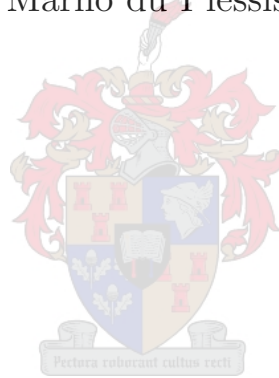


# Reinforcement learning for inventory management in information-sharing pharmaceutical supply chains

by

Marno du Plessis



Thesis presented in fulfilment of the requirements for the degree of  
**Master of Engineering (Industrial Engineering)**  
in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof JH van Vuuren  
Co-supervisor: Dr J van Eeden

March 2020

# Declaration

By submitting this thesis 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: March 2020

---

# Abstract

A general lack of information sharing across the various tiers of pharmaceutical supply chains in developing countries continues to compromise the availability of essential medicines. In the South African public health-care context, recent efforts aimed at improving information sharing in the pharmaceutical supply chain have been plagued by several implementation problems. It is conjectured that the true potential impact of information sharing remains unclear in the South African public health-care domain. The objective in this thesis is to elucidate conceptually how information sharing may benefit inventory management in a pharmaceutical supply chain.

A number of hypothetical information-sharing scenarios are proposed in this thesis and their relative effectiveness is evaluated within a simulation modelling environment. The first of these scenarios does not involve any information sharing and serves as a benchmark. The scope of information sharing is further increased incrementally over the remaining scenarios. An agent-based pharmaceutical supply chain simulation model is further established in this thesis in order to evaluate the impact of information sharing in the context of the information-sharing scenarios. This simulation model is implemented as a concept demonstrator and takes as input any user-specified supply chain network. The concept demonstrator is capable of modelling the high-level operation of a pharmaceutical supply chain over time, with a particular focus on the flow of inventory.

A reinforcement learning approach is adopted towards discovering effective inventory replenishment policies, specifically informed by information sharing, during each of the aforementioned information-sharing scenarios. The effectiveness of these policies is measured in respect of the total number of stock-outs and product expiries observed in the supply chain. A comparative analysis of the information-sharing scenarios is performed in the context of a hypothetical supply chain network experiencing a fluctuating demand pattern. This analysis reveals that stock-outs may be mitigated substantially when allowing health-care facilities that are located in close proximity to one another to share inventory among each other. It is also shown that the types of, and granularity of, information shared are instrumental in determining the relative effectiveness of information sharing.



---

# Opsomming

'n Algemene gebrek aan die deel van inligting oor verskillende vlakke van farmaseutiese voorsieningskettings in ontwikkelende lande belemmer die beskikbaarheid van noodsaaklike medisyne. In die Suid-Afrikaanse openbare gesondheidsorg-konteks is onlangse pogings om die deel van inligting in die farmaseutiese voorsieningsketting te verbeter, geteister deur verskeie implementeringsprobleme. Daar word vermoed dat die werklike potensiële impak van inligting-deling in Suid-Afrikaanse openbare gesondheidsorg onduidelik is. Die doel van hierdie tesis is om konseptueel toe te lig hoe die deel van inligting voorraadbestuur in 'n farmaseutiese voorsieningsketting kan beoordeel.

'n Aantal hipotetiese scenario's word vir die deel van inligting in hierdie tesis voorgestel en die relatiewe doeltreffendheid daarvan word in 'n simulasiemodelleringsomgewing beoordeel. Die eerste van hierdie scenario's behels geen inligting-deling nie en dien as maatstaf. Die omvang van inligting-deling word in die daaropvolgende scenario's inkrementeel verhoog. 'n Agent-gebaseerde simulasiemodel vir farmaseutiese voorsieningskettings word verder in hierdie tesis daargestel om die impak van inligting-deling in die konteks van die bostaande scenario's te evalueer. Hierdie simulasiemodel word as 'n konsepdemonstrasie geïmplementeer en neem 'n gebruikersgespesifiseerde voorsieningskettingnetwerk as toevoer. Die konsepdemonstrasie-model is daartoe in staat om die hoëvlak-werking van 'n farmaseutiese voorsieningsketting oor tyd te modelleer, met 'n spesifieke fokus op die vloeï van voorraad.

'n Versterkingsleerbenadering word gevolg om vir elk van die bogenoemde scenario's doeltreffende voorraadaanvullingsbeleide te ontdek deur spesifiek gebruik te maak van inligting-deling. Die doeltreffendheid van hierdie beleide word gemeet in terme van die totale aantal voorraadtekorte en produkverstrykings wat in die voorsieningsketting waargeneem word. 'n Vergelykende studie van die inligting-delingscenario's word in die konteks van 'n hipotetiese voorsieningskettingnetwerk met 'n wisselende vraagpatroon uitgevoer. Uit hierdie ontleding volg dit dat voorraadtekorte aansienlik verlaag kan word indien nabygeleë gesondheidsorgfasiliteite voorraad onder mekaar kan uitruil. Daar word ook aangetoon dat die tipes inligting en die mate van inligting-deling instrumenteel is in die bepaling van die relatiewe doeltreffendheid van die deel van inligting.



# Acknowledgements

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

- My supervisor, Professor JH van Vuuren, for his guidance and continual support, and the time and effort he invested in developing me as a researcher as well as a person. I feel privileged to have been under his tutelage over the past three years.
- My co-supervisor, Dr J van Eeden, for his input and willingness to share his wealth of knowledge involving the fields of supply chain management and logistics.
- The Industrial Engineering Department at Stellenbosch University as well as the *Stellenbosch Unit for Operations Research in Engineering* (SUnORE), for the opportunity to use the office space and computational facilities.
- The Bill and Melinda Gates Foundation for the financial support received over the past two years.





---

# Table of Contents

<b>Abstract</b>	<b>iii</b>
<b>Opsomming</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>List of Acronyms</b>	<b>xv</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xix</b>
<b>List of Algorithms</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem description . . . . .	6
1.3 Research objectives . . . . .	7
1.4 Scope delimitation . . . . .	8
1.5 Research methodology . . . . .	9
1.6 Thesis organisation . . . . .	10
<b>I Literature review</b>	<b>11</b>
<b>2 Supply chain management</b>	<b>13</b>
2.1 An introduction to supply chain management . . . . .	14
2.2 Supply chain management strategies . . . . .	15
2.3 The bullwhip effect . . . . .	19
2.3.1 Causes of the bullwhip effect . . . . .	19
2.3.2 Preventing the bullwhip effect . . . . .	20

2.4	Information sharing in supply chains . . . . .	21
2.4.1	Types of shared information . . . . .	22
2.4.2	Barriers to information sharing . . . . .	24
2.4.3	Previous supply chain information sharing studies . . . . .	24
2.5	Demand-driven supply chain management . . . . .	26
2.6	Supply chain collaboration . . . . .	28
2.7	Inventory management . . . . .	29
2.8	Measuring supply chain performance . . . . .	34
2.9	Pharmaceutical supply chains . . . . .	36
2.9.1	Global challenges in pharmaceutical supply chains . . . . .	36
2.9.2	Inventory management in pharmaceutical supply chains . . . . .	38
2.9.3	A perspective on the South African pharmaceutical supply chain . . . . .	39
2.10	Chapter summary . . . . .	41
<b>3</b>	<b>Computer simulation and agent-based modelling</b>	<b>43</b>
3.1	An introduction to computer simulation modelling . . . . .	44
3.2	Basic simulation modelling concepts . . . . .	45
3.3	Prevailing simulation modelling paradigms . . . . .	46
3.3.1	Discrete-event modelling . . . . .	46
3.3.2	System dynamics modelling . . . . .	47
3.3.3	Agent-based modelling . . . . .	47
3.3.4	Dynamic systems modelling . . . . .	47
3.4	Typical steps in a sound simulation study . . . . .	47
3.5	Simulation input modelling . . . . .	49
3.6	Verification and validation of a simulation model . . . . .	50
3.6.1	Model verification . . . . .	50
3.6.2	Model validation . . . . .	51
3.7	Developing an agent-based model . . . . .	52
3.7.1	Definition of an agent . . . . .	52
3.7.2	Designing an agent-based model . . . . .	53
3.7.3	Agent-based modelling of supply chains . . . . .	54
3.8	Chapter summary . . . . .	56
<b>4</b>	<b>Reinforcement learning</b>	<b>57</b>
4.1	An introduction to machine learning . . . . .	57
4.2	Reinforcement learning . . . . .	58

