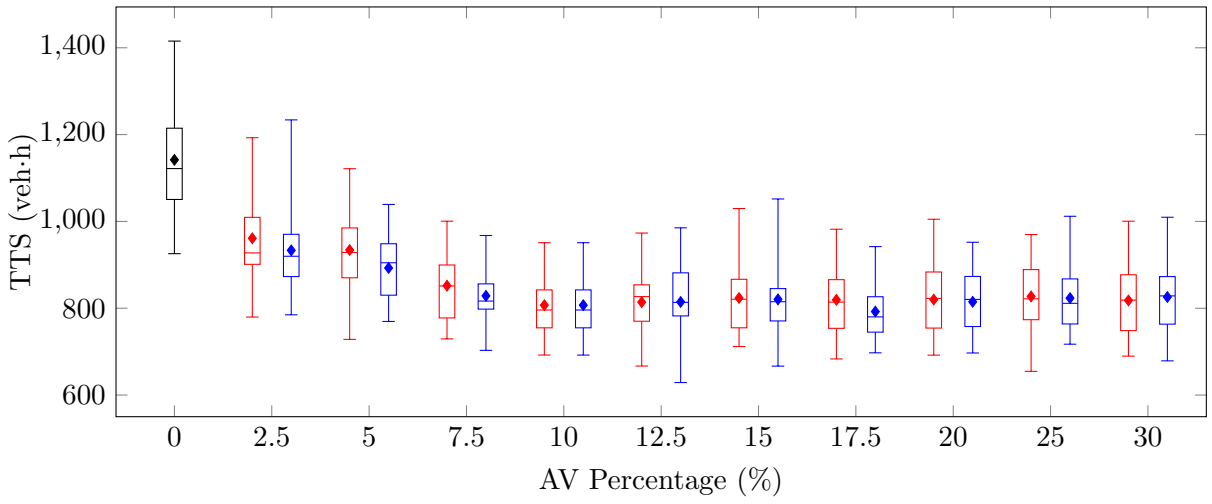
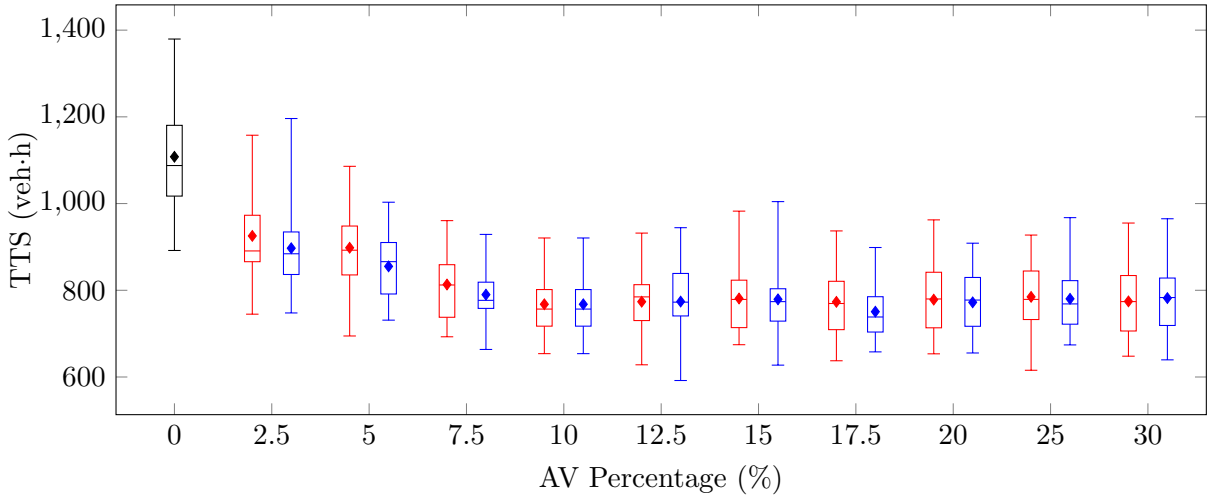


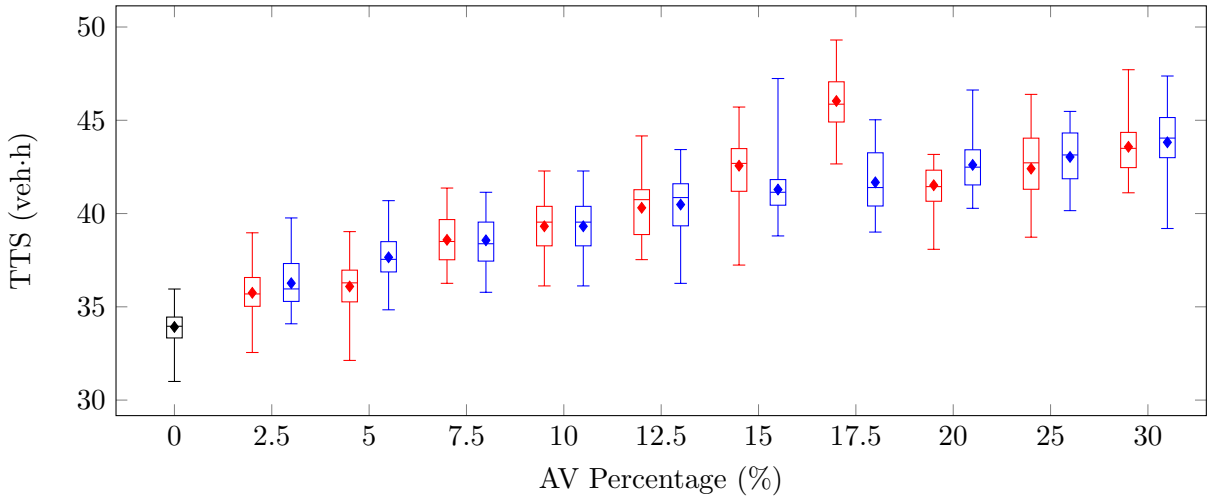
FIGURE 11.11: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 1.*



(a)



(b)



(c)

FIGURE 11.12: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 2.*

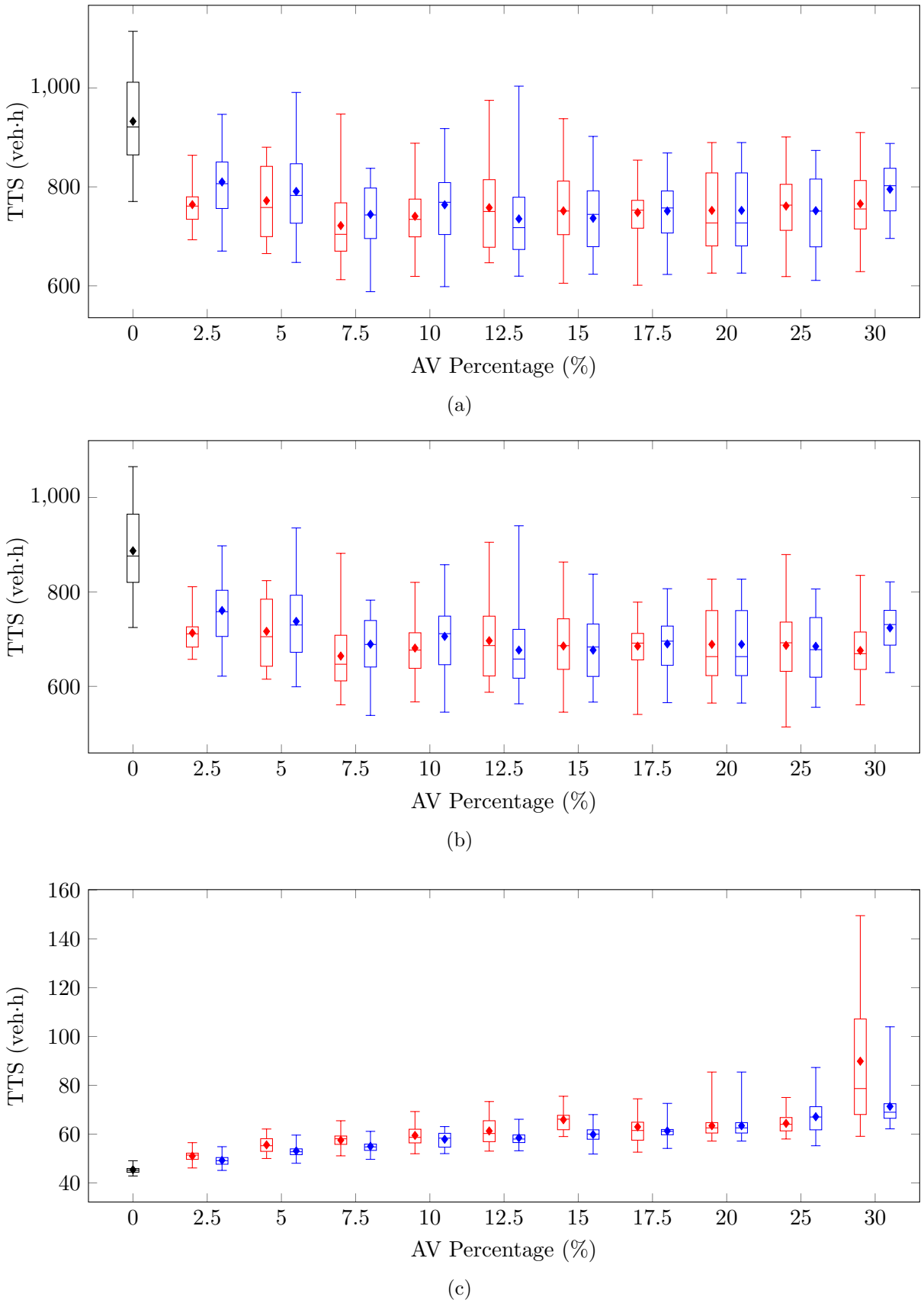


FIGURE 11.13: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 3.*

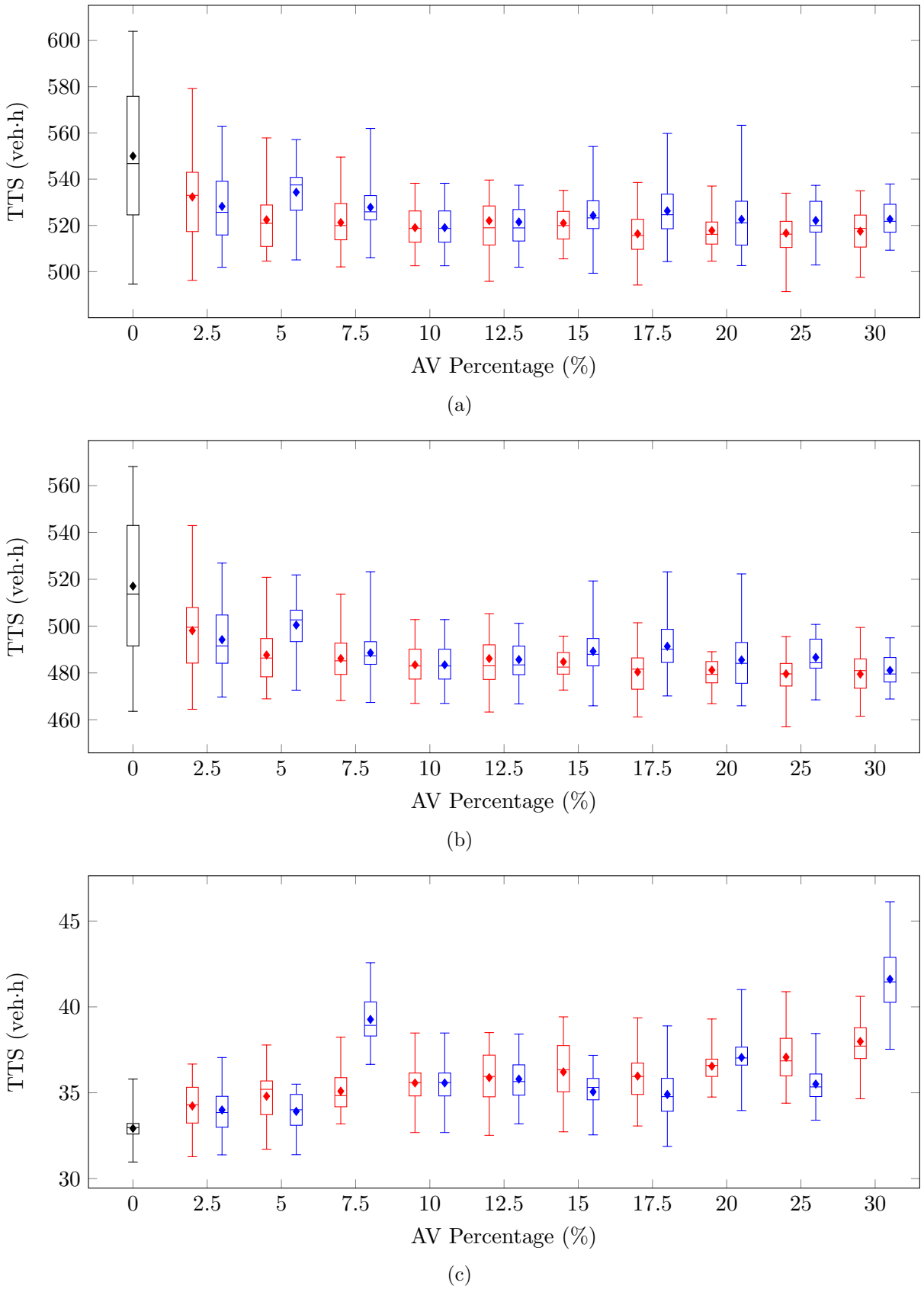


FIGURE 11.14: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying AV percentages in Scenario 4.*

As in the case of the Q-Learning implementation, the improvements in respect of the TTSHW came at the expense of increases in the TTSOR, as may clearly be seen in the box plots of Figures 11.11(c), 11.12(c), 11.13(c) and 11.14(c). The trend of increase in the TTSOR as the percentage of AVs increases observed in the Q-Learning implementations is again clearly visible for the kNN-TD learning implementations, although these increases are generally smaller than those in the Q-Learning implementations, as may be deduced from the mean TTSOR-values presented in Tables 11.11–11.14. These increases may again be explained by the fact that, as the number of AVs on the on-ramp increases, the number of vehicles that travel at slow speeds on the on-ramp increases, which is reflected in the TTSOR PMI.

The extrapolated policies again performed very similarly to the individually trained policies, especially in respect of the TTS and TTSHW PMIs when considering the kNN-TD learning implementation. These similarities are very clear in the box plots of Figures 11.11–11.14. The similarities in the performances of the individually trained and extrapolated policies are corroborated by the mean values of the TTS and TTSHW PMIs, from which a maximum difference of 5.64% was found between these policies in respect of the TTS, while this difference increased to 6.24% in respect of the TTSHW. In both of these cases, however, the individually trained policy achieved the smaller TTS and TTSHW-values. These similarities strengthen the argument that states have an intrinsic value, which is, to a certain extent learnt by the RL agent regardless of the composition of the traffic flow.

Perhaps surprisingly, in Scenario 2, the performances of the individually trained and the extrapolated policies were also very similar in respect of the TTSOR, as may be seen in the box plots of Figure 11.12(c), where the increases in the travel times by vehicles joining the highway from the on-ramp increased at a very similar rate in respect of both the individually trained and extrapolated policies. Two explanations are offered for this observation: In Scenario 2 there is reduced on-ramp demand, which implies that the increase in the absolute number of AVs present in the simulation model is not as large as that in Scenarios 1 and 3, and as a result, low speeds have to be assigned to all vehicles in order to achieve relatively large metering rates (even if the proportion of AVs in the traffic flow increases). Secondly, function approximation is employed in the kNN-TD learning implementation, which allows for more accurate state-action pair value estimates than in the more coarsely discretised Q-Learning implementation, which may result in more effective action selection for each state of traffic flow along the highway.

The increase in the TTSOR-values returned by the extrapolated policy over and above the increase measured in respect of the individual policies when large proportions of the traffic flow are AVs is, however, very clear in Scenario 1, as may be seen in Figure 11.11(c), as the individually trained policies consistently achieve smaller mean TTSOR-values than the extrapolated policy when AV percentages greater than 10% were simulated. These increases are, however, not as pronounced as in the Q-Learning implementation (perhaps due to improved state-action value estimation), as may be deduced from the mean TTSOR-values presented in Table 11.11. The increases may thus again be explained by the fact that in the case of smaller AV percentages in the traffic flow, smaller speed values have to be assigned to the AVs in order to implement the required level of on-ramp flow metering, while these speed values may increase as more AVs are present in the traffic flow (still achieving a similar metering rate). Due to the fact that these increases in the assigned speeds are not implemented in the extrapolated policies, the increase in the TTSOR over and above those recorded for the individually trained policies are observed.

In order to assess whether the above-mentioned effect could be countered by employing a policy trained with a larger proportion of AVs on the road, the policy employed for the performance comparison of the extrapolated and individually trained policies in Scenario 3 was learnt with the proportion of AVs set to 20%. As may be seen in Figure 11.13(c), employing the policy trained

TABLE 11.11: *AV percentage parameter evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 1 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	1 548.46	1 457.28	1 482.55	1 474.47	1 480.86	1 486.70	1 516.38	1 489.79	1 547.79	1 540.40
	Indiv.	1 528.80	1 495.40	1 496.58	1 474.47	1 488.67	1 416.64	1 487.91	1 478.92	1 481.55	1 502.30
TTSHW	Extra.	1 494.70	1 400.59	1 422.95	1 410.95	1 415.38	1 418.56	1 443.35	1 413.86	1 463.62	1 452.26
	Indiv.	1 476.12	1 437.73	1 435.60	1 410.95	1 425.78	1 351.35	1 423.86	1 412.92	1 406.82	1 434.04
TTSOR	Extra.	53.76	56.69	59.60	63.52	65.48	68.14	73.03	75.94	84.17	88.14
	Indiv.	52.68	57.66	60.98	63.52	62.89	65.29	64.05	66.00	74.74	68.26

TABLE 11.12: *AV percentage parameter evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 2 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	933.33	892.76	828.55	807.09	814.45	820.46	792.30	814.81	823.33	825.84
	Indiv.	961.14	934.13	851.88	807.09	813.92	823.60	819.32	819.93	827.20	817.98
TTSHW	Extra.	897.07	855.11	789.98	767.77	773.97	779.17	750.63	772.20	780.29	782.02
	Indiv.	925.39	898.04	813.29	767.77	773.61	781.04	773.28	778.41	784.80	774.40
TTSOR	Extra.	36.27	37.66	38.57	39.32	40.48	41.29	41.67	42.61	43.03	43.82
	Indiv.	35.75	36.09	38.59	39.32	40.31	42.56	46.03	41.52	42.41	43.58

TABLE 11.13: *AV percentage parameter evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 3 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	809.85	790.76	744.35	763.77	735.32	736.60	751.26	752.39	751.92	795.20
	Indiv.	764.12	772.13		740.58	758.00	751.48	748.19	752.39	761.22	765.78
TTSHW	Extra.	760.60	737.62	689.36	705.87	676.83	676.75	689.97	688.96	684.80	723.85
	Indiv.	713.11	716.54	664.21	681.11	696.71	685.57	685.21	688.96	686.82	675.92
TTSOR	Extra.	49.26	53.14	55.00	57.90	58.49	59.85	61.29	63.44	67.12	71.35
	Indiv.	51.02	55.59	57.51	59.47	61.29	65.91	62.98	63.44	74.40	89.86

TABLE 11.14: *AV percentage parameter evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 4 of §5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	528.21	534.35	527.83	519.03	521.49	524.26	526.26	522.60	522.14	522.70
	Indiv.	532.31	522.43	521.23	519.03	522.03	520.98	516.37	517.74	516.67	517.47
TTSHW	Extra.	494.21	500.44	488.56	483.46	485.70	489.21	491.36	485.54	486.63	481.08
	Indiv.	498.08	487.63	486.14	483.46	486.15	484.76	480.40	481.20	479.60	479.49
TTSOR	Extra.	34.00	33.91	39.27	35.57	35.80	35.06	34.90	37.06	35.51	41.61
	Indiv.	34.23	34.80	35.09	35.57	35.88	36.21	35.97	36.55	37.07	37.99

with a larger proportion of AVs in the traffic flow did, in fact, prevent the large increases in respect of the TTSOR recorded over and above those in observed for the individually trained policies when large proportions of the traffic flow consist of AVs. As may have been expected, this did, however, result in larger speed assignments (even in cases where few AVs are present in the traffic flow), reducing the effectiveness of the RM employed, as may be seen from the mean TTS and TTSHW-values returned by the extrapolated policy when compared with those achieved by the individually trained policies, presented in Table 11.13. As expected, larger speed values are assigned to the AVs in the extrapolated policies, resulting in smaller metering rates on the on-ramp, and thus enabling the extrapolated policy to return smaller TTSOR-values than the individually trained policies in the cases where the AV percentage was smaller than 20%, as is evident from the mean values in Table 11.13 as well as the box plots in Figure 11.13(c).

In respect of Scenario 4, the trends in the performances of the individually trained and extrapolated policies were again largely similar, as may be seen in Figure 11.14. The performances of the individually trained policies were, however, more consistent, and the improvements in respect of both the TTS and TTSHW where smoother and more predictable, while the increases in the TTSOR also followed a more regular pattern than those of the extrapolated policies. Finally, as may be seen in Table 11.14, the individually trained policies were typically able to achieve smaller TTS and TTSHW-values than the extrapolated policies.

A complete comparison of AV percentage and on-ramp length

In order to asses the effects that changes in both the on-ramp length and AV percentage have on the performance of RM by AVs, each combination of on-ramp length and AV percentage was evaluated in the context of Scenarios 2 and 3. Note that, due to the significant computational expense of training a policy individually for each of the 90 combinations, the results reported in this section were generated employing individually trained policies in respect of the AV percentage, while extrapolating over the on-ramp length. The policies were again trained for an on-ramp length of 250 metres. The choice of extrapolation over the on-ramp length was informed by the observation that the difference in performance between the individually trained policies and extrapolated policies was not as large in respect of the on-ramp lengths as in respect of the AV percentages. Furthermore, this comparison was performed only for the kNN-TD RM by AVs implementation due to the finding that the kNN-TD implementation was typically able to achieve smaller TTS-values than the Q-Learning implementation. The results of this comparison are shown in the form of surface plots in Figure 11.15.

As expected, the largest TTS-values were achieved by the combination of the smallest AV percentage of 2.5% and the shortest on-ramp length of 100 metres, as may be seen in Figures 11.15(a) and 11.15(b). From these highest points, the TTS then slopes downward in all directions as both the on-ramp length and the AV percentage increase. Note, however, that this slope is steeper in respect of the on-ramp length than in respect of the AV percentage in both scenarios, indicating that the length of the on-ramp may have a more prominent influence on the performance of RM by AVs. An explanation for this observation is that a small number of AVs on a relatively long stretch of on-ramp have the potential to influence more human-driven vehicles (due to the fact that these AVs spend longer amounts of time on the on-ramp) than relatively large numbers of AVs on a short on-ramp (if the on-ramp is short, there is only limited space for the AVs to limit the flow of human-driven vehicles). Furthermore, the decay in the TTS as the on-ramp length increases in respect of Scenario 2 is approximately linear, while the decay in the TTS as the on-ramp length increases in respect of Scenario 3 is again approximately exponential. This may be due to the fact that the on-ramp demand in Scenario 3 is significantly larger than that

in Scenario 2, and as a result, the point at which all human-driven vehicles are affected by AVs is reached sooner in Scenario 3, thus resulting in the fast initial decay, which then slows as the on-ramp lengths increase. In Scenario 2, however, the point at which all human-driven vehicles are affected by AVs is reached only at a later stage due to the lower on-ramp demand, and therefore, the influence of a longer on-ramp is approximately linear. Note, however, that the no-control case of 0% AVs is not included in these plots, and that, as in all previous parameter evaluations, a significant decrease in the TTS is expected to occur between 0% AVs and 2.5% AVs.

The trends in respect of the TTSHW were again, as expected, similar to those in respect of the TTS, as may be seen in Figures 11.15(c) and 11.15(d). The highest points on the surfaces are again at the shortest on-ramp length of 100 metres and the smallest AV percentage of 2.5%. Notably, however, the TTSHW-values at an on-ramp length of 500 meters are almost constant, especially in Scenario 2. This observation may be as a result of all human-driven vehicles being affected by AVs in cases where the on-ramp is sufficiently long (due to AVs spending long times travelling along the on-ramp at slow speeds), and as a result, the increases in the effectiveness of RM by AVs are limited as the number of AVs increases (as similar numbers of vehicles are caught behind AVs at all AV percentages).

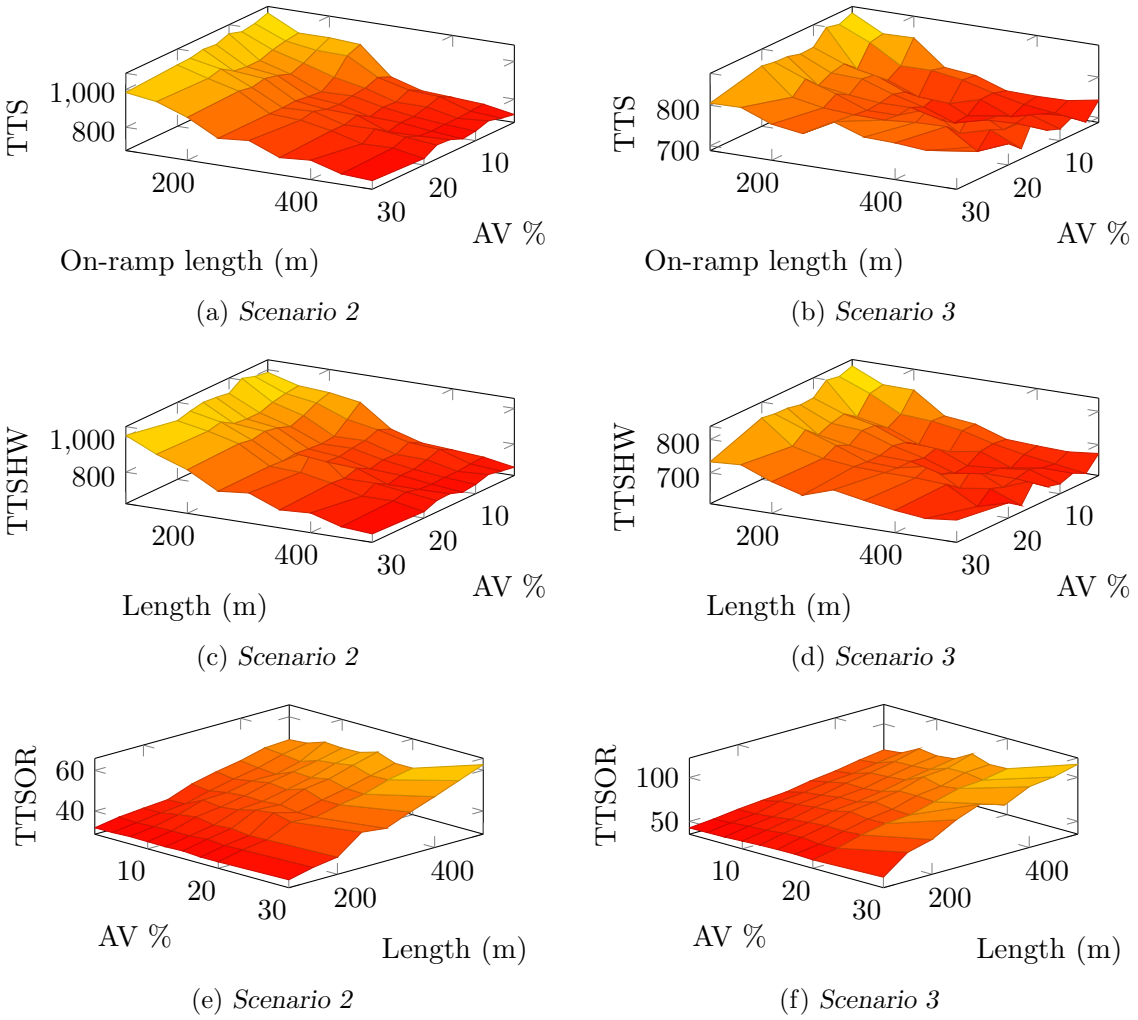
As expected, an increase is observed in respect of the TTSOR as both the on-ramp length and the AV percentage increase, with the largest TTSOR-value recorded at the combination of an on-ramp length of 500 metres and an AV percentage of 30%, as may be seen in Figures 11.15(e) and 11.15(f). These increases may again be explained by the observation that as the on-ramp length increases, the amount of time that AVs, and the vehicles following these AVs, spend on the on-ramp increases. The same reasoning may apply to the increases observed as the AV percentage increases, because, as the number of AVs increases, so too does the number of human-driven vehicles affected by the AVs. As expected, the slope of the surface corresponding to Scenario 3 in Figure 11.15(f) is steeper than that corresponding to Scenario 2 in Figure 11.15(e). The reason for this observation is again that, due to the larger on-ramp demand in Scenario 3, the number of vehicles affected by AVs, and thus the magnitude of the metering rate, increases at a faster rate than in Scenario 2, which represents a lower on-ramp demand.

11.5.4 Traffic Demand Parameter Evaluation

The aim in this section is to investigate the effect of variations in traffic demand on the performance of an RM by AV policy, so as to assess the robustness of policies in various situations of traffic demand. For this assessment, the policies learnt by Q-Learning and kNN-TD learning in Scenario 2 are employed in Scenarios 1, 3 and 4 of §5.3.2 and compared with policies learnt by the algorithms in each of the scenarios. This comparison is performed within the context of the benchmark model of §5.1.2 with an on-ramp length ℓ_{OR} of 250 metres. For the sake of completeness, this comparison was performed in respect of all ten varying levels of AV percentages as in the previous section. Note that the extrapolation was, however, performed only in respect of the varying traffic demand, while the individually trained policies in respect of the AV percentage in Scenario 2 were employed for the extrapolation in respect of the traffic demand across the scenarios.

Q-Learning

As may have been expected, the extrapolated policies again proved to be relatively robust against variations in the traffic demand, especially when considering the TTS and TTSHW PMIs, as



(a) *Scenario 2* (b) *Scenario 3*

(c) *Scenario 2* (d) *Scenario 3*

(e) *Scenario 2* (f) *Scenario 3*

FIGURE 11.15: *Surface plots relating AV percentage and on-ramp length, in respect of (a) and (b) the TTS, (c) and (d) the TTSHW, and (e) and (f) the TTSOR in Scenarios 2 and 3, respectively, for kNN-TD RM by AVs implementations.*

may be seen in the box plots in Figures 11.16, 11.17 and 11.18. Although the performances of the individually trained and extrapolated policies were generally similar in respect of the TTS and TTSHW, the individually trained policies were typically able to achieve smaller TTS and TTSHW-values than the extrapolated policies, as may be deduced from the mean PMI-values presented in Tables 11.15–11.17. These similarities in the performances may again be explained by the observation that the policies, when trained in the context of Scenario 2, are trained with a relatively low on-ramp demand, and as a result, AVs are instructed to travel at low speeds along the on-ramp in order to achieve the required metering rates. These relatively large metering rates are effective in reducing the TTS and TTSHW in all scenarios, even in the cases of relatively large on-ramp traffic demand, as may be seen in the box plots corresponding to the TTS and TTSHW PMIs in Figures 11.16, 11.17 and 11.18.