Table of Contents	xi
4.2.1 Evaluative feedback . . . . .	60
4.2.2 Formulation of the reinforcement learning problem . . . . .	62
4.2.3 Reinforcement learning solution approaches . . . . .	67
4.3 Chapter summary . . . . .	70
<b>II Pharmaceutical supply chain modelling</b>	<b>71</b>
<b>5 Information sharing in a pharmaceutical supply chain</b>	<b>73</b>
5.1 Investigating the impact of information sharing . . . . .	73
5.2 Five information-sharing scenarios . . . . .	74
5.2.1 Scenario 1: No information sharing . . . . .	74
5.2.2 Scenario 2: Intra-neighbourhood information sharing between clinics . . .	75
5.2.3 Scenario 3: Limited inter-tier information sharing . . . . .	77
5.2.4 Scenario 4: Information sharing between warehouses . . . . .	78
5.2.5 Scenario 5: Extended inter-tier information sharing . . . . .	79
5.3 Chapter summary . . . . .	80
<b>6 An agent-based pharmaceutical supply chain simulation model</b>	<b>81</b>
6.1 Model framework . . . . .	81
6.1.1 Model input . . . . .	84
6.1.2 Inventory replenishment orders . . . . .	89
6.1.3 The prescriptive paradigm . . . . .	90
6.1.4 The graphical user interface . . . . .	91
6.1.5 Model output . . . . .	92
6.2 Model verification and validation . . . . .	93
6.2.1 Verification of the simulation model . . . . .	93
6.2.2 Validation of the simulation model . . . . .	94
6.3 Chapter summary . . . . .	95
<b>7 Reinforcement learning in a pharmaceutical supply chain</b>	<b>97</b>
7.1 The state space . . . . .	97
7.2 The action space . . . . .	102
7.3 The reward function . . . . .	103
7.4 Learning rate . . . . .	106
7.5 Exploration rate and action selection . . . . .	106
7.6 Chapter summary . . . . .	109

---

<b>8</b>	<b>Experimental design</b>	<b>111</b>
8.1	Experimental design overview . . . . .	111
8.2	The pharmaceutical supply chain network . . . . .	113
8.2.1	Facilities . . . . .	114
8.2.2	Connections . . . . .	114
8.2.3	Neighbourhoods . . . . .	115
8.2.4	Inventory . . . . .	116
8.2.5	Demand . . . . .	117
8.3	Q-learning algorithmic implementation . . . . .	118
8.3.1	The state space . . . . .	118
8.3.2	The action space . . . . .	123
8.3.3	The reward function . . . . .	124
8.3.4	Learning rate and action selection . . . . .	125
8.4	Experimental procedure . . . . .	125
8.5	Statistical analysis . . . . .	126
8.6	Chapter summary . . . . .	129
<b>9</b>	<b>Results</b>	<b>131</b>
9.1	Analysis of results . . . . .	131
9.2	Statistical analysis of the impact of fluctuating demand . . . . .	132
9.2.1	Experiment 1 . . . . .	132
9.2.2	Experiment 2 . . . . .	138
9.2.3	Experiment 3 . . . . .	143
9.2.4	Experiment 4 . . . . .	148
9.2.5	Experiment 5 . . . . .	150
9.2.6	Synopsis of the relative effectiveness of information sharing . . . . .	152
9.3	Chapter summary . . . . .	155
<b>III</b>	<b>Conclusion</b>	<b>157</b>
<b>10</b>	<b>Conclusion and summary</b>	<b>159</b>
10.1	Thesis summary . . . . .	159
10.2	Appraisal of contributions . . . . .	161
<b>11</b>	<b>Suggestions for future work</b>	<b>163</b>
11.1	Suggestions involving the simulation model . . . . .	163

Table of Contents	xiii
11.2 Scope enlargement of information sharing . . . . .	164
11.3 Solution approach suggestions . . . . .	165
<b>References</b>	<b>167</b>



## List of Acronyms

**ANOVA:** Analysis of variance

**ARV:** Antiretroviral

**CMS:** Central medical store

**CPFR:** Collaborative planning, forecasting and replenishment

**DCM:** Demand chain management

**DDSCM:** Demand-driven supply chain management

**EDLP:** Everyday low price

**EOQ:** Economic order quantity

**GUI:** Graphical user interface

**IT:** Information technology

**KPI:** Key performance indicator

**LSD:** Least significant difference

**MDP:** Markov decision process

**SSP:** Stop Stock-outs Project

**SVS:** Stock Visibility Solution

**TB:** Tuberculosis

**VMI:** Vendor-managed inventory

**WHO:** World Health Organisation





---

## List of Figures

1.1	An illustrative schematic representation of a simple pharmaceutical supply chain	2
1.2	A living bridge emerges from the self-organising behaviour of ants . . . . .	6
2.1	A cost-responsiveness efficient frontier . . . . .	17
2.2	A schematic representation of a generic push-pull-based supply chain . . . . .	18
2.3	Inventory level as a function of time according to the basic EOQ model . . . . .	31
2.4	Inventory level as a function of time according to a stochastic inventory model .	33
4.1	The agent-environment interaction in reinforcement learning . . . . .	62
5.1	No information sharing between the facilities in a pharmaceutical supply chain .	75
5.2	Information sharing between clinics in the same neighbourhood . . . . .	76
5.3	Clinics share information in their neighbourhoods and with their direct suppliers	77
5.4	Intra-neighbourhood information sharing between warehouses and clinics . . . . .	79
5.5	Extended inter-tier information sharing . . . . .	80
6.1	A screenshot of the animation portion of the simulation model GUI . . . . .	92
8.1	The layout of the experimental pharmaceutical supply chain network . . . . .	116
9.1	The learning progression of the Q-learning algorithm during Scenario 1 . . . . .	133
9.2	The number of end-user stock-outs observed during Experiment 1 . . . . .	133
9.3	The inventory level of Clinic E during low demand in Scenario 1 . . . . .	135
9.4	The inventory level of Clinic A during low demand in Scenario 1 . . . . .	135
9.5	The amount of inventory in Neighbourhood 1 during Experiment 1 . . . . .	136
9.6	The inventory level of the hospital during a replication run of Experiment 1 . . .	137
9.7	The inventory level of the warehouse during a replication run of Experiment 1 . .	138
9.8	The inventory level of the manufacturer during a replication run of Experiment 1	139
9.9	Unfulfilled demand observed at each health-care facility during Experiment 2 . .	140
9.10	The number of end-user stock-outs observed during Scenarios 1 and 2 . . . . .	141