While the performances in respect of the TTS and TTSHW are generally similar for the TTS and TTSHW PMIs, the performances differ significantly in respect of the TTSOR, especially in Scenarios 1 and 3, as may be seen in Figures 11.16(c) and 11.17(c), while these differences are not as significant in Scenario 4, as is evident in Figure 11.18(c). As may be seen in the box plots of Figures 11.16(c) and 11.17(c), the performances of the individually trained and extrapolated

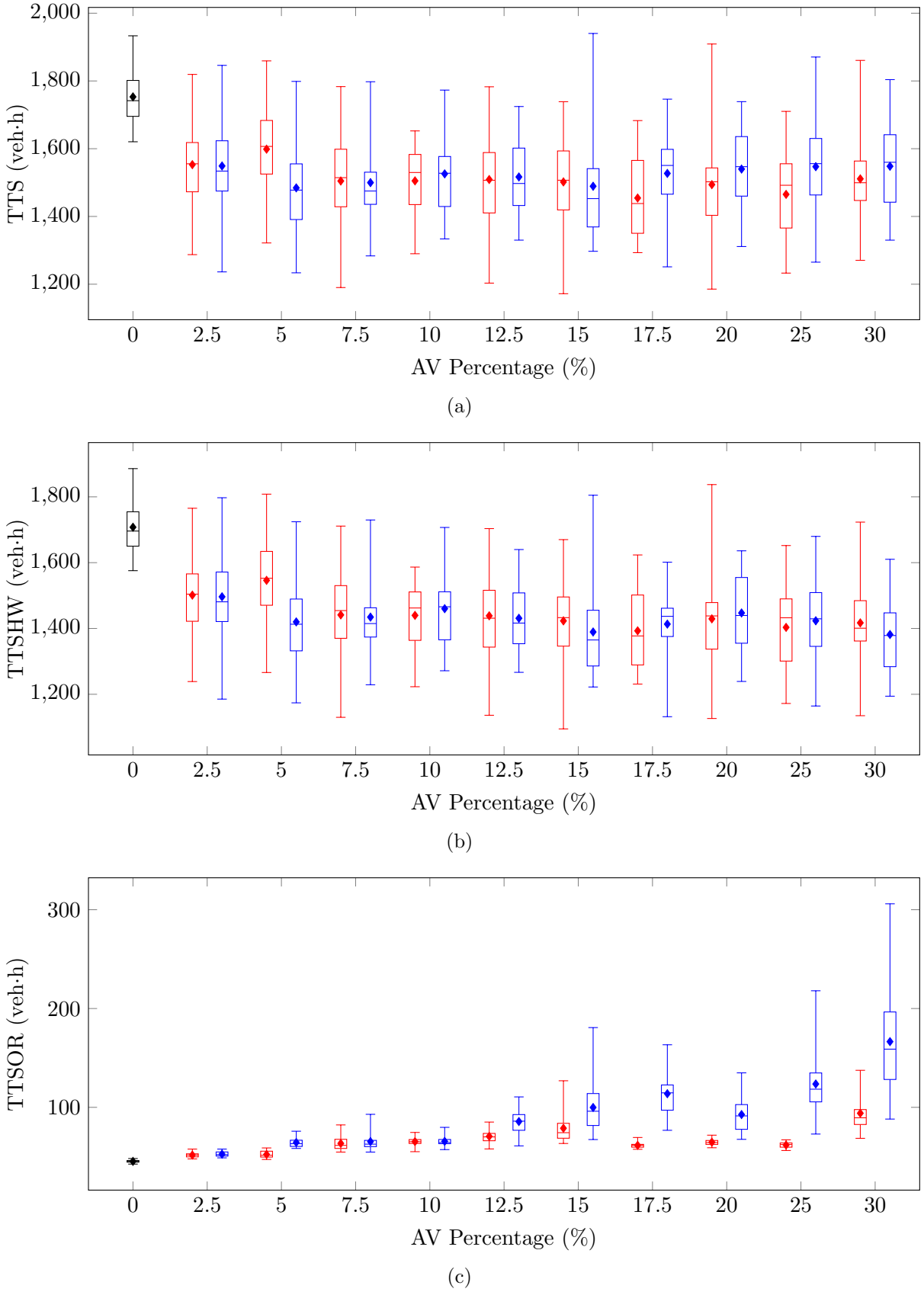
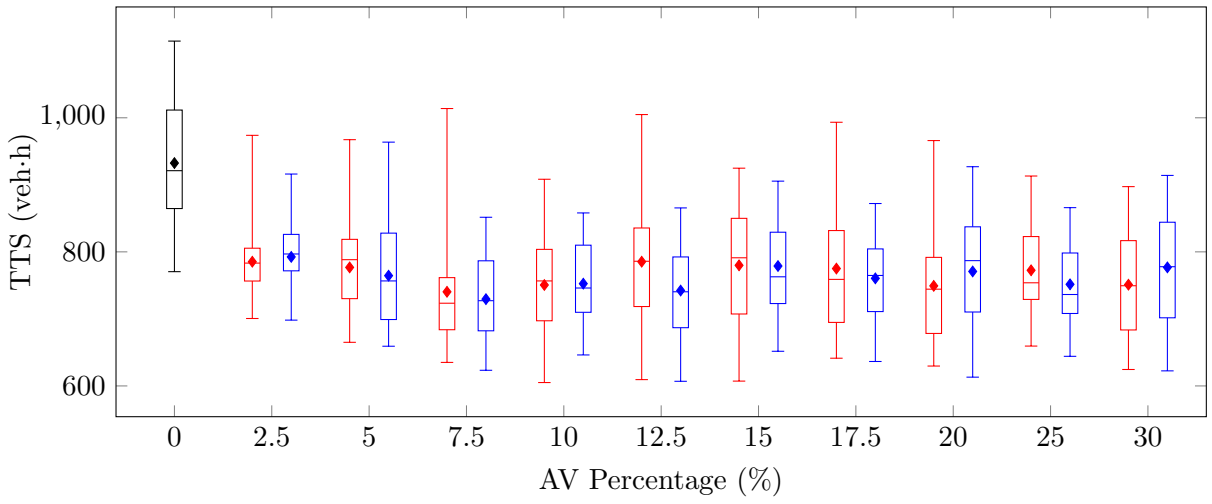
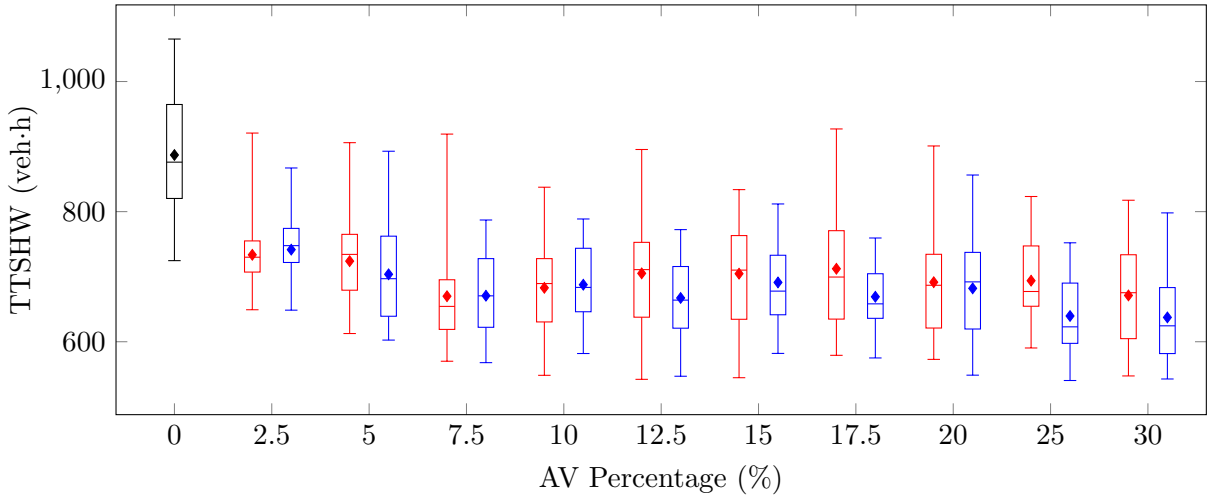


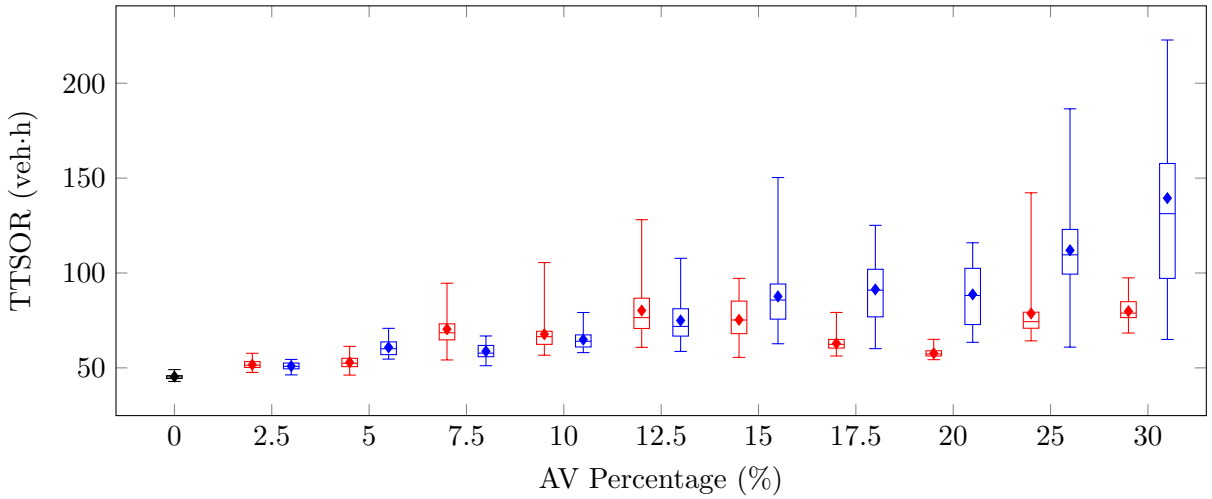
FIGURE 11.16: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying traffic demands in Scenario 1.*



(a)



(b)



(c)

FIGURE 11.17: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying traffic demands in Scenario 3.*

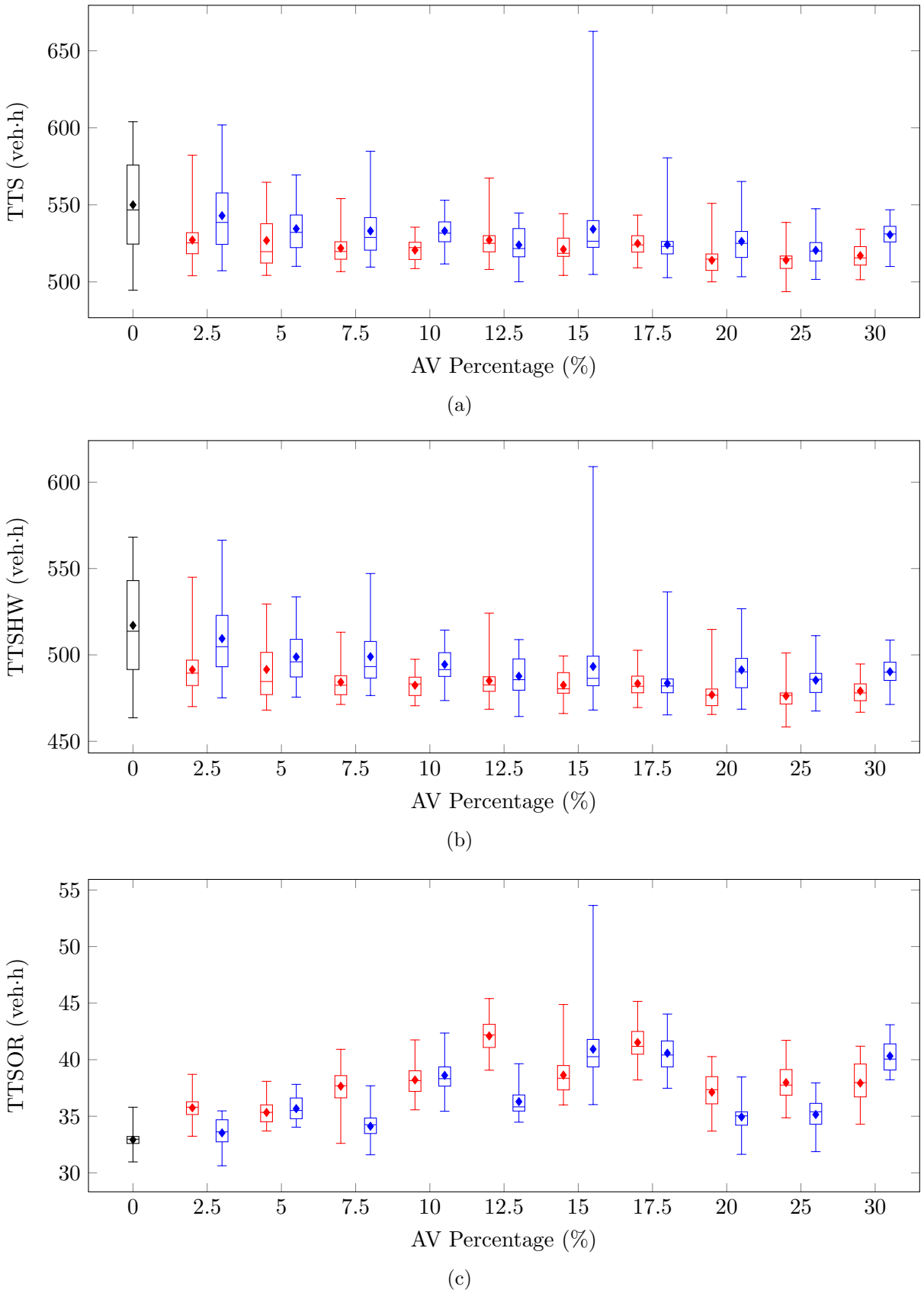


FIGURE 11.18: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying traffic demands in Scenario 4.*

TABLE 11.15: *Traffic demand evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 1 of 5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	1 548.97	1 484.36	1 499.83	1 525.71	1 516.62	1 489.20	1 527.05	1 539.84	1 546.97	1 548.13
	Indiv.	1 522.63	1 598.59	1 504.65	1 505.07	1 509.04	1 501.88	1 454.39	1 493.93	1 465.15	1 511.05
TTSHW	Extra.	1 496.32	1 419.91	1 434.50	1 460.15	1 431.00	1 389.37	1 413.26	1 447.16	1 423.35	1 381.64
	Indiv.	1 501.25	1 546.23	1 441.17	1 439.77	1 438.45	1 423.04	1 392.85	1 429.21	1 403.21	1 417.04
TTSOR	Extra.	52.65	64.45	65.33	65.56	85.62	99.84	113.80	92.68	123.62	166.49
	Indiv.	51.68	52.35	63.48	65.30	70.59	78.84	61.54	64.73	61.85	94.05

TABLE 11.16: *Traffic demand evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 3 of 5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	792.46	764.49	729.48	752.55	742.29	778.87	760.47	770.78	751.57	776.85
	Indiv.	785.29	776.80	740.39	750.67	785.31	779.91	775.07	749.25	772.53	751.03
TTSHW	Extra.	741.56	703.68	670.72	687.66	667.30	691.18	669.13	682.06	639.58	637.39
	Indiv.	733.62	723.89	669.99	682.85	705.09	704.60	712.17	691.54	693.86	671.18
TTSOR	Extra.	50.90	60.81	58.76	64.89	74.98	87.69	91.34	88.72	111.99	139.45
	Indiv.	51.67	52.91	70.40	67.72	80.22	75.31	62.89	57.71	78.67	79.85

TABLE 11.17: *Traffic demand evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 4 of 5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	542.96	534.52	533.07	533.07	523.98	534.20	524.18	526.26	520.49	530.59
	Indiv.	527.17	526.88	521.88	520.60	527.14	521.13	524.94	514.02	514.17	517.05
TTSHW	Extra.	509.43	498.84	498.95	494.46	487.69	493.27	483.60	491.32	485.34	490.27
	Indiv.	491.42	491.55	484.22	482.42	485.04	482.49	483.41	476.89	476.20	479.11
TTSOR	Extra.	33.53	35.68	34.12	38.61	36.30	40.93	40.58	34.94	35.15	40.32
	Indiv.	35.75	35.34	37.65	38.21	42.10	38.63	41.52	37.13	37.98	37.93

policies are relatively similar when small proportions of AVs are present in the traffic flow, while the differences in the performances of these policies are amplified at AV percentages of more than 12.5%. The reason for this may again be that, due to the fact that the extrapolated policies were trained in the context of a relatively small on-ramp demand in Scenario 2, the best-performing actions represent relatively small speeds. These small speed values are required in order to achieve the desired metering rate. In the cases of large on-ramp demand, as in Scenarios 1 and 3, however, the number of AVs on the on-ramp naturally increases, and larger speeds may be assigned while achieving similar metering rates as more vehicles are likely to be affected by AVs. The increases in respect of the TTSOR observed for the extrapolated policies may therefore, again be attributed to a lack of adjustment in respect of the speeds assigned to the AVs when larger numbers of AVs are present in the traffic flow than the case in which the policy was trained.

Interestingly, the opposite is observed in respect of the TTSOR in Scenario 4, as may be seen in Figure 11.17(c). Due to the fact that Scenario 4 represents the smallest overall traffic demand on both the on-ramp and the highway, even lower speeds may be required in order to achieve the best-performing level of on-ramp metering than those learnt when training the policies in the context of Scenario 2. As may be seen in the box plots in Figure 11.18, the individually trained policies consistently achieve smaller TTS and TTSHW-values than the extrapolated policies, while the TTSOR-values recorded for the individually trained policies are generally larger than those of the extrapolated policies.

kNN-TD Learning

The performances of the individually trained and extrapolated policies under varying conditions of traffic demand were again generally very similar in the kNN-TD RM by AVs implementation, indicating that the policies learnt by the kNN-TD agent in Scenario 2 are relatively robust against changes in the traffic demand. As may be seen in Figures 11.19(a), 11.20(a) and 11.21(a), the trends observed in respect of the TTS for the individually trained and extrapolated policies follow very similar patterns. This similarity in the TTS-values is also evident from the mean TTS-values presented in Tables 11.18–11.20.

As for the TTS, the trends observed by the individually trained and extrapolated policies were again very similar in respect of the TTSHW, as may have been expected, as the improvements achieved in the TTS are typically achieved along the highway when RM is applied. These similarities in the trends are again very clear in the box plots in Figures 11.19(b), 11.20(b) and 11.21(b). These similarities on the TTSHW-values are again also reflected in the mean TTSHW-values presented in Tables 11.18–11.20.

Unlike for Q-Learning, the mean TTSOR-values recorded for the individually trained and extrapolated policies were also generally similar, especially in Scenarios 1 and 3, as may be seen in Figures 11.19(c) and 11.20(c). This observation is corroborated by the mean TTSOR-values presented in Tables 11.18 and 11.19. The expected reason for this improved robustness in respect of the TTSOR by kNN-TD learning when compared with Q-Learning may again be due to the fact that continuous function approximation is employed in the kNN-TD learning implementation, which provides the agent with more detailed state information, based on which more appropriate actions may be selected. Although the mean values of the TTSOR are generally similar for the individually trained and extrapolated policies of the kNN-TD implementations, it is evident from the box plots in Figures 11.19(c) and 11.20(c) that the extrapolated policies result in significant increases in the variances of the travel times of vehicles joining the highway from the on-ramp. This increase in the variances may again be attributed to the fact that

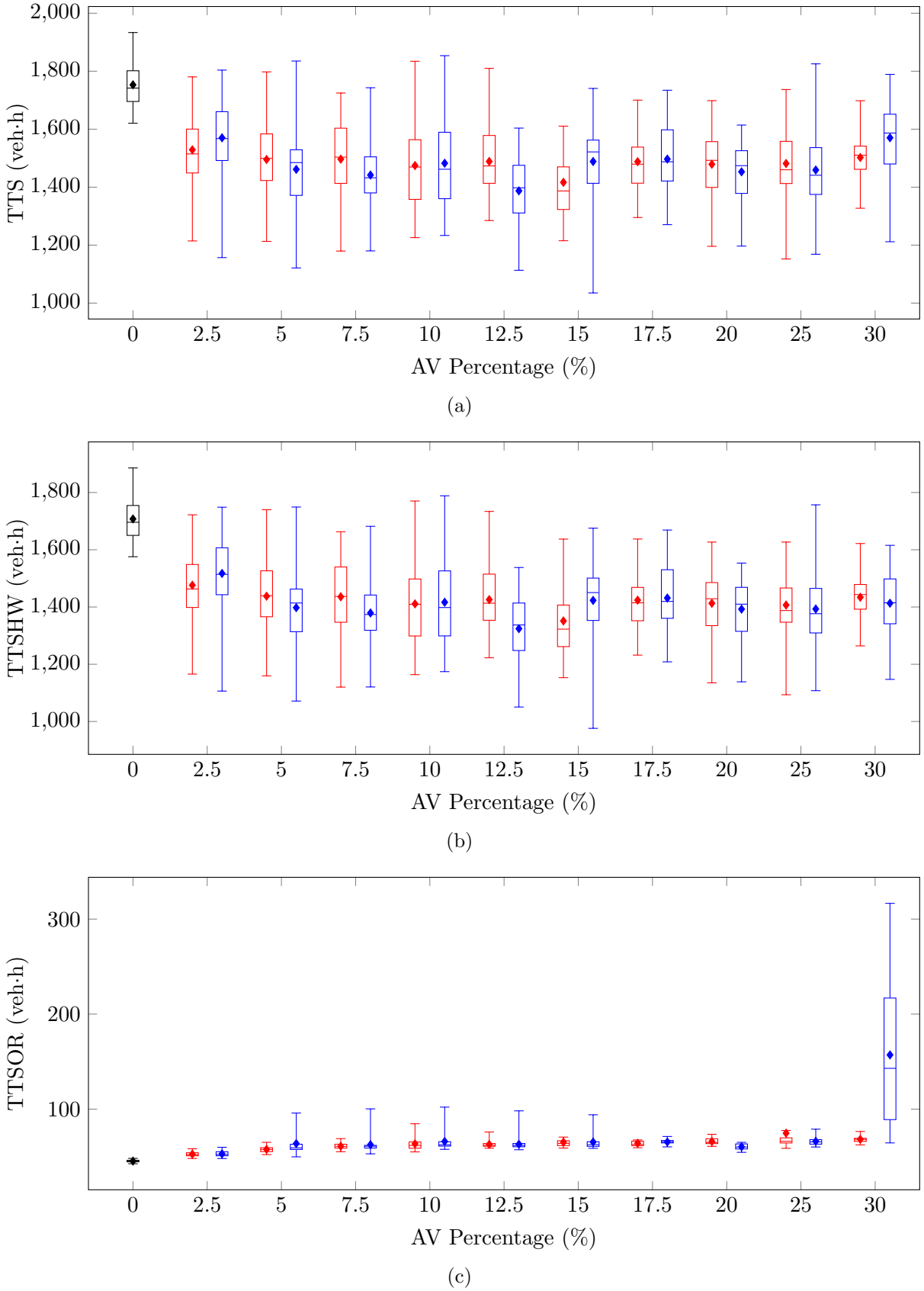


FIGURE 11.19: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying traffic demands in Scenario 1.*

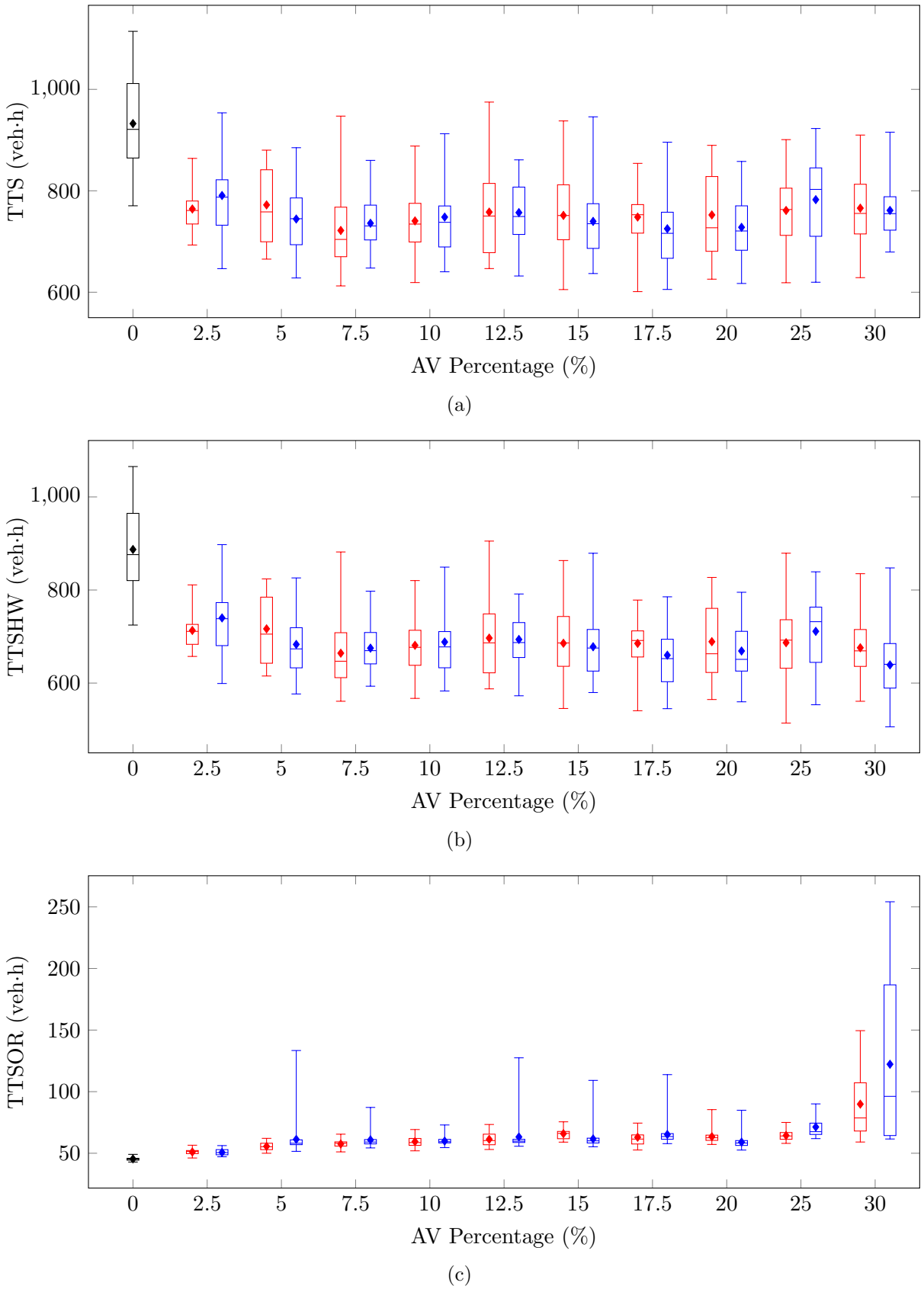


FIGURE 11.20: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying traffic demands in Scenario 3.*

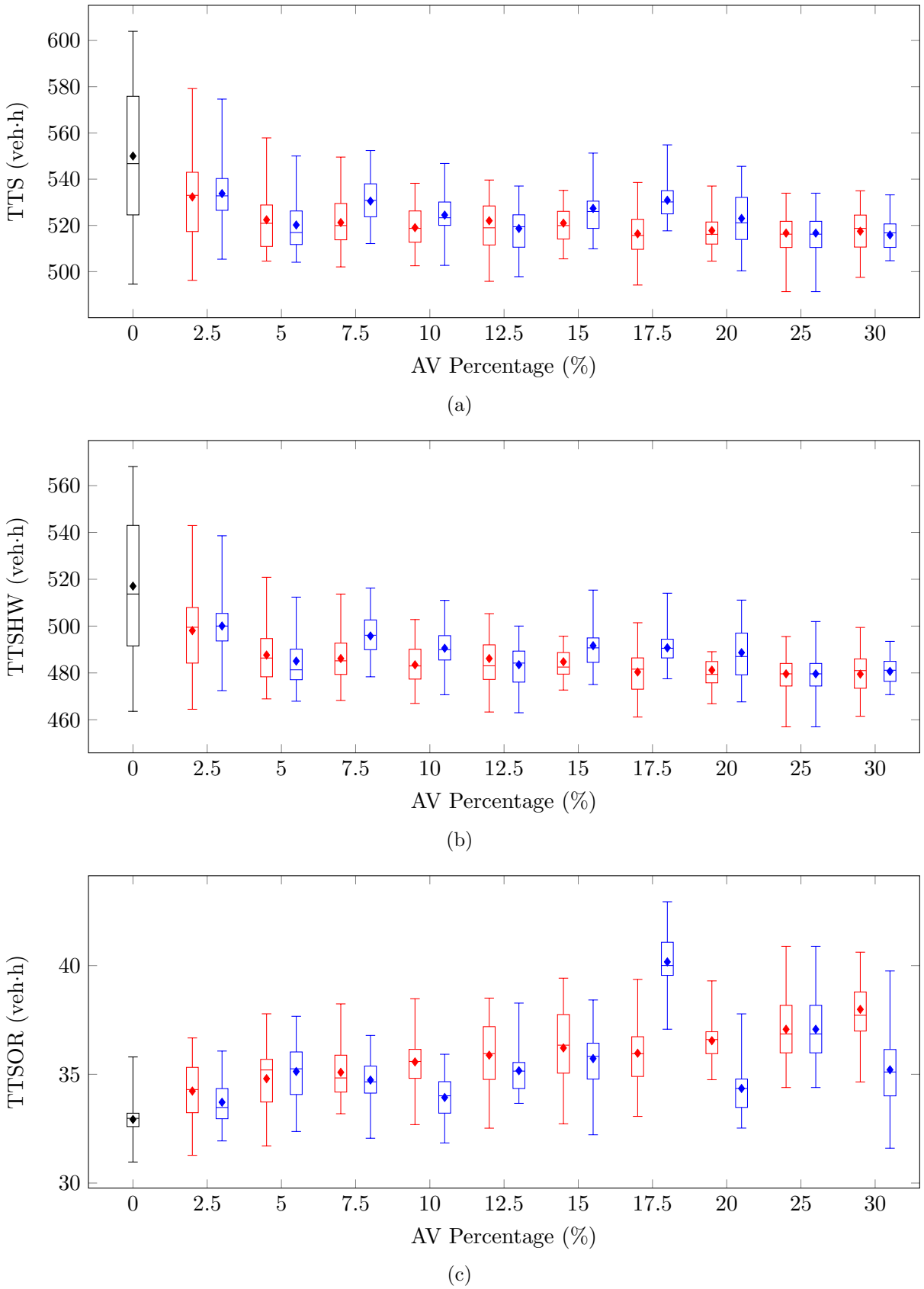


FIGURE 11.21: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) with varying traffic demands in Scenario 4.*

TABLE 11.18: *Traffic demand evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 1 of 5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	1 570.26	1 461.40	1 441.79	1 482.54	1 387.24	1 488.42	1 496.82	1 452.90	1 459.11	1 570.32
	Indiv.	1 528.80	1 495.40	1 496.58	1 474.47	1 488.67	1 416.64	1 487.91	1 478.92	1 481.55	1 502.30
TTSHW	Extra.	1 517.29	1 397.66	1 379.20	1 416.57	1 324.24	1 423.04	1 431.67	1 392.52	1 392.97	1 413.26
	Indiv.	1 476.12	1 437.73	1 435.60	1 410.95	1 425.78	1 351.35	1 423.86	1 412.92	1 406.82	1 434.04
TTSOR	Extra.	52.97	63.74	62.59	65.97	63.00	65.38	65.45	60.39	66.14	157.06
	Indiv.	52.68	57.66	60.98	63.52	62.89	65.29	64.05	66.00	74.74	68.26

TABLE 11.19: *Traffic demand evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 3 of 5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	790.67	744.32	735.96	748.13	756.97	739.67	725.14	727.99	782.51	761.27
	Indiv.	764.12	772.13		740.58	758.00	751.48	748.19	752.39	761.22	765.78
TTSHW	Extra.	740.00	683.07	675.12	688.25	693.74	678.04	659.90	668.96	711.19	639.02
	Indiv.	713.11	716.54	664.21	681.11	696.71	685.57	685.21	688.96	686.82	675.92
TTSOR	Extra.	50.67	61.26	60.84	59.88	63.23	61.63	65.24	59.03	71.33	122.25
	Indiv.	51.02	55.59	57.51	59.47	61.29	65.91	62.98	63.44	74.40	89.86

TABLE 11.20: *Traffic demand evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h) for Scenario 4 of 5.3.2.*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	533.76	520.19	530.51	524.48	518.66	527.35	530.89	523.01	516.67	515.87
	Indiv.	532.31	522.43	521.23	519.03	522.03	520.98	516.37	517.74	516.67	517.47
TTSHW	Extra.	500.05	485.06	495.78	490.54	483.50	491.63	490.72	488.67	479.60	480.66
	Indiv.	498.08	487.63	486.14	483.46	486.15	484.76	480.40	481.20	479.60	479.49
TTSOR	Extra.	33.72	35.13	34.74	33.93	35.17	35.72	40.17	34.35	37.07	35.21
	Indiv.	34.23	34.80	35.09	35.57	35.88	36.21	35.97	36.55	37.07	37.99

according to the extrapolated policies, the AVs are often assigned smaller speed limits in order to achieve larger metering rates in cases were fewer AVs are present in the traffic flow, resulting in increases in the variances of the box plots corresponding to the extrapolated policies in the scenarios where the number of AVs in the traffic flow is larger than in the scenario in which the policy was learnt.

In respect of Scenario 4, a trend similar to that observed for the Q-Learning implementation is observed. As may be seen in the box plots in Figure 11.21, the individually trained policies were generally able to achieve smaller TTS and TTSHW-values than the extrapolated policies. This was typically achieved by applying larger metering rates than in the extrapolated policies, resulting in smoother traffic flow along the highway while, naturally, increases in the TTSOR were recorded. Furthermore, the increases in respect of the TTSOR for the individually trained policies are again more gradual than for the extrapolated policies, as was the case in the parameter evaluations in respect of the AV percentages. These observations again indicate that, in cases where the on-ramp and highway traffic demands are small, low speeds yield the best performance if RM by AVs is employed, as illustrated by the individually trained policies in both the Q-Learning and kNN-TD implementations.

11.6 Algorithmic Comparison

For the purpose of consistency, the original benchmark model of §5.1.2, with an on-ramp length of 250 metres is employed for the purpose of the algorithmic comparison in this section. Furthermore, the algorithmic comparison is again performed within the context of the four scenarios of traffic flow of §5.3.2. In order to ascertain which AV percentage to employ for the comparison of RM by AVs with the conventional RM methods, a statistical comparison was performed in order to determine at which AV percentage the improvements in respect of the TTS ceased to be statistically significant in each of the four scenarios of traffic flow of §5.3.2. This statistical comparison was performed for both the Q-Learning and kNN-TD learning implementations. The results of the ANOVA and Levene tests performed in this respect are presented in Table 11.21.

TABLE 11.21: *The p-values for the ANOVA and Levene statistical tests performed in order to ascertain whether statistical differences occur at various levels of AV percentages. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Scenario 1		Scenario 2	
Algorithm	ANOVA	Levene's Test	ANOVA	Levene's Test
Q-Learning	2.7350×10^{-3}	9.9980×10^{-1}	3.2641×10^{-14}	1.5180×10^{-2}
kNN-TD	1.1031×10^{-1}	4.6317×10^{-1}	1.1102×10^{-16}	5.1141×10^{-1}
	Scenario 3		Scenario 4	
Algorithm	ANOVA	Levene's Test	ANOVA	Levene's Test
Q-Learning	1.8229×10^{-1}	5.5390×10^{-2}	3.6763×10^{-8}	9.5512×10^{-5}
kNN-TD	3.6024×10^{-1}	5.0717×10^{-1}	1.2507×10^{-4}	2.2689×10^{-4}

As may be seen in the table, no statistically significant differences were observed between any pairs of AV percentages in respect of Scenario 3 for the Q-Learning implementation, and in Scenarios 1 and 3 for the kNN-TD implementation. Statistical differences were, however, detected at a 5% level of significance between the performances of at least some pair of AV percentages in Scenarios 1, 2 and 4 for Q-Learning, while statistical differences between the performances of at least some pair of AV percentages were detected at a 5% level of significance in Scenarios 2

and 4 for the kNN-TD learning implementations. Furthermore, Levene's test revealed that the variances of the algorithmic output data are only statistically indistinguishable at a 5% level of significance in respect of Scenario 1 for Q-Learning and Scenario 2 for kNN-TD learning. Therefore, the Fisher LSD *post hoc* test was performed in order to ascertain between which pairs of AV percentages these differences occur, as may be seen in Tables 11.22 and 11.25, while the Games-Howell test was performed for this purpose in respect of Scenario 2 and 4 for Q-Learning and Scenario 4 for kNN-TD learning, as may be seen in Tables 11.23, 11.24 and 11.26. As may be seen in these tables, performing RM by AVs with 10% of the traffic flow comprising AVs is never outperformed by a higher percentage of AVs at a 5% level of significance in respect of the TTS. As a result, the AV percentage in the traffic flow is set to 10% in all the remaining comparisons performed with respect to RM by AVs in this chapter.

Due to the fact that in the conventional RM implementations of Chapter 6, the kNN-TD learning RM implementation returned the best performance, the kNN-TD RM implementation was chosen as the conventional RM implementation (without queue limits) against which to measure the performance of the novel RM technique. In order to avoid ambiguity with the kNN-TD implementation for RM by AVs, the kNN-TD implementation for conventional RM will henceforth be referred to as *conventional ramp metering* (CRM). Similarly, the novel RM technique is also compared with the Q-Learning implementation with the addition of queue limits, as the Q-Learning implementation returned the most favourable results when queue limits were employed. The Q-Learning implementation for conventional RM with queue limits will henceforth be referred to as CRM-QL in order to avoid ambiguity with the Q-Learning implementation for RM by AVs.

11.6.1 Scenario 1

From the results of the ANOVA performed on the algorithmic outputs in respect of Scenario 1, presented in Table 11.27, it is evident that there are statistical differences at a 5% level of significance between at least some pair of algorithmic outputs in respect of all seven PMIs. Furthermore, the Levene test revealed that the variances of the algorithmic output data are only statistically indistinguishable in respect of the TTS PMI, while the variances in respect of the other six PMIs were found to be statistically different at a 5% level of significance. As a result, the Fisher LSD *post hoc* test was performed in order to ascertain between which pairs of algorithmic outputs these differences occur in respect of the TTS, while the Games-Howell test was performed for this purpose in respect of the other six PMIs.

As may clearly be seen in Figure 11.22(a), all four algorithmic implementations were able to achieve improvements over the no-control case in respect of the TTS. This is corroborated by the p-values in Table 11.28, from which it is evident that CRM achieved the best performance, outperforming all other algorithms at a 5% level of significance, as it achieved an improvement of 20.21% over the no-control case. CRM is followed in the order of relative algorithmic performances by Q-Learning and kNN-TD for RM by AVs, as they achieved 14.14% and 15.89% improvements over the no-control case, respectively, thus outperforming CRM-QL at a 5% level of significance. Finally, the order of relative algorithmic performances is completed by CRM-QL, which achieved an improvement of 10.95% over the no-control case.

As expected, the improvements achieved by the RM implementations were achieved by vehicles travelling along the highway, as may clearly be seen in Figure 11.22(b). From the box plots in the figure it is evident that CRM achieved the largest improvement over the no-control case. This finding is confirmed by the p-values in Table 11.29. CRM did, in fact, achieve the smallest TTSHW-value of 606.16 veh·h, thereby again outperforming all other algorithms at a 5% level of

TABLE 11.22: *Differences in respect of the total time spent in the system (TTS) by the Q-Learning algorithm with varying AV percentages in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TTS									
AV Percentage	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	25%	30%
2.5%	—	1.8227×10^{-1}	1.5839×10^{-1}	$1.6203 times 10^{-1}$	1.9967×10^{-1}	1.3597×10^{-1}	4.1952×10^{-3}	8.5062×10^{-2}	1.0644×10^{-2}	2.2101×10^{-1}
5%		—	6.3128×10^{-3}	6.5473×10^{-3}	9.1925×10^{-3}	4.9478×10^{-3}	3.2310×10^{-5}	2.3821×10^{-3}	1.1597×10^{-4}	1.0867×10^{-2}
7.5%			—	9.9021×10^{-1}	8.9766×10^{-1}	9.3550×10^{-1}	1.4215×10^{-1}	7.5389×10^{-1}	2.4834×10^{-1}	8.5128×10^{-1}
10%				—	9.0739×10^{-1}	9.2574×10^{-1}	1.3886×10^{-1}	7.4458×10^{-1}	2.4337×10^{-1}	8.6092×10^{-1}
12.5%					—	8.3403×10^{-1}	1.1057×10^{-1}	6.5843×10^{-1}	1.9967×10^{-1}	9.5305×10^{-1}
15%						—	1.6534×10^{-1}	8.1607×10^{-1}	2.8295×10^{-1}	7.8839×10^{-1}
17.5%							—	2.4781×10^{-1}	7.5290×10^{-1}	9.8101×10^{-1}
20%								—	3.9998×10^{-1}	6.1643×10^{-1}
25%									—	1.7988×10^{-1}
30%										—
Mean	1 522.63	1 598.59	1 504.65	1 505.07	1 509.04	1 501.88	1 454.39	1 493.93	1 465.15	1 511.05

TABLE 11.23: *Differences in respect of the total time spent in the system (TTS) by the Q-Learning algorithm with varying AV percentages in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTS									
AV Percentage	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	25%	30%
2.5%	—	6.2845×10^{-3}	5.5883×10^{-3}	1.7165×10^{-7}	1.2852×10^{-4}	1.7943×10^{-5}	2.0218×10^{-5}	3.6601×10^{-2}	8.9047×10^{-9}	1.8626×10^{-8}
5%		—	9.9988×10^{-1}	1.6199×10^{-2}	5.1608×10^{-1}	1.6755×10^{-1}	8.0194×10^{-2}	1.0000×10^{-0}	7.2989×10^{-4}	1.4865×10^{-3}
7.5%			—	2.5899×10^{-1}	9.0706×10^{-1}	5.6375×10^{-1}	4.6781×10^{-1}	9.9998×10^{-1}	3.0832×10^{-2}	4.5731×10^{-2}
10%				—	9.9901×10^{-1}	1.0000×10^{-0}	9.9999×10^{-1}	1.2372×10^{-1}	9.7844×10^{-1}	9.9046×10^{-1}
12.5%					—	9.9985×10^{-1}	9.9991×10^{-1}	7.2226×10^{-1}	8.2231×10^{-1}	8.7249×10^{-1}
15%						—	9.9999×10^{-1}	3.4498×10^{-1}	9.9427×10^{-1}	9.9743×10^{-1}
17.5%							—	2.5844×10^{-1}	9.7231×10^{-1}	9.8597×10^{-1}
20%								—	1.372×10^{-2}	1.9555×10^{-2}
25%									—	9.9999×10^{-1}
30%										—
Mean	948.81	873.67	863.19	815.05	830.49	815.88	818.04	873.29	795.41	797.25

TABLE 11.24: *Differences in respect of the total time spent in the system (TTS) by the Q-Learning algorithm with varying AV percentages in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTS									
AV Percentage	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	25%	30%
2.5%	—	9.9999×10^{-1}	8.2436×10^{-2}	4.5851×10^{-1}	1.0000×10^{-0}	6.4523×10^{-1}	9.9911×10^{-1}	5.0227×10^{-3}	4.3186×10^{-3}	4.5859×10^{-2}
5%		—	9.4374×10^{-1}	7.5336×10^{-1}	9.9999×10^{-1}	8.6045×10^{-1}	9.9993×10^{-1}	3.9159×10^{-2}	3.7558×10^{-2}	1.9719×10^{-1}
7.5%			—	9.9994×10^{-1}	6.9008×10^{-1}	9.9999×10^{-1}	9.6364×10^{-1}	1.1778×10^{-1}	1.0323×10^{-1}	6.2199×10^{-1}
10%				—	2.3279×10^{-1}	9.9999×10^{-1}	5.6039×10^{-1}	1.4153×10^{-1}	1.1575×10^{-1}	7.6143×10^{-1}
12.5%					—	4.4411×10^{-1}	9.9731×10^{-1}	6.1176×10^{-4}	4.2408×10^{-4}	8.4715×10^{-3}
15%						—	8.2568×10^{-1}	1.5828×10^{-1}	1.3788×10^{-1}	7.4636×10^{-1}
17.5%							—	1.4024×10^{-3}	8.1809×10^{-4}	2.1250×10^{-2}
20%								—	1.0000×10^{-0}	9.5842×10^{-1}
25%									—	9.5924×10^{-1}
30%										—
Mean	527.17	526.88	521.88	520.60	527.14	521.13	524.94	514.02	514.17	517.05

TABLE 11.25: *Differences in respect of the total time spent in the system (TTS) by the kNN-TD algorithm with varying AV percentages in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TTS									
AV Percentage	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	25%	30%
2.5%	—	2.1485×10^{-1}	8.6482×10^{-7}	$1.0224 times 10^{-11}$	6.8739×10^{-11}	9.2672×10^{-10}	2.9759×10^{-10}	3.5079×10^{-10}	2.3589×10^{-9}	2.0760×10^{-10}
5%		—	1.8603×10^{-4}	1.3433×10^{-8}	7.0280×10^{-8}	6.5306×10^{-7}	2.4783×10^{-7}	2.8528×10^{-7}	1.4409×10^{-6}	1.8202×10^{-7}
7.5%			—	4.0153×10^{-2}	8.1650×10^{-2}	1.9417×10^{-1}	1.3504×10^{-1}	1.4254×10^{-1}	2.5701×10^{-1}	1.1976×10^{-1}
10%				—	7.5362×10^{-1}	4.4784×10^{-1}	5.7406×10^{-1}	5.5496×10^{-1}	3.5536×10^{-1}	6.1672×10^{-1}
12.5%					—	6.5601×10^{-1}	8.0388×10^{-1}	7.8208×10^{-1}	5.4132×10^{-1}	8.5189×10^{-1}
15%						—	8.4370×10^{-1}	8.6589×10^{-1}	8.6855×10^{-1}	7.9581×10^{-1}
17.5%							—	9.7744×10^{-1}	7.1688×10^{-1}	9.5086×10^{-1}
20%								—	7.3811×10^{-1}	9.2836×10^{-1}
25%									—	6.7139×10^{-1}
30%										—
Mean	961.14	934.13	851.88	807.09	813.92	823.60	819.32	819.93	827.20	817.98

TABLE 11.26: *Differences in respect of the total time spent in the system (TTS) by the kNN-TD algorithm with varying AV percentages in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Games-Howell test p-values: TTS									
AV Percentage	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	25%	30%
2.5%	—	5.0179×10^{-1}	3.0644×10^{-1}	9.1098×10^{-2}	5.7606×10^{-1}	2.4231×10^{-1}	2.2789×10^{-2}	3.8756×10^{-2}	2.440×10^{-2}	3.5989×10^{-2}
5%		—	9.9999×10^{-1}	9.8121×10^{-1}	9.9999×10^{-1}	9.9998×10^{-1}	6.6366×10^{-1}	8.4328×10^{-1}	6.9134×10^{-1}	8.1853×10^{-1}
7.5%			—	9.9877×10^{-1}	9.9999×10^{-1}	9.9999×10^{-1}	8.3114×10^{-1}	9.5507×10^{-1}	8.5557×10^{-1}	9.4096×10^{-1}
10%				—	9.9808×10^{-1}	9.9916×10^{-1}	9.9269×10^{-1}	9.9994×10^{-1}	9.9600×10^{-1}	9.9979×10^{-1}
12.5%					—	9.9999×10^{-1}	8.9029×10^{-1}	9.7016×10^{-1}	9.0898×10^{-1}	9.6076×10^{-1}
15%						—	8.1816×10^{-1}	9.5296×10^{-1}	8.4309×10^{-1}	9.3797×10^{-1}
17.5%							—	9.9994×10^{-1}	9.9999×10^{-1}	9.9999×10^{-1}
20%								—	9.9999×10^{-1}	9.9999×10^{-1}
25%									—	9.9999×10^{-1}
30%										—
Mean	532.31	522.43	521.23	519.03	522.03	520.98	516.37	517.74	516.67	517.47

TABLE 11.27: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 1. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

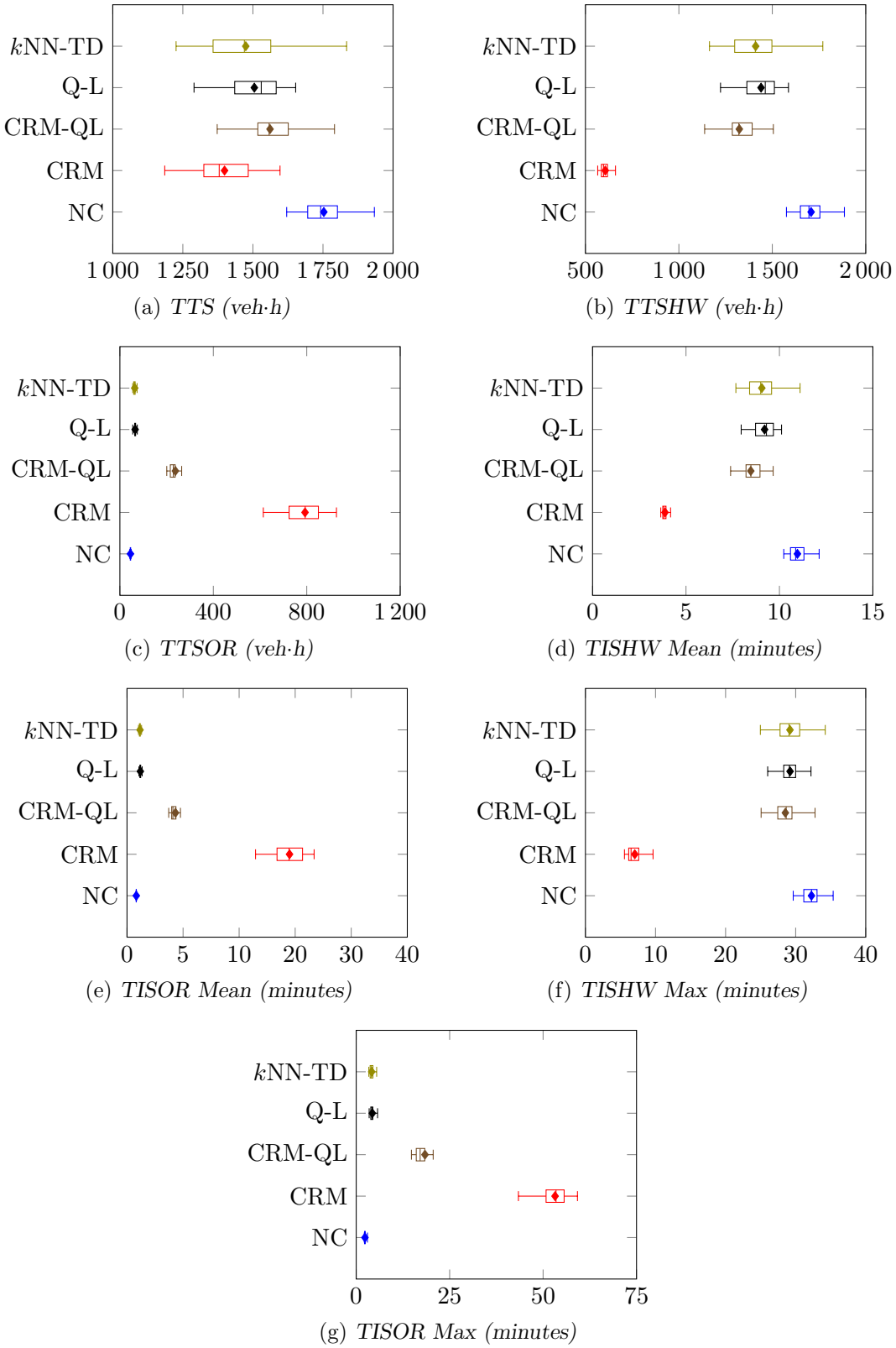
	Mean value					p-value	
PMI	No Control	CRM	CRM-QL	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	1 753.01	1 398.80	1 561.05	1 505.07	1 474.47	$< 1 \times 10^{-17}$	9.9214×10^{-2}
TTSHW	1 707.70	606.16	1 323.47	1 439.77	1 410.95	$< 1 \times 10^{-17}$	2.2427×10^{-6}
TTSOR	45.31	792.64	237.58	65.30	63.52	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Mean	10.96	3.88	8.47	9.21	9.06	$< 1 \times 10^{-17}$	1.6690×10^{-6}
TISOR Mean	1.66	28.99	8.64	2.38	2.31	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Max	32.25	7.04	28.53	29.20	29.19	$< 1 \times 10^{-17}$	2.3221×10^{-2}
TISOR Max	2.34	53.21	18.40	4.31	4.14	$< 1 \times 10^{-17}$	1.2939×10^{-5}

significance. CRM-QL returned the next-best TTSHW-value of 1 323.47 veh·h, which was small enough for CRM-QL to outperform Q-Learning and kNN-TD learning for RM by AVs at a 5% level of significance, as these implementations returned TTSHW-values of 1 505.07 veh·h and 1 474.47 veh·h, respectively. Q-Learning and kNN-TD were again found to perform statistically indistinguishably at a 5% level of significance, while they were both able to outperform the no-control case, which returned a TTSHW-value of 1 707.70 veh·h.

As may have been expected, the ordering of the relative algorithmic performances in respect of the TTSOR is exactly opposite to that in respect of the TTSHW, as may be seen in the box plots in Figure 11.22(c). Naturally, the no-control case achieved the smallest TTSOR-value of 45.31 veh·h, outperforming all the algorithmic implementations at a 5% level of significance, as may be deduced from the p-values in Table 11.30. The no-control case is followed in the order of relative algorithmic performances by Q-Learning and kNN-TD for RM by AVs, as these algorithms returned TTSOR-values of 65.30 veh·h and 63.52 veh·h, respectively, outperforming both CRM and CRM-QL at a 5% level of significance. Expectedly, due to the addition of the queue limitation, CRM-QL was able to outperform CRM in respect of the TTSOR, as these algorithms returned TTSOR-values of 237.58 veh·h and 792.64 veh·h, respectively.

From the box plots in Figures 11.22(d) and 11.22(f), it is clear that the ordering of the relative algorithmic performances in respect of both the mean and maximum TISHW is the same as that in respect of the TTSHW. CRM again returned the best performance in respect of both these PMIs, achieving mean and maximum TISHW-values of 3.88 minutes and 7.04 minutes, respectively, outperforming all other algorithms at a 5% level of significance. CRM-QL achieved the second-best performance in respect of both of these PMIs, as may be deduced from the p-values in Tables 11.31 and 11.33, returning mean and maximum TISHW-values of 8.47 minutes and 28.53 minutes, respectively. CRM-QL was able to outperform both Q-Learning and kNN-TD for RM by AVs in respect of the mean TISHW, as these algorithms achieved mean TISHW-values of 9.21 minutes and 9.06 minutes, respectively, while the performances of these three algorithms were statistically indistinguishable in respect of the maximum TISHW, as Q-Learning and kNN-TD for RM by AVs returned values of 29.20 minutes and 29.19 minutes, respectively. Finally, all the algorithmic implementations were able to outperform the no-control case at a 5% level of significance in respect of both these PMIs.

In respect of the mean and maximum TISOR, the order of relative algorithmic performances was, as expected, the same as in respect of the TTSOR. These trends are again clearly visible in the box plots of Figures 11.22(e) and 11.22(g). The no-control case returned the smallest mean TISOR-value of 1.66 minutes, outperforming all algorithmic implementations, as may be inferred from the p-values in Table 11.32. The RM by AV implementations were able to achieve the next-best performances as Q-Learning and kNN-TD learning returned mean TISOR-values



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 11.22: *PMI results for the no-control case (NC), the CRM and CRM-QL control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 1.*

of 2.38 minutes and 2.31 minutes, respectively, outperforming both CRM and CRM-QL, while their performances were found to be statistically indistinguishable. CRM-QL was finally able to outperform CRM in respect of the mean TISOR as these implementations achieved values of 8.64 minutes and 28.99 minutes, respectively. In respect of the maximum TISOR, the ordering of relative algorithmic performances is the same, as statistical differences were found at a 5% level of significance between all algorithms except the RM by AV implementations, as the no-control case, Q-Learning and kNN-TD for RM by AVs, CRM-QL and CRM achieved maximum TISOR-values of 2.34 minutes, 4.31 minutes, 4.14 minutes, 18.40 minutes and 53.21 minutes, respectively.

TABLE 11.28: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	2.1782×10^{-10}	3.3307×10^{-15}	$< 1 \times 10^{-17}$
CRM		—	4.5743×10^{-8}	2.2807×10^{-4}	7.9333×10^{-3}
CRM-QL			—	4.8289×10^{-2}	2.4776×10^{-3}
Q-Learning				—	2.7827×10^{-1}
kNN-TD					—
Mean	1 753.01	1 398.80	1 561.05	1 505.07	1 474.47

TABLE 11.29: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	5.1992×10^{-14}	1.3352×10^{-11}	4.1422×10^{-12}	3.0975×10^{-13}
CRM		—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	5.9730×10^{-14}
CRM-QL			—	2.6663×10^{-4}	4.4566×10^{-2}
Q-Learning				—	8.9458×10^{-1}
kNN-TD					—
Mean	1 707.70	606.16	1 323.47	1 439.77	1 410.95

TABLE 11.30: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	1.3320^{-15}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	1.1668×10^{-13}
CRM		—	$< 1 \times 10^{-17}$	1.0436×10^{-15}	1.0214×10^{-15}
CRM-QL			—	3.4420×10^{-16}	2.0095×10^{-15}
Q-Learning				—	6.8710×10^{-1}
kNN-TD					—
Mean	45.31	792.64	237.58	65.30	63.52

TABLE 11.31: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	5.6266×10^{-13}	1.1647×10^{-11}	1.6424×10^{-12}	$< 1 \times 10^{-17}$
CRM		—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	5.3069×10^{-14}
CRM-QL			—	2.2011×10^{-4}	2.0343×10^{-2}
Q-Learning				—	9.4147×10^{-1}
kNN-TD					—
Mean	10.96	3.88	8.47	9.21	9.06

TABLE 11.32: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	1.9980×10^{-15}	2.1090×10^{-15}	4.6629×10^{-14}	4.9959×10^{-15}
CRM		—	$< 1 \times 10^{-17}$	5.3289×10^{-15}	1.1100×10^{-15}
CRM-QL			—	5.5509×10^{-15}	1.8208×10^{-14}
Q-Learning				—	5.3579×10^{-1}
kNN-TD					—
Mean	1.66	28.99	8.64	2.38	2.31

TABLE 11.33: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISHW Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	1.3021×10^{-11}	1.4906×10^{-10}	1.1502×10^{-10}	4.1934×10^{-6}
CRM		—	$< 1 \times 10^{-17}$	1.2841^{-11}	$< 1 \times 10^{-17}$
CRM-QL			—	5.4782×10^{-1}	7.7095×10^{-1}
Q-Learning				—	9.9999×10^{-1}
kNN-TD					—
Mean	32.25	7.04	28.53	29.20	29.19

TABLE 11.34: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 1. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	7.719×10^{-15}	8.66708×10^{-14}	4.2188×10^{-14}	3.3196×10^{-14}
CRM		—	$< 1 \times 10^{-17}$	6.4059×10^{-14}	6.9167×10^{-14}
CRM-QL			—	1.9390×10^{-12}	1.4384×10^{-12}
Q-Learning				—	7.8358×10^{-1}
kNN-TD					—
Mean	2.34	53.21	18.40	4.31	4.14

11.6.2 Scenario 2

As may be seen from the results of the ANOVA performed in respect of Scenario 2, presented in Table 11.35, statistical differences again occur at a 5% level of significance between at least some pair of algorithmic outputs in resect of all seven PMIs. The results from the Levene test furthermore indicate that the variances of at least some pair of algorithms' output are statistically distinguishable at a 5% level of significance in respect of all seven PMIs. Therefore, the Games-Howell test was employed in order to ascertain between which pairs of algorithmic outputs these differences occur in respect of all seven PMIs.

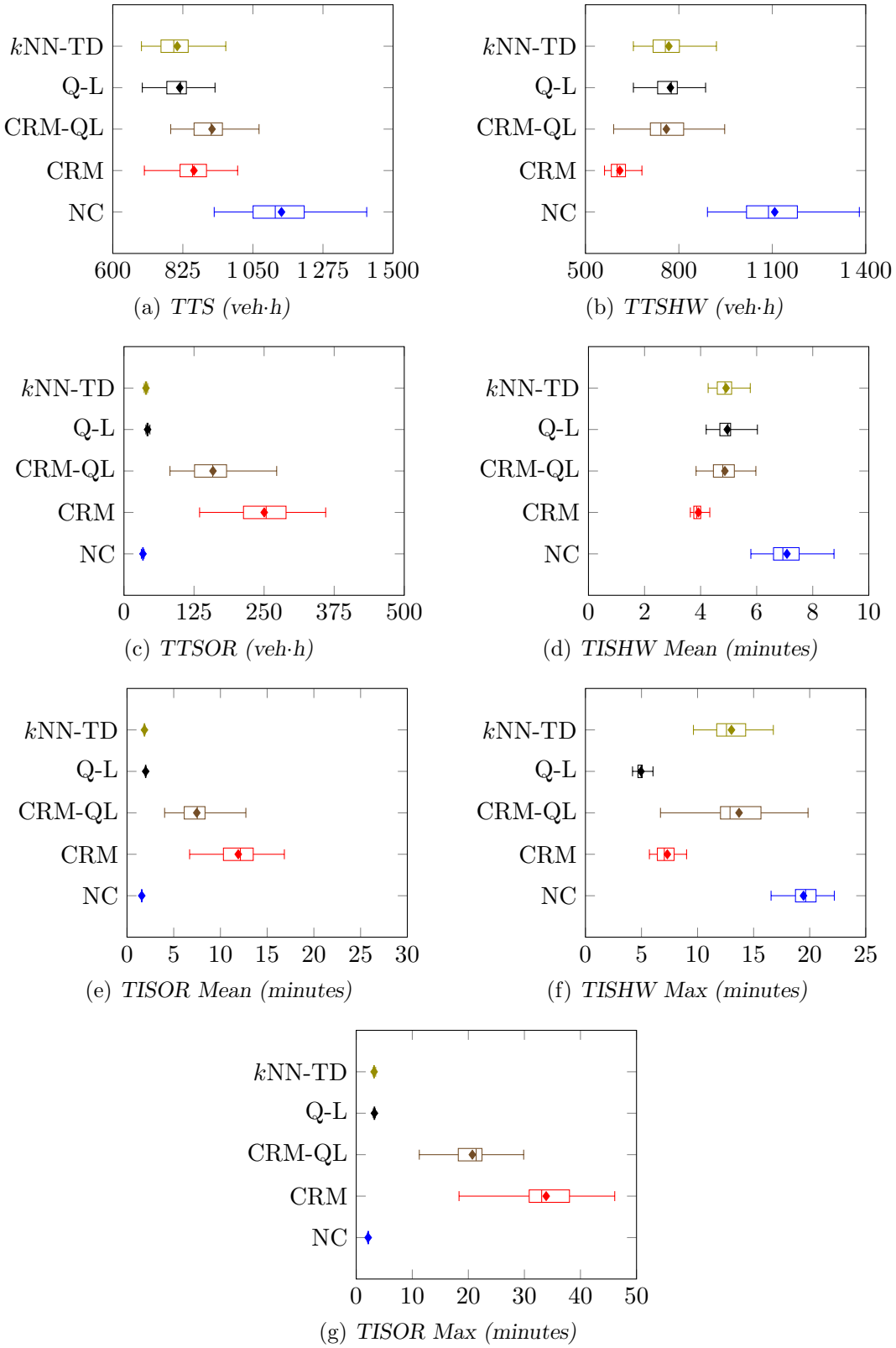
TABLE 11.35: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 2. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					*p*-value	
PMI	No Control	CRM	CRM-QL	Q-Learning	*k*NN-TD	ANOVA	Levene's Test
TTS	1 141.80	860.61	918.68	815.05	807.09	$< 1 \times 10^{-17}$	5.0111×10^{-3}
TTSHW	1 107.88	610.40	759.88	772.90	767.77	$< 1 \times 10^{-17}$	1.7983×10^{-5}
TTSOR	33.92	250.21	158.79	42.14	39.32	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Mean	7.08	3.92	4.87	4.95	4.91	$< 1 \times 10^{-17}$	1.5196×10^{-5}
TISOR Mean	1.58	11.91	7.47	2.00	1.87	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISHW Max	19.45	7.31	13.70	13.89	13.02	$< 1 \times 10^{-17}$	3.6292×10^{-3}
TISOR Max	2.13	33.89	20.72	3.25	3.19	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$

All four algorithmic implementations were again able to improve on the no-control case in respect of the TTS in Scenario 2, as may be seen in Figure 11.23(a). These improvements are corroborated by the *p*-values presented in Table 11.36, from which it is evident that all of the algorithms outperformed the no-control case at a 5% level of significance. Q-Learning and *k*NN-TD for RM by AVs and CRM achieved the largest improvements over the no-control case of 28.62%, 29.31% and 24.63%, respectively, outperforming CRM-QL at a 5% level of significance, while their performances were found to be statistically indistinguishable at a 5% level of significance. The order of relative algorithmic performances is completed by CRM-QL, which was able to reduce the TTS by 19.54% when compared with the no-control case.

As expected from RM implementations, the improvements were again achieved along the highway, as is clearly visible in the box plots in Figure 11.23(b). As in Scenario 1, CRM was able to achieve the smallest TTSHW-value of 610.40 veh·h, outperforming all other algorithms as well as the no-control case at a 5% level of significance, as may be inferred from the *p*-values in Table 11.37. CRM-QL returned the next-smallest TTSHW-value of 759.88 veh·h, but this value was not small enough to outperform Q-Learning or *k*NN-TD for RM by AVs at a 5% level of significance, as the latter two algorithms returned TTSHW-values of 772.90 veh·h and 767.77 veh·h, respectively. As may be seen in Table 11.37, the performances of these three algorithms were found to be statistically indistinguishable, while all three algorithms outperformed the no-control case, which achieved a TTSHW-value of 1 107.88 veh·h, at a 5% level of significance.

Interestingly, statistical differences were found between all algorithms at a 5% level of significance in respect of the TTSOR, as may seen from the *p*-values in Table 11.38. The no-control case returned the smallest TTSOR-value of 33.92 veh·h, as expected, outperforming all algorithms at a 5% level of significance. The no-control case is followed in the order of relative algorithimic performances by *k*NN-TD learning for RM by AVs, which returned a TTSOR-value of 39.32 veh·h, outperforming Q-Learning, CRM-QL and CRM at a 5% level of significance. The next-best performance was achieved by Q-Learning for RM by AVs, which achieved a TTSOR-value of 42.14 veh·h, outperforming CRM-QL and CRM, which achieved values of 158.79 veh·h and 250.21 veh·h, respectively, in respect of the TTSOR. Finally, CRM-QL was able to outperform



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 11.23: *PMI results for the no-control case (NC), the CRM and CRM-QL control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 2.*

CRM at a 5% level of significance. This ordering of the relative algorithmic performances is also visible in the box plots of Figure 11.23(c).

The trends in respect of the mean and maximum TISHW PMIs are the same as those in respect of the TTSHW, as may be seen in the box plots in Figures 11.23(d) and 11.23(f). CRM achieved the smallest mean and maximum TISHW-values of 3.92 minutes and 7.31 minutes, respectively, outperforming all other algorithmic implementations at a 5% level of significance, as may be deduced from the results of the Games-Howell tests, presented in Tables 11.39 and 11.41. CRM-QL, Q-Learning and kNN-TD learning were again found to perform statistically indistinguishably, as they returned values of 4.87 minutes, 4.95 minutes and 4.91 minutes, respectively, in respect of the mean TISHW, while these values increased to 13.70 minutes, 13.89 minutes and 13.02 minutes, respectively. This dominance by CRM was, however, to be expected, as no limitations are placed on the build-up of on-ramp queues in that implementation, while the build-up of an on-ramp queue is less likely in the RM by AV implementations. All of the algorithms were, nevertheless, again able to outperform the no-control case at a 5% level of significance in respect of both these PMIs.

In respect of the mean TISOR, statistical differences were again found at a 5% level of significance between all algorithmic implementations, as may be seen in Table 11.40. The no-control case achieved the smallest mean TISOR-value of 1.58 minutes, outperforming all algorithmic implementations. The no-control case is followed by kNN-TD learning for RM by AVs in the order of relative algorithmic performances, which returned a value of 1.87 minutes, outperforming the other three implementations. The effectiveness of RM by AVs in maintaining acceptable on-ramp travel times is again illustrated by Q-Learning for RM by AVs, which achieved the next-best performance, returning a mean TISOR-value of 2.00 minutes, outperforming both CRM and CRM-QL at a 5% level of significance. Finally CRM-QL was able to outperform CRM at a 5% level of significance, as these algorithms returned mean TISOR-values of 7.47 minutes and 11.91 minutes, respectively. The order of relative algorithmic performances in respect of the maximum TISOR is exactly the same, except that the performances of Q-Learning and kNN-TD learning for RM by AVs were found to be statistically indistinguishable at a 5% level of significance, as may be deduced from the p-values presented in Table 11.42. These orderings of the relative algorithmic performances are also clear in the box plots of Figures 11.23(e) and 11.23(g).

TABLE 11.36: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	2.2531×10^{-10}	6.3827×10^{-13}	$< 1 \times 10^{-17}$
CRM		—	3.7122×10^{-2}	8.5394×10^{-2}	6.0547×10^{-2}
CRM-QL			—	9.0532×10^{-6}	1.2023×10^{-5}
Q-Learning				—	9.9269×10^{-1}
kNN-TD					—
Mean	1 141.80	860.61	918.68	815.05	807.09

11.6.3 Scenario 3

As in Scenarios 1 and 2, statistical differences were again detected at a 5% level of significance between at least some pair of algorithmic output data in respect of all seven PMIs, as may be seen from the p-values returned by the ANOVA, presented in Table 11.43. Levene's test

TABLE 11.37: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	6.2017×10^{-13}	$< 1 \times 10^{-17}$
CRM		—	1.7124×10^{-11}	$< 1 \times 10^{-17}$	1.6949×10^{-11}
CRM-QL			—	9.4892×10^{-1}	9.9449×10^{-1}
Q-Learning				—	9.9864×10^{-1}
kNN-TD					—
Mean	1 107.88	610.40	759.88	772.90	767.77

TABLE 11.38: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance..*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	3.3309×10^{-14}	3.2199×10^{-14}	$< 1 \times 10^{-17}$	2.9109×10^{-12}
CRM		—	4.2808×10^{-9}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM-QL			—	1.1102×10^{-13}	1.3989×10^{-13}
Q-Learning				—	6.5631×10^{-9}
kNN-TD					—
Mean	33.92	250.21	158.79	42.14	39.32

TABLE 11.39: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	6.0651×10^{-13}	$< 1 \times 10^{-17}$
CRM		—	3.9715×10^{-12}	$< 1 \times 10^{-17}$	1.7264×10^{-12}
CRM-QL			—	9.2403×10^{-1}	9.9543×10^{-1}
Q-Learning				—	9.9418×10^{-1}
kNN-TD					—
Mean	7.08	3.92	4.87	4.95	4.91

TABLE 11.40: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	5.5509×10^{-15}	$< 1 \times 10^{-17}$	7.4163×10^{-14}
CRM		—	5.9179×10^{-10}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM-QL			—	1.5099×10^{-14}	$< 1 \times 10^{-17}$
Q-Learning				—	8.4386×10^{-12}
kNN-TD					—
Mean	1.58	11.91	7.47	2.00	1.87

TABLE 11.41: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	2.7923×10^{-9}	1.3245×10^{-11}	1.36976×10^{-11}
CRM		—	1.5064×10^{-10}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM-QL			—	9.9885×10^{-1}	8.8355×10^{-1}
Q-Learning				—	3.9426×10^{-1}
kNN-TD					—
Mean	19.45	7.31	13.70	13.89	13.02

TABLE 11.42: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 2. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	2.6650×10^{-15}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM		—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	4.6630×10^{-15}
CRM-QL			—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Q-Learning				—	1.0159×10^{-1}
kNN-TD					—
Mean	2.13	33.89	20.72	3.25	3.19

furthermore revealed that, as in Scenario 2, the variances of at least some pair of algorithmic output differ statistically at a 5% level of significance in respect of all seven PMIs. Therefore, the Games-Howell test was again employed in order to determine between which pairs of algorithmic outputs these differences occur in respect of all seven PMIs.

TABLE 11.43: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 3. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					p-value	
PMI	No Control	CRM	CRM-QL	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	932.46	829.02	815.96	750.57	740.58	$< 1 \times 10^{-17}$	1.8472×10^{-2}
TTSHW	887.07	518.66	676.46	682.85	681.11	$< 1 \times 10^{-17}$	7.7012×10^{-9}
TTSOR	45.40	310.36	139.50	67.72	59.47	$< 1 \times 10^{-17}$	3.3089×10^{-12}
TISHW Mean	6.18	3.60	4.70	4.74	4.74	$< 1 \times 10^{-17}$	2.7650×10^{-10}
TISOR Mean	1.63	11.47	5.02	2.48	2.19	$< 1 \times 10^{-17}$	5.6116×10^{-9}
TISHW Max	22.19	7.14	13.02	14.90	15.10	$< 1 \times 10^{-17}$	7.9012×10^{-6}
TISOR Max	2.37	26.76	13.10	4.71	3.96	$< 1 \times 10^{-17}$	1.1626×10^{-4}

All of the RM implementations were again able to achieve improvements over the no-control case in respect of the TTS, as may clearly be seen in the box plots in Figure 11.24(a). Furthermore, from the figure, it is evident that the RM by AV implementations achieved the largest improvements in respect of the TTS. This observation is confirmed by the p-values presented in Table 11.44. As may be deduced from the table, Q-Learning and kNN-TD learning for RM by AVs achieved the smallest TTS-values of 750.57 veh·h and 740.58 veh·h, respectively, outperforming all other algorithms at a 5% level of significance, while their performances were found to be statistically indistinguishable at a 5% level of significance. These implementations are followed in the order of relative algorithmic performances by CRM and CRM-QL which achieved

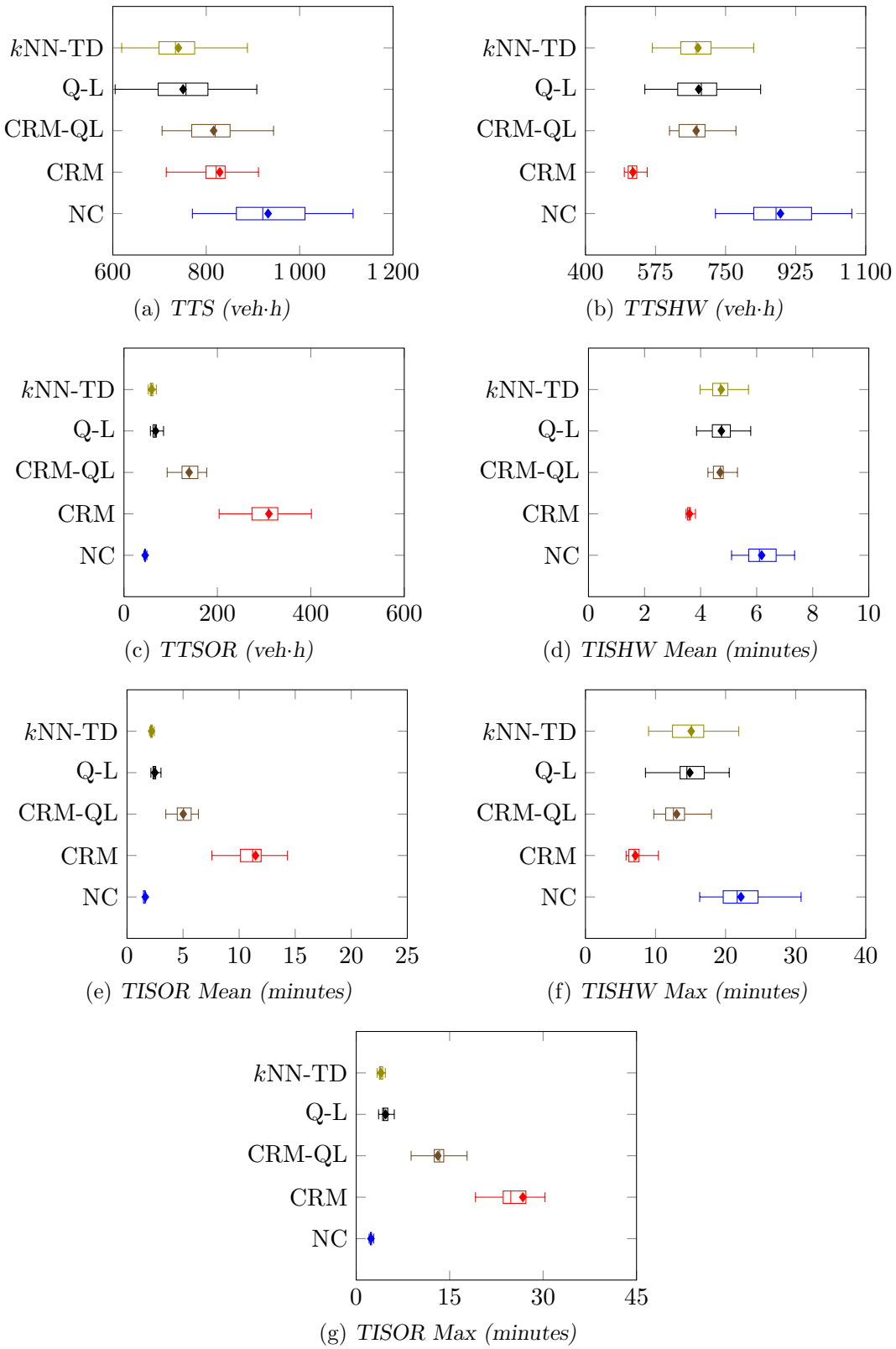
TTS-values of 829.02 veh·h and 815.96 veh·h, respectively, both outperforming the no-control case, which achieved a TTS-value of 932.46 veh·h, while their performances were again found to be statistically similar in respect of the TTS at a 5% level of significance.