---

9.11	The inventory level of Neighbourhood 1 during Experiment 2 . . . . .	143
9.12	The mean daily amount of inventory held by the hospital and the warehouse . .	144
9.13	The total amount of inventory in Neighbourhood 2 during Experiment 3 . . . . .	145
9.14	Unfulfilled demand observed at each health-care facility during Experiment 3 . .	146
9.15	The impact of a moving average sample window on the number of stock-outs . .	147
9.16	The inventory level of the hospital during three replication runs of Experiment 4	149
9.17	The mean daily amount of inventory held by the manufacturer . . . . .	151
9.18	The inventory level of the manufacturer during Experiments 4 and 5 . . . . .	151
9.19	Unfulfilled demand observed during each scenario . . . . .	153

---

## List of Tables

6.1	The structure of the <code>table_facilities</code> database table . . . . .	84
6.2	The structure of the <code>table_products</code> database table . . . . .	85
6.3	The structure of the <code>table_connections</code> database table . . . . .	86
6.4	The structure of the <code>table_inventory</code> database table . . . . .	87
6.5	The structure of the <code>table_manufacturers</code> database table . . . . .	87
6.6	The structure of the <code>table_starting_inventory</code> database table . . . . .	87
6.7	The structure of the <code>table_demand</code> database table . . . . .	89
6.8	The structure of the <code>table_neighbourhoods</code> database table . . . . .	89
6.9	The structure of the <code>table_events</code> database table . . . . .	89
7.1	The state space design of each agent according to Scenario 1 . . . . .	99
7.2	The state space design of each agent according to Scenario 2 . . . . .	100
7.3	The state space design of each agent according to Scenario 3 . . . . .	101
7.4	The state space design of each agent according to Scenario 4 . . . . .	102
7.5	The state space design of each agent according to Scenario 5 . . . . .	103
8.1	The characteristics of each <b>Facility</b> agent . . . . .	114
8.2	The primary supplier-customer connections in the experimental network . . . . .	115
8.3	The delivery lead times between supplier-customer pairs in Neighbourhood 3 . . . . .	116
8.4	The triangular distribution parameter values of the two demand classes . . . . .	117
8.5	A summary of the end-user demand conditions considered in all experiments . . . . .	118
8.6	The discretisation of the manufacturer agent's state space . . . . .	120
8.7	The discretisation of the warehouse agent's state space . . . . .	121
8.8	The discretisation of the hospital agent's state space . . . . .	122
8.9	The discretisation of the clinic agent's state space . . . . .	123
8.10	The size of each agent's state space according to the five scenarios . . . . .	123
8.11	The starting inventory levels and corresponding remaining shelf-lives . . . . .	126

9.1	Differences in the number of stock-outs observed during Scenario 2 . . . . .	140
9.2	A portion of the policy learnt by Clinic A during the training run of Scenario 2 .	141
9.3	A selection of actions taken by the clinics in Neighbourhood 2 . . . . .	142
9.4	The states perceived and actions taken by the warehouse during Experiment 3 .	146
9.5	Differences in respect of the total number of end-user stock-outs . . . . .	154

---

## List of Algorithms

4.1	The policy iteration algorithm . . . . .	68
4.2	The value iteration algorithm . . . . .	69
4.3	The Q-learning algorithm . . . . .	70



---



---

## CHAPTER 1

---

# Introduction

### Contents

1.1 Background . . . . .	1
1.2 Problem description . . . . .	6
1.3 Research objectives . . . . .	7
1.4 Scope delimitation . . . . .	8
1.5 Research methodology . . . . .	9
1.6 Thesis organisation . . . . .	10

## 1.1 Background

Perennial stock-outs and shortages of critical medicines are commonplace in developing countries and continue to compromise the quality of health care services. Developing nations furthermore carry a considerable burden of life-threatening diseases and the treatment of such diseases is significantly complicated by medicine stock-outs and shortages. The sobering truth, however, is that stock-outs are preventable, but overcoming these deficiencies and their damaging consequences demands major improvements in the management of pharmaceutical supply chains of developing countries.

The statistics paint a concerning picture of the magnitude of the medicine stock-out crisis. The Global AIDS Response Progress Reporting programme, for example, reported that 38 of 108 low- and middle-income countries experienced stock-outs of *antiretroviral* (ARV) medicines in 2013 [164]. In a different study conducted amongst 1 200 clinics in 30 countries worldwide, the *World Health Organisation* (WHO) [165] disclosed that Africa was the region with the largest incidence of ARV medicine stock-outs during the period 2005–2013. In South Africa, a survey conducted in 2015 by the *Stop Stock-outs Project* (SSP) consortium [44] revealed that approximately one in four health-care facilities suffered from stock-outs of either ARV or *tuberculosis* (TB) medicines during the three-month period preceding the survey. A staggering 70% of these stock-outs lasted longer than one month, highlighting the supply chain’s inability to remedy the root causes of stock-outs rapidly.

The ramifications of medicine stock-outs are far-reaching and are the most severe on the subsequently untreated patients. Treatment interruptions or failure to start treatment as a direct result of stock-outs may lead to increased drug resistance, aggravation of disease, transmission of disease (in the case of communicable diseases) or even death [44, 63, 118]. The impact of

stock-outs is particularly harsh on impoverished communities in rural areas which depend solely on public health care services. To keep up with their prescriptions, these indigent patients are forced to make frequent and costly trips to their local health-care facilities. If they are confronted with stock-outs at these facilities, they are turned away and forced to visit even farther facilities, with no guarantee of stock availability at these facilities either [70].

A *supply chain* is the construct supporting and executing the delivery of a product or a service to a consumer. There are many definitions of the notion of a supply chain in the literature. Mentzer *et al.* [112], for example, describe a *supply chain* as a set constituting at least three entities directly involved in both the upstream and downstream flow of goods, services, money and information from a source to a customer. A *pharmaceutical supply chain*, by implication, is a network of organisations involved in the delivery of pharmaceuticals (medicines or drugs) from various sources to patients. A schematic of a simple pharmaceutical supply chain — delivering a single drug — is shown in Figure 1.1. The flow of materials in the pharmaceutical supply chain is initiated by a number of suppliers that deliver raw materials to the manufacturer of the drug. Thereafter, the product is delivered to an intermediate storage facility (warehouse) which, in turn, distributes the drug to clinics. Finally, the product concludes its passage through the supply chain when it is consumed by the end user (patient) who acquires the drug at a clinic.

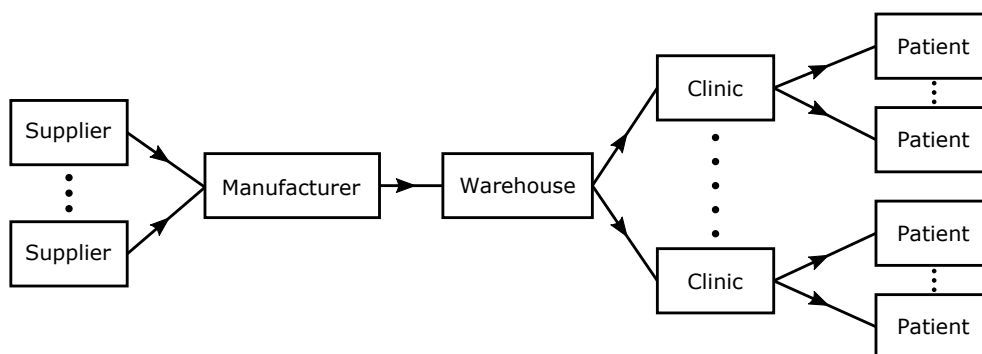


FIGURE 1.1: An illustrative schematic representation of a simple pharmaceutical supply chain.