In respect of the TTSHW, CRM again achieved the smallest value as was the case in Scenarios 1 and 2, improving on the no-control case by 41.53%, and outperforming all other algorithms at a 5% level of significance, as may be deduced from the p-values presented in Table 11.45. CRM was followed in the order of algorithmic performances by CRM-QL, Q-Learning and kNN-TD learning, whose performances were found to be statistically indistinguishable from one another at a 5% level of significance as they achieved improvements of 23.74%, 23.02% and 23.22%, respectively. These three algorithms were, however, all able to outperform the no-control case at a 5% level of significance. This order of the relative algorithmic performances is also very clear in the box plots of Figure 11.24(b).

RM by AVs was again the best-performing RM technique in respect of the TTSOR, as may clearly be seen in the box plots of Figure 11.24(c). Taking the natural increase in respect of the travel times of vehicles joining the highway from the on-ramp due to RM into account, it was expected that the no-control case, having returned a TTSOR-value of 45.40 veh·h, would again outperform all RM algorithms at a 5% level of significance in respect of the TTSOR. This expectation is confirmed by the p-values in Table 11.46. The no-control case is followed in the ordering of relative algorithmic performances by kNN-TD learning for RM by AVs, achieving a TTSOR-value of 59.47 veh·h, outperforming all other algorithmic implementations at a 5% level of significance. Q-Learning for RM by AVs, which was able to achieve a TTSOR-value of 67.72 veh·h, was also able to outperform both CRM and CRM-QL at a 5% level of significance. Finally, as expected, CRM-QL was able to outperform CRM at a 5% level of significance, as these implementations returned values of 139.50 veh·h and 310.36 veh·h, respectively, in respect of the TTSOR.

The order of relative algorithmic performances in respect of the mean TISHW is the same as that in respect of the TTSHW, as may be seen in the box plots in Figure 11.24(d). From the p-values in Table 11.47, it may be deduced that CRM again achieved the best performance in respect of the mean TISHW, improving on the no-control case by 41.75%, and outperforming all other algorithms at a 5% level of significance. CRM is followed in the order of relative algorithmic performances by CRM-QL, Q-Learning and kNN-TD learning for RM by AVs, as these implementations were able to reduce the mean TISHW by 23.95%, 23.29% and 23.35%, respectively, when compared with the no-control case. As may have been expected due to the similarity in the magnitude of the reductions achieved by these three implementations, their performances were found to be statistically indistinguishable at a 5% level of significance. A similar trend emerged in respect of the maximum TISHW, as may be seen in the box plots in Figure 11.24(f). CRM was again able to achieve the smallest maximum TISHW-value, improving on the no-control case by 67.82%, and outperforming all other algorithms. The performances of CRM-QL, Q-Learning and kNN-TD learning for RM by AVs were, however, found to differ statistically at a 5% level of significance, as CRM-QL, which achieved an improvement of 41.32% over the no-control case, outperformed both Q-Learning and kNN-TD for RM by AVs at a 5% level of significance, as may be inferred from the p-values in Table 11.49. The performances of the latter two implementations, which achieved improvements of 32.85% and 31.95% respectively, over the no-control case were again found to be statistically indistinguishable at a 5% level of significance.

All algorithms were found to perform statistically distinguishably in respect of both the mean and maximum TISOR, as may be deduced from the p-values presented in Tables 11.48 and 11.50. The no-control case again returned the smallest mean and maximum TISOR-values of



(a) *TTS (veh·h)*

(b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)*

(d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)*

(f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 11.24: *PMI results for the no-control case (NC), the CRM and CRM-QL control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 3.*

1.63 minutes and 2.37 minutes, respectively. The no-control case was followed in the order of relative algorithmic performances by kNN-TD learning for RM by AVs, which was able to limit the mean and maximum TISOR-values to 2.19 minutes and 3.96 minutes, respectively, although RM is applied. Similarly, Q-Learning was able to achieve relatively small mean and maximum TISOR-values of 2.48 minutes and 4.71 minutes, respectively, outperforming both the conventional RM techniques. Q-Learning for RM by AVs is followed in the order of relative algorithmic performances by CRM-QL, which returned mean and maximum TISOR-values of 5.02 minutes and 13.10 minutes, respectively, while the order of relative algorithmic performances is completed by CRM, which achieved mean and maximum TISOR-values of 11.47 minutes and 26.76 minutes, respectively. These trends are also evident in the box plots of in Figures 11.24(e) and 11.24(g).

TABLE 11.44: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	3.1995×10^{-5}	2.9345×10^{-6}	1.6807×10^{-10}	1.5364×10^{-11}
CRM		—	9.0447×10^{-1}	3.4091×10^{-4}	1.8453×10^{-5}
CRM-QL			—	3.5404×10^{-3}	2.6374×10^{-4}
Q-Learning				—	9.8387×10^{-1}
kNN-TD					—
Mean	932.46	829.02	815.96	750.57	740.58

TABLE 11.45: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	6.9611×10^{-14}	1.3231×10^{-12}	6.0164×10^{-12}	5.1204×10^{-13}
CRM		—	$< 1 \times 10^{-17}$	1.7267×10^{-12}	5.2236×10^{-13}
CRM-QL			—	9.9377×10^{-1}	9.9785×10^{-1}
Q-Learning				—	9.9998×10^{-1}
kNN-TD					—
Mean	887.07	518.66	676.46	682.85	681.11

TABLE 11.46: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	1.2990×10^{-14}	2.8921×10^{-13}	$< 1 \times 10^{-17}$
CRM		—	$< 1 \times 10^{-17}$	5.9286×10^{-14}	2.9979×10^{-15}
CRM-QL			—	$< 1 \times 10^{-17}$	8.1934×10^{-14}
Q-Learning				—	3.7446×10^{-4}
kNN-TD					—
Mean	45.40	310.36	139.50	67.72	59.47

TABLE 11.47: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	4.3632×10^{-14}	8.7152×10^{-13}	2.2232×10^{-12}	$< 1 \times 10^{-17}$
CRM		—	$< 1 \times 10^{-17}$	3.2152×10^{-14}	1.8474×10^{-14}
CRM-QL			—	9.9168×10^{-1}	9.9322×10^{-1}
Q-Learning				—	9.9999×10^{-1}
kNN-TD					—
Mean	6.18	3.60	4.70	4.74	4.74

TABLE 11.48: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	1.3320×10^{-15}	1.6875×10^{-14}	1.0469×10^{-13}
CRM		—	5.3846×10^{-14}	2.5202×10^{-14}	6.8830×10^{-15}
CRM-QL			—	$< 1 \times 10^{-17}$	2.3981×10^{-15}
Q-Learning				—	1.3833×10^{-4}
kNN-TD					—
Mean	1.63	11.47	5.02	2.48	2.19

TABLE 11.49: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	1.4480×10^{-10}	4.5940×10^{-10}
CRM		—	9.6656×10^{-13}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-1}$
CRM-QL			–	4.8718×10^{-2}	2.7769×10^{-2}
Q-Learning				—	9.9905×10^{-1}
kNN-TD					—
Mean	22.19	7.14	13.02	14.90	15.10

TABLE 11.50: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 3. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	1.1435×10^{-14}	1.1058×10^{-13}	3.7459×10^{-13}
CRM		—	2.9208×10^{-9}	3.8636×10^{-14}	2.6645×10^{-14}
CRM-QL			—	9.0816×10^{-14}	7.6605×10^{-14}
Q-Learning				—	4.6990×10^{-4}
kNN-TD					—
Mean	2.37	26.76	13.10	4.71	3.96

11.6.4 Scenario 4

As in all previous scenarios, the performances of at least some pair of algorithms were found to be statistically distinguishable at a 5% level of significance in respect of all seven PMIs in Scenario 4, as may be inferred from the p-values presented in Table 11.51. As may be deduced from the p-values returned by Levene's test, statistical differences were again found between the variances of at least some pair of algorithmic output in respect of all seven PMIs. Therefore, the Games-Howell *post hoc* test was again employed in respect of all seven PMIs in order to ascertain between which pairs of algorithmic outputs these differences occur.

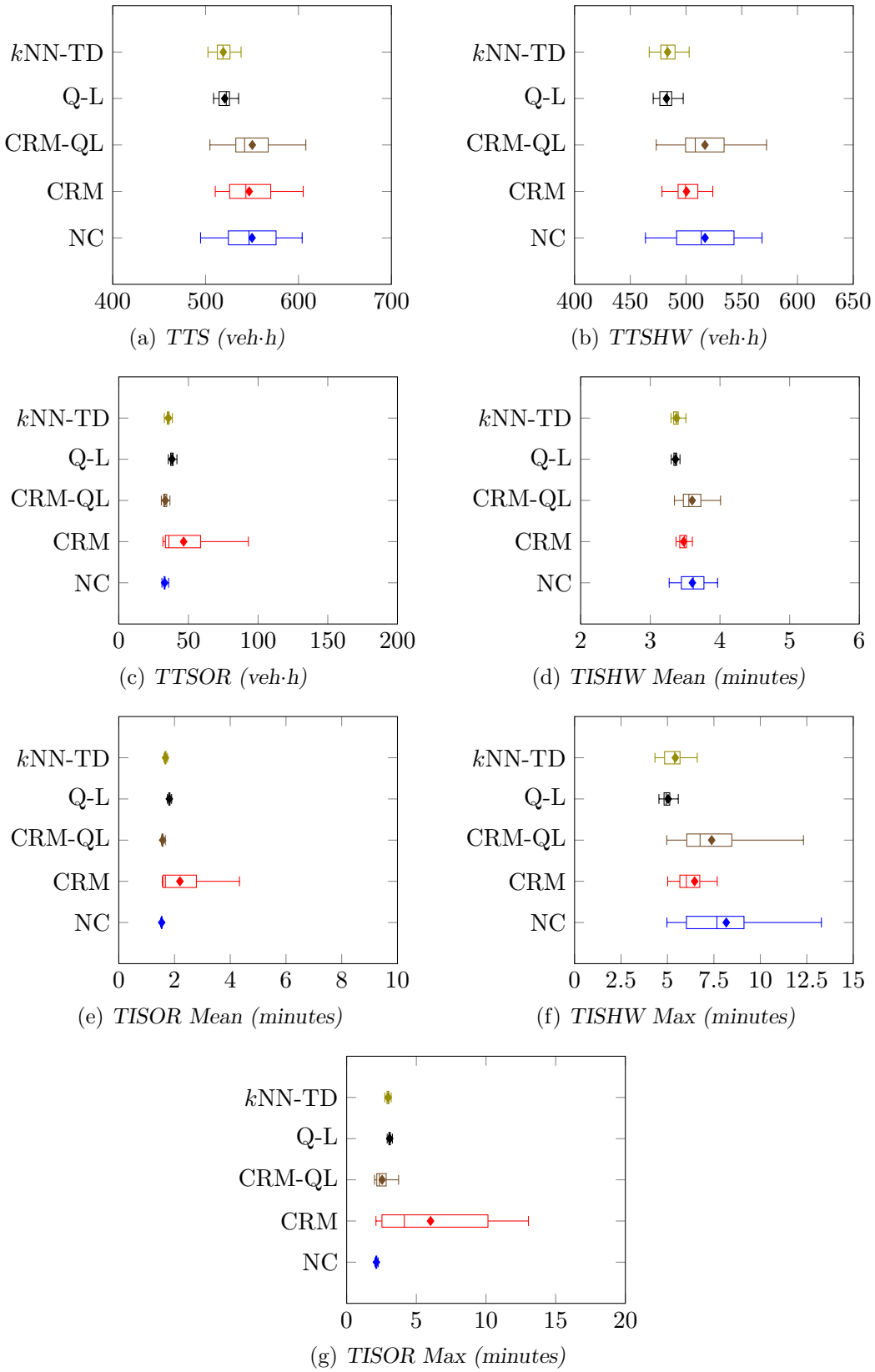
TABLE 11.51: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests in Scenario 4. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

PMI	Mean value					p-value	
	No Control	CRM	CRM-QL	Q-Learning	kNN-TD	ANOVA	Levene's Test
TTS	550.00	546.93	550.31	520.63	519.03	4.4156×10^{-10}	3.1264×10^{-13}
TTSHW	517.07	500.40	516.95	482.42	483.46	6.7613×10^{-14}	$< 1 \times 10^{-17}$
TTSOR	32.93	46.53	33.36	38.21	35.57	3.2251×10^{-9}	$< 1 \times 10^{-17}$
TISHW Mean	3.60	3.48	3.60	3.36	3.38	3.3307×10^{-16}	$< 1 \times 10^{-17}$
TISOR Mean	1.54	2.19	1.57	1.81	1.68	7.2924×10^{-10}	$< 1 \times 10^{-17}$
TISHW Max	8.16	6.46	7.38	5.04	5.42	1.6659×10^{-12}	6.3539×10^{-11}
TISOR Max	2.13	6.02	2.55	3.09	2.98	4.3299×10^{-15}	$< 1 \times 10^{-17}$

Interestingly, in respect of the TTS in Scenario 4, only the RM by AV implementations were able achieve improvements over the no-control case, as may be seen in the box plots in Figure 11.25(a). This observation is corroborated by the p-values presented in Table 11.52. As may be seen in the table, both Q-Learning and kNN-TD learning for RM by AVs were able to outperform CRM, CRM-QL and the no-control case at a 5% level of significance, while their performances were found to be statistically indistinguishable. As may have been expected from the box plots in Figure 11.25(a), the performances of CRM, CRM-QL and the no-control case were also found to be statistically indistinguishable at a 5% level of significance.

Perhaps surprisingly, the RM by AV implementations were also able to achieve the largest improvements in respect of the TTSHW in Scenario 4. Q-Learning and kNN-TD learning for RM by AVs were, in fact, able to reduce the TTSHW by 6.70% and 6.50%, respectively, when compared with the no-control case. These improvements were large enough for both these implementations to outperform CRM, CRM-QL and the no-control case at a 5% level of significance, while their performances were found to be statistically indistinguishable, as may be deduced from the p-values in Table 11.53. CRM, which was able to achieve an improvement of 3.22% over the no-control case, was also able to outperform CRM-QL at a 5% level of significance, while its performance was found to be statistically similar to that of the no-control case. Finally, the performances of CRM-QL and the no-control case were found to be statistically indistinguishable at a 5% level of significance, as CRM-QL was able to achieve an improvement of only 0.23% over the no-control case. These trends are also clearly visible in the box plots in Figure 11.25(b).

In respect of the TTSOR, the no-control case again returned the best performance, achieving a TTSOR-value of 32.93 veh·h and outperforming all algorithms at a 5% level of significance except for CRM-QL, which achieved a TTSOR-value of 33.36 veh·h. CRM-QL was, in fact, able to outperform all other algorithms at a 5% level of significance, as may be inferred from the p-values in Table 11.54. The kNN-TD learning algorithm for RM by AVs achieved the next-best performance, outperforming both Q-Learning for RM by AVs and CRM at a 5% level of significance, as it returned a TTSOR-value of 35.57 veh·h. Finally, Q-Learning for RM by AVs



(a) *TTS (veh·h)* (b) *TTSHW (veh·h)*

(c) *TTSOR (veh·h)* (d) *TISHW Mean (minutes)*

(e) *TISOR Mean (minutes)* (f) *TISHW Max (minutes)*

(g) *TISOR Max (minutes)*

FIGURE 11.25: *PMI results for the no-control case (NC), the CRM and CRM-QL control strategies, the Q-Learning algorithm (Q-L) and the kNN-TD algorithm for the RM implementation in Scenario 4.*

outperformed CRM at a 5% level of significance, as these algorithms achieved TTSOR-values of 38.21 veh·h and 46.53 veh·h, respectively. These trends in the algorithmic performances may also be seen in the box plots of Figure 11.25(c).

As for the TTSHW, the RM by AV implementations again achieved the smallest values in respect of both the mean and maximum TISHW PMIs, outperforming all other implementations at a 5% level of significance in respect of both these PMIs, as may be deduced from the p-values in Tables 11.55 and 11.57. Vehicles travelling along the highway only took on average 3.36 minutes and 3.38 minutes, respectively, when Q-Learning and kNN-TD learning for RM by AVs were employed, while these values increased to 5.04 minutes and 5.42 minutes, respectively, in respect of the maximum TISOR. The performances of the two RM by AV implementations were again found to be statistically indistinguishable at a 5% level of significance. CRM achieved the next-best performance in respect of the mean and maximum TISHW, achieving values of 3.48 minutes and 6.46 minutes, respectively, outperforming both CRM-QL and the no-control case in respect of the mean TISHW, and outperforming only the no-control case in respect of the maximum TISHW. Finally, the performance of CRM-QL, which achieved mean and maximum TISHW-values of 3.60 minutes and 7.38 minutes, respectively, was found to be statistically on par with that of the no-control case in respect of both these PMIs. These orderings of the relative algorithmic performances are again visible in the box plots of Figures 11.25(d) and 11.25(f).

As may be seen in Figures 11.25(e) and 11.25(g), the trends in the algorithmic performances in respect of the mean and maximum TISOR PMIs are again similar to that in respect of the TTSOR. The no-control case achieved the smallest mean and maximum TISOR-value of 1.54 minutes and 2.13 minutes, respectively, outperforming all other algorithms at a 5% level of significance, as may be deduced from Tables 11.56 and 11.58. The no-control case was followed in the order of relative algorithmic performances by CRM-QL, which achieved values of 1.57 minutes and 2.55 minutes, respectively, in respect of the mean and maximum TISOR, outperforming all other implementations at a 5% level of significance. The kNN-TD learning implementation for RM by AVs followed CRM-QL in the order of relative algorithmic performances, as it returned mean and maximum TISOR-values of 1.68 minutes and 2.98 minutes, respectively, thereby outperforming both Q-Learning for RM by AVs and CRM at a 5% level of significance. Finally, the performances of Q-Learning for RM by AVs and CRM were found to be statistically indistinguishable at a 5% level of significance in respect of the mean TISOR, as these implementations achieved values of 1.18 minutes and 2.19 minutes, respectively. In respect of the maximum TISOR, however, Q-Learning for RM by AVs was again able to outperform CRM at a 5% level of significance, as these implementations returned maximum TISOR-values of 3.09 minutes and 6.02 minutes, respectively.

TABLE 11.52: *Differences in respect of the total time spent in the system (TTS) by all vehicles in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTS			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	9.9332×10^{-1}	9.9999×10^{-1}	1.8215×10^{-4}	9.4168×10^{-5}
CRM		—	9.8917×10^{-1}	3.2494×10^{-5}	1.4579×10^{-5}
CRM-QL			—	7.3739×10^{-5}	3.6575×10^{-5}
Q-Learning				—	9.5571×10^{-1}
kNN-TD					—
Mean	550.00	546.93	550.31	520.63	519.03

TABLE 11.53: *Differences in respect of the total time spent in the system by vehicles travelling along the highway only (TTSHW) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSHW			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	5.9648×10^{-2}	9.9999×10^{-1}	9.7181×10^{-6}	1.7975×10^{-5}
CRM		—	4.1276×10^{-2}	1.3186×10^{-8}	4.6854×10^{-7}
CRM-QL			—	3.0276×10^{-6}	5.7990×10^{-6}
Q-Learning				—	9.8891×10^{-1}
kNN-TD					—
Mean	517.07	500.40	516.95	482.42	483.46

TABLE 11.54: *Differences in respect of the total time spent in the system by vehicles joining the highway from the on-ramp (TTSOR) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSOR			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	3.2264×10^{-3}	7.1375×10^{-1}	$< 1 \times 10^{-17}$	1.0406×10^{-10}
CRM		—	4.5567×10^{-3}	1.2967×10^{-7}	2.3343×10^{-2}
CRM-QL			—	1.3879×10^{-11}	2.2176×10^{-6}
Q-Learning				—	9.2379×10^{-9}
kNN-TD					—
Mean	32.93	46.53	33.36	38.21	35.57

TABLE 11.55: *Differences in respect of the mean time spent in the system by vehicles travelling along the highway only (TISHW Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	1.4588×10^{-2}	9.9999×10^{-1}	1.3559×10^{-6}	4.6307×10^{-6}
CRM		—	1.2933×10^{-2}	1.2179×10^{-10}	8.5745×10^{-8}
CRM-QL			—	8.6863×10^{-7}	3.0434×10^{-6}
Q-Learning				—	6.4904×10^{-1}
kNN-TD					—
Mean	3.60	3.48	3.60	3.36	3.38

TABLE 11.56: *Differences in respect of the mean time spent in the system by vehicles joining the highway from the on-ramp (TISOR Mean) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	2.1368×10^{-3}	1.5008×10^{-4}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM		—	3.7262×10^{-3}	1.3807×10^{-1}	1.9752×10^{-2}
CRM-QL			—	1.4808×10^{-11}	1.2161×10^{-11}
Q-Learning				—	1.3101×10^{-11}
kNN-TD					—
Mean	1.54	2.19	1.57	1.81	1.68

TABLE 11.57: *Differences in respect of the maximum time spent in the system by vehicles travelling along the highway only (TISHW Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISHW Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	2.7965×10^{-2}	6.9115×10^{-1}	5.9426×10^{-6}	6.2115×10^{-5}
CRM		—	2.1478×10^{-1}	3.2368×10^{-5}	7.2253×10^{-3}
CRM-QL			—	1.9300×10^{-6}	6.9490×10^{-5}
Q-Learning				—	1.7074×10^{-1}
kNN-TD					—
Mean	8.16	6.46	7.38	5.04	5.42

TABLE 11.58: *Differences in respect of the maximum time spent in the system by vehicles joining the highway from the on-ramp (TISOR Max) in Scenario 4. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISOR Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	5.0067×10^{-5}	1.5587×10^{-3}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM		—	2.6295×10^{-4}	2.0593×10^{-3}	1.3373×10^{-3}
CRM-QL			—	3.9755×10^{-5}	1.0598×10^{-3}
Q-Learning				—	7.7770×10^{-4}
kNN-TD					—
Mean	2.13	6.02	2.55	3.09	2.98

11.6.5 Discussion

Although it was outperformed in respect of the TTS by CRM in Scenario 1, while its performance was statistically indistinguishable at a 5% level of significance from CRM in Scenario 2 and from Q-Learning for RM by AVs in Scenarios 2–3, kNN-TD learning for RM by AVs consistently achieved the smallest TTS-values in Scenarios 2–4, while it achieved the second-smallest TTS-value in Scenario 1. Furthermore, kNN-TD for RM by AVs was outperformed in respect of the TTSHW only by CRM and CRM-QL in Scenario 1 and by CRM in Scenarios 2 and 3, while outperforming both CRM and CRM-QL in respect of the TTSHW in Scenario 4. Its poorer performance in respect of the TTSHW may, however, have been expected, as in the novel method of RM there is less emphasis on protecting the highway traffic flow than in the conventional RM methods. This is evident from the fact that kNN-TD for RM by AVs was able to outperform CRM and CRM-QL in respect of the TTSOR in Scenarios 1–3, while also outperforming Q-Learning for RM by AVs in respect of the TTSOR in Scenarios 2–4. The kNN-TD for RM by AVs implementation was, in fact, outperformed only once in respect of the TTSOR by CRM-QL in Scenario 4.

Q-Learning for RM by AVs performed similarly to the kNN-TD for RM by AVs implementation, as their performances were found to be statistically indistinguishable at a 5% level of significance in Scenarios 1–4 when considering both the TTS and TTSHW PMIs, and in Scenario 1 when considering the TTSOR. The kNN-TD for RM by AVs implementation was, however, generally able to achieve marginally smaller TTS and TTSHW-values than the Q-Learning implementation, while outperforming the Q-Learning implementation in respect of the TTSOR in Scenarios 2–4. Q-Learning for RM by AVs was nevertheless able to outperform CRM-QL in respect of the TTS in all four scenarios, while also outperforming CRM in respect of the TTS

in Scenarios 3 and 4. Furthermore, Q-Learning for RM by AVs was able to outperform CRM in all four scenarios when considering the TTSOR, while outperforming CRM-QL in respect of the TTSOR in Scenarios 1-3.

In summary, the novel method of RM by AVs demonstrated the ability to perform at least as well, or better than the conventional RM techniques, except in Scenario 1, when the heaviest traffic conditions prevail. Taking into account the expectation that CRM was expected to achieve the smallest TTSOR-value (due to the fact that in the original RM implementations more focus is placed on protecting the highway flow at all cost), the novel RM technique demonstrated its ability to perform statistically on-par with RM with an additional queue limit in respect of the travel times of vehicles travelling along the highway only, while outperforming the conventional RM techniques in respect of the travel times of vehicles joining the highway from the on-ramp. The expected reason for this finding is that, in the novel RM approach, vehicles never come to a complete stand-still on the on-ramp, preventing, to a certain extent, the build up of long on-ramp queues which often occurs when conventional RM techniques are employed. Finally, due to its overall superior performance in respect of the travel times of vehicles joining the highway from the on-ramp, the kN-TD for RM by AVs implementation was judged to be the best performing implementation, based on the results from all four scenarios analysed in this section.

11.7 Chapter Summary

This chapter opened in §11.1 with a description of the novel concept of employing AVs for the purpose of RM. This was followed in §11.2 by a thorough description of the RM problem in the context of AVs, as an RL problem which may be solved using RL algorithms. Thereafter, the implementations of the Q-Learning and kNN-TD RL algorithms were detailed in §11.3 and §11.4, respectively.

The focus then shifted to a thorough parameter evaluation performed in the context of Q-Learning and kNN-TD learning in §11.5. Initially, the focus of this parameter evaluation was on determining the best-performing target density values for each of these algorithms, as well as whether the algorithm should be triggered by vehicles passing a specific point or according to a fixed time schedule in §11.5.1. Once these superior target densities and the trigger method had been determined, the impact of varying on-ramp lengths on the performance of the novel RM by AVs technique was assessed in §11.5.2. This was followed by an investigation in §11.5.3 of the effect of varying the composition of AVs and human-driven vehicles on the algorithmic performances. Finally, the parameter evaluation closed in §11.5.4 with an investigation of the robustness of the policies learnt by the RL agents under varying conditions of traffic demand.

Once the parameter evaluations had been completed, and the best-performing parameter combinations determined, an algorithmic comparison was performed in §11.6. This comparison was performed in the context of the four scenarios of varying traffic demand of §5.3.2 within the benchmark simulation model of §5.1.2. The novel technique of RM by AVs was compared statistically with the best-performing implementations of conventional RM and conventional RM with queue limits, as determined in Chapter 6. It was found that the kNN-TD for RM by AVs implementation generally yielded the most favourable results over al four scenarios of traffic demand.

CHAPTER 12

Ramp Metering by Autonomous Vehicles on the N1

Contents

The purpose of this chapter is to provide a detailed description of the implementation of RM by AVs in the context of the case study simulation model of Chapter 9. The chapter opens in §12.1 with a description of the implementations of Q-Learning and kNN-TD learning for RM by AVs within the context of the simulation model. The focus then shifts in §12.2 to a thorough parameter evaluation aimed at determining the best-performing target density values for the algorithms in §12.2.1 and the influence that varying compositions of mixed human-driven and autonomous traffic flow have on the performances of the algorithms in §12.2.2. Thereafter, a thorough algorithmic comparison between the novel RM by AV implementations' performances and those of the best-performing conventional RM techniques is performed in §12.3, followed by a discussion on some of the key findings in §12.4. The chapter finally closes in §12.5 with a brief summary of the work included in the chapter.

12.1 Algorithmic Implementations

As in the implementations of conventional RM in Chapter 10, RM by AVs may be applied at three on-ramps in the case study section of the N1, at the R300 on-ramp at O_2, the Brackenfell Boulevard on-ramp at O_3 and the Okavango Road on-ramp at O_4, as may be seen in Figure 12.1. The state spaces of the RM by AVs agents remain unchanged from the implementations in the benchmark simulation model of §5.1.2 discussed in Chapter 11.

The R300 RM by AVs agent thus receives information on the downstream density ρ_{ds} at the section of highway directly downstream of the on-ramp where the traffic flows of the vehicles

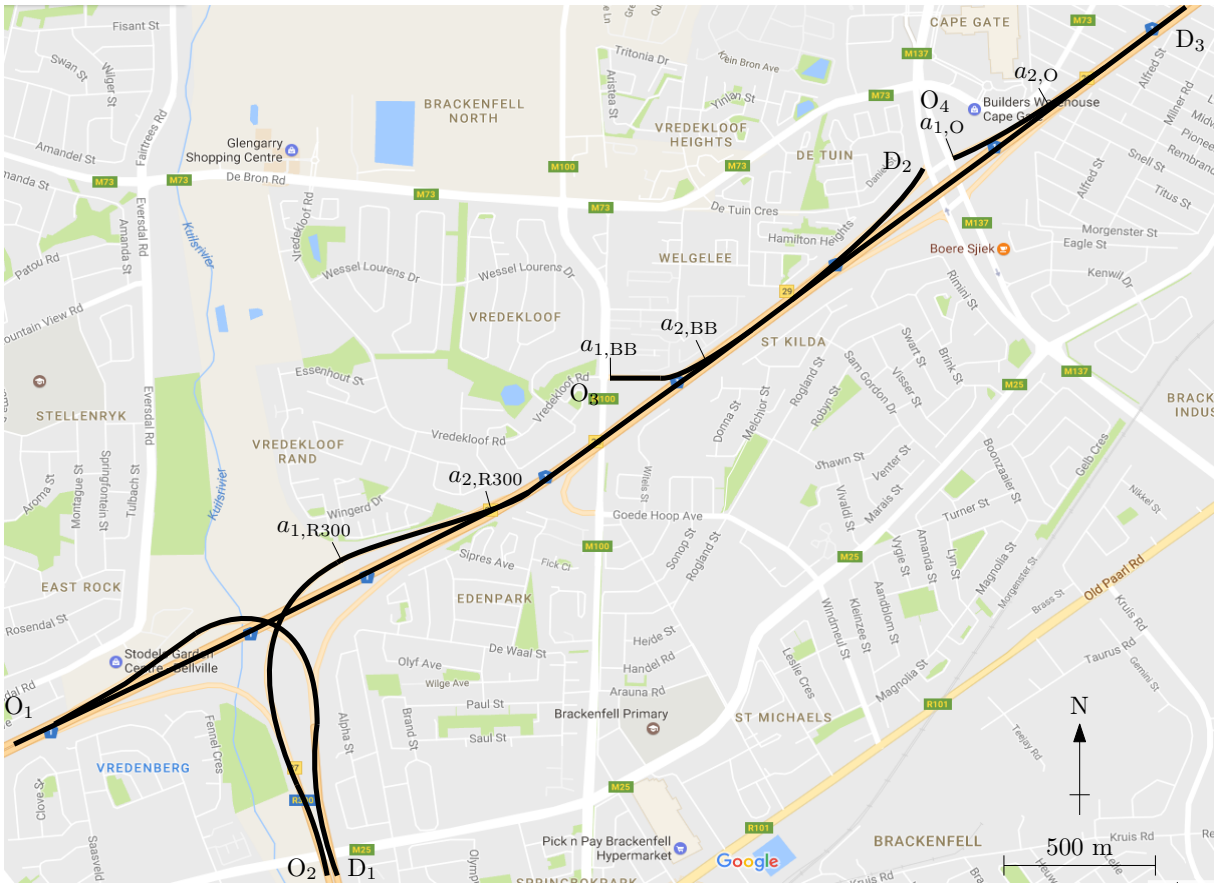


FIGURE 12.1: *The locations at which RM by AVs (indicated by the actions) may be applied in the case study area.*

joining the N1 from the R300 and the vehicles travelling along the N1 merge. The upstream density ρ_{us} is measured on the section of highway between the R300 off-ramp at D_1 and the R300 on-ramp at O_2, while the state of traffic flow on the on-ramp w is the number of vehicles present on the R300 on-ramp, as well as those vehicles present in the queue buffer (in cases where there is not sufficient space available on the on-ramp for vehicles to enter the highway network).

The downstream density for the Brackenfell Boulevard RM by AVs agent is again measured at the section where the traffic flows of those vehicles travelling along the N1 and those vehicles joining the N1 from the Brackenfell Boulevard on-ramp merge. The upstream density is measured on the section of highway between the R300 on-ramp at O_2 and the Brackenfell Boulevard on-ramp at O_3. Finally, the on-ramp queue length is again measured as the sum of the number of vehicles present on the Brackenfell Boulevard on-ramp and the number of vehicles present in the queue buffer waiting to to enter the road network as soon as sufficient space becomes available.

Similarly, for the Okavango Road RM agent, the downstream density is measured at the section of highway where the highway and the Okavango Road on-ramp traffic flows merge, while the upstream density is measured on the section of the N1 between the Okavango Road off-ramp at D_2 and the Okavango Road on-ramp at O_4. Finally, as was the case for both the R300 and Brackenfell Boulevard RM agents, the queue length is the sum of the number of vehicles present on the on-ramp and the number of vehicles in the queue buffer waiting to enter the road network at the Okavango Road on-ramp.

The action spaces of the RM agents also remain unchanged from that employed in the benchmark simulation model implementations in Chapter 11. The RM is thus again enforced by AVs travelling slowly along the respective on-ramps, specifically on the sections of the on-ramps as shown in Figure 12.1. The application areas in which RM is employed on the on-ramps are naturally limited by the existing highway traffic infrastructure. Due to the finding in Chapter 11 that longer on-ramps yield more effective RM by AVs, the aim when defining the application areas was to make these application areas as long as possible. Due to the fact that the R300 on-ramp is a dual carriageway for most of the length of the on-ramp, and the expectation that RM by AVs is more effective when there exists only a single lane (because if multiple lanes are available, human-driven vehicles will simply overtake AVs travelling slowly) RM by AVs is employed on the R300 on-ramp from the point (as shown in the figure by $a_{1,\mathrm{R300}}$) where the dual lanes merge into a single lane. The AVs then travel at the assigned speed for a distance of $\ell_{\mathrm{OR,R300}} = 400$ metres until they are assigned the nominal speed limit of 120 km/h at $a_{2,\mathrm{R300}}$. Due to the fact that the on-ramps at the Brackenfell Boulevard and Okavango Road interchanges are single-lane on-ramps, AVs are assigned the on-ramp speeds $a_{1,\mathrm{BB}}$ and $a_{1,\mathrm{O}}$, as soon as they enter the on-ramp, as may be seen in Figure 12.1. The AVs then travel at these speeds until just before the merge section, where they are assigned speed limits of 120 km/h at $a_{2,\mathrm{BB}}$ and $a_{2,\mathrm{O}}$, respectively. According to these implementations the RM by AVs is applied for $\ell_{\mathrm{OR,BB}} = 324$ metres and $\ell_{\mathrm{OR,O}} = 330$ metres, respectively at the Brackenfell Boulevard and Okavango Road on-ramps. Finally, the reward function for all three agents remains unchanged from that presented in (6.2).

12.2 Parameter Evaluations

This section is devoted to a thorough parameter evaluation with the aim of determining the best-performing target densities in respect of the Q-Learning and kNN-TD learning RM by AVs implementations in §12.2.1. Furthermore, the aim in this section is to determine the best combination of on-ramps at which RM by AVs should be applied in the case study area so as to achieve the best results. Once the best-performing combinations of on-ramps and target densities have been found, the focus shifts in §12.2.2 to the effect that varying proportions of AVs in the traffic flow have on the performance of the RM by AVs implementations.

12.2.1 Target Density Parameter Evaluations

In order to determine the best-performing target densities at each on-ramp, as well as the best-performing combinations of on-ramps at which RM by AVs should be employed, the same step-wise approach, as followed for the conventional RM implementations in Chapter 10 is again employed in this section. This parameter evaluation is again performed with 10% of the traffic flow comprising AVs, while the remaining 90% of the traffic flow are human-driven vehicles, as was the case in the implementations in the context of the benchmark simulation model of §5.1.2 in Chapter 11. Furthermore, due to the finding that the vehicle-triggered implementations of RM by AVs consitently performed better than the time-triggered implementations, the parameter evaluations performed in respect of the case study are all performed within the context of vehicle-triggered implementations.

Q-Learning

The initial focus in the parameter evaluations is again to determine the best-performing target density at the R300 on-ramp, which is the first on-ramp in the case study area. An initial rough parameter evaluation of target densities between 24 veh/km and 34 veh/km revealed that setting the target density to 31 veh/km yielded the best performance. As a result the unit interval around 31 veh/km was investigated in intervals of 0.1 veh/km, as may be seen in Table 12.1. As is evident in the table, setting the target density to 30.9 veh/km resulted in the overall-smallest TTS-value. Therefore, the target density is set to 30.9 veh/km for all further investigations and comparisons performed including a Q-Learning RM by AVs agent at the R300 on-ramp conducted in this chapter.

TABLE 12.1: *Parameter evaluation results for Q-Leaning for RM by AVs at the R300 on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	30.0	30.5	30.6	30.7	30.8	30.9	31.0
—	1 891.94	1 930.07	1 902.58	1 873.89	1 884.09	1 859.44	1 874.54
	Target density $\hat{\rho}$						
Combination	31.1	31.2	31.3	31.4	31.5	32.0	
—	1 934.61	1 903.08	1 933.14	1 885.96	1 930.46	1 927.47	

Once the best-performing target density for the agent at the R300 on-ramp had been found, the focus shifted to the Brackenfell Boulevard on-ramp. Two scenarios were investigated. In the first of these there is only a single RM by AVs agent at the Brackenfell Boulevard on-ramp, while in the second there are RM by AVs agents at both the Brackenfell Boulevard and R300 on-ramps. The initial target density investigation revealed that employing only the single RM by AVs agent at the Brackenfell Boulevard on-ramp consistently yielded smaller TTS-values than the combined case. Furthermore, it was found that the best-performing target density was 31 veh/km. Therefore, the unit interval around 31 veh/km was again investigated in intervals of 0.1 veh/km, as may be seen in Table 12.2. This more detailed investigation revealed that the best-performing target density for a Q-Learning RM by AVs agent at the Brackenfell Boulevard on-ramp is 31.2 veh/km. The target density is thus set to 31.2 veh/km for all further investigations and comparisons performed with a Q-Learning RM by AVs agent in this chapter.

TABLE 12.2: *Parameter evaluation results for Q-Learning for RM by AVs at the Brackenfell Boulevard on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	30	30.5	30.6	30.7	30.8	30.9	31.0
Alone	1 914.04	1 973.31	1 933.61	1 978.90	1 907.08	1 874.56	1 878.95
R300	1 952.21	—	—	—	—	—	1 938.41
	Target density $\hat{\rho}$						
Combination	31.1	31.2	31.3	31.4	31.5	32	
Alone	1 892.16	1 860.73	1 916.45	1 941.12	1 924.85	1 887.95	
R300	—	—	—	—	—	1 991.65	

Due to the finding that the smallest TTS-values achieved by the Q-Learning RM by AVs agents at the R300 and Brackenfell Boulevard were so similar, three cases were considered when deter-

mining the best-performing target density for the Q-Learning RM by AVs agent at the Okavango Road on-ramp. In the first of these, only a single RM by AVs agent is employed at the Okavango Road on-ramp, while in the second and third cases the RM by AVs agent works together with an RM by AVs agent at either the Brackenfell Boulevard or R300 on-ramp, respectively. The initial parameter evaluation of target densities between 24 veh/km and 34 veh/km revealed that the best-performing combination of RM by AVs agents in the Q-Learning implementations for the case study area is an agent at the Brackenfell Boulevard on-ramp and an agent at the Okavango Road on-ramp, while the target density for the Okavango Road agent is taken as 31 veh/km. The surrounding unit interval was therefore again investigated in increments of 0.1 veh/km as shown in Table 12.3. As may be seen from the TTS-values in the table, the best performance is achieved when having a Q-Learning RM by AVs agent with a target density of 31.2 veh/km at the Brackenfell Boulevard on-ramp, together with a Q-Learning RM by AVs agent with a target density of 30.6 veh/km at the Okavango Road on-ramp. This is therefore the combination of RM by AVs agents and their respective target densities employed for all further comparisons involving Q-Learning RM by AVs agents conducted in this chapter.

TABLE 12.3: *Parameter evaluation results for Q-Learning for RM by AVs at the Okavango Road on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	30	30.5	30.6	30.7	30.8	30.9	31.0
Alone	1 926.00	—	—	—	—	—	1 973.73
Brackenfell	1 888.53	1 914.61	1 836.96	1 875.10	1 881.16	1 881.17	1 843.05
R300	1 962.39	—	—	—	—	—	1 870.19
	Target density $\hat{\rho}$						
Combination	31.1	31.2	31.3	31.4	31.5	32	
Alone	—	—	—	—	—	1 880.06	
Brackenfell	1 920.74	1 894.72	1 909.30	1 950.92	1 969.22	1 879.59	
R300	—	—	—	—	—	1 967.82	

kNN-TD learning

In the kNN-TD learning parameter evaluation for determining the best-performing target densities, the initial focus was again on the R300 on-ramp. Target densities ranging from 24 veh/km to 34 veh/km were again investigated in unit intervals. The results of this investigation indicated that the smallest TTS-value is achieved with a target density of 26 veh/km, and as a result, the surrounding unit interval was considered in 0.1 veh/km increments, as may be seen in Table 12.4. This finer investigation revealed that the smallest TTS-value may be achieved by setting the target density to 26.5 veh/km. This is the target density employed at the R300 on-ramp for all further comparisons conducted in this chapter with a kNN-TD learning RM by AVs at the R300 on-ramp.

The focus of the target density parameter evaluation then shifted to the Brackenfell Boulevard on-ramp, where two scenarios were again investigated. In the first of these, only a single RM by AVs agent is employed at the Brackenfell Boulevard on-ramp, while in the second scenario, the case of employing RM by AVs agents at both the R300 and Brackenfell Boulevard on-ramps is considered. An initial rough parameter evaluation of target densities ranging from 24 veh/km to 34 veh/km revealed that, as was the case in the Q-Learning implementation, having only a

TABLE 12.4: *Parameter evaluation results for kNN-TD for RM by AVs at the R300 on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	25.0	25.5	25.6	25.7	25.8	25.9	26.0
—	1 943.29	1 941.75	1 860.44	1 872.39	1 894.43	1 941.59	1 868.40
	Target density $\hat{\rho}$						
Combination	26.1	26.2	26.3	26.4	26.5	26.6	27.0
—	1 904.58	1 858.09	1 914.09	1 932.39	1 826.84	1 903.69	1 879.34

single RM by AVs agent at the Brackenfell Boulevard consistently yielded smaller TTS-values, as may be seen from the results in Table 12.5. The initial investigation furthermore revealed that the smallest TTS-value was achieved when setting the target density for the Brackenfell Boulevard agent to 28 veh/km. Therefore, the surrounding unit interval was again investigated in increments of 0.1 veh/km. From the results of this finer investigation it is evident that setting the target density to 27.6 veh/km yielded the best performance. As a result this is the target density setting employed for a *k*NN-TD learning RM by AVs agent at the Brackenfell Boulevard on-ramp in all further comparisons conducted in this chapter.

TABLE 12.5: *Parameter evaluation results for kNN-TD for RM by AVs at the Brackenfell Boulevard on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	27	27.5	27.6	27.7	27.8	27.9	28.0
Alone	1 942.23	1 864.93	1 832.95	1 878.64	1 909.45	1 858.30	1 841.58
R300	1 954.80	—	—	—	—	—	1 925.77
	Target density $\hat{\rho}$						
Combination	28.1	28.2	28.3	28.4	28.5	29	
Alone	1 854.07	1 908.60	1 960.79	1 948.19	1 921.06	1 938.88	
R300	—	—	—	—	—	1 877.05	

As in the Q-Learning implementation, the smallest TTS-values achieved by the *k*NN-TD learning RM by AVs agents at the R300 and Brackenfell Boulevard on-ramps were again very similar, and, as a result, three cases were again investigated when considering the RM by AVs agent at the Okavango Road on-ramp. In the first of these, there is only a single RM by AVs agent operating at the Okavango Road on-ramp, while in the second and third cases RM by AVs is employed at the Okavango Road on-ramp in combination with an agent at the Brackenfell Boulevard or R300 on-ramp, respectively. As in the Q-Learning implementation, the combination of RM by AVs agents at the Brackenfell Boulevard and Okavango Road on-ramps consistently yielded the smallest TTS-values, as may be seen in Table 12.6, while the overall smallest TTS-value was achieved at a target density of 26 veh/km. The surrounding unit interval was thus again considered in intervals of 0.1 veh/km, as may be seen in the table. The results of this finer investigation revealed that setting the target density at the Okavango Road on-ramp to 26 veh/km, does indeed result in the smallest TTS-value. Therefore the combination of *k*NN-TD RM by AVs agents at the Brackenfell Boulevard on-ramp and the Okavango Road on-ramp, with target densities of 27.6 veh/km and 26 veh/km, respectively, is employed for all further comparisons involving *k*NN-TD RM by AVs implementations, conducted in this chapter.

TABLE 12.6: *Parameter evaluation results for kNN-TD for RM by AVs at the Okavango Road on-ramp, measured as the TTS by the vehicles (in veh·h).*

	Target density $\hat{\rho}$						
Combination	25	25.5	25.6	25.7	25.8	25.9	26.0
Alone	1 896.31	—	—	—	—	—	1 830.74
Brackenfell	1 865.49	1 894.04	1 814.73	1 860.99	1 836.74	1 799.72	1 765.89
R300	1 888.40	—	—	—	—	—	1 885.47
	Target density $\hat{\rho}$						
Combination	26.1	26.2	26.3	26.4	26.5	27	
Alone	—	—	—	—	—	1 895.84	
Brackenfell	1 792.78	1 878.85	1 876.44	1 863.18	1 836.10	1 1825.81	
R300	—	—	—	—	—	1 1874.49	

12.2.2 AV Percentage Parameter Evaluations

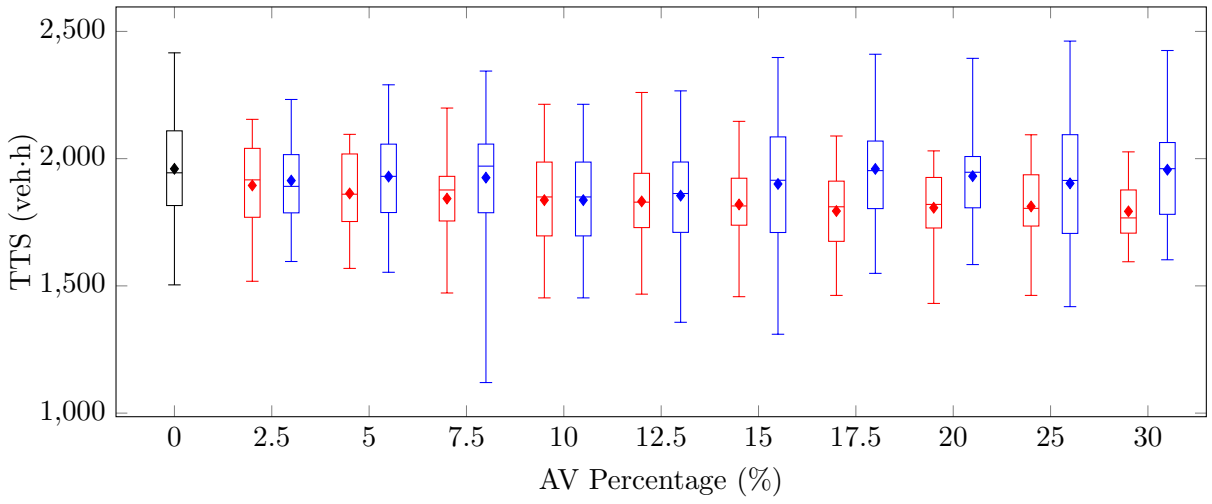
In this section, a parameter evaluation is performed with the aim of determining the effect that varying proportions of AVs present in the traffic flow have on the performance of the RM by AVs in the context of the real-world case study. Furthermore, the rigidity of policies is investigated in respect of applying policies in contexts of different AV percentages than that in which they have been trained. Similarly to the investigation performed in respect of the benchmark simulation model of §5.1.2, an initial investigation of AV percentages between 2.5% and 20% was performed in intervals of 2.5%. Thereafter, AV percentages of 25% and 30% were also again considered for the sake of completeness. In order to assess the rigidity of the policies learnt, the performances of the policies obtained by training the Q-Learning and kNN-TD learning algorithms with an AV percentage of 10% were compared with the individually trained policies for each of the various AV percentages.

Q-Learning

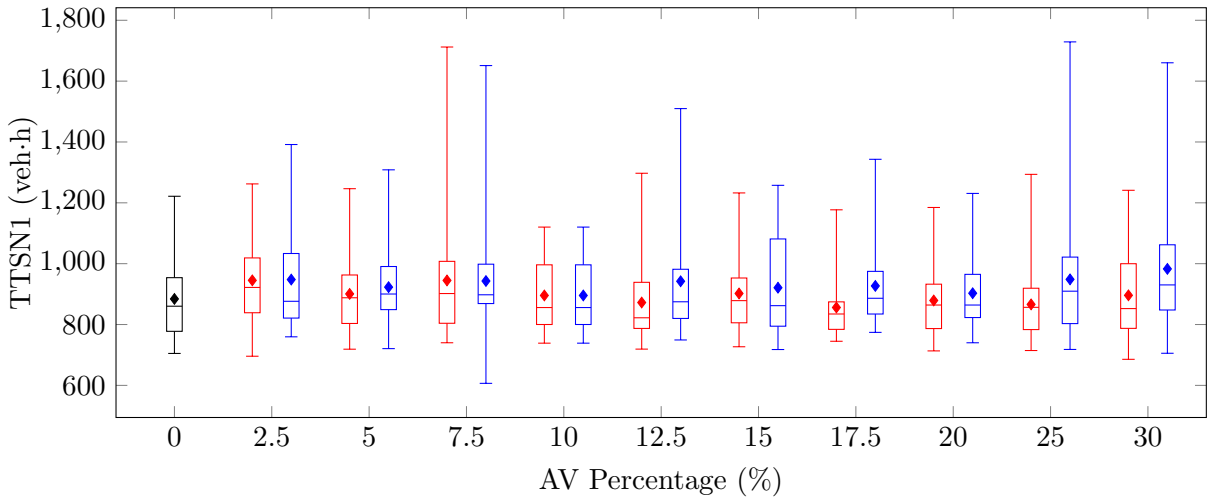
As may have been expected, the individually trained policies in respect of the Q-Learning implementation were able to consistently achieve improvements over the no-control case in respect of the TTS for all AV percentages, as is evident from the results in Table 12.7. Furthermore, the individually trained policies consistently achieved smaller TTS-values than the extrapolated policy. This trend is also evident in Figure 12.2(a). Unlike the results obtained for the benchmark simulation model of §5.1.2, however, the trend in the improvements in respect of the TTS is not approximately exponential, but rather follows a piecewise linear distribution, as the rate of improvement remains relatively constant up to an AV percentage of 7.5%, and then decreases significantly once the AV percentage exceeds 7.5%. The reduction in the rate of decrease in respect of the TTS may again be attributed to the finding that as the number of AVs in the traffic flow increases, a point at which all human-driven vehicles are affected by AVs is reached, after which the gains which may be achieved by larger numbers of AVs are limited. Although not performing as well as the individually trained policies, the extrapolated policy never returned a TTS-value larger than that of the no-control case.

Interestingly, in respect of the TTSN1, the performances of RM by AVs remained largely similar to that of the no-control case, as may be seen in Figure 12.2(b). Furthermore, no clear trend emerges in respect of the TTSN1 as the AV percentage increases, except that the individually trained policies generally achieved smaller TTSN1-values than the extrapolated policy. When considering the TTSR300-values, however, it is evident that both the individually trained policies and the extrapolated policy were able to achieve noticeable improvements over the no-control case, as may be seen from the box plots in Figure 12.2(c). An explanation for these observations may be that, although the vehicles travelling along the N1 only and those vehicles joining the N1 from the R300 on-ramp all benefit from the RM by AVs implementations at the Brackenfell Boulevard and Okavango Road on-ramps, the vehicles travelling along the N1 experience the most severe congestion at the R300 on-ramp, and therefore, no significant improvements are observed for the vehicles entering the simulated area along the N1. The improvements in the traffic flow along the N1 after the R300 on-ramp due to RM by AVs at the Brackenfell Boulevard and Okavango Road on-ramps are, however, clearly reflected in the TTSR300 PMI. This may be corroborated by the finding from Chapter 10, that when RM was employed at the R300 on-ramp in the kNN-TD implementation, this had the largest impact on the travel times of the vehicles travelling along the N1 only, while in the cases of ALINEA, PI-ALINEA and Q-Learning, where RM was applied only at the Okavango Road on-ramp, there was no significant impact on the travel times of the vehicles entering the network along the N1, suggesting that the vehicles travelling along the N1 only do, in fact, experience the most severe congestion at the R300 on-ramp. Finally, as may be seen from the results in Table 12.7, the individually trained policies were again able to consistently achieve smaller TTSR300-values than the extrapolated policy, as expected.

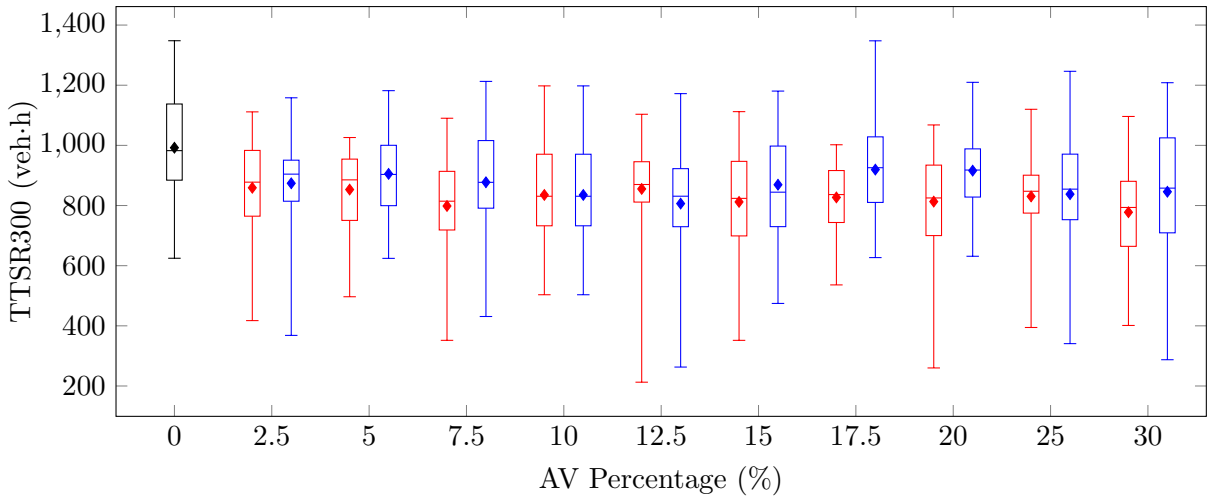
As expected, increases were observed in both the TTSBB and TTSO due to the fact that RM by AVs is applied at the Brackenfell Boulevard and Okavango Road on-ramps, as may be deduced from the box plots in Figures 12.3(a) and 12.3(b). As the demand at the Brackenfell Boulevard on-ramp is significantly smaller than that at the Okavango road on-ramp, the increases in the travel times observed by the vehicles joining the N1 from the Brackenfell Boulevard on-ramp are not as large as those observed for vehicles joining the N1 from the Okavango Road on-ramp. The increases in the travel times observed for those vehicles joining the N1 from the Brackenfell Boulevard on-ramp follow an approximately exponential growth, as the rate of increase decays once an AV percentage of 5% has been reached. Two explanations are offered for this observation. First, the vehicles joining the N1 from the Brackenfell Boulevard on-ramp benefit from the improved traffic flow along the N1 due to RM by AVs at the Okavango Road on-ramp, which may compensate for the increased travel times while travelling along the on-ramp. Secondly, the vehicles joining the N1 from the Brackenfell Boulevard on-ramp spend a comparatively long time travelling along the remaining stretch of highway compared to the travel times along the on-ramp, and as a result, an increase in the travel time along the on-ramp may not reflect as clearly when considering the total travel time by these vehicles. When considering the travel times of vehicles joining the N1 from the Okavango Road on-ramp, an approximately linear relationship between the travel time and the AV percentage is observed, as the travel times increase together with the AV percentage, because naturally more vehicles are affected by slow-travelling AVs. This approximately linear increase is simlar to that observed in respect of the TTSOR-values returned by the Q-Learning implementation in the context of the benchmark model of §5.1.2 in Chapter 11. Note again, however, that the individually trained policies were able to generally achieve smaller TTSO-values than the extrapolated policy, as the learning agent notices that similar metering rates may be achieved by enforcing larger speed limits when larger percentages of AVs are present in the system.



(a)

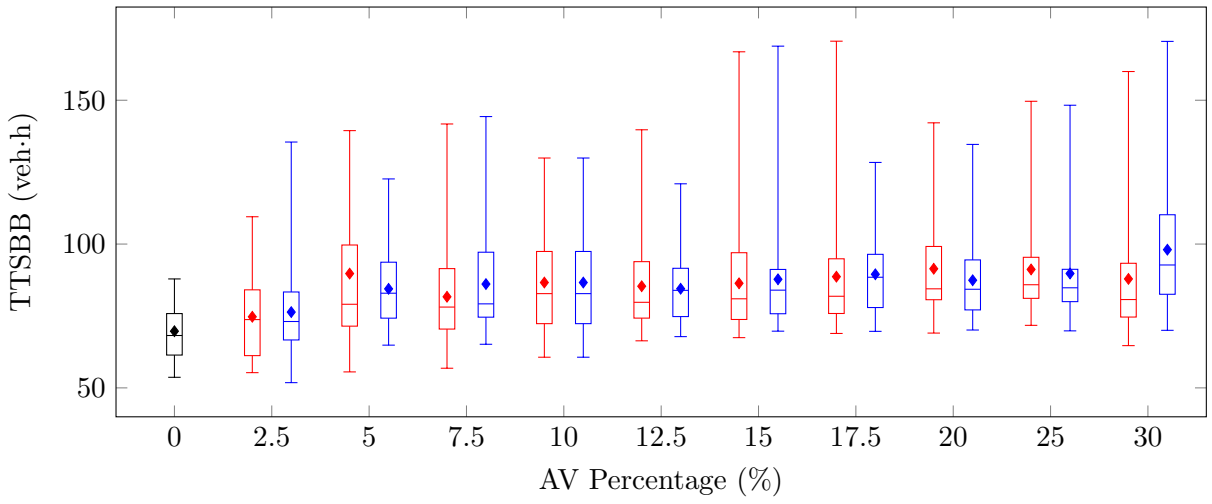


(b)

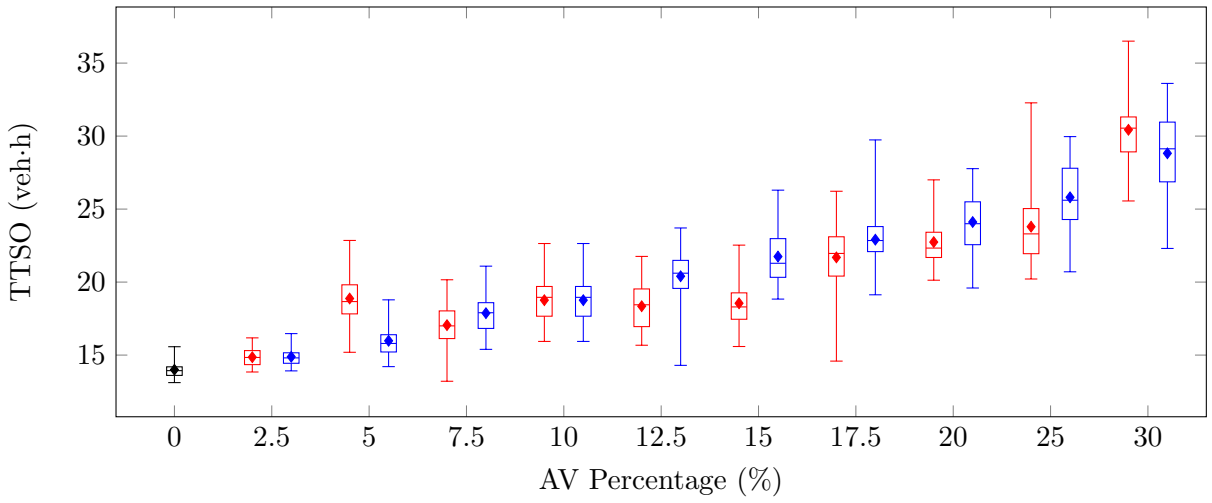


(c)

FIGURE 12.2: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) at on-ramps where RM by AVs is not applied.*



(a)



(b)

FIGURE 12.3: *A comparison of the performance of Q-Learning for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) at on-ramps where RM by AVs is applied.*

TABLE 12.7: *Traffic demand evaluation results for the vehicle-triggered Q-Learning RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	1 913.78	1 929.48	1 925.35	1 836.96	1 854.55	1 900.49	1 959.58	1 931.25	1 902.75	1 956.51
	Indiv.	1 894.65	1 863.18	1 842.95	1 836.96	1 831.88	1 820.03	1 794.09	1 806.98	1 812.24	1 792.82
TTSN1	Extra.	947.79	922.88	943.00	895.53	942.14	920.98	926.86	902.84	948.46	982.82
	Indiv.	945.17	900.60	944.88	895.53	872.28	902.06	855.98	879.01	866.15	895.97
TTSR300	Extra.	873.95	905.17	877.38	835.15	806.45	869.12	919.48	915.97	837.79	845.74
	Indiv.	858.96	853.08	798.41	835.15	855.17	811.99	826.95	813.02	830.29	777.68
TTSBB	Extra.	76.35	84.42	86.09	86.62	84.46	87.76	89.49	87.42	89.76	98.05
	Indiv.	74.72	89.73	81.67	86.62	85.34	86.36	88.66	91.41	91.18	87.88
TTSO	Extra.	14.88	15.98	17.87	18.76	20.40	21.75	22.91	24.11	25.81	28.82
	Indiv.	14.86	18.88	17.05	18.76	18.35	18.55	21.69	22.74	23.79	30.43

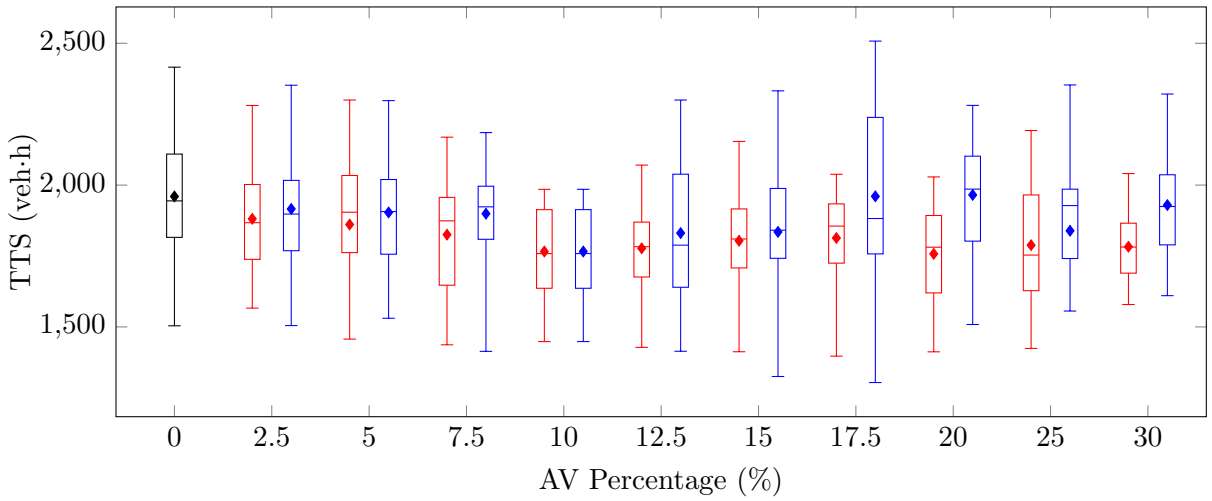
kNN-TD learning

As in the case of the Q-Learning for RM by AVs implementation, the individually trained policies of the kNN-TD learning RM for AVs implementation were consistently able to achieve smaller TTS-values than the extrapolated policy, as may clearly be seen in Figure 12.4(a). Furthermore, the individually trained policies were always able to reduce the TTS when compared with the no-control case. The decrease in the TTS does not, however, follow an approximately exponential decline as it did in the simplified case of the benchmark simulation model of §5.1.2 in Chapter 11. Instead, similarly to the Q-Learning implementatino in the case study, the decrease in the TTS is more gradual, as approximately linear decreases in the TTS are observed until 10% of the traffic flow comprises AVs, at which point the TTS-values achieved stabilise. This observation is corroborated by the mean TTS-values presented in Table 12.8, and may again be due to all human-driven vehicles being affected by AVs once the 10% mark has been reached, at which point the possible further improvements are limited. The extrapolated policy was also generally able to achieve improvements in the TTS when compared with the no-control case, except for an AV percentage of 20%. The differences in the performance of the individually trained and extrapolated policies in the context of this real-world case study are larger than for the simplified benchmark model. A similar trend was observed for the Q-Learning implementation in the context of the case study. This larger discrepancy between the individually trained and extrapolated policies may be due to the fact that, with a larger study area, more stochasticity and complexity is introduced and, as a result, more factors influence the performance of a policy. This, in turn, may result in larger uncertainty around the quality of actions chosen within various scenarios of AV percentages. The individually trained policies are naturally more fine-tuned to the specific scenario in which the agent operates, thus achieving better and more consistent results.

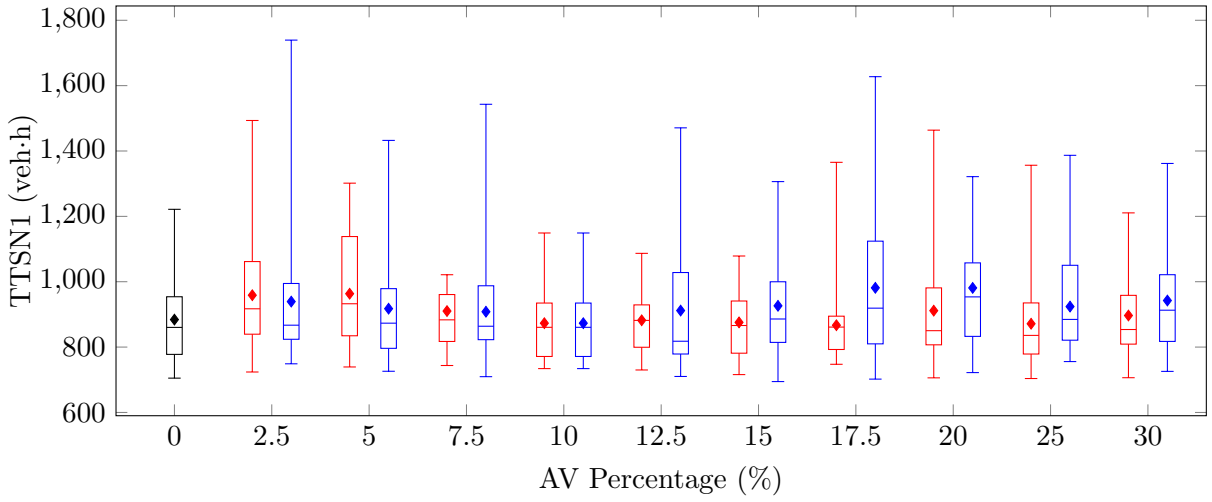
As in the case of Q-Learning, the performances of the policies learnt by the kNN-TD agents were again largely similar to that of the no-control case in respect of the TTSN1-values, as may be seen in the box plots in Figure 12.4(b). As expected, the individually trained policies were typically able to achieve smaller TTSN1-values than the extrapolated policy, as may be seen in the results presented in Table 12.8. The lack of improvement in the TTSN1, which is similar to that observed for the Q-Learning for RM by AVs implementation may again be attributed to the fact that RM is not applied at the R300 on-ramp, where severe congestion therefore still prevails. This congestion has the largest impact on the vehicles entering the system along the N1 and, as a result, no significant improvements were recorded in respect of the TTSN1.

When considering the TTSR300, however, clear improvements over the no-control case were observed at all of the various AV percentages for both the individually trained policies and the extrapolated policy, as may be seen in the box plots of Figure 12.4(c). These results are again similar to those recorded for the Q-Learning implementation. This improvement in respect of the TTSR300 may again be ascribed to the fact that most of the vehicles joining the N1 from the R300 benefit from the improved traffic flow along the N1 due to RM by AVs at both the Brackenfell Boulevard and Okavango Road on-ramps, resulting in the improved travel times. As may be seen in Table 12.8, the individually trained policies were again able to achieve smaller TTSR300-values than the extrapolated policies, which may be the result of more effective RM in the case of individually trained policies, resulting from the more scenario-specific training of the algorithm.