Traditional definitions of supply chains, however, all emphasise the activities responsible for the downstream movement of products in a supply chain network *en route* to consumers [89]. Organisations will, for example, concentrate on streamlining production processes and distribution practices in order to improve the efficiency with which goods are moved downstream from one facility to another in a supply chain. Not disregarding the importance of these activities, proponents of the so-called *demand chain management* (DCM) notion suggest that the principal focus of such an approach is misplaced. DCM is a relatively new concept that champions the philosophy that the consumer should drive the upstream processes (such as manufacturing and distribution) in a supply chain [89]. In contrast to traditional supply chains, the consumer is considered as the starting point in a supply chain and not as the final destination. Although products flow through a supply chain toward the end user, it is indeed the particular demands of the end user that largely dictate the upstream activities. Consider the hypothetical situation where the demand for a particular drug increases sharply as a result of an unexpected disease outbreak. If suppliers, manufacturers and distributors fail to increase and accelerate their operations accordingly, patients may fail to receive proper treatment. Manufacturers, on the other hand, cannot overcompensate for stock-outs through overproducing, because superfluous stock reflect unnecessary expenditures and expose the drugs to the risk of expiration.



Fisher [49] posited that any supply chain performs two distinct functions. The first is the *physical function* which reflects the transformation of raw materials to finished products, and the transportation of these goods along a supply chain. The performance of the physical function is a determinant of a supply chain's efficiency. Manufacturing, delivery and inventory storage outlays include expenditures associated with the physical function and these are classified as physical costs. The second function is the *market mediation* function and its purpose is to ensure that customer demand is satisfied. Market mediation costs are incurred when supply exceeds demand or *vice versa*. In the case of oversupply, superfluous stocks may be sold at a loss or even discarded. Undersupply of stock, on the other hand, reflects lost sales opportunities. Evidently, neither a surplus nor a shortage of stock is desirable in a supply chain.

Through a juxtaposition of the physical function and the market mediation function, Fisher suggested that organisations may emphasise one function at the expense of the other. An organisation subject to predictable demand can, however, take deliberate measures to avoid both under- and overproduction. As a result, the firm can devote its attention to supply chain efficiency and the restriction of physical costs. Firms faced with unpredictable demand, on the other hand, prioritise market mediation costs over physical costs, because they place a premium on customer satisfaction as opposed to supply chain efficiency. In other words, the desire to satisfy customer demand predominates the level of efficiency utilised to meet demand.

In view of Fisher's perspective, De Treville *et al.* [39] defined a *demand chain* as a supply chain that accentuates market mediation to a greater degree than its function to optimise physical efficiency (the physical function) of a supply chain. Adopting this view of a demand chain in a pharmaceutical environment would seem a natural fit. Considering the uncompromising risks posed by medicine stock-outs, it may be argued that market mediation should enjoy a stronger preference than the physical efficiency of a supply chain. To build the case for a *pharmaceutical demand chain* is, however, not as straightforward as it would seem at first glance and poses an intriguing conundrum for the various entities in a pharmaceutical supply chain.

Consider, for example, a health-care facility dispensing medicines to a large population of patients. Whereas this facility may pursue a customer service level of 100% in a bid to avoid stock-outs, a drug manufacturer upstream in the supply chain may have a different, conflicting objective. The manufacturer may exclusively pursue profit maximisation and have little regard for the service level attained by the downstream health-care facility. In other words, the manufacturer may act solely in its own best interests, even if this would compromise the supply chain's ability to successfully satisfy customer (patient) demand. This phenomenon suggests that the transformation from a traditional pharmaceutical supply chain to a pharmaceutical demand chain (*i.e.* placing a premium on market mediation) may be met with resistance from stakeholders with contrasting objectives.

Causal factors of medicine stock-outs in developing countries are numerous and tend to vary across different pharmaceutical supply chains [81]. The predominant sources of pharmaceutical supply chain under-performance in public health sectors include fragmented accountability amongst stakeholders [166], superfluous complexity [75, 166], funding complexities and inadequacies [24, 81, 166], insufficient inventory management in the face of information shortages [11, 81] and incompetent distribution systems [15, 24]. A lack of data capturing and data sharing, however, is recognised as one of the largest stumbling blocks toward pharmaceutical supply chain improvement in developing nations [166]. In South Africa, for example, stock levels at medicine depots could be aggregated to a national level, but the absence of inventory data at health-care facilities render the supply chain convoluted and difficult to manage [11]. Developing countries may not have access to the resources required to readily adopt information systems in their pharmaceutical supply chains, but the irrefutable benefits of information sharing cannot

be ignored. Sharing supply chain information, such as demand forecasts and inventory levels, across an entire network may enable supply chains to better balance supply and demand, drive greater accountability and enhance overall supply chain performance at a lower cost [53, 123, 166]. End-to-end information visibility empowers supply chain entities to plan for events, instead of reacting to them [59]. Drug manufacturers can, for example, proactively increase production operations if a sudden increase in forecasted demand at clinics is made known to them in a timely fashion. Otherwise, manufacturers would be forced to wait for orders from intermediate storage facilities that may belatedly reflect the sudden demand increase. Information sharing is, presumably, a reasonable starting point for supply chain reform, because it allows organisations to work in unison to the benefit of the entire supply chain.

Initiatives utilising the benefits of information sharing in pharmaceutical supply chains have successfully been introduced in some African countries in recent years. The Senegalese contraceptive supply chain, for example, suffered predominantly as a result of inadequate inventory management and poor distribution practices. A study conducted in 2011 revealed that at least 60% of contraceptive stock-outs occurred at warehouses and health-care facilities, despite stock availability at a national level. In a drive to remedy this stock-out dilemma, the supply chain adopted a new system according to which dedicated logisticians purposefully utilise stock data to manage inventory and curb stock-outs. Within the first six months of implementation, stock-outs diminished to less than 2% across 140 health-care facilities in Senegal [37].

A large-scale anti-malaria drug stock-out crisis in Africa, on account of a dearth of proper information management, furthermore gave birth to the so-called *SMS for Life* programme in 2009. This programme offers a web-based reporting system that allows health facility workers to report stock levels on a weekly basis through a simple SMS message. The utilisation of accurate and real-time stock level data subsequently alleviated the stock-out predicaments in Kenya and Tanzania. The system is geared for implementation in more African countries [10, 57].

Another example illustrating the power of mobile technology in respect of supply chain information sharing is embodied in South Africa's so-called *Stock Visibility Solution* (SVS) programme. The SVS is a mobile phone-based reporting system employed in more than 3 000 clinics across South Africa. Clinic staff report stock levels on a daily basis by means of a mobile application. The frequent capturing of accurate stock level data is aimed at enabling dispensaries to identify poorly performing facilities and to purposefully manage inventory [77].

Although the extent to which organisations have adjusted their operations in the three aforementioned cases is lesser known, the results build a strong case for information sharing as a means to eradicate stock-outs in pharmaceutical supply chains.