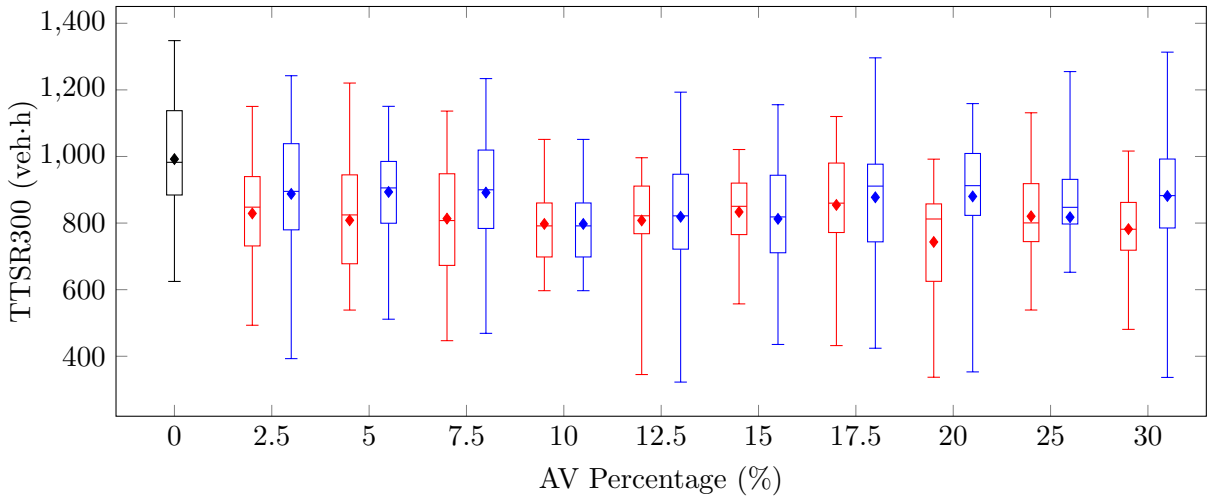
Increases over the no-control case were again recorded in respect of both the TTSBB and TTSO PMIs, as may be seen in Figures 12.5(a) and 12.5(b). These increases were again expected due to the fact that RM by AVs is applied at both the Brackenfell Boulevard and Okavango



(a)

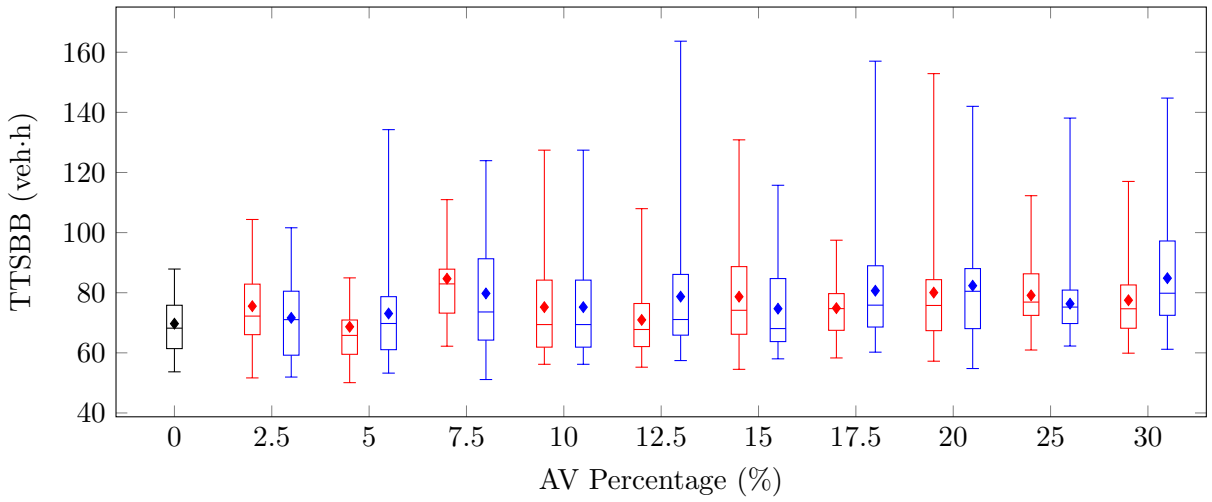


(b)

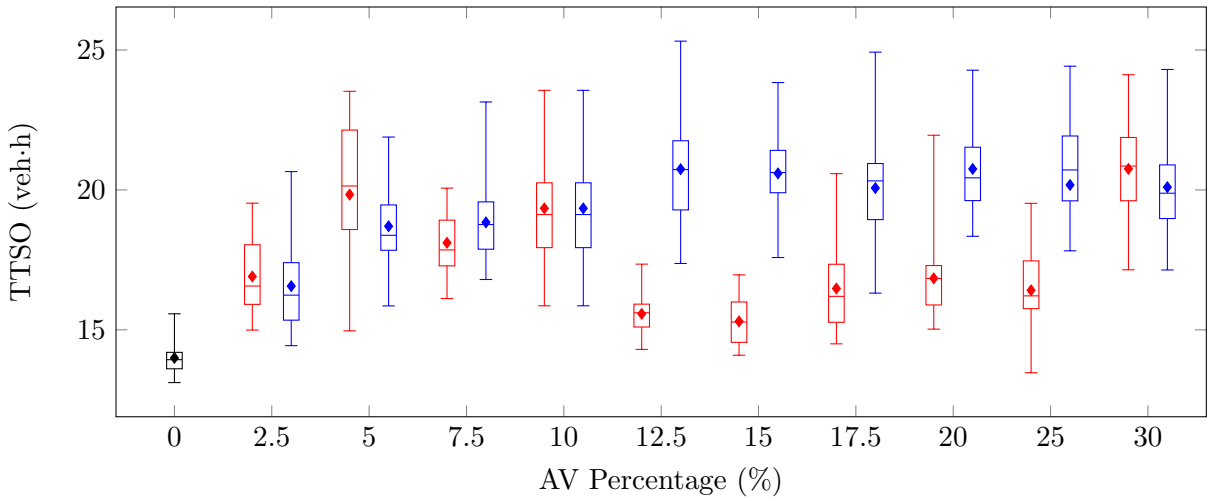


(c)

FIGURE 12.4: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) at on-ramps where RM by AVs is not applied.*



(a)



(b)

FIGURE 12.5: *A comparison of the performance of kNN-TD for RM by AVs in respect of individually trained policies (indicated in red) and extrapolated policies (indicated in blue) at on-ramps where RM by AVs is applied.*

TABLE 12.8: *Traffic demand evaluation results for the vehicle-triggered kNN-TD RM by AVs implementation, measured in terms of the TTS by the vehicles (in veh·h).*

PMI	Policy	AV Percentage (%) 2.5	5	7.5	10	12.5	15	17.5	20	25	30
TTS	Extra.	1 915.94	1 903.57	1 899.20	1 765.89	1 830.80	1 834.93	1 960.39	1 965.31	1 838.91	1 929.59
	Indiv.	1 881.00	1 861.01	1 827.25	1 765.89	1 777.17	1 803.75	1 813.56	1 752.45	1 788.19	1 777.60
TTSN1	Extra.	939.16	917.49	908.14	873.14	911.71	926.22	981.30	981.01	923.88	942.43
	Indiv.	958.72	963.27	910.20	873.14	881.96	875.58	866.89	911.44	871.39	896.60
TTSR300	Extra.	887.62	893.41	891.46	797.38	818.87	812.55	877.43	880.10	817.57	881.31
	Indiv.	828.92	808.36	813.34	797.38	807.84	833.36	854.43	743.27	820.43	781.88
TTSBB	Extra.	71.63	73.08	79.76	75.21	78.70	74.69	80.66	82.34	76.34	84.84
	Indiv.	75.53	68.66	84.67	75.21	70.95	78.67	74.87	80.34	79.12	77.51
TTSO	Extra.	16.56	18.70	18.84	19.34	20.74	20.59	20.07	20.75	20.18	20.10
	Indiv.	16.91	19.83	18.11	19.34	15.57	15.30	16.48	16.84	16.41	20.75

Road on-ramps. As in the case of Q-Learning for RM by AVs, the increases in respect of the TTSBB are comparatively small when compared with those in respect of the TTSO. A reason for the realtively small increases observed in respect of the TTSBB may again be that most of the vehicles joining the N1 from the Brackenfell Boulevard on-ramp benefit from the improved traffic flow along the N1 as a result of RM by AVs at the Okavango Road on-ramp. Furthermore, due to the fact that these vehicles spend a significant amount of time in the system once they have entered the traffic flow on the N1 from the on-ramp, the proportion of time spent on the on-ramp is relatively small when compared with the total time spent in the system by these vehicles, which may result in the increase in the travel times along the highway not being reflected as clearly in the TTSBB PMI. As may be deduced from the results in Table 12.8, the individually trained policies were again more effective in limiting the increase in the TTSO than the extrapolated policy. The increases in the travel times along the on-ramp by the vehicles entering the highway from the Okavango Road on-ramp are more pronounced. Because the Okavango Road on-ramp is the last on-ramp in the case study area, the vehicles entering the highway from the Okavango Road on-ramp travel along the highway only for a relatively short distance and, as a result, the time spent on the on-ramp comprises a larger proportion of the total time spent in the system by these vehicles. As may be seen in Figure 12.5(b), the increase in the TTSO recorded for the extrapolated policy follows an approximately exponential rate, as the rate of increase slows once an AV percentage of 12.5% is reached. The trend in respect of the TTSO recorded for the individually trained policies is more irregular. As may be seen in the figure, the individually trained policies typically achieve significantly smaller TTSO-values than the extrapolated policy, especially when the proportion of AVs in the traffic flow is larger than 10%. This may be due to the fact that the learning agent realises that, due to the larger numbers of AVs present in the traffic flow, the required metering rate may be achieved by assigning larger speed values to the AVs, which is not the case for the extrapolated policy.

12.3 Algorithmic Comparison

This section is devoted to a thorough algorithmic performance comparison of the novel RM by AVs technique with the best-performing conventional RM and conventional RM with queue limits implementations within the context of the case study simulation model of Chapter 9. In order to ascertain which AV percentage to employ for the comparison of RM by AVs with the conventional RM methods, a statistical comparison was performed so as to determine at which AV percentage the improvements in respect of the TTS ceased to be statistically significant for both the Q-Learning and kNN-TD RM by AVs implementations. The results of the ANOVA and Levene tests performed in this respect are presented in Table 12.9.

TABLE 12.9: *The p-values for the ANOVA and Levene statistical tests performed in order to ascertain whether statistical differences occur at various levels of AV percentages. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

Algorithm	ANOVA	Levene's Test
Q-Learning	3.5636×10^{-2}	4.2326×10^{-1}
kNN-TD	2.5389×10^{-2}	1.2255×10^{-1}

As may be seen in the table, statistical differences exist at a 5% level of significance between at least some pair of AV percentages in respect of both the Q-Learning and kNN-TD learning RM by AVs implementations. Furthermore, Levene's test revealed that the variances achieved in respect of the various AV percentages do not differ statistically at a 5% level of significance.

Therefore, the Fisher LSD *post hoc* test was performed in order to ascertain between which pairs of AV percentages statistical differences occur for both the Q-Learning and kNN-TD learning implementations, as may be seen in Tables 12.10 and 12.11.

As shown in Table 12.10, when employing Q-Learning for RM by AVs, an AV percentage as small as 5% is never outperformed at a 5% level of significance in respect of the TTS by a larger AV percentage. When considering the kNN-TD RM by AVs implementation, however, the smallest AV percentage which is never outperformed at a 5% level of significance by a larger AV percentage rises to 10%, as may be seen in Table 12.11. As a result, an AV percentage of 10% was employed for all further algorithmic performance comparisons conducted in this chapter.

Due to the fact that in the conventional RM implementations of Chapter 10, the kNN-TD implementation returned the best performance in the context of the case study simulation model of Chapter 9, the kNN-TD RM implementation was chosen as the conventional RM implementation (without queue limits) against which to measure the performance of the novel RM technique in the context of a real-world scenario. As in the previous chapter, the kNN-TD implementation for conventional RM is again referred to as CRM in order to avoid ambiguity with the kNN-TD implementation for RM by AVs. Similarly, the novel RM technique is again also compared with the kNN-TD RM implementation with the addition of queue limits, as the kNN-TD RM implementation with queue limits returned the most favourable results when queue limits were employed. The kNN-TD implementation for conventional RM with queue limits is again referred to as CRM-QL, in order to avoid ambiguity with the kNN-TD implementation for RM by AVs.

From the results of the ANOVA performed on the various algorithmic outputs in the context of the case study simulation model of Chapter 9, presented in Table 12.12, it is evident that there are statistical differences between at least some pair of algorithmic outputs in respect of all thirteen PMIs. Furthermore, the Levene test revealed that the variances of the algorithmic output data are statistically indistinguishable in respect of the TTS, TTSR300 and mean and maximum TISR300 PMIs, while the variances in respect of the other nine PMIs were found to be statistically different at a 5% level of significance. As a result, the Fisher LSD *post hoc* test was performed in order to ascertain between which pairs of algorithmic outputs these differences occur in respect of the TTS, TTSR300 and mean and maximum TISR300 PMIs, while the Games-Howell test was performed for this purpose in respect of the other nine PMIs.

As may clearly be seen from the box plots in Figure 12.6(a), all four algorithmic implementations were able to outperform the no-control case in respect of the TTS. This observation is corroborated by the p-values presented in Table 12.13. Interestingly, however, none of the algorithmic implementations were able to outperform one another at a 5% level of significance, as may be deduced from the table. CRM-QL nevertheless achieved the smallest TTS-value of 1 750.33 veh·h, followed by kNN-TD for RM by AVs, which returned a TTS-value of 1 765.89 veh·h. The next-smallest TTS-value of 1 768.29 veh·h, was achieved by CRM, while Q-Learning returned the largest TTS-value of 1 836.96 veh·h.

In respect of the TTSN1, the conventional RM techniques achieved the best performance, outperforming both the RM by AVs implementations and the no-control case at a 5% level of significance, as may be deduced from the p-values presented in Table 12.14. CRM achieved the best performance, returning a 31.41% improvement over the no-control case, followed by CRM-QL which achieved a 30.48% improvement over the no-control case. As may have been expected, the performances of CRM and CRM-QL were statistically indistinguishable at a 5% level of significance. The AV by RM implementations, on the other hand, performed statistically similar to the no-control case at a 5% level of significance, as Q-Learning for RM by AVs returned a 1.29% increase in TTSN1, while kNN-TD for RM by AVs achieved a 1.24% improvement over the no-control case in respect of the TTSN1. These trends are also clearly visible in the box plots

TABLE 12.10: *Differences in respect of the total time spent in the system (TTS) by the Q-Learning algorithm with varying AV percentages for the case study. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TTS									
AV Percentage	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	25%	30%
2.5%	—	4.6227×10^{-1}	2.2750×10^{-1}	1.7821×10^{-1}	1.4310×10^{-1}	8.1913×10^{-2}	1.9312×10^{-2}	4.1177×10^{-2}	5.4833×10^{-2}	1.7851×10^{-2}
5%		—	6.3635×10^{-1}	5.4004×10^{-1}	4.6461×10^{-1}	3.1350×10^{-1}	1.0708×10^{-1}	1.8962×10^{-1}	2.3427×10^{-1}	1.0081×10^{-1}
7.5%			—	8.8862×10^{-1}	7.9587×10^{-1}	5.9212×10^{-1}	2.5391×10^{-1}	4.0077×10^{-1}	4.7297×10^{-1}	2.4181×10^{-1}
10%				—	9.0555×10^{-1}	6.9226×10^{-1}	3.1669×10^{-1}	4.8367×10^{-1}	5.6343×10^{-1}	3.0261×10^{-1}
12.5%					—	7.8165×10^{-1}	3.7729×10^{-1}	5.6065×10^{-1}	6.4611×10^{-1}	3.6151×10^{-1}
15%						—	5.4444×10^{-1}	7.6049×10^{-1}	8.5553×10^{-1}	5.2495×10^{-1}
17.5%							—	7.6310×10^{-1}	6.7145×10^{-1}	9.7634×10^{-1}
20%								—	9.0228×10^{-1}	7.4060×10^{-1}
25%									—	6.4997×10^{-1}
30%										—
Mean	1 894.65	1 863.18	1 842.95	1 836.96	1 831.88	1 820.03	1 794.09	1 806.98	1 812.24	1 792.82

TABLE 12.11: *Differences in respect of the total time spent in the system (TTS) by the kNN-TD learning algorithm with varying AV percentages for the case study. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Fisher LSD test p-values: TTS									
AV Percentage	2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	25%	30%
2.5%	—	6.9438×10^{-1}	1825.2×10^{-1}	2.4272×10^{-2}	4.1985×10^{-2}	1.2965×10^{-1}	1.8560×10^{-1}	1.5682×10^{-2}	3.8895×10^{-2}	3.2458×10^{-2}
5%		—	4.8229×10^{-1}	4.2305×10^{-2}	4.8357×10^{-2}	2.6089×10^{-1}	2.6089×10^{-1}	3.2533×10^{-2}	4.5683×10^{-2}	4.2403×10^{-2}
7.5%			—	2.4384×10^{-1}	3.4500×10^{-1}	6.7264×10^{-1}	8.1825×10^{-1}	4.8334×10^{-2}	4.6656×10^{-1}	4.0214×10^{-1}
10%				—	8.2451×10^{-1}	4.5696×10^{-1}	3.4912×10^{-1}	8.6834×10^{-1}	6.6116×10^{-1}	7.4256×10^{-1}
12.5%					—	6.0143×10^{-1}	4.7465×10^{-1}	6.9841×10^{-1}	8.2850×10^{-1}	9.1499×10^{-1}
15%						—	8.4713×10^{-1}	3.6318×10^{-1}	7.5975×10^{-1}	6.7766×10^{-1}
17.5%							—	2.7063×10^{-1}	6.1813×10^{-1}	5.4298×10^{-1}
20%								—	5.4587×10^{-1}	6.2120×10^{-1}
25%									—	9.1250×10^{-1}
30%										—
Mean	1 881.00	1 861.01	1 827.25	1 765.89	1 777.17	1 803.75	1 813.56	1 752.45	1 788.19	1 777.60

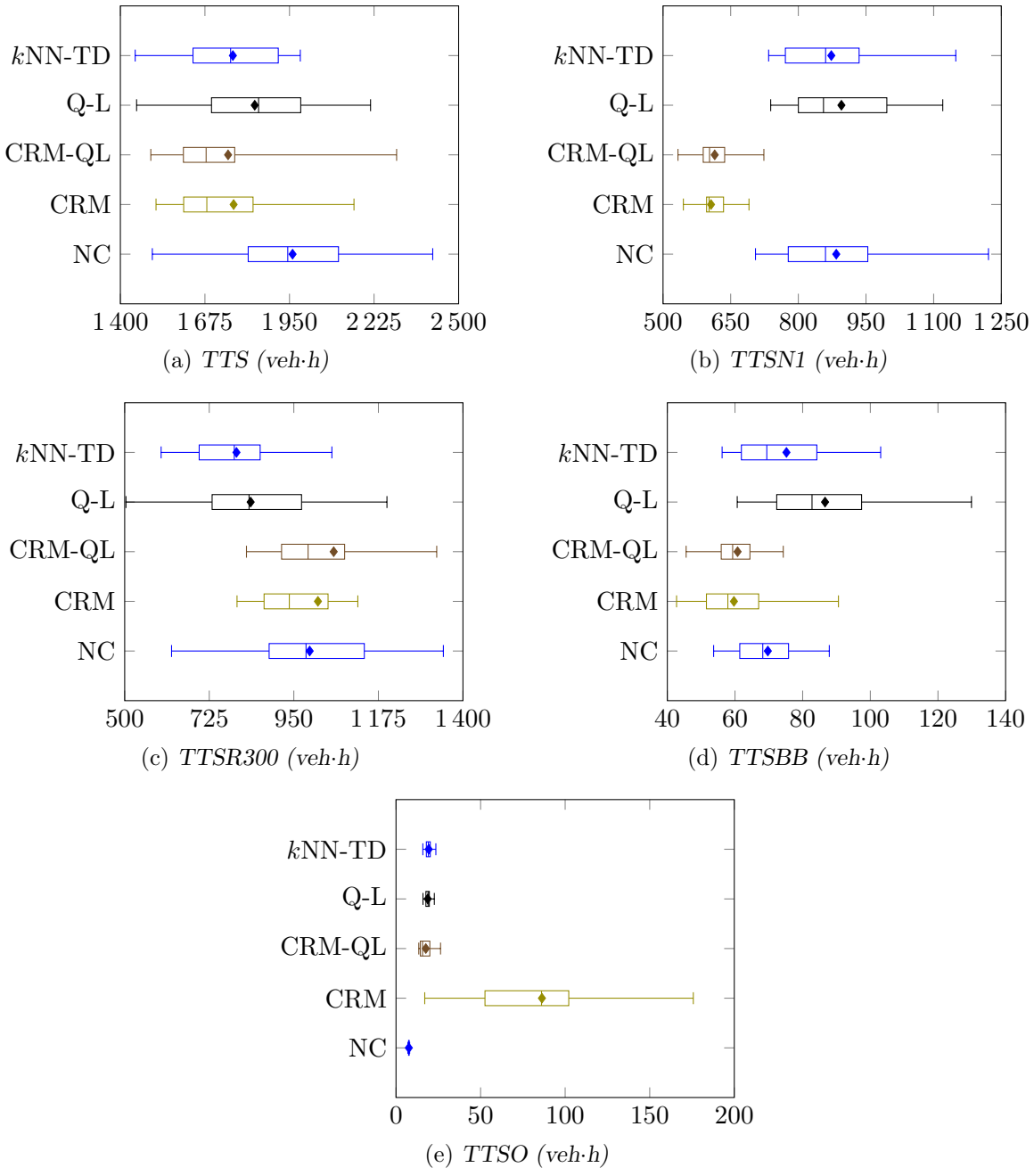
TABLE 12.12: *The mean values of all PMIs, as well as the p-values for the ANOVA and Levene statistical tests, for the case study. A p-value less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

	Mean value					*p*-value	
PMI	No Control	CRM	CRM-QL	Q-Learning	*k*NN-TD	ANOVA	Levene's Test
TTS	1 960.01	1 768.29	1 750.33	1 836.96	1 765.889	6.5034×10^{-2}	7.3629×10^{-1}
TTSN1	884.11	606.44	614.63	895.53	873.14	$< 1 \times 10^{-17}$	4.5023×10^{-10}
TTSR300	992.19	1 014.18	1 056.11	835.15	797.34	1.7529×10^{-8}	2.9710×10^{-1}
TTSBB	69.71	59.69	60.76	86.62	75.21	1.3223×10^{-13}	3.6151×10^{-5}
TTSO	14.00	86.27	17.62	18.76	19.34	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISN1 Mean	1.24	0.89	0.90	1.25	1.22	$< 1 \times 10^{-17}$	2.2109×10^{-9}
TISN1 Max	5.30	3.88	4.70	8.09	8.41	3.9308×10^{-7}	1.1554×10^{-3}
TISR300 Mean	8.84	14.32	14.77	7.37	7.11	$< 1 \times 10^{-17}$	7.0096×10^{-2}
TISR300 Max	25.42	42.03	42.28	20.71	22.66	$< 1 \times 10^{-17}$	6.6702×10^{-2}
TISBB Mean	2.01	1.72	1.73	2.51	2.18	$< 1 \times 10^{-17}$	3.9660×10^{-5}
TISBB Max	5.05	4.46	4.50	5.42	5.24	4.3645×10^{-3}	3.6562×10^{-2}
TISO Mean	0.82	5.03	1.04	1.10	1.15	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
TISO Max	1.50	18.43	6.41	4.00	4.03	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$

of Figure 12.6(b). Note, however, that although the RM by AVs implementations were found to perform statistically on par with the no-control case, these implementations achieved a more stable traffic flow along the N1, as may be deduced from the smaller interquartile ranges in the box plots corresponding to these implementations, when compared with that of the no-control case.

When considering the TTSR300, the RM by AVs implementations achieved the best performances, as they outperformed all other implementations at a 5% level of significance, while their performances were again found to be statistically indistinguishable at a 5% level of significance. The *k*NN-TD for RM by AVs implementation achieved the smallest TTSR300-value of 797.34 veh·h, followed by Q-Learning for RM by AVs, which returned a TTSR300-value of 835.15 veh·h. The increases in the TTSR300 over the no-control case recorded for CRM and CRM-QL were, however, to be expected, as RM is applied at the R300 on-ramp in both of these implementations. As may be seen from the *p*-values in Table 12.15, these increases were, however, not large enough for the algorithmic performances to be classified as statistically different from that of the no-control case at a 5% level of significance. This similarity in the algorithmic performances of the no-control case, CRM and CRM-QL is also reflected in the TTSR300 values as the no-control case, CRM and CRM-QL achieved values of 992.19 veh·h, 1 014.18 veh·h and 1 056.11 veh·h, respectively. These trends in the relative algorithmic performances are also evident in the box plots in Figure 12.6(c).

As may be seen in the box plots in Figure 12.6(d), CRM and CRM-QL returned the best performances in respect of the TTSBB, outperforming the no-control case and both the RM by AVs implementations at a 5% level of significance, as they achieved 14.37% and 12.84% improvements over the no-control case, respectively. The no-control case achieved the next best performance, outperforming Q-Learning for RM by AVs, which resulted in a 24.58% increase in the TTSBB, at a 5% level of significance, while it was found to perform statistically indistinguishably from *k*NN-TD for RM by AVs, although *k*NN-TD for RM by AVs resulted in a 7.89% increase over the no-control case. Finally, as may be deduced from the *p*-values in Table 12.16, Q-Learning and *k*NN-TD for RM by AVs again performed statistically indistinguishably. The increases recorded for the RM by AVs implementations were to be expected, as RM by AVs is applied at the Brackenfell Boulevard on-ramp in both these implementations.



(a) *TTS (veh·h)*

(b) *TTSN1 (veh·h)*

(c) *TTSR300 (veh·h)*

(d) *TTSBB (veh·h)*

(e) *TTSO (veh·h)*

FIGURE 12.6: *Total time spent in the system PMI results for the no-control case (NC), conventional RM (CRM), conventional RM with queue limits (CRM-QL), Q-Learning for RM by AVs and kNN-TD for RM by AVs applied to the case study model of Chapter 9.*

As expected, the no-control case achieved the smallest TTSO-value of 14.00 veh·h, outperforming all algorithmic implementations at a 5% level of significance, as may be deduced from the results of the Games-Howell test, presented in Table 12.17. CRM-QL, Q-Learning and kNN-TD for RM by AVs returned the next best performances, achieving TTSO-values of 17.62 veh·h, 18.76 veh·h and 19.34 veh·h, respectively, thereby outperforming CRM, while their performances were found to be statistically indistinguishable from one another at a 5% level of significance. From the box plots in Figure 12.6(e), it is, however, evident that the traffic flow along the Okavango Road on-ramp is more stable in the case of RM by AVs, as may be deduced from the smaller interquartile ranges associated with the corresponding box plots, when compared with those corresponding to the conventional RM implementations. Finally, the order of relative algorithmic performances is completed by CRM, which achieved a TTSO-value of 86.27 veh·h.

TABLE 12.13: *Differences in respect of the total time spent in the system (TTS) by all vehicles in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTS			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	5.4097×10^{-4}	1.6401×10^{-4}	2.4614×10^{-2}	4.6309×10^{-4}
CRM		—	7.4078×10^{-1}	2.0702×10^{-1}	9.6464×10^{-1}
CRM-QL			—	1.1202×10^{-1}	7.7450×10^{-1}
Q-Learning				—	1.9164×10^{-1}
Mean	1 960.01	1 768.29	1 750.33	1 836.96	1 765.89

TABLE 12.14: *Differences in respect of the total time spent in the system by vehicles entering the system from the N1 (TTSN1) in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSN1			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	9.1958×10^{-12}	1.5508×10^{-11}	9.9663×10^{-1}	9.9686×10^{-1}
CRM		—	9.3376×10^{-1}	6.1950×10^{-13}	1.5309×10^{-13}
CRM-QL			—	$< 1 \times 10^{-17}$	2.8077×10^{-13}
Q-Learning				—	9.4483×10^{-1}
Mean	884.11	606.44	614.63	895.53	873.14

TABLE 12.15: *Differences in respect of the total time spent in the system by vehicles entering the system from the R300 (TTSR300) in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TTSR300			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	6.4006×10^{-1}	1.7516×10^{-1}	1.0378×10^{-3}	5.5776×10^{-5}
CRM		—	3.7287×10^{-1}	1.9994×10^{-4}	8.3433×10^{-6}
CRM-QL			—	5.7353×10^{-6}	1.5531×10^{-7}
Q-Learning				—	4.2202×10^{-1}
Mean	992.19	1 014.18	1 056.11	835.15	797.34

As for the TTSN1, the conventional RM approaches again achieved the best performances in respect of both the mean and maximum TISN1 PMIs, as may be seen in the box plots in Figures 12.7(a) and 12.7(b). This observation is corroborated by the p-values in Tables 12.18 and

TABLE 12.16: *Differences in respect of the total time spent in the system by vehicles entering the system from Brackenfell Boulevard (TTSBB) in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

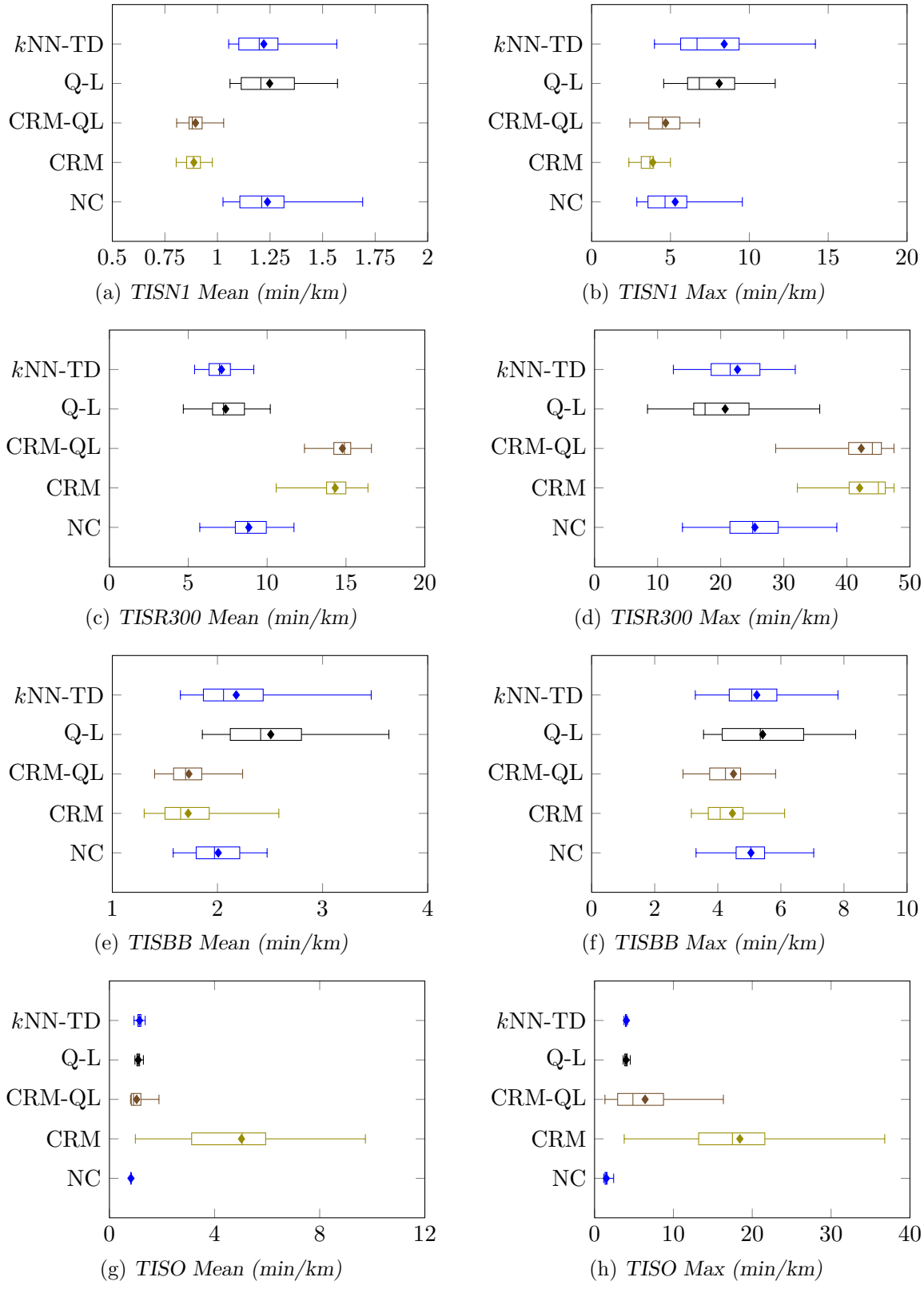
		Games-Howell test p-values: TTSBB			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	4.6529×10^{-3}	2.7231×10^{-3}	5.3264×10^{-4}	5.2126×10^{-1}
CRM		—	9.9249×10^{-1}	1.1146×10^{-7}	7.1081×10^{-4}
CRM-QL			—	1.4409×10^{-7}	7.5538×10^{-4}
Q-Learning				—	9.4689×10^{-2}
Mean	69.71	59.69	60.76	86.62	75.21

TABLE 12.17: *Differences in respect of the total time spent in the system by vehicles entering the system from Okavango Road (TTSO) in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TTSO			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	5.7257×10^{-10}	9.4946×10^{-4}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM		—	1.6378×10^{-9}	2.6015×10^{-9}	3.1365×10^{-9}
CRM-QL			—	6.7213×10^{-1}	2.9234×10^{-1}
Q-Learning				—	6.7499×10^{-1}
Mean	14.00	86.27	17.62	18.76	19.34

12.19, from which it may be deduced that CRM and CRM-QL were both able to outperform all other implementations at a 5% level of significance in respect of the mean TISN1, while outperforming both the RM by AVs implementations in respect of the maximum TISN1. Furthermore, CRM, which achieved the smallest maximum TISN1-value was also able to outperform the no-control case in respect of the maximum TISN1. The performances of both the RM by AVs implementations were found not to differ statistically from one another and the no-control case at a 5% level of significance in respect of the mean TISN1. When considering the maximum TISN1, however, the no-control case was able to outperform both the RM by AVs implementations at a 5% level of significance. These increases in the travel times of the vehicles entering the system on the N1 are also visible in the box plots in Figure 12.7(b). This increase may again be the result of unresolved congestion problems at the bottleneck at the R300 on-ramp, at which no RM is applied in both the RM by AVs implementations.

The order of relative algorithmic performances in respect of the mean and maximum TISR300 is the same as that in respect of the TTSR300. The Q-Learning for RM by AVs and KNN-TD for RM by AVs implementations achieved the smallest mean TISR300-values of 7.37 min/km and 7.11 min/km, respectively, outperforming both the conventional RM techniques and the no-control case at a 5% level of significance. From the p-values in Table 12.20 it may be deduced that the no-control case, which achieved a mean TISR300-value of 8.84 min/km, outperformed both CRM and CRM-QL at a 5% level of significance, as the latter two implementations achieved mean TISR300-values of 14.32 min/km and 14.77 min/km, respectively. The ordering of relative algorithmic performances in respect of the maximum TISR300 is the same as that in respect of the mean TISR300, except that the no-control case and kNN-TD for RM by AVs were found to perform statistically similarly at a 5% level of significance, as may be deduced from the results of the Fisher LSD test presented in Table 12.21. These trends are also clear in the box plots in Figures 12.7(c) and 12.7(d). The decreases in the travel times achieved by the vehicles joining



(a) *TISN1 Mean (min/km)* (b) *TISN1 Max (min/km)*

(c) *TISR300 Mean (min/km)* (d) *TISR300 Max (min/km)*

(e) *TISBB Mean (min/km)* (f) *TISBB Max (min/km)*

(g) *TISO Mean (min/km)* (h) *TISO Max (min/km)*

FIGURE 12.7: *Mean and maximum time spent in the system PMI results for the no-control case (NC), conventional RM (CRM), conventional RM with queue limits (CRM-QL), Q-Learning for RM by AVs and kNN-TD for RM by AVs applied to the case study model of Chapter 9.*

the N1 from the R300 when RM by AVs is employed may be attributed to the benefits that these vehicles experience when travelling past the metered Brackenfell Boulevard and Okavango Road on-ramps. The increases in the travel times of the vehicles joining the N1 from the R300 when conventional RM methods are employed may, on the other hand, be attributed to the fact that RM is employed at the R300 on-ramp in both these implementations, and, as a result, an increase in these travel times is to be expected.

CRM and CRM-QL achieved the smallest mean TISBB-values of 1.72 min/km and 1.73 min/km, respectively, outperforming the no-control case, as well as both RM by AVs implementations at a 5% level of significance. This reduction in the travel times of the vehicles entering the N1 from the Brackenfell Boulevard on-ramp may be attributed to the improved traffic flow along the N1 as a result of the RM employed at the Okavango Road on-ramp in both these implementations. Due to the fact that RM by AVs is applied at the Brackenfell Boulevard on-ramp in both the Q-Learning and kNN-TD learning implementations, it was expected that increases in the mean TISBB would be recorded for both these implementations. Although kNN-TD learning for RM by AVs achieved a larger mean TISBB-value of 2.18 in/km, compared with 2.01 min/km returned by the no-control case, the performances of these two implementations were found to be statistically indistinguishable at a 5% level of significance, as may be deduced from the p-values in Table 12.22. The no-control case was, however, able to outperform Q-Learning for RM by AVs, which achieved a mean TISBB-value of 2.51 min/km, at a 5% level of significance. This ordering of the relative algorithmic performances is also clear in the box plots in Figure 12.7(e). Perhaps surprisingly, none of the algorithmic implementations were able to outperform the no-control case at a 5% level of significance in respect of the maximum TISBB, as is evident from the p-values in Table 12.23. This similarity in the performances is also visible in the box plots in Figure 12.7(f). CRM and CRM-QL which were again able to achieve improvements in the maximum TISBB were, however, both able to outperform both Q-Learning and kNN-TD learning for RM by AVs at a 5% level of significance in respect of the maximum TISBB. The finding that, although RM by AVs is applied at the Brackenfell Boulevard on-ramp, the performances of the RM by AVs implementations did not differ statistically from that of the no-control case, may again be attributed to improved traffic flow along the N1 as a result of RM by AVs being employed at the Okavango Road on-ramp. Furthermore, the increases in travel times along the on-ramp are not as large as in conventinoal RM because RM is now applied by AVs and the vehicles never come to a stand still along the on-ramp.