Inventory replenishment policies are commonly employed by supply chain entities to establish two key decision variables, namely a *reorder point* and a *reorder quantity*. It would seem reasonable that the values of these two variables should be informed by pertinent supply chain elements, such as delivery lead times and customer demand. A health-care facility may, for example, increase its reorder quantity in the face of significantly increased patient demand. When demand subsequently declines, however, it would be uneconomical to sustain the large reorder quantities. The parameters of an inventory replenishment policy should therefore be dynamic in order to both satisfy demand and do so economically. Now consider a large group of health-care facilities that order from the same supplier. If the health-care facilities experience varying demand over time, they will also adapt their ordering behaviours over time. As a result, the supplier may experience volatile demand that makes it difficult to control its own inventory effectively. This phenomenon may even extend to the supplier's suppliers and ultimately abound an entire supply chain network. As a result, it is extremely difficult to coordinate inventory management decisions across an entire supply chain network. A suitable mechanism for readily

adapting the parameters of inventory replenishment policies at the various entities in a large and complex supply chain may be found within the notion of *self-organisation*.

Amid multiple and occasionally vague definitions of self-organisation, De Wolf and Holvoet [40] provided the following working definition of self-organisation: “Self-organisation is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control.” The aforementioned ‘structure’ may be of a spatial, temporal or functional nature, while a lack of ‘external control’ marks the absence of direction, manipulation, interference, coordination, pressures or involvement from outside the system [40]. Self-organisation, in other words, empowers entities in a system to wholly control their own operations and to acquire coordinated structure in the management of these operations, without any control instructions being imposed explicitly from outside the system.

A self-organising process exhibits four key characteristics. First, it displays an increase in order [40, 135]. This characteristic embodies the ‘organisation’ element of the process as a result of the constituents of a system ‘organising’ themselves in order to improve their collective performance in respect of a particular function [67]. The second feature of a self-organising process is that its components are autonomous (void of external control) [40, 67, 133]. The third property is that of adaptability or robustness. In the face of perturbations or change imposed on the collective system, a self-organising process is expected to adapt accordingly in order to restore itself and to maintain organisation autonomously [40, 67]. Finally, self-organisation is a dynamic process since it occurs over time [40, 67, 133, 135].

Self-organisation may lead to the related phenomenon of *emergence*. The meaning of self-organisation and emergence is often confused in the literature and incorrectly perceived as synonymous [40]. Based on the historical use of the concept in the relevant literature, De Wolf and Holvoet [40] provided the following definition for emergence:

“A system exhibits emergence when there are coherent emergents at the macro-level that dynamically arise from the interactions between the parts at the micro-level. Such emergents are novel with regards to the individual parts of the system.”

In this definition, the term ‘macro-level’ refers to a system as a whole while ‘micro-level,’ on the other hand, considers a system from the point of view of its individual constituents. Emergence may also be described as the phenomenon where structure not explicitly represented at a lower level (entity level), emerges at a higher level (global system) [119]. For sufficiently large systems, any individual constituent may be removed or replaced without damaging the emerging structure [68]. It is important to recognise that emergence may occur spontaneously (without expecting it) and it may have either good or bad consequences [119]. An intriguing example of self-organisation and emergence found in nature is illustrated in Figure 1.2. A colony of ants often use their bodies to build a living bridge in order to allow them to cross a gap in their path. This is achieved through self-organising behaviour where each ant follows a set of two simple rules. First, it slows down as it reaches the gap and secondly, it stops when it feels another ant walking over it. The ants continue in this manner until they have formed a complete bridge. Since no individual ant is representative of the bridge, the bridge is said to be an *emergent* resulting from the local interactions between the ants.

The concepts of self-organisation and emergence may be applicable to the context of pharmaceutical supply chains in the following way: Suppose each facility in a pharmaceutical supply chain dynamically adapts its own inventory replenishment policy in order to prevent stock-outs locally (*i.e.* inventory management occurs by self-organisation). In other words, each facility seeks to achieve and maintain an operating ‘structure’ within the larger supply chain that empowers it



FIGURE 1.2: A living bridge emerges from the self-organising behaviour of ants [64, 107].

to restrict stock-outs actively. In an information-sharing supply chain, these facilities remain autonomous and may simply base their decisions on what they observe at surrounding facilities. Emergence may subsequently manifest itself as an effective, global supply chain management policy that arises from the local interactions between facilities.

Inventory management is a continuous process filled with sequential and dependent decisions. For example, a decision either to place an order or not to place an order on any given day, will typically influence the ordering decision made on the following day(s). Given the presence of random variables such as demand, delivery lead times and storage capacities, it is evident that many factors may influence an inventory management policy. Subsequently, the question arises as to how an inventory manager can develop and adopt a fluid policy that is sufficient for many different supply chain permutations. One solution may be found within the machine learning paradigm of *reinforcement learning*. It is an approach that enables an agent to learn particular behaviour for specific situations through interaction with its environment. In an inventory management context, reinforcement learning may be employed by inventory managers to learn optimal (or near-optimal) ordering decisions for every possible situation.

## 1.2 Problem description

The problem considered in this thesis is concerned with the performance of traditional pharmaceutical supply chains in developing countries and how these may be improved by means of established, demand-driven supply chain management policies. Poor pharmaceutical supply chain management practices, compounded by a lack of information visibility across the multiple tiers of the supply chain, give rise to medicine stock-outs and adverse patient health outcomes. Since manufacturers dictate the pace at which pharmaceuticals are injected into supply chains, the visibility of inventory levels and demand forecasts at downstream storage facilities and health-care facilities may inform effective production and distribution regimes. Storage depots, in turn, can readily adapt procurement and distribution practices based on the real-time visibility of stock data at health-care facilities.

In this thesis, a tool for discovering and evaluating the effectiveness of inventory replenishment policies in information-sharing pharmaceutical supply chains is established within the modelling paradigm of agent-based computer simulation. The proposed simulation model is capable of accommodating embedded reinforcement learning in order to allow agents to learn effective inventory replenishment policies based on information shared within the supply chain. The simulation model is able to pronounce on the effectiveness of the policies learnt in the form of appropriate *key performance indicator* (KPI) values, which may be used by decision makers to compare the relative effectiveness of different information-sharing protocols.

### 1.3 Research objectives

The following ten research objectives are pursued in this thesis:

- I To *conduct* a thorough review of the literature related to:
  - (a) The notion of a supply chain, with a specific focus on:
    - (i) the constituents of supply chain management,
    - (ii) information sharing in a supply chain,
    - (iii) the concept of demand-driven supply chain management,
    - (iv) inventory replenishment policies in a supply chain, and
    - (v) a global and a local perspective on pharmaceutical supply chain management.
  - (b) Simulation modelling techniques, with a particular focus on supply chain modelling within an agent-based context.
  - (c) The machine learning paradigm of reinforcement learning.
- II To *define* nested hypothetical information-sharing scenarios in a pharmaceutical supply chain. The design of these scenarios are informed by the research conducted in pursuit of Objective I(a). The relative effectiveness of information sharing should be evaluated in the context of these scenarios.
- III To *identify* a suitable reinforcement learning algorithm capable of successfully learning effective inventory replenishment policies. This selection should be informed by the research conducted in pursuit of Objective I(c).
- IV To *establish* suitable KPIs for evaluating the relative effectiveness of inventory replenishment policies in a pharmaceutical supply chain. These KPIs should sufficiently pronounce on the performance of a pharmaceutical supply chain as a whole, as well as on the performance of individual facilities within the supply chain.
- V To *design* an agent-based pharmaceutical supply chain simulation model that may be used as a tool for discovering effective inventory replenishment policies by implementing the reinforcement learning algorithm of Objective III in the context of the information-sharing scenarios of Objective II. Furthermore, the simulation model should serve as a test bed for evaluating the relative effectiveness of inventory replenishment protocols with respect to the KPIs of Objective IV. The simulation model should take as input the topology of a pharmaceutical supply chain network.
- VI To *verify* and *validate* the simulation model of Objective V according to generally accepted modelling guidelines researched in pursuit of Objective I(b).

- VII To *formulate* the inventory management problem considered in this thesis as a reinforcement learning problem. The formulation of the problem is informed by the guidelines researched in pursuit of Objective I(a)
- VIII To *apply* the simulation model of Objectives V–VI to the information-sharing scenarios of Objective II in the context of an experimental design. The reinforcement learning algorithm of Objective III is employed to solve an instance of the inventory management problem formulated in pursuit of Objective VII during each experiment.
- IX To *compare* statistically the relative effectiveness of the information-sharing scenarios of Objective II based on the experiments conducted during the experimental design of Objective VIII. The relative effectiveness of information sharing is measured in terms of the KPIs identified in pursuit of Objective IV.
- X To *suggest* possible avenues of future work that may follow on the work contained in this thesis.

## 1.4 Scope delimitation

The overarching objective of this study is to demonstrate how information sharing may potentially benefit inventory management in a pharmaceutical supply chain context. Subsequently, the research aims to provide sufficient evidence that information sharing is a worthwhile avenue to pursue in the quest for supply chain reform. Since the number of information instances that may be shared in a pharmaceutical supply chain is large, only a small selection of information-sharing permutations is considered in this thesis. Although the results of this study may not pronounce on the impact of information sharing absolutely, it may provide guidance towards identifying the most prominent information-sharing configurations in a pharmaceutical supply chain.

In this research, it is assumed that facility-specific information may be shared across multiple tiers of a supply chain, without any conflicts of interest. Furthermore, it is assumed that shared information is accurate at all times and is shared in real-time. An investigation is also made into the possibility of supply chain peers sharing inventory between them in order to satisfy demand in the short-term future. For instance, a clinic may occasionally choose to order inventory from a nearby clinic, as opposed to ordering from a supplier upstream. Although there are several possible inventory-sharing schemes available for implementation, only one instance is considered in this thesis. Finally, the practical implications for implementing each of the information-sharing scenarios considered is not taken into account. The human and financial resources involved in the installation and subsequent management of information-sharing technologies are, for example, not considered.

The proposed simulation model serves strictly as an evaluation tool and not as an optimisation engine employed in pursuit of optimal supply chain configurations. In the guise of a concept demonstrator, the proposed simulation model therefore adopts a particular level of abstraction that makes it suitable for evaluating the impact of information sharing conceptually. Reinforcement learning is employed in the proposed concept demonstrator in order to demonstrate how agents may learn effective inventory replenishment policies based on the information provided to them. The aim of each reinforcement learning agent is to learn an inventory management policy that minimises stock-outs and expiries locally, whilst maintaining reasonable inventory levels. Supply chain performance is measured with respect to stock-outs and expiries only, and not in

terms of monetary cost. The impact of holding cost is, however, taken into account during the reinforcement learning process so as to ensure that inventory levels are never excessively high.

## 1.5 Research methodology

The execution of research toward this thesis is segmented into five distinct stages. The first stage comprises a thorough literature review specifically aimed at the areas of the academic literature identified in Objective I. In the first place, the literature study provides a comprehensive understanding of the notion of supply chain management. Considerable attention is afforded to the concepts of demand-driven supply chain management and information sharing, with a focus on how these notions may be applied to improve supply chain performance. Traditional, as well as contemporary inventory replenishment policies, are reviewed. The review of the supply chain literature also extends to the unique characteristics of pharmaceutical supply chains in order to highlight the similarities and differences between this type of supply chain and a commercial supply chain. The second branch of the literature review provides an overview of computer simulation modelling, with a focus on agent-based modelling and generally accepted guidelines for simulation model verification and validation. The literature study concludes with a review of the field of reinforcement learning and, in particular, some reinforcement learning solution approaches.

During the second stage of this research, a number of hypothetical information-sharing scenarios in a pharmaceutical supply chain are designed and proposed in fulfilment of Objective II. The first of these scenarios does not involve any information sharing and serves as a benchmark scenario. The scope of information sharing is enlarged incrementally over each of the remaining scenarios.

The third stage of the research pertains to the design and formulation of an agent-based pharmaceutical supply chain computer simulation model in pursuit of Objective V. This simulation model is capable of modelling the high-level operation of a pharmaceutical supply chain over time and accommodates the information-sharing scenarios of Objective II. The model takes as input a pharmaceutical supply chain network and the user may define, amongst others, the constituent facilities, the connections between facilities, delivery lead times and the nature of end-user demand. Furthermore, the reinforcement learning algorithm identified in pursuit of Objective III is embedded in the simulation model in order to allow agents to learn inventory replenishment policies based on one of the various information-sharing scenarios. The simulation model is finally verified and validated in fulfilment of Objective VI.

The fourth stage of this study involves the design of the reinforcement learning problem considered in this thesis, in pursuit of Objective VII. A reinforcement learning problem instance is developed for each of the information-sharing scenarios of Objective II. Although this formulation is carried out in the context of pharmaceutical supply chain management, it provides valuable insight into how reinforcement learning can be applied to inventory management problems in general.

A set of simulation experiments is designed and executed during the fifth stage of the research, in fulfilment of Objective VIII, so as to determine the relative effectiveness of information sharing according to the scenarios of Objective II. This experimental design involves the learning of agents for each of the scenarios, and the subsequent implementation of the policies learnt so as to evaluate their relative effectiveness. This set of experiments is carried out in respect of a hypothetical pharmaceutical supply chain network that is subjected to a fluctuating end-user demand pattern. The results are analysed statistically with respect to the KPIs of Objective IV, in pursuit of Objective IX.

The thesis closes with a summary of the contributions, as well as recommendations with respect to possible future work and improvements that may follow on the work documented in this study, in fulfilment of Objective X.

## 1.6 Thesis organisation

Including the current introductory chapter, this thesis comprises a total of eleven chapters. Chapters 2, 3 and 4 are included in Part I and they are devoted to a brief review of the literature topics pertinent to this thesis. The notion of supply chain management is reviewed in the second chapter and an overview of computer simulation modelling is provided in Chapter 3. Part I concludes in Chapter 4 with a brief review of the machine learning paradigm of reinforcement learning. Part II of this thesis opens in Chapter 5 with the formulation of a number of hypothetical information-sharing scenarios in a pharmaceutical supply chain. The architecture of the proposed agent-based pharmaceutical supply chain simulation model is described next in Chapter 6. Chapter 7 is devoted to the formulation of the inventory management reinforcement learning problem addressed in this thesis. The experimental design approach followed in this thesis is delineated in Chapter 8 and the numerical results are subsequently presented in Chapter 9. The thesis finally closes in Part III with a summary of the research contributions and suggestions for possible future work in Chapters 10 and 11, respectively.