The increases recorded for all four algorithmic implementations in respect of the TTSO are, as expected, also reflected in the mean and maximum TISO PMI-values. Due to the fact that RM is employed at the Okavango Road on-ramp in all four algorithmic implementations, the no-control case achieved the smallest mean and maximum TISO-values of 0.82 min/km and 1.50 min/km, respectively, outperforming all four algorithmic implementations at a 5% level of significance, as may be deduced from the p-values in Tables 12.24 and 12.25. Interestingly, in respect of the mean TISO, CRM-QL, Q-Learning for RM by AVs and kNN-TD for RM by AVs performed statistically similarly at a 5% level of significance, as these algorithms achieved values of 1.04 min/km, 1.10 min/km and 1.15 min/km, respectively. All three of these implementations were furthermore able to outperform CRM, which returned a mean TISO-value of 5.03 min/km, at a 5% level of significance. These trends are also clear in the box plots of Figure 12.7(g). When considering the maximum TISO-values, the RM by AVs implementations were both able to outperform both conventional RM implementations at a 5% level of significance, as Q-Learning and kNN-TD for RM by AVs achieved values of 4.00 min/km and 4.03 min/km, respectively. The RM by AVs implementations were followed in the order of relative algorithmic performances by CRM-QL, which achieved a maximum TISO-value of 6.41 min/km, thereby outperforming CRM at a 5% level of significance, for which a value of 18.43 min/km was recorded. These

trends are again clearly visible in the box plots in Figure 12.7(h). The improved performances of the RM by AVs implementations in respect of the TISO PMIs, when compared with the conventional RM implementations, may again be attributed to the fact that even when RM is enforced, the vehicles never stop when travelling along the on-ramp, resulting in a faster and more stable traffic flow along the on-ramp (indicated by the smaller interquartile ranges of the box plots corresponding to the RM by AVs implementations).

TABLE 12.18: *Differences in respect of the mean time spent in the system by vehicles entering the system from the N1 in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

Games-Howell test p-values: TISN1 Mean					
No Control	—	1.0259×10^{-11}	1.3704×10^{-11}	9.9901×10^{-1}	9.9225×10^{-1}
CRM		—	9.7083×10^{-1}	4.3154×10^{-13}	7.4163×10^{-14}
CRM-QL			—	2.5013×10^{-13}	8.4821×10^{-14}
Q-Learning				—	9.4659×10^{-1}
Mean	1.24	0.89	0.90	1.25	1.22

TABLE 12.19: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the N1 in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

Algorithm	No Control	Games-Howell test p-values: TISN1 Max CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	4.5598×10^{-2}	7.8569×10^{-1}	4.7633×10^{-3}	1.1860×10^{-3}
CRM		—	2.3479×10^{-1}	2.2346×10^{-6}	5.8375×10^{-3}
CRM-QL			—	1.4252×10^{-4}	3.4505×10^{-2}
Q-Learning				—	9.9927×10^{-1}
Mean	5.30	3.88	4.70	8.09	8.41

TABLE 12.20: *Differences in respect of the mean time spent in the system by vehicles entering the system from the R300 in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

Algorithm	No Control	Fisher LSD test p-values: TISR300 Mean CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	8.0459×10^{-6}	1.8817×10^{-7}
CRM		—	1.5168×10^{-1}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM-QL			—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Q-Learning				—	3.9989×10^{-1}
Mean	8.84	14.32	14.77	7.37	7.11

12.4 Discussion

Although they were outperformed in respect of the TTSN1 and TTSBB PMIs by both CRM and CRM-QL, the performances of the RM by AVs implementations were consistently at least statistically on par with those of the conventional RM implementations in respect of the TTS, TTSR300 and TTSO PMIs. The kNN-TD for RM by AVs implementation did, in fact, achieve the smallest travel times of the vehicles joining the N1 from the R300 on-ramp, while achieving the second-smallest TTS-value. Furthermore, kNN-TD for RM by AVs was found to perform

TABLE 12.21: *Differences in respect of the maximum time spent in the system by vehicles entering the system from the R300 in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Fisher LSD test p-values: TISR300 Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	5.8191×10^{-3}	1.0267×10^{-1}
CRM		—	8.8117×10^{-1}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
CRM-QL			—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Q-Learning				—	2.4923×10^{-1}
Mean	25.42	42.03	42.28	20.70	22.66

TABLE 12.22: *Differences in respect of the mean time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISBB Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	1.8497×10^{-3}	1.6289×10^{-4}	6.2482×10^{-5}	3.3239×10^{-1}
CRM		—	9.9997×10^{-1}	4.9744×10^{-9}	1.0704×10^{-4}
CRM-QL			—	3.1099×10^{-9}	3.7837×10^{-5}
Q-Learning				—	4.5316×10^{-2}
Mean	2.01	1.72	1.73	2.51	2.18

TABLE 12.23: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Brackenfell Boulevard in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISBB Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	5.8588×10^{-2}	7.5067×10^{-2}	2.3060×10^{-1}	5.4991×10^{-1}
CRM		—	9.0989×10^{-1}	2.2517×10^{-3}	1.3329×10^{-2}
CRM-QL			—	3.2104×10^{-3}	1.8027×10^{-2}
Q-Learning				—	5.4642×10^{-1}
Mean	5.05	4.46	4.50	5.42	5.24

TABLE 12.24: *Differences in respect of the mean time spent in the system by vehicles entering the system from Okavango Road in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Mean			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	2.3104×10^{-10}	8.3369×10^{-4}	4.5519×10^{-15}	3.8859×10^{-15}
CRM		—	6.7484×10^{-10}	1.0962×10^{-9}	1.4408×10^{-9}
CRM-QL			—	7.0218×10^{-1}	1.9561×10^{-1}
Q-Learning				—	1.9923×10^{-1}
Mean	0.82	5.03	1.04	1.10	1.15

statistically on par with the no-control case in respect of the TTSN1 and mean TISN1 PMIs, implying that although there was no improvement in respect of these PMIs, kNN-TD for RM by AVs was not detrimental to the performance in respect of these PMIs. Perhaps more surprising, however, is the finding that kNN-TD for RM by AVs performed statistically on par with the no-

TABLE 12.25: *Differences in respect of the maximum time spent in the system by vehicles entering the system from Okavango Road in the case of RM by AVs. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance.*

		Games-Howell test p-values: TISO Max			
Algorithm	No Control	CRM	CRM-QL	Q-Learning	kNN-TD
No Control	—	2.5918×10^{-11}	4.5555×10^{-6}	$< 1 \times 10^{-17}$	1.4796×10^{-11}
CRM		—	5.5164×10^{-8}	1.0382×10^{-9}	1.1207×10^{-9}
CRM-QL			—	2.8433×10^{-2}	3.1236×10^{-2}
Q-Learning				—	9.9472×10^{-1}
Mean	1.50	18.43	6.41	4.00	4.03

control case in respect of the TTSBB and mean and maximum TISBB PMIs, while RM by AVs was employed at the Brackenfell Boulevard on-ramp. This implies that, although the Brackenfell Boulevard was a metered on-ramp in this implementation, the increases in the travel times by the vehicles entering the system from the Brackenfell Boulevard on-ramp were not large enough to be classified as statistically different from the no-control case at a 5% level of significance. This may be due to the fact that the increases in travel times along the on-ramp when RM by AVs is employed are smaller than when conventional RM techniques are employed, as well as that the vehicles joining the N1 from the Brackenfell Boulevard on-ramp enjoy the benefit of RM by AVs being employed at the Okavango Road on-ramp. Although the increases in travel times of the vehicles joining the N1 from the Okavango Road on-ramp were statistically significantly different from those of the no-control case, kNN-TD for RM by AVs was never outperformed by any other algorithmic implementation in this regard, while outperforming CRM in respect of the TTSO, mean and maximum TISO PMIs, and outperforming CRM-QL in respect of the maximum TISO. Furthermore, the traffic flow along the Okavango Road on-ramp was significantly more stable when RM by AVs is applied when compared with conventional RM techniques, as may be deduced from the smaller interquartile ranges of the corresponding box plots in Figures 12.6(e), 12.7(g) and 12.7(h).

The performance of Q-Learning for RM by AVs was generally statistically similar to that of kNN-TD for RM by AVs, kNN-TD outperformed Q-Learning for RM by AVs only once, in respect of the mean TISBB. Although the performances of Q-Learning and kNN-TD learning for RM by AVs were generally statistically similar, kNN-TD for RM by AVs was able to achieve smaller values for eight of the thirteen PMIs. Most notable, however, is the finding that in respect of the TTSBB and mean TISBB PMIs, Q-Learning for RM by AVs was, in fact, outperformed by the no-control case, while kNN-TD for RM by AVs performed statistically on par with the no-control case in respect of these PMIs. Based on this observation, as well as the fact that kNN-TD for RM by AVs achieved a smaller TTS-value than Q-Learning for RM by AVs, kNN-TD for RM by AVs is judged to be the better performing of the RM by AVs implementations within the context of this case study.

In summary, the novel RM by AVs control measure was proven to perform statistically on par with the best-performing conventional RM techniques in the context of the case study, as the improvements in respect of the total time spent in the system by all vehicles were statistically indistinguishable for all four algorithmic implementations. Furthermore, when considering specifically the Okavango Road on-ramp, which was the only on-ramp at which RM was applied in all four implementations, one may conclude that, similarly to the findings in the context of the benchmark model of §5.1.2 in Chapter 11, RM by AVs is able to reduce the travel times along the on-ramp when compared with conventional RM techniques, while still achieving effective metering rates. Finally, the traffic flow conditions along the on-ramp are significantly more

stable when RM by AVs is employed, which may result in shorter on-ramp queue formation, limiting the possibility on an on-ramp queue propagating backwards into the arterial network feeding into the highway network.

12.5 Chapter Summary

This chapter opened in §12.1 with a detailed description of the implementations Q-Learning and kNN-TD learning for RM by AVs in the context of the case study simulation model of Chapter 9. The focus then shifted in §12.2.1 to a thorough parameter evaluation for determining the best-performing target density values as well as the best-performing combination of RM by AVs agents in the case study area. Thereafter, the influence of varying the percentage of AVs in the traffic flow has on the performance of RM by AVs was investigated in §12.2.2. This was followed in §12.3, by a thorough algorithmic performance comparison of the novel RM by AVs implementations with the best-performing conventional RM techniques, as identified in Chapter 10. A discussion highlighting some of the key findings of the algorithmic performance comparison was finally presented in §12.4.

Part IV

Conclusion

CHAPTER 13

Summary and Conclusions

Contents

This summative chapter comprises two sections. In §13.1, a chapter-by-chapter overview of the work contained in this dissertation is provided, while the novel contributions of the dissertation are highlighted in §13.2.

13.1 Dissertation Contents

Apart from Part IV, the current part, this dissertation comprises a total of twelve chapters. The last eleven of these have been partitioned into three parts. The only stand-alone chapter, Chapter 1, was devoted to providing the reader with a general background detailing the context of the work to follow in the remainder of the document. After an opening background section, in which the need for improved highway traffic control was illustrated in respect of both local and international congestion statistics, the formal problem statement considered in the dissertation was stated. Thereafter, the twelve research objectives to be pursued in the dissertation were presented. This was followed by a description of the dissertation scope which delimited the existing highway control measures considered to RM and VSLs only, while highlighting some of the other prominent highway traffic control measures which were omitted for various reasons. Once the dissertation scope had been defined, the research methodology to be followed in pursuit of the research objectives was outlined. The chapter finally closed with an explanation of how the material presented later in the dissertation is organised into the various chapters, forming cohesive dissertation parts.

The first part of the dissertation, entitled *Literature Review*, comprised three further chapters. The first of these, Chapter 2, was devoted to a comprehensive discussion on machine learning paradigms in general, with a particular focus on RL, in fulfilment of Objectives I(a) and I(b) of §1.3. The chapter opened with a brief review of machine learning in general, highlighting the various major machine learning paradigms in the literature, namely supervised learning, unsupervised learning, RL and evolutionary learning. Once these paradigms had been described, the focus shifted to a description of the working of RL, which was identified as the most suitable machine learning technique for solving highway traffic control problems. First, the four components of a typical RL problem were introduced, after which various notions related to the concept of

evaluative feedback were discussed. An expansive, formal definition of the RL problem was provided next, detailing the agent-environment interface, the notions of goals, rewards and returns within RL, as well as the underlying Markov property and the subsequent relationship of RL problems to MDPs. Thereafter, the key notion of a value function, based on the well-known Bellman equation [22], was described in detail. Some considerations related to the trade-off between exploration of the sate-action space and exploitation of what has previously been learnt were presented next. This was followed by the description of a number of algorithms which may be used to solve RL problems, including policy iteration [154], value iteration [154], Q-Learning [170], SARSA [154] and R-MART [179]. Finally, two generalisations were outlined which facilitate the application of RL to problems that have continuous state and action spaces, namely the use of weighted k nearest neighbours [94] and ANNs for value function approximation.

The second chapter of Part I, Chapter 3, contained a review of basic traffic flow theory, as well as some prominent existing highway traffic control measures, with a specific focus on control measures in which AVs were employed or machine learning algorithms were applied to solve the control problems posed by these control measures, in fulfilment of Objectives I(c), I(d) and I(e). After introducing general principles pertaining to macroscopic and microscopic traffic flow theory, the focus shifted to a review of existing highway traffic control measures. The underlying principles of RM, often seen as the most effective highway traffic control measure, were discussed at length, and this was followed by a description of some of the most successful solution methods for solving the RM problem, such as the well-known ALINEA control law [112]. Another notable solution approach is the MPC approach followed by Hegyi *et al.* [53]. Dynamic speed limits, which may be employed in order to improve the flow of vehicles already on the highway, were reviewed next. Various approaches drawing from control theory that have been employed to solve the VSL control problem were also discussed, such as that proposed by Carlson *et al.* [25], and the paradigm of MPC, as proposed by Hegyi *et al.* [53]. A description of a number of applications where AVs were employed for improving the traffic flow along a highway followed, which included the advisory algorithm of Schakel and van Arem [141] and the hierarchical MPC approach of Roncoli *et al.* [137]. The chapter culminated in a review of instances where RL techniques have been employed for solving the highway traffic control problems.

In the third and final chapter of Part I, Chapter 4, basic concepts pertaining to computer simulation modelling were highlighted, in fulfilment of Objective I(f). Some of the concepts common to all types of simulation models were introduced, and this was followed by a description of the prevailing simulation modelling paradigms, which include agent-based modelling, discrete-event modelling, system dynamics modelling and dynamic systems modelling. Thereafter, the twelve steps typically carried out during a simulation study, as suggested by Banks *et al.* [9, 10], were discussed. A discussion followed on various techniques which may be employed in the process of model verification and model validation, after which some of the advantages as well as some of the drawbacks of computer simulation modelling were highlighted. The chapter closed with a discussion on the various prevailing traffic simulation modelling paradigms, while providing examples of commercially available software environments within each of these paradigms.

The focus in the second part of this dissertation, entitled *Current Technologies*, was on implementing existing highway control measures within a microscopic traffic modelling environment and culminating in a novel implementation of a combination of highway traffic control measures. This part comprised six chapters. The first of these, Chapter 5, was devoted to a detailed explanation of the simulation modelling environment within the AnyLogic Road Traffic Library [5], as well as to a description of the benchmark simulation model developed for the present study within this simulation environment, in fulfilment of Objective II. The chapter opened with a

description of the various entities involved in the simulation model building process, culminating in a detailed description of the simple, hypothetical, benchmark highway network used later in the dissertation as a test-bed and concept demonstrator for the working of the RL algorithms. This was followed by a description of the model verification and validation techniques employed so as to ensure that the benchmark simulation model built was, in fact, a valid representation of the traffic flow on such a highway network, in fulfilment of dissertation Objective VI. The chapter finally closed with a description of the experimental design, with a specific focus on the simulation model warm-up period, as well as some of the general parameter specifications employed and the statistical analysis to be performed later in the dissertation in respect of the model output data.

The second chapter of Part II, Chapter 6, was devoted to thorough descriptions of the RM implementations within the benchmark simulation model of Chapter 5, in partial fulfilment of Objectives III and IV. The chapter opened with a description of the adjustments required to the ALINEA and PI-ALINEA control laws, which were originally designed for application in a macroscopic traffic environment, in order to render them applicable to a microscopic traffic simulation modelling environment. Thereafter, the RM problem was formulated as an RL problem. This formulation included a thorough description of the state and action spaces employed, as well as the reward function chosen in order to provide feedback to the learning agent. The implementations of the Q-Learning and kNN-TD RL algorithms for solving the RM control problem were outlined next. This was followed by a description of the parameter evaluations performed in respect of each of the four RM implementations, aimed at determining the best-performing parameter combinations for each of the implementations (measured according to the total time spent in the system by all vehicles). Once these parameter combinations had been found, the relative performances of the four RM implementations were compared statistically in the context of four scenarios of varying traffic demand within the benchmark simulation model of Chapter 5, in partial fulfilment of Objective VII. The results achieved by the four algorithms revealed that in these RM implementations, long on-ramp queues (for which RM is notorious) often build up. As a result, limitations on the allowable on-ramp queue length were imposed in all four RM implementations, and the subsequent algorithmic performances were again compared statistically. From the results of these algorithmic performance comparisons, it was found that the kNN-TD algorithm was generally the best-performing algorithm over all four scenarios of traffic demand when queue limitations are not applied, while in the case of queue limitations, the Q-Learning algorithm returned the most favourable performance.

The third chapter of Part II, Chapter 7, was dedicated to a detailed description of the implementation of Q-Learning and kNN-TD learning for solving the VSL control problem by RL for the first time within a microscopic traffic modelling paradigm, in partial fulfilment of Objectives III and IV. First, the feedback MTFC controller of Müller *et al.* [105], which was used as a benchmark against which to measure the RL implementations was introduced. This was followed by the formulation of the VSL control problem as an RL problem. This formulation included descriptions of the state and action spaces, as well as the reward function employed in order to provide feedback to the learning agent. This was followed by detailed descriptions of the Q-Learning and kNN-TD learning implementations for solving the VSL problem. Thereafter, the computational results of these implementations were presented, starting with a complete parameter evaluation in order to determine the best-performing target density and speed limit adjustment rule. A statistical comparison of the relative algorithmic performances was carried out next within the context of the four scenarios of varying traffic demand mentioned above, in partial fulfilment of Objective VII. The results of the algorithmic comparison revealed again that the kNN-TD learning algorithm achieved the best performance over all of the varying scenarios of traffic demand.

The fourth chapter of Part II, Chapter 8, was devoted to a thorough description of the MARL implementations for solving the RM and VSL problems simultaneously, adopting RL approaches for the first time, within the context of the benchmark simulation model and in final fulfilment of Objectives III and IV. The chapter opened with an introduction to the feedback controller of Carlson *et al.* [24] for integrated RM and VSLs. This was followed by a brief introduction to the vast field of MARL, containing an explanation of the notions of employing either independent or cooperative learning agents, and culminating in a detailed description of the MARLIN-ATSC approach developed by El-Tantawy *et al.* [157] for solving the traffic signal timing problem according to a decentralised MARL approach. This was followed by a detailed description of the three approaches towards solving MARL problems adopted in this dissertation, namely independent learners, hierarchical MARL and maximax MARL, with the latter two approaches invoking the "*principle of locality of interaction among agents*," defined by Nair *et al.* [106] in their approximation of global value functions. Once the algorithmic implementations pertaining to each of these MARL approaches had been described, an evaluation was carried out in order to determine the best-performing combination of reward functions which should be employed within each of these MARL implementations. Thereafter, the relative performances of the three MARL implementations were compared with one another, as well as with kNN-TD RM (the best-performing single-agent RM approach in the case without queue limits) or the feedback controller for integrated RM and VSLs (in the case where queue limits were enforced), in partial fulfilment of Objective VII. These comparisons were again performed in each of the four scenarios of varying traffic demand within the benchmark simulation model. Although the results of the algorithmic performance comparison revealed that the improvements achieved by the MARL implementations over and above those of the single RM agent were typically not large enough to be of statistical significance, it was found that the maximax MARL approach typically achieved a better trade-off in respect of the travel time reductions achieved on the highway and the travel time increases for vehicles joining the highway from the on-ramp. It was therefore decided that the maximax MARL algorithm generally yielded the most favourable results out of all the algorithms over all of the traffic scenarios simulated.

The fifth chapter of Part II, Chapter 9, was devoted to a thorough description of the simulation model built in order to represent a realistic case study area, in partial fulfilment of Objective IX. The chapter opened with a description of the area under consideration for the case study, as well as a detailed description of the simulation model developed as a test bed for the evaluation of the relative algorithmic performances within the context of this case study. This was followed by a description of the input data obtained for the purpose of this case study, detailing the sources of these data, as well as where the data-collecting sensors are located. Thereafter, the model output data were briefly described. This was followed by a detailed model validation, carried out based on real-world measurements, in order to ensure that the simulation model is a valid representation of the underlying real-world system. The chapter finally closed with a description of the experimental design employed, with a specific focus on the simulation warm-up period as well as certain general parameter specifications employed in the simulation model.

The sixth and final chapter of Part II, Chapter 10, was devoted to a detailed description of the RM, VSL and MARL algorithmic implementations in the context of the real-world case study, in partial fulfilment of Objective X. The chapter opened with a detailed description of the algorithmic implementations of the various RM agents implemented within the case study area. This description was followed by a thorough, step-wise parameter evaluation conducted for the ALINEA, PI-ALINEA, Q-Learning and kNN-TD implementations, with the aim of determining the best-performing value for the target density in each of these implementations. Once these target densities had been determined, a relative algorithmic performance comparison was carried out, in fulfilment of Objective X. This initial comparison was again followed by a comparison of

the relative algorithmic performances taking into account queue limitations at the respective on-ramps. Thereafter, a similar description followed for the VSL implementations, again initially describing the implementations, followed by a step-wise parameter evaluation in order to determine the best-performing VSL update rules, and finally culminating in a thorough algorithmic performance comparison. As was the case for the VSL implementations within the benchmark model, the kNN-TD VSL implementation again yielded the most favourable performance in respect of VSLs in the case study. A thorough description of the implementations of the MARL approaches within the case study simulation model followed. Again the section opened with a description of the algorithmic implementations, and this was followed by a reward function evaluation similar to the one performed in Chapter 8. Once the best-performing combinations of reward functions had been determined, the customary algorithmic performance comparison followed, in fulfilment of Objective X. Queue limitations were thereafter again enforced within the context of the MARL implementations, and a thorough algorithmic performance comparison again followed taking the queue limitations into account. It followed, from the relative algorithmic performance comparisons carried out in this chapter, that the hierarchical MARL approach yielded the most favourable performance within the context of this real-world case study.

In the third part of this dissertation, entitled *Future Technologies*, the focus shifted from the currently available technology, towards a future perspective of employing AVs for improving the traffic flow along highways by means of a novel highway traffic control measure, in fulfilment of Objective V. The novel concept of employing AVs to perform RM was introduced in Chapter 11. The chapter opened with a description of the basic concepts on which the novel method of RM by AVs is based. This was followed by a description of the formulation of the RM by AVs problem as an RL problem, which may be solved using RL algorithms. A thorough description of the Q-Learning and kNN-TD RL algorithms for solving the RM by AVs RL problem was provided next, before the focus shifted to an extensive parameter evaluation. The aim in this parameter evaluation was to determine the effects that various important parameters, such as the target density at the bottleneck, the length of an on-ramp, the percentage of AVs present in the traffic flow and finally the traffic demand have on the effectiveness of RM by AVs. A thorough algorithmic performance comparison followed in which the efficacy of RM by AVs was compared with those of the best-performing conventional RM implementations, taking into account the implementations with and without on-ramp queue restrictions, in fulfilment of Objective VIII. This algorithmic comparison revealed that RM by AVs typically results in even shorter on-ramp queues than conventional RM with the additional queue limits, while performing at least statistically on-par with RM with queue limits in respect of the travel times achieved by vehicles travelling along the highway only. Finally, the kNN-TD algorithm for RM by AVs was found to return the most favourable performance over all four scenarios of varying traffic demand.

The second chapter of Part II, Chapter 12, was devoted to a detailed description of RM by AVs in the context of the real-world case study, in partial fulfilment of Objective XI. The chapter opened with a detailed description of the algorithmic implementations of the various RM by AVs agents implemented within the case study area. This description was followed by a thorough, step-wise parameter evaluation conducted for the Q-Learning and kNN-TD implementations with the aim of determining the best-performing value for the target density as well as the best-performing combination of RM by AVs agents in the case study area (for each of these implementations). The target density parameter evaluation was again followed by a parameter evaluation in which the effects of varying percentages of AVs in the traffic flow were investigated. Once the best-performing target densities and AV percentages had been determined, a relative algorithmic performance comparison was carried out, comparing the performances of RM by AVs and the best-performing conventional RM implementations, in final fulfilment of Objective XI. It followed, from the relative algorithmic performance comparisons carried out in this chapter,

that kNN-TD for RM by AVs approach yielded the most favourable performance within the context of this real-world case study.

13.2 Appraisal of Dissertation Contributions

This section contains a brief summary and appraisal of the main contributions of this dissertation. Seven novel contributions are contained in this dissertation.

Contribution 1 *The development of a microscopic traffic simulation modelling framework within the AnyLogic software environment which realistically represents vehicles travelling along, joining and leaving a highway.*

The simulation modelling framework of Chapter 5 incorporates individual vehicle attributes such as vehicle length, preferred speed, maximum acceleration and maximum deceleration. The user may define the routes to be followed by the individual vehicles, as the framework supports turning and lane changing of the vehicles. Furthermore, the user may easily adjust the vehicle arrival rates, vehicle attributes and the turning probabilities. The road network topology is also easily adjustable, as the number of lanes, the lane width, the specification of left-hand or right-hand driving and the general appearance of the road network may be adjusted. Furthermore, the framework allows for the implementation of functions for controlling phase lengths at specific traffic signals or adjusting speed limits at specified locations contained in the study area. Simulation replication visualisations may be observed during model runs, and various output data may be monitored and recorded during simulation model execution.

Contribution 2 *The successful implementation of RL for solving the VSL problem within a microscopic traffic simulation environment.*

Although RL has been applied previously to the VSL problem by Li *et al.* [85], Walraven *et al.* [166] and Zhu and Ukkusuri [179], all three of these implementations were within a macroscopic traffic simulation modelling environment. In a macroscopic traffic modelling paradigm, however, it is often difficult to capture some of the important, realistic characteristics of traffic flow, such as shockwave propagation, or the spill-back effect of heavy congestion. These deficiencies were overcome by working within a microscopic traffic modelling paradigm, which is intrinsically able to capture such features, because individual vehicles are simulated as they travel along the road network. To the author's best knowledge the work presented in this dissertation is the first successful implementation of RL for solving the control problem posed by VSLs within a microscopic traffic modelling paradigm.

The results obtained from this implementation have confirmed the homogenising effect that VSLs have been claimed to exert on traffic flow, to which the improvements in traffic safety observed at various real-world installations have been attributed [23]. Furthermore, the results have shown that if VSLs are implemented correctly, the homogenisation of traffic flow may yield statistically significant improvements in terms of the travel times of the average road user.

Contribution 3 *The successful implementation of three MARL approaches towards solving the RM and VSL problems simultaneously in an online manner using RL.*

Although the RM and VSL problems have previously been solved simultaneously in an MPC context by Hegyi *et al.* [53] and in a feedback control approach by Carlson *et al.* [24], they have not yet been solved simultaneously using RL. Furthermore, the MPC approach followed by Hegyi *et al.* [53] as well as the feedback control capproach of Carlson *et al.* [24] were based on macroscopic traffic simulation models. As stated above, it may be difficult to capture some of the important features of traffic flow when adopting a macroscopic traffic simulation paradigm. Furthermore, the MPC controller involved prediction of future traffic flows, according to which an RM schedule and a VSL assignment was determined. Depending on the control interval employed, this may limit the responsiveness of the control strategy to changes in the current traffic situation. To the author's best knowledge, the work presented in this dissertation is the first example of an approach to solving the RM and VSL problems simultaneously in an online manner within a microscopic traffic modelling environment.

The results obtained from the RM, VSL and MARL implementations within the context of the benchmark simulation model of Chapter 5 were summarised in a journal article [143] which has been submitted for publication.

Contribution 4 *A demonstration of the practical working of the RL and MARL approaches to solving the RM and VSL problems within the context of a South African real-world case study.*

Adopting the modelling framework developed in Chapter 5, a valid simulation model of a real-world section of the N1 national highway outbound out of Cape Town, South Africa was built and validated using real-world data. This simulation model served as a test bed for the evaluation of the RL approaches to solving the RM and VSL problems (which were proven to be effective in the hypothetical benchmark model in Chapters 6–8) in the context of a real-world case study.

The results of this case study showed that, although the improvements achievable by the RM and VSL agents in respect of travel times in the real-world case study were not as considerable as in the simplified hypothetical benchmark model, these methods are able to improve on the current traffic flow situation without the need for capacity expansion.

A second journal article [142], on the results and findings of the case study, has been prepared and submitted for publication.

Contribution 5 *The development and successful implementation of a novel method of RM by AVs within the context of a simplified highway network.*

Conventional RM techniques are notorious for the build-up of long on-ramp queues due to stop-and-go traffic which is induced by the traffic light placed at the on-ramp tasked with performing RM. Various queue limitation strategies, such as that of Smaragdis and Papageorgiou [150], have been proposed in the literature. While often being effective in limiting the build-up of an on-ramp queue to a pre-specified value, these queue limitations often inhibit the performance of the RM controller, as well as requiring significant infrastructure expansions in order to accurately measure the on-ramp queue length.

A novel highway traffic control measure was developed in an attempt to address these shortcomings by performing RM for various percentages of AVs to which specific instructions pertaining to the speed at which they should travel along the on-ramp are issued. Due to the fact that the vehicles are not expected to come to a complete stop along the on-ramp (as in conventional RM), it was envisioned that the build-up of on-ramp queues would be addressed, while RM could still

take place as AVs travelling slowly along the on-ramp hold up the human-driven vehicles behind them.

An RL approach was adopted towards solving the novel method of RM by AVs in the context of the hypothetical benchmark simulation model, and the results of a thorough algorithmic performance comparison revealed that the novel method of RM by AVs returned favourable results under various traffic conditions. Furthermore, the novel method of RM by AVs did, in fact, address the build-up of on-ramp queues and, as a result, shortened the travel times along the on-ramp, while maintaining a more stable traffic flow.

Contribution 6 *The successful implementation of the novel method of RM by AVs of Contribution 5 in the context of a South African real-world case study.*

The novel method of RM by AVs was also implemented in a simulation model of the real world case study area described in Chapter 9 in order to assess its performance in the context of a more realistic scenario. The novel method of RM by AVs was again compared statistically with the best-performing conventional RM implementations with and without the addition of a queue limitation. The findings of this comparison showed that although the novel method of RM by AVs was unable to outperform the conventional RM implementations in respect of the total time spent in the system by all vehicles, the novel method performed at least statistically on par at a 5% level of significance with the best-performing conventional RM implementations.

A third paper, on the development and implementation of the novel method of RM by AVs in the context of both the hypothetical benchmark model and the real world case study is being prepared for submission.

Contribution 7 *The suggestion of a number of ideas for novel future work following on the contributions of this dissertation.*

The last contribution of this dissertation is proffered in the next chapter, Chapter 14. These suggestions are made in an effort to guide highway traffic control-related research in the short to medium term by documenting suitable avenues of investigation as possible follow-up work to the contributions of this dissertation.

CHAPTER 14

Suggestions for Future Work

Contents

This final chapter contains suggestions for seven avenues of further investigation as possible follow-up work on the contributions of this dissertation. In each case, the suggestion is stated formally and then elaborated upon and motivated briefly.

14.1 Scope Enlargement Suggestions

This section contains two suggestions for future work related to natural scope enlargements of the models considered in this dissertation.

Suggestion 1 *Enlarging the scope of control measures included in the simulated environment.*

The scope of existing highway traffic control measures considered in this dissertation was limited to methods for RM and VSLs only. It is suggested that this scope be enlarged to consider additional highway traffic control measures, such as dynamic lane assignments (especially when working with larger numbers of AVs), the use of variable message signs for routing suggestions or the implementation of vehicle platooning. It is expected that consideration of such additional control measures may yield further improvements of the traffic flow along a highway. It is envisioned in the case of dynamic lane assignments that these further increases may be achieved through a combination of (1) more equal lane utilisation, and (2) vehicles entering their exit lanes at more suitable points, thereby avoiding unnecessary weaving in highway traffic. In the case of using variable message signs for routing suggestions, it is envisioned that more drivers may be convinced to follow alternative routes, which may relieve pressure on the highway traffic flow, thereby ensuring that congestion due to over-utilisation of the highway does not occur.

Suggestion 2 *Implementation of an integrated control approach employing autonomous vehicles on the on-ramp and on the highway.*

Various approaches towards employing AVs travelling along the highway in order to improve the traffic flow along the highway have been proposed. Examples of such approaches are the advisory algorithm of Schakel and Van Arem [141], or the hierarchical MPC approach of Roncoli *et al.*[137]. Based on the promising results of integrating conventional RM and VSLs achieved by Carlson *et al.* [24], as well as the results of the MARL approaches adopted in this dissertation, it may be interesting to investigate an integrated approach, providing instructions to AVs travelling along both the highway and the on-ramp. It is envisioned that in such an approach, RM by AVs may be performed at the on-ramps, while the AVs travelling along the highway may receive a variety of instructions pertaining to the speed at which they should travel, the headway to the leading vehicle that should be maintained or the lane in which the AV should travel. Due to the possibly large action space of RL agents when considering the range of instructions that may be given to AVs travelling along the highway, a natural step towards such an integrated solution may be incorporating RM by AVs in the hierarchical MPC control approach of Roncoli *et al.* [137].

Suggestion 3 *Development of a feedback-based controller for RM by AVs.*

From the work performed in respect of both conventional RM and VSLs, reviewed in Chapter 3, it is evident that online controllers for RM and VSLs were initially feedback-based, as is the case in ALINEA, PI-ALINEA and the MTFC controllers by Carlson *et al.* [25] and Müller *et al.* [105]. It may therefore be a natural extension to design and implement a feedback controller for RM by AVs, the performance of which may then be measured against that achieved by the RL algorithms presented in this dissertation.

Suggestion 4 *Implementation of a highway traffic density estimation based on floating-car data gathered from individual vehicles.*

The work on highway traffic control methods conducted in this dissertation, as well as the work mentioned in the literature review on highway traffic control methods in Chapter 3, is largely based on controlling highway density, with the aim of maintaining the traffic density at bottleneck locations as close to the critical density (at which maximum traffic flow occurs). In this dissertation, the traffic densities on the respective stretches of highway were read directly from the microscopic traffic simulation models. Obtaining accurate traffic density measures is, however, not as simple in a real-world implementation. It is therefore suggested that a technique for obtaining traffic density estimates from real-world vehicles measurements, such as those developed by Bekiaris-Liberis *et al.* [14], Fountoulakis *et al.* [38], Rempe *et al.* [129] and Roncoli *et al.* [134], for example, is implemented in order to attain traffic densities as in a real-world scenario with a view to showcase the applicability and possibilities for implementation of the traffic control measures in real-world scenarios.

14.2 Solution Methodology Suggestions

This section contains a further three suggestions for future work related to solution techniques which may be employed in order to better solve the RM and VSL control problems considered in this dissertation.

Suggestion 5 *Evaluating the effectiveness of using an ANN for function approximation in conjunction with Q-Learning.*

Only one approach towards continuous value function approximation, namely using weighted k nearest neighbours, was considered in this dissertation. A natural extension of the work presented in this dissertation would, however, be to incorporate alternative function approximation methods. ANNs are often used for function approximation in conjunction with back propagation. In such a scenario, the error term E used in the training of the neural network by the backprogpagation algorithm (see Algorithm 2.7), based on the update rule in Q-Learning, is given by

$$E = r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t). \tag{14.1}$$

An expected advantage of using an ANN for value function approximation is that the resulting value function is truly continuous, whereas the value function approximation achieved when considering k nearest neighbours as in kNN-TD learning is still a piecewise linear approximation of the function value from a set of discretised centres. Thus, using an ANN for value function approximation may yield further reductions in the TTS due to a more accurate representation of the value function.