**Part I**

**Literature review**



---



---

## CHAPTER 2

---

# Supply chain management

### Contents

2.1	An introduction to supply chain management . . . . .	14
2.2	Supply chain management strategies . . . . .	15
2.3	The bullwhip effect . . . . .	19
2.3.1	<i>Causes of the bullwhip effect</i> . . . . .	19
2.3.2	<i>Preventing the bullwhip effect</i> . . . . .	20
2.4	Information sharing in supply chains . . . . .	21
2.4.1	<i>Types of shared information</i> . . . . .	22
2.4.2	<i>Barriers to information sharing</i> . . . . .	24
2.4.3	<i>Previous supply chain information sharing studies</i> . . . . .	24
2.5	Demand-driven supply chain management . . . . .	26
2.6	Supply chain collaboration . . . . .	28
2.7	Inventory management . . . . .	29
2.8	Measuring supply chain performance . . . . .	34
2.9	Pharmaceutical supply chains . . . . .	36
2.9.1	<i>Global challenges in pharmaceutical supply chains</i> . . . . .	36
2.9.2	<i>Inventory management in pharmaceutical supply chains</i> . . . . .	38
2.9.3	<i>A perspective on the South African pharmaceutical supply chain</i> . . . . .	39
2.10	Chapter summary . . . . .	41

The objective in this chapter is to provide the reader with an overview of the broad field of supply chain management. Brief, general introductions to supply chain management and its relevant strategies are provided in §2.1 and §2.2, respectively. This is followed by a description of the notorious bullwhip effect in §2.3, with a particular focus on the causes of, and antidotes to, this well-documented phenomenon. An overview of the role of information sharing in supply chains is next provided in §2.4. In §2.5 the focus shifts to the newer concept of demand-driven supply chain management, while the significance of supply chain collaboration is discussed in §2.6. Considerable attention is afforded to the practices of inventory management and performance measurement in supply chains in §2.7 and §2.8, respectively. Some of the most prominent challenges in the pharmaceutical supply chains of developing countries are next described in §2.9. The chapter finally closes in §2.10 with a brief summary of the material included in this chapter.

## 2.1 An introduction to supply chain management

Market competition has increased and intensified globally in the last half-century and has forced organisations to adapt and improve their internal operations in order to remain competitive [62]. Initially, the first phase of these improvement efforts pertained to organisations focussing on marketing techniques aimed at capturing and maintaining customer loyalty. Since customer needs change over time, however, the next shift in focus was placed on understanding customer requirements accurately, and translating these needs into precise product or service specifications [115]. The quest to provide products and services according to these exact user specifications furthermore called for a renewed emphasis on proper engineering and manufacturing functions. The need for understanding customer requirements properly became more pronounced when the market demand for new products and services started to increase more rapidly. As a result, manufacturing functions had to become more adept at enhancing their flexibility so that they could respond to these ever-changing market requirements [115]. This increased focus on flexibility has compelled manufacturers to become more involved with their respective suppliers so as to ensure that they receive high-quality materials. Although these improvement strategies evolved over time, businesses soon learnt that these methodologies should be integrated in order to deliver a high-quality product or service, at a feasible cost. This need for integration has brought about a paradigm shift where organisations realised that managing their own businesses exclusively was not sufficient for maintaining competitive advantage [115]. The notion that businesses' focus should extend to their upstream and downstream partners in order to increase collective competitiveness, gave birth to the concept of a *supply chain*.

A *supply chain* is an integrated network of all the business entities that are involved, through upstream and downstream connections, in the different activities associated with the transformation of raw materials to the finished product (or service) that serves to fulfil consumer demand [32, 33, 62]. A supply chain is typically characterised by the bidirectional flows of information, materials and money between the relevant supply chain entities [32]. All of these flows incur financial costs and should therefore be coordinated and managed effectively in order to enhance overall supply chain performance at a reasonable cost [32, 98]. The practice of managing a supply chain is commonly known as *supply chain management*.

*Supply chain management* involves the active management of the movement and coordination of material, information and financial flows across the entire supply chain, in a manner that maximises the supply chain value [32, 115]. Effective supply chain management is achieved through the systemic and strategic coordination of business functions, with the objective of improving the profitability, performance and competitiveness of the individual organisations and the entire supply chain overall [87, 112]. Simchi-Levi *et al.* [140] provided a more tangible definition of supply chain management by highlighting its operational functions:

“Supply chain management is a set of approaches used to efficiently integrate suppliers, manufacturers, warehouses and stores so that merchandise is produced and distributed at the right quantities, to the right locations, and at the right time in order to minimise system-wide costs while satisfying service-level requirements.”

A notable characteristic of supply chain management is the emphasis placed on the supply chain as a whole. By implication, effective supply chain management transcends organisational boundaries and asks of organisations to collaborate for the benefit of themselves and of the entire supply chain. This view correlates strongly with the modern phenomenon that supply chains, as opposed to individual organisations, compete against one another [87].

Supply chain management is typically regarded as a complex task and this complexity may be attributed to two fundamental characteristics of a supply chain [140]. The first property is related to the *size of a supply chain*. Supply chain complexity increases as the size of a supply chain grows, because it becomes all the more difficult to optimise or coordinate a supply chain system-wide. The second source of supply chain management complexity is the *uncertainty* inherent to any supply chain. Supply chain uncertainties, such as customer demand or delivery lead times, are pervasive and complicate supply chain management because they are intrinsically linked to all supply chain management activities. Supply chain design should, therefore, be aimed at eliminating as much uncertainty as possible [140].

Monczka *et al.* [115] provided an extensive description of typical supply chain management activities. The activity of *purchasing* is, for example, concerned with the identification and selection of suppliers, the management of supplier relationships and the function of collaborating with them to support the various manufacturing functions [26, 86, 115]. At the heart of supply chain management lies the notion of *demand and supply planning* [32, 33]. Demand planning captures the nature of customer demand (what is demanded and when), whereas supply planning focuses on the alignment of procurement and manufacturing operations in order to fulfil demand successfully. Manufacturing units are typically involved in *production planning, scheduling and control* activities [115]. These processes include the establishment of production schedules as well as the control and monitoring of real-time manufacturing processes. The activity of *inventory control* involves all the processes concerned with the management of inventory levels, with the overall objective of satisfying customer demand [33].

The activities of *receiving, materials handling and storage* involve the physical receipt of goods from suppliers and the subsequent storage of these materials [115]. *Quality control* is performed in a supply chain to ensure that product standards are maintained and, more recently, includes an emphasis on the prevention of defects, instead of simply detecting them [115]. Since the flow of materials is an essential component in any supply chain, *transportation* is a substantial supply chain management function [26, 32]. Apart from the physical distribution of materials, the transportation activity also involves the preparation of outgoing orders through labelling and packaging. The *warehousing* function involves the temporary storage of inventory before it is distributed to a customer [32]. The activity of *order processing* represents the link between an organisation and its external customers [115]. When a new order is received, an organisation first has to determine whether it possesses the production capacity required to fulfil the customer demand on time. Finally, *customer service* encompasses the set of activities aimed at establishing and maintaining good customer relationships [32, 86].

## 2.2 Supply chain management strategies

According to Chopra and Meindl [32], any organisation should ensure that its competitive strategy is aligned with a proper supply chain design so that a so-called strategic fit may be achieved. Consider a computer store that, for example, advertises its supposed ability to respond to customer demand for computers faster than its competitors. This computer store, however, operates on a make-to-order system and does not carry any finished computers in-store. Suppose furthermore that the computer store's suppliers of computer components seek to minimise distribution costs and employ a less expensive, slower means of transport. Although the store might advertise its competitive strategy as one that responds quickly to customer orders, the slow means of transportation employed by the upstream suppliers may render this strategy infeasible. Supply chain design and supply chain strategy, therefore, play defining roles in determining the responsiveness and efficiency of a supply chain [32].