Suggestion 6 *Exploring the effectiveness of using an ANN for value function approximation which is trained by means of a population-based metaheuristic.*

An alternative approach to back propagation for training ANNs is that of employing population-based metaheuristics, such as a genetic algorithm for example. When adopting this approach, the weights of the neural network are typically the variables that the metaheuristic adjusts with the aim of achieving the highest possible prediction accuracy, measured according to some PMI. In respect of highway traffic control problems, this approach may be implemented as follows. The ANN may, in conjunction with RL, be used as a value function approximator, which may, in turn, be used to determine the best action for each state. The objective function minimised by the metaheuristic when adopting this approach would be the total time spent in the system by all the vehicles. It is expected that this approach may perform well due to the fact that the objective is to minimise the TTS directly, instead of aiming to achieve a specific target density or maximise the outflow out of a bottleneck, thereby minimising the TTS indirectly. Due to the fact, however, that a simulation run would have to be performed for each individual of the population during each generation, it is expected that this approach will be computationally very expensive.

Suggestion 7 *Exploring the effectiveness of adopting a hybrid approach towards RL in conjunction with back propagation and metaheuristics aimed at training an ANN for value function approximation.*

Due to the large expected computational expense of training an ANN purely using a metaheuristic, as detailed in Suggestion 4, the use of a novel hybrid approach is proposed. According to such an approach, the kNN-TD learning algorithm may be employed for the approximately 300 learning episodes it requires until the TTS-values begin to converge. Once this convergence sets in, the table of the centre-action pairs, together with their approximated Q-values, may be used as the training set for an ANN trained using back propagation in an offline manner. Once this training is complete, the newly trained ANN may be employed in conjunction with a population-based metaheuristic in order to evaluate whether further improvements in respect

of the TTS are possible by adjusting the weights of the ANN according to the metaheuristic algorithm, as described in Suggestion 4. It is envisioned that giving the metaheuristic a good starting solution in this manner will significantly reduce the number of generations required to obtain a good soltion, and provide a good platform from which to embark on a search for further reductions in the TTS. To the author's best knowledge, such a hybrid approach has not yet been employed.

References

[1] Alessandri A, Di Febbraro A, Ferrara A & Punta E, 1998, *Optimal control of freeways via speed signalling and ramp metering*, Control Engineering Practice, **6(6)**, pp. 771–780.

[2] Alessandri A, Di Febbraro A, Ferrara A & Punta E, 1999, *Nonlinear optimization for freeway control using variable-speed signaling*, IEEE Transactions on Vehicular Technology, **48(6)**, pp. 2042–2052.

[3] Alpaydin E, 2014, *Introduction to machine learning*, 2nd Edition, MIT Press, Cambridge (MA).

[4] Amirjamshidi G & Roorda MJ, 2017, *Multi-objective calibration of traffic microsimulation models*, Transportation Letters, **9(1)**, pp. 1–9.

[5] AnyLogic, 2017, *Multimethod simulation software*, [Online], [Cited January 2017], Available from `http://www.anylogic.com/`.

[6] Armstrong RA & Hilton AC, 2010, *One-way analysis of variance (ANOVA)*, pp. 626–652 in Curless L (Ed), *Statistical analysis in microbiology: StatNotes*, John Wiley & Sons, Hoboken (NJ).

[7] Audet C & Dennis JE, 2002, *Analysis of generalized pattern searches*, SIAM Journal on Optimization, **13(3)**, pp. 889–903.

[8] Balci O, 1997, *Verification, validation and accreditation of simulation models*, Proceedings of the 29th Conference on Winter Simulation, Atlanta (GA), pp. 135–141.

[9] Banks J, Carson JS, Nelson BL & Nicol DM, 2005, *Discrete-event system simulation*, 4th Edition, Pearson, Upper Saddle River (NJ).

[10] Banks J, 1999, *Introduction to simulation*, Proceedings of the 31st Conference on Winter Simulation, Phoenix (AZ), pp. 7–13.

[11] Baskar LD, De Schutter B & Hellendoorn H, 2012, *Traffic management for automated highway systems using model-based predictive control*, IEEE Transactions on Intelligent Transportation Systems, **13(2)**, pp. 838–847.

[12] Bates D, 2010, *Great crawl of China: Vendors cash in on 60-mile traffic jam that's lasted 11 days — with no end in sight*, [Online], [Cited August 2017], Available from `http://www.dailymail.co.uk/news/article-1306058/China-traffic-jam-enters-11th-day-officials-admit-weeks.html`.

[13] Behrisch M, Bieker L, Erdmann J & Krajzewicz D, 2011, *SUMO — Simulation of urban mobility: An overview*, Proceedings of the 3rd International Conference on Advances in System Simulation, Barcelona, pp. 55–60.

[14] Bekiaris-Liberis N, Roncoli C & Papageorgiou M, 2017, *Traffic state estimation per lane in highways with connected vehicles*, Transportation Research Procedia, **27**, pp. 921–928.

[15] Bellemans T, De Schutter B & De Moor B, 2002, *Model predictive control with repeated model fitting for ramp metering*, Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems, Singapore, pp. 236–241.

[16] Bonabeau E, 2002, *Agent-based modeling: Methods and techniques for simulating human systems*, Proceedings of the National Academy of Sciences, **99**, pp. 7280–7287.

[17] Borshchev A & Filippov A, 2004, *From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools*, Proceedings of the 22nd International Conference of the System Dynamics Society, Oxford, No page numbers.

[18] Boxill SA & Yu L, 2000, *An evaluation of traffic simulation models for supporting ITS development*, (Unpublished) Technical Report, Center for Transportation Training and Research, Houston (TX).

[19] Bunchome A, 2016, *AI system that correctly predicted last 3 US elections says Donald Trump will win*, [Online], [Cited August 2017], Available from `http://www.independent.co.uk/news/world/americas/us-elections/mogia-ai-system-that-correctly-predicted-last-3-us-elections-says-donald-trump-will-win-artificial-a7384671.html`.

[20] Burghout W, Koutsopoulos HN & Andreasson I, 2006, *A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation*, Proceedings of the 9th Intelligent Transportation Systems Conference, Toronto, pp. 1102–1107.

[21] Buşoniu L, Babuška R & De Schutter B, 2008, *A comprehensive survey of multiagent reinforcement learning*, IEEE Transactions on Systems Management and Cybernetics Part C: Applications and Reviews, **38(2)**, pp. 156–172.

[22] Busoniu L, Babuska R, De Schutter B & Ernst D, 2010, *Reinforcement learning and dynamic programming using function approximators*, CRC Press, Boca Raton (FL).

[23] Carlson RC, Papamichail I, Papageorgiou M & Messmer A, 2010, *Optimal motorway traffic flow control involving variable speed limits and ramp metering*, Transportation Science, **44(2)**, pp. 238–253.

[24] Carlson RC, Papamichail I & Papageorgiou M, 2014, *Integrated feedback ramp metering and mainstream traffic flow control on motorways using variable speed limits*, Transportation Research Part C: Emerging Technologies, **46**, pp. 209–221.

[25] Carlson RC, Papamichail I & Papageorgiou M, 2011, *Local feedback-based mainstream traffic flow control on motorways using variable speed limits*, IEEE Transactions on Intelligent Transportation Systems, **12(4)**, pp. 1261–1276.

[26] Chandler RE, Herman R & Montroll EW, 1958, *Traffic dynamics: Studies in car following*, Operations Research, **6(2)**, pp. 165–184.

[27] Chu L, Liu HX, Recker W & Zhang HM, 2004, *Performance evaluation of adaptive ramp-metering algorithms using a microscopic traffic simulation model*, Journal of Transportation Engineering, **130(3)**, pp. 330–338.

[28] Connor M, 2011, *Automobile sensors may usher in self-driving cars*, [Online], [Cited March 2016], Available from `http://www.edn.com/design/automotive/4368069/Automobile-sensors-may-usher-in-self-driving-cars`.

[29] Davarynejad M, Hegyi A, Vrancken J & van den Berg J, 2011, *Motorway ramp-metering control with queuing consideration using Q-learning*, Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems, Washington (DC), pp. 1652–1658.

[30] Davenport R, Hoelscher M, Bak J & McCormick A, 2015, *Traffic gridlock sets new records for traveler misery*, [Online], [Cited March 2016], Available from `http://mobility.tamu.edu/ums/media-information/press-release/`.

[31] Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM & Thrun S, 2017, *Dermatologist-level classification of skin cancer with deep neural networks*, Nature, **542(7639)**, pp. 115–118.

[32] Fares A & Gomaa W, 2014, *Freeway ramp-metering control based on reinforcement learning*, Proceedings of the 11th IEEE International Conference on Control and Automation, Taichung, pp. 1226–1231.

[33] Fausett L, 1994, *Fundamentals of neural networks: Architectures, algorithms, and applications*, Prentice-Hall, Englewood Cliffs (NJ).

[34] Ferrara A, Sacone S & Siri S, 2018, *An overview of traffic control schemes for freeway systems*, pp. 193–234 in Ferrara A, Sacone S & Siri S (Eds), *Freeway traffic modelling and control*, Springer International Publishing, Cham.

[35] Field A, 2016, *Contrasts and post hoc tests for one-way independent ANOVA using SPSS*, [Online], [Cited May 2017], Available from `http://www.discoveringstatistics.com/docs/contrasts.pdf`.

[36] Forbes, 2015, *10 Worst traffic jams in history*, [Online], [Cited August 2017], Available from `https://www.yahoo.com/news/10-worst-traffic-jams-in-history-235149097.html`.

[37] Forbes T & Simpson ME, 1968, *Driver-and-vehicle response in freeway deceleration waves*, Transportation Science, **2(1)**, pp. 77–104.

[38] Fountoulakis M, Bekiaris-Liberis N, Roncoli C, Papamichail I & Papageorgiou M, 2017, *Highway traffic state estimation with mixed connected and conventional vehicles: Microscopic simulation-based testing*, Transportation Research Part C: Emerging Technologies, **78**, pp. 13–33.

[39] Games PA & Howell JF, 1976, *Pairwise multiple comparison procedures with unequal n's and/or variances: A Monte Carlo study*, Journal of Educational Statistics, **1(2)**, pp. 113–125.

[40] Gazis DC, Herman R & Potts RB, 1959, *Car-following theory of steady-state traffic flow*, Operations Research, **7(4)**, pp. 499–505.

[41] Gazis DC, Herman R & Rothery RW, 1961, *Nonlinear follow-the-leader models of traffic flow*, Operations Research, **9(4)**, pp. 545–567.

[42] Ghods AH, Fu L & Rahimi-Kian A, 2010, *An efficient optimization approach to real-time coordinated and integrated freeway traffic control*, IEEE Transactions on Intelligent Transportation Systems, **11(4)**, pp. 873–884.

[43] Gomes G & Horowitz R, 2006, *Optimal freeway ramp metering using the asymmetric cell transmission model*, Transportation Research Part C: Emerging Technologies, **14(4)**, pp. 244–262.

[44] Google DeepMind, 2016, *AlphaGo — The first computer program to ever beat a professional player at the game of Go*, [Online], [Cited May 2016], Available from `https://deepmind.com/alpha-go`.

[45] Gordon R, 1996, *Algorithm for controlling spillback from ramp meters*, Transportation Research Record, **1554**, pp. 162–171.

[46] Greenshields B, 1934, *A study of traffic capacity*, Proceedings of the 14th Annual Meeting of the Highway Research Board, Washington (DC), pp. 448–477.

[47] Hall M & Willumsen L, 1980, *SATURN — A simulation-assignment model for the evaluation of traffic management schemes*, Traffic Engineering and Control, **21(4)**, pp. 168–180.

[48] Hall RW & Caliskan C, 1999, *Design and evaluation of an automated highway system with optimized lane assignment*, Transportation Research Part C: Emerging Technologies, **7(1)**, pp. 1–15.

[49] Hall RW & Lotspeich D, 1996, *Optimized lane assignment on an automated highway*, Transportation Research Part C: Emerging Technologies, **4(4)**, pp. 211–229.

[50] Haykin SS, 1994, *Neural networks: A comprehensive foundation*, Macmillan, Englewood Cliffs (NJ).

[51] Hayter AJ, 1986, *The maximum familywise error rate of Fisher's least significant difference test*, Journal of the American Statistical Association, **81(396)**, pp. 1000–1004.

[52] Hegyi A, 2004, *Model predictive control for integrating traffic control measures*, PhD thesis, Delft University of Technology, Delft.

[53] Hegyi A, De Schutter B & Hellendoorn H, 2005, *Model predictive control for optimal coordination of ramp metering and variable speed limits*, Transportation Research Part C: Emerging Technologies, **13(3)**, pp. 185–209.

[54] Helbing D, Hennecke A, Shvetsov V & Treiber M, 2001, *MASTER: Macroscopic traffic simulation based on a gas-kinetic, non-local traffic model*, Transportation Research Part B: Methodological, **35(2)**, pp. 183–211.

[55] Herman R, Montroll EW, Potts RB & Rothery RW, 1959, *Traffic dynamics: Analysis of stability in car following*, Operations Research, **7(1)**, pp. 86–106.

[56] Hidas P, 2002, *Modelling lane changing and merging in microscopic traffic simulation*, Transportation Research Part C: Emerging Technologies, **10(5)**, pp. 351–371.

[57] Hoogendoorn S & Knoop V, 2012, *Traffic flow theory and modelling*, pp. 125–159 in van Wee B, Annema J & Bannister D (Eds), *The transport system and transport policy: An introduction*, Edward Elgar Publishing Limited, Cheltenham.

[58] Hornik K, Stinchcombe M & White H, 1989, *Multilayer feedforward networks are universal approximators*, Neural Networks, **2(5)**, pp. 359–366.

[59] Howell JF & Games PA, 1974, *The effects of variance heterogeneity on simultaneous multiple-comparison procedures with equal sample size*, British Journal of Mathematical and Statistical Psychology, **27(1)**, pp. 72–81.

[60] Howell JF & Games PA, 1973, *The robustness of the analysis of variance and the Tukey WSD test under various patterns of heterogeneous variances*, Journal of Experimental Education, **41(4)**, pp. 33–37.

[61] Hueper J, Dervisoglu G, Muralidharan A, Gomes G, Horowitz R & Varaiya P, 2009, *Macroscopic modeling and simulation of freeway traffic flow*, IFAC Proceedings Volumes, **42(15)**, pp. 112–116.

[62] IBM ILOG, 2016, *IBM CPLEX Optimizer*, [Online], [Cited November 2016], Available from `https://www.ibm.com/software/commerce/optimization/cplex-optimizer/`.

[63] Ingalls RG, 2008, *Introduction to simulation*, Proceedings of the 40th Conference on Winter Simulation, Miami (FL), pp. 17–26.

[64] Institute for Transportation Systems, 2016, *SUMO — Simulation of urban mobility*, [Online], [Cited November 2016], Available from `http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/`.

[65] Jacobson LN, Henry KC & Mehyar O, 1989, *Real-time metering algorithm for centralized control*, Transportation Research Record, **1232**, pp. 17–26.

[66] Jones SL, Sullivan AJ, Cheekoti N, Anderson MD & Malave D, 2004, *Traffic simulation software comparison study*, (Unpublished) Technical Report, University Transportation Center for Alabama, Tuscaloosa (AL).

[67] Kang KP, Chang GL & Zou N, 2004, *Optimal dynamic speed-limit control for highway work zone operations*, Transportation Research Record, **1877**, pp. 77–84.

[68] Kato S, Tsugawa S, Tokuda K, Matsui T & Fujii H, 2002, *Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications*, IEEE Transactions on Intelligent Transportation Systems, **3(3)**, pp. 155–161.

[69] Kesting A, Treiber M, Schönhof M & Helbing D, 2008, *Adaptive cruise control design for active congestion avoidance*, Transportation Research Part C: Emerging Technologies, **16(6)**, pp. 668–683.

[70] Khamis MA & Gomaa W, 2012, *Enhanced multiagent multi-objective reinforcement learning for urban traffic light control*, Proceedings of the 11th International Conference on Machine Learning and Applications, Boca Raton (FL), pp. 586–591.

[71] Khamis MA & Gomaa W, 2014, *Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on a cooperative multi-agent framework*, Engineering Applications of Artificial Intelligence, **29**, pp. 134–151.

[72] Kim K, Medanić J & Cho DI, 2008, *Lane assignment problem using a genetic algorithm in the automated highway systems*, International Journal of Automotive Technology, **9(3)**, pp. 353–364.

[73] Knoop VL, Duret A, Buisson C & Van Arem B, 2010, *Lane distribution of traffic near merging zones influence of variable speed limits*, Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Madeira Island, pp. 485–490.

[74] Kotsialos A & Papageorgiou M, 2004, *Efficiency and equity properties of freeway network-wide ramp metering with AMOC*, Transportation Research Part C: Emerging Technologies, **12(6)**, pp. 401–420.

[75] Kotsialos A, Papageorgiou M, Mangeas M & Haj-Salem H, 2002, *Coordinated and integrated control of motorway networks via non-linear optimal control*, Transportation Research Part C: Emerging Technologies, **10(1)**, pp. 65–84.

[76] Kotusevski G & Hawick K, 2009, *A review of traffic simulation software*, Research Letters in the Information and Mathematical Sciences, **13**, pp. 35–54.

[77] Krajzewicz D, Hertkorn G, Rössel C & Wagner P, 2002, *SUMO (Simulation of Urban Mobility) — An open-source traffic simulation*, Proceedings of the 4th Middle East Symposium on Simulation and Modelling, Sharjah, pp. 183–187.

[78] Kuyer L, Whiteson S, Bakker B & Vlassis N, 2008, *Multiagent reinforcement learning for urban traffic control using coordination graphs*, Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Antwerp, pp. 656–671.

[79] Kwon E & Stephanedes YJ, 1994, *Comparative evaluation of adaptive and neural-network exit demand prediction for freeway control*, Transportation Research Record, **1446**, pp. 66–66.

[80] Lau R, 1997, *Ramp metering by zone — The minnesota algorithm*, (Unpublished) Technical Report, Minnesota Department of Transportation, Saint Paul (MN).

[81] Law A & Kelton WD, 2000, *Simulation modelling and analysis*, 3rd Edition, McGraw-Hill, Boston (MA).

[82] Law AM, 2008, *How to build valid and credible simulation models*, Proceedings of the 40th Conference on Winter Simulation, Miami (FL), pp. 39–47.

[83] Lenz H, 1999, *Design of nonlinear, discrete control laws to flatten inhomogeneous flow in macroscopic models*, PhD thesis, Technical University Munich, Munich.

[84] Lenz H, Sollacher R & Lang M, 2001, *Standing waves and the influence of speed limits*, Proceedings of the 2001 European Control Conference, Porto, pp. 1228–1232.

[85] Li Z, Liu P, Xu C, Duan H & Wang W, 2017, *Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks*, IEEE Transactions on Intelligent Transportation Systems, **18(11)**, pp. 3204–3217.

[86] Lighthill MJ & Whitham GB, 1955, *On kinematic waves II: A theory of traffic flow on long crowded roads*, Proceedings of the 1178th Meeting of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences, London, pp. 317–345.

[87] Lindley JA, 2004, *Traffic analysis toolbox volume III: Guidelines for applying traffic microsimulation modelling software*, (Unpublished) Technical Report TR 1956-371, U.S. Department of Transportation, Georgetown Pike (VA).

[88] Lindo Systems Inc., 2016, *Lindo*, [Online], [Cited October 2016], Available from `http://www.lindo.com/`.

[89] Litman T, 2015, *Autonomous vehicle implementation predictions: Implications for transport planning*, (Unpublished) Technical Report, Victoria Transport Policy Institute, Victoria.

[90] Logghe S, 2003, *Dynamic modeling of heterogeneous vehicular traffic*, PhD thesis, Katholieke Universiteit Leuven, Leuven.

[91] Maerivoet S & De Moor B, 2005, *Traffic flow theory*, (Unpublished) Technical Report 02.50, Katholieke Universiteit Leuven, Leuven.

[92] Mannering FL & Kilareski WP, 1990, *Elements of traffic analysis*, pp. 127–166 in Mannering FL & Kilareski WP (Eds), *Principles of highway engineering and traffic analysis*, John Wiley & Sons, New York (NY).

[93] Marsland S, 2013, *Machine learning: An algorithmic perspective*, CRC Press, Boca Raton (FL).

[94] Martín JA, de Lope J & Maravall D, 2011, *Robust high performance reinforcement learning through weighted k-nearest neighbors*, Neurocomputing, **74(8)**, pp. 1251–1259.

[95] Masher DP, Ross D, Wong P, Tuan P, Zeidler H & Petracek S, 1975, *Guidelines for design and operation of ramp control systems*, (Unpublished) Technical Report, NCHRP 3–22, SRI Project 3340, Stanford Research Institute, Menid Park (CA).

[96] May AD, 1990, *Traffic flow fundamentals*, Prentice-Hall, Englewood Cliffs (NJ).

[97] McCulloch WS & Pitts W, 1943, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics, **5(4)**, pp. 115–133.

[98] McMillin B & Sanford KL, 1998, *Automated highway systems*, IEEE Potentials, **17(4)**, pp. 7–11.

[99] Mercedes-Benz, 2016, *Assistance systems: Making life easier*, [Online], [Cited April 2016], Available from `http://www.mercedes-benz.co.za/content/south_africa/mpc/mpc_south_africa_website/en/home_mpc/passengercars/home/new_cars/models/e-class/_w212/facts_/comfort/assistancesystems.html`.

[100] Messner A & Papageorgiou M, 1990, *METANET: A macroscopic simulation program for motorway networks*, Traffic Engineering and Control, **31(8–9)**, pp. 466–470.

[101] Microsoft, 2017, *SQL Server 2016*, [Online], [Cited February 2017], Available from `https://www.microsoft.com/en-us/sql-server/sql-server-2016`.

[102] Mitchell TM, 1997, *Machine learning*, McGraw-Hill, New York (NY).

[103] Mitchell TM, 2006, *The discipline of machine learning*, Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh (PA).

[104] Montgomery DC & Runger GC, 2011, *Applied statistics and probability for engineers*, 5th Edition, John Wiley & Sons, Singapore.

[105] Müller ER, Carlson RC, Kraus W & Papageorgiou M, 2015, *Microsimulation analysis of practical aspects of traffic control with variable speed limits*, IEEE Transactions on Intelligent Transportation Systems, **16(1)**, pp. 512–523.

[106] Nair R, Varakantham P, Tambe M & Yokoo M, 2005, *Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs*, Proceedings of the Association for the Advancement of Artificial Intelligence, Pittsburgh (PA), pp. 133–139.

[107] Noland RB, 2001, *Relationships between highway capacity and induced vehicle travel*, Transportation Research Part A: Policy and Practice, **35(1)**, pp. 47–72.

[108] OpenStreetMap, 2017, *Welcome to OpenStreetMap*, [Online], [Cited February 2017], Available from `https://www.openstreetmap.org/#map=5/51.522/-0.088`.

[109] Papacostas CS & Prevedourous PD, 2001, *Transportation software*, pp. 626–652 in Curless L (Ed), *Transportation engineering and planning*, Prentice-Hall, Upper Saddle River (NJ).

[110] Papageorgiou M, Blosseville JM & Hadj-Salem H, 1990, *Modelling and real-time control of traffic flow on the southern part of Boulevard Périphérique in Paris: Part II: Coordinated on-ramp metering*, Transportation Research Part A: Policy and Practice, **24(5)**, pp. 361–370.

[111] Papageorgiou M, Blosseville J & Hadj-Salem H, 1998, *La fluidification des Rocades de l'Ile de France: Un Projet d'importance*, (Unpublished) Technical Report No 1998-17, Dynamic Systems and Simulation Laboratory, Technincal University of Crete, Chania.

[112] Papageorgiou M, Hadj-Salem H & Blosseville JM, 1991, *ALINEA: A local feedback control law for on-ramp metering*, Transportation Research Record, **1320**, pp. 2043–2067.

[113] Papageorgiou M, Hadj-Salem H & Middelham F, 1997, *ALINEA local ramp metering: Summary of field results*, Transportation Research Record, **1603**, pp. 90–98.

[114] Papageorgiou M, Kosmatopoulos E & Papamichail I, 2008, *Effects of variable speed limits on motorway traffic flow*, Transportation Research Record, **2047**, pp. 37–48.

[115] Papageorgiou M & Kotsialos A, 2000, *Freeway ramp metering: An overview*, Proceedings of the IEEE Intelligent Transportation Systems Conference, Dearborn (MI), pp. 228–239.

[116] Papageorgiou M, Papamichail I, Messmer A & Wang Y, 2010, *Traffic simulation with METANET*, pp. 399–430 in Barceló J (Ed), *Fundamentals of traffic simulation*, Springer, New York (NY).

[117] Papamichail I, Kotsialos A, Margonis I & Papageorgiou M, 2010, *Coordinated ramp metering for freeway networks: A model-predictive hierarchical control approach*, Transportation Research Part C: Emerging Technologies, **18(3)**, pp. 311–331.

[118] Papamichail I & Papageorgiou M, 2008, *Traffic-responsive linked ramp-metering control*, IEEE Transactions on Intelligent Transportation Systems, **9(1)**, pp. 111–121.

[119] Pegden CD, Sadowski RP & Shannon RE, 1995, *Introduction to simulation using SIMAN*, McGraw-Hill, New York (NY).

[120] Perraki G, Roncoli C, Papamichail I & Papageorgiou M, 2017, *Evaluation of an MPC strategy for motorway traffic comprising connected and automated vehicles*, Proceedings of the 20th International Conference on Intelligent Transportation Systems, Maui, pp. 1–7.

[121] Pipes LA, 1953, *An operational analysis of traffic dynamics*, Journal of Applied Physics, **24(3)**, pp. 274–281.

[122] Prevedouros PD & Li H, 2000, *Comparison of freeway simulation with INTEGRATION, KRONOS, and KWaves*, Proceedings of the 4th International Symposium on Highway Capacity, Maui (HI), pp. 96–107.

[123] PTV Group, 2016, *PTV Vissim*, [Online], [Cited November 2016], Available from `http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/`.

[124] Quadstone, 2016, *Paramics*, [Online], [Cited October 2016], Available from `http://www.paramics-online.com/`.

[125] Rakha H, Hellinga B, Van Aerde M & Perez W, 1996, *Systematic verification, validation and calibration of traffic simulation models*, Proceedings of the 75th Annual Meeting of the Transportation Research Board, Washington (DC), pp. 1–14.

[126] Ramaswamy D, Medanic JV, Perkins WR & Benekohal RF, 1997, *Lane assignment on automated highway systems*, IEEE Transactions on Vehicular Technology, **46(3)**, pp. 755–769.

[127] Rao B & Varaiya P, 1993, *Flow benefits of autonomous intelligent cruise control in mixed manual and automated traffic*, Transportation Research Record, **1408**, pp. 36–43.

[128] Ratrout NT & Rahman SM, 2009, *A comparative analysis of currently used microscopic and macroscopic traffic simulation software*, Arabian Journal for Science and Engineering, **34(1B)**, pp. 121–133.

[129] Rempe F, Franeck P, Fastenrath U & Bogenberger K, 2016, *Online freeway traffic estimation with real floating car data*, Proceedings of the 19th International Conference on Intelligent Transportation Systems, Rio de Janeiro, pp. 1838–1843.

[130] Rezaee K, 2014, *Decentralized coordinated optimal ramp metering using multi-agent reinforcement learning*, PhD thesis, University of Toronto, Toronto.

[131] Rezaee K, Abdulhai B & Abdelgawad H, 2012, *Application of reinforcement learning with continuous state space to ramp metering in real-world conditions*, Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage (AK), pp. 1590–1595.

[132] Rezaee K, Abdulhai B & Abdelgawad H, 2013, *Self-learning adaptive ramp metering: Analysis of design parameters on a test case in Toronto, Canada*, Transportation Research Record, **2396**, pp. 10–18.

[133] Richards PI, 1956, *Shock waves on the highway*, Operations Research, **4(1)**, pp. 42–51.

[134] Roncoli C, Bekiaris-Liberis N & Papageorgiou M, 2015, *Highway traffic state estimation using speed measurements: Case studies on NGSIM data and highway A20 in the Netherlands*, Transportation Research Record, **2559**, pp. 1–21.

[135] Roncoli C, Papageorgiou M & Papamichail I, 2015, *Traffic flow optimisation in presence of vehicle automation and communication systems — Part I: A first-order multi-lane model for motorway traffic*, Transportation Research Part C: Emerging Technologies, **57**, pp. 241–259.

[136] Roncoli C, Papageorgiou M & Papamichail I, 2015, *Traffic flow optimisation in presence of vehicle automation and communication systems — Part II: Optimal control for multi-lane motorways*, Transportation Research Part C: Emerging Technologies, **57**, pp. 260–275.

[137] Roncoli C, Papamichail I & Papageorgiou M, 2016, *Hierarchical model predictive control for multi-lane motorways in presence of vehicle automation and communication systems*, Transportation Research Part C: Emerging Technologies, **62**, pp. 117–132.

[138] SANRAL, 2009, *Gauteng Freeway Improvement Project*, [Online], [Cited August 2017], Available from `http://www.nra.co.za/live/content.php?Session_ID=ba5532d579e74187850e66750126832a&Item_ID=260`.

[139] Sargent RG, 2005, *Verification and validation of simulation models*, Proceedings of the 37th Conference on Winter Simulation, Orlando (FL), pp. 130–143.

[140] Savrasovs M, 2011, *Urban transport corridor mesoscopic simulation*, Proceedings of the European Council for Modelling and Simulation, Krakow, pp. 587–593.

[141] Schakel WJ & Van Arem B, 2014, *Improving traffic flow efficiency by in-car advice on lane, speed, and headway*, IEEE Transactions on Intelligent Transportation Systems, **15(4)**, pp. 1597–1606.

[142] Schmidt-Dumont T & van Vuuren JH, 2017, *A case for the adoption of decentralised reinforcement learning for the control of traffic flow on South African highways*, Journal of the South African Institution of Civil Engineering, **(Submitted)**.

[143] Schmidt-Dumont T & van Vuuren JH, 2017, *Decentralised reinforcement learning for ramp metering and variable speed limits on highways*, IEEE Transactions on Intelligent Transportation Systems, **(Submitted)**.

[144] Schranck D, Eisele B & Lomax T, 2012, *TTI's 2012 urban mobility report*, (Unpublished) Technical Report, Texas A&M Transportation Institute, College Station (TX).

[145] Schriber TJ, Brunner DT & Smith JS, 2013, *Inside discrete-event simulation software: How it works and why it matters*, Proceedings of the 2013 Winter Simulation Conference, Washington (DC), pp. 424–438.

[146] Schultz BB, 1985, *Levene's test for relative variation*, Systematic Biology, **34(4)**, pp. 449–456.

[147] Shannon R, 1975, *Systems simulation: The art and science*, Prentice-Hall, Englewood Cliffs (NJ).

[148] Shannon RE, 1998, *Introduction to the art and science of simulation*, Proceedings of the 30th Conference on Winter Simulation, Washington (DC), pp. 7–14.

[149] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V & Lanctot M, 2016, *Mastering the game of Go with deep neural networks and tree search*, Nature, **529(7587)**, pp. 484–489.

[150] Smaragdis E & Papageorgiou M, 2003, *Series of new local ramp metering strategies*, Transportation Research Record, **1856**, pp. 74–86.

[151] Smulders S, 1990, *Control of freeway traffic flow by variable speed signs*, Transportation Research Part B: Methodological, **24(2)**, pp. 111–132.

[152] Spall JC & Chin DC, 1994, *A model-free approach to optimal signal light timing for system-wide traffic control*, Proceedings of the 33rd IEEE Conference on Decision and Control, Lake Buena Vista (FL), pp. 1868–1875.

[153] Steinmetz K, 2016, *Smarter cars are already here*, Time, March, pp. 30–33.

[154] Sutton RS & Barto AG, 1998, *Reinforcement learning: An introduction*, MIT Press, Cambridge (MA).

[155] Szepesvári C, 2010, *Algorithms for reinforcement learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning, **4(1)**, pp. 1–103.

[156] Talebpour A & Mahmassani HS, 2016, *Influence of connected and autonomous vehicles on traffic flow stability and throughput*, Transportation Research Part C: Emerging Technologies, **71**, pp. 143–163.

[157] El-Tantawy S, Abdulhai B & Abdelgawad H, 2013, *Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto*, IEEE Transactions on Intelligent Transportation Systems, **14(3)**, pp. 1140–1150.

[158] Tesla Motors, 2016, *Model S software version 7.0*, [Online], [Cited March 2016], Available from `https://www.teslamotors.com/presskit/autopilot`.

[159] Tokic M & Palm G, 2011, *Value-difference based exploration: Adaptive control between epsilon-greedy and softmax*, Proceedings of the 34th Annual German Conference on AI, Berlin, pp. 335–346.

[160] TomTom, 2017, *The TomTom Traffic Index*, [Online], [Cited August 2017], Available from `https://www.tomtom.com/en_gb/trafficindex/list?citySize=LARGE&continent=ALL&country=ALL`.

[161] Transportation Research Board, 2000, *Highway capacity manual*, Washington (DC).

[162] Van Aerde M, Hellinga B, Baker M & Rakha H, 1996, *INTEGRATION: An overview of traffic simulation features*, Transportation Research Record, **1464**, pp. 122–130.

[163] Van Arem B, Van Driel CJ & Visser R, 2006, *The impact of cooperative adaptive cruise control on traffic-flow characteristics*, IEEE Transactions on Intelligent Transportation Systems, **7(4)**, pp. 429–436.

[164] Varaiya P, 1993, *Smart cars on smart roads: Problems of control*, IEEE Transactions on Automatic Control, **38(2)**, pp. 195–207.

[165] Vella M, 2016, *Why you shouldn't be be allowed to drive*, Time, March, pp. 24–29.

[166] Walraven E, Spaan MT & Bakker B, 2016, *Traffic flow optimization: A reinforcement learning approach*, Engineering Applications of Artificial Intelligence, **52**, pp. 203–212.

[167] Wang Y, Kosmatopoulos EB, Papageorgiou M & Papamichail I, 2014, *Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study*, IEEE Transactions on Intelligent Transportation Systems, **15(5)**, pp. 2024–2039.

[168] Wardrop J, 1952, *Some theoretical aspects of road traffic research*, Proceedings of the Conference of the Institute of Civil Engineers, London, pp. 325–378.

[169] Watkins CJCH & Dayan P, 1992, *Q-learning*, Machine Learning, **8(3–4)**, pp. 279–292.

[170] Watkins CJCH, 1989, *Learning from delayed rewards*, PhD thesis, University of Cambridge, Cambridge.

[171] Wattleworth JA, 1967, *Peak period analysis and control of a freeway system with discussion*, Highway Research Record, **157**, pp. 1–21.

[172] Wavetronix LLC, 2017, *SmartSensor HD: Arterial and freeway*, [Online], [Cited June 2017], Available from `https://www.wavetronix.com/en/products/3-smartsensor-hd`.

[173] Wen K, Qu S & Zhang Y, 2009, *A machine learning method for dynamic traffic control and guidance on freeway networks*, Proceedings of the IEEE International Asia Conference on Informatics in Control, Automation and Robotics, Wuhan, pp. 67–71.

[174] Williams LJ & Abdi H, 2010, *Fisher's least significant difference (LSD) test*, Encyclopedia of Research Design, pp. 1–6.

[175] Winston WL, 2004, *Operations research: Applications and algorithms*, 4th Edition, Brooks/Cole, Belmont (CA).

[176] World Health Organisation, 2015, *Global status report on road safety 2015*, Technical Report ISBN 9789241565066, World Health Organisation, Geneva.

[177] Yang Q & Koutsopoulos HN, 1996, *A microscopic traffic simulator for evaluation of dynamic traffic management systems*, Transportation Research Part C: Emerging Technologies, **4(3)**, pp. 113–129.

[178] Zhang HM & Ritchie SG, 1997, *Freeway ramp metering using artificial neural networks*, Transportation Research Part C: Emerging Technologies, **5(5)**, pp. 273–286.

[179] Zhu F & Ukkusuri SV, 2014, *Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach*, Transportation Research Part C: Emerging Technologies, **41**, pp. 30–47.