Fisher [49] argued in his influential work that an effective supply chain strategy should be informed by the nature of the corresponding product demand pattern. The majority of supply chain problems stem from a discrepancy between the supply chain type and the product kind. Based on its demand profile, a product may be classified as either primarily functional or primarily innovative [49]. Functional products are everyday, common products that fulfil basic human needs, such as bread and milk. These kinds of products enjoy stable and predictable demand because basic human needs do not change significantly over time. Functional products also embrace low profit margins because of the large market competition that exists amongst retailers for these products. Innovative products, on the other hand, are original commodities offered in addition to functional products and do not necessarily fulfil basic human needs. Examples of innovative products include a new sports car or a new personal computer. The pure originality of innovative products make their demand highly unpredictable, but an initial lack of competition for these products allows retailers the opportunity to realise higher profit margins. When competitors start, however, to introduce similar innovative products, the original competitive advantage held by the retailer decays rapidly. This phenomenon makes the life cycles of innovative products relatively short and forces organisations to introduce new innovations to the market frequently. Fisher [49] argued that functional and innovative products each require their own type of supply chain, respectively.

According to Fisher [49], any supply chain performs a *physical function* and a *market mediation function*. The physical function comprises the physical activities of manufacturing, distribution and storage of stock across an entire supply chain. The costs associated with these physical functions are classified as *physical costs*. The market mediation function, on the other hand, targets the successful fulfilment of customer demand through avoiding both stock shortages and stock surpluses. *Market mediation costs* are incurred when surplus stock is marked down and sold at a loss, or when sales opportunities are lost as a result of demand exceeding supply. Fisher [49] argued that organisations with functional products can prioritise their physical function and aim to minimise their physical costs because market mediation can be achieved with reasonable accuracy. Considering the low profit margins of most functional products, the physical function arguably presents the best opportunity for financial savings in the supply chain. Fisher [49] suggested that a so-called *efficient supply chain* is the best suited for functional products. Efficient supply chains can position themselves to minimise their physical costs because the demand for functional products is fairly stable. In order to minimise their physical costs, efficient supply chains pursue cost-effective manufacturing, distribution, information transfer and the exploitation of economies of scale [94].

The unpredictable demand for innovative products, on the other hand, intensifies the risk of stock-outs or oversupply in a supply chain. The higher profit margins associated with innovative products, coupled with the pressure to capture market share early, further magnifies the costs associated with stock shortages. This phenomenon compels a supply chain with innovative products to prioritise its market mediation function over its physical function [49]. A supply chain that embraces this approach is classified as a *responsive supply chain*. Responsive supply chains are orientated towards responding to fluctuating customer needs in a timeous fashion [94]. The focus of responsive supply chains lies in speed and flexibility, and not on the minimisation of physical costs. Inventory and manufacturing capacities are, instead, positioned strategically in the supply chain to safeguard against unpredictable demand [49]. Although responsiveness is typically traded for cost in a supply chain, responsive supply chains are geared to manage supply uncertainties reasonably well, meet shorter lead times, achieve high service levels and respond to a large variety of product orders [32]. The trade-off between responsiveness and cost in a supply chain is illustrated in Figure 2.1. Increased supply chain responsiveness can typically be achieved through larger investments in manufacturing and storage functions. Consider, for

example, a supply chain where retailers carry large amounts of stock in an attempt to maximise responsiveness. Although the retailers may carry enough stock to satisfy customer demand, their level of responsiveness is traded for the increased costs associated with holding inventory.

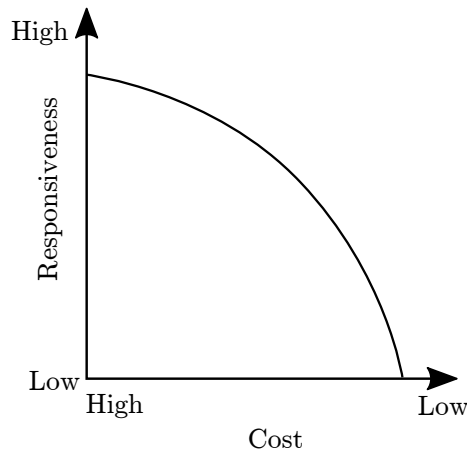


FIGURE 2.1: *The cost-responsiveness efficient frontier [32].*

### Push-based and pull-based supply chain processes

Supply chain processes are traditionally classified as either *push-based* or *pull-based* [32]. The difference between these two processes lies in the timing of their execution relative to a newly placed customer order. Push-based processes are always executed in anticipation of customer demand. A computer manufacturer that, for example, manufactures finished computers before receiving any customer orders for them, follows a push-based approach. These finished products are said to be pushed down the supply chain by the manufacturer in anticipation of future customer demand. Pull-based processes, on the other hand, are performed in direct response to a new customer order. According to a pull-based process, the aforementioned computer manufacturer would only produce a finished computer in direct response to a customer order. Hence, the nature of a supply chain process (push or pull) dictates the production and inventory management activities in a supply chain [120].

In a push-based supply chain, the manufacturing entity decides when, and what quantity of, a product is injected into a supply chain. These manufacturing and distribution functions are based on long-term demand forecasts that are typically informed by historical orders received from downstream facilities, such as warehouses and retailers. This reliance on long-term forecasts often leaves a push-based supply chain ill-equipped to respond effectively to changes in product demand or demand volumes [34, 140]. When actual demand is at odds with demand forecasts, the risk of stock-outs or stock surpluses is increased. Stock shortages result in lost sales opportunities, while obsolescence or inventory carrying costs may be incurred in the case of stock surpluses. A notable benefit associated with push-based processes is the attainment of economies of scale through the manufacturing and transportation of large lot sizes [34, 120].

A pull-based supply chain process, on the other hand, is governed by actual demand and not by forecast demand [140]. According to a pull-based process, the upstream activities of production and distribution are initiated only in direct response to actual customer demand downstream. Hence, inventory is pulled from the upstream operations in the supply chain towards the end

users. A purely pull-based supply chain would not carry any inventory (in storage), because it will always be moving towards the end user [140].

Pull-based systems are, however, difficult to implement in practice. Manufacturing and/or transportation lead times may, for example, be so large that it becomes infeasible for a supply chain to respond to customer demand promptly. Consider the example of the computer manufacturer once more. When this manufacturer follows a pull-based approach, it can only start to assemble computer components once it has received an order for a computer. This assembly process will undeniably consume a period of time and the question arises whether the customer would tolerate the duration of this delay.

The successful implementation of pull-based supply chains hinges strongly on the rapid flow of information between supply chain partners [140]. When actual customer demand data can be transferred across the entire supply chain in real time, all the supply chain partners can orientate themselves so as to respond to real demand effectively. This increased responsiveness makes pull-based systems attractive because they induce less variability, less inventory and lower costs than push-based systems [140]. In contrast to push-based supply chains, pull-based systems can typically not benefit from economies of scale, because supply planning is performed in the short term and does not involve significantly large batches [140].

Supply chains are, however, seldom purely push-based or pull-based. A push-based supply chain can, for example, push stock down towards a retailer, but the end user would have to pull this stock from the retailer's shelves. As a result, a supply chain typically contains both push-based and pull-based processes, and is classified as a *push-pull-based* supply chain. The push and pull phases are typically separated by a so-called *push-pull boundary* or *decoupling point*. This decoupling point represents the point in time when a customer order is placed. Alternatively, the decoupling point is also described as the furthest point upstream in a supply chain where a customer order has a direct influence on inventory-level decisions [120]. A schematic of a generic push-pull-based supply chain configuration is shown in Figure 2.2. The push phase comprises the standardised leg of the supply chain and contains the push processes that are executed in anticipation of customer orders. The pull phase, on the other hand, entails the customised processes aimed at responding to specific customer orders. In other words, materials are pushed downstream in a supply chain towards the decoupling point, irrespective of customer demand. Only when a customer order is placed, the stock is pulled from the decoupling point towards the end user. The location of the push-pull boundary, therefore, plays a critical role in a supply chain's ability to balance supply and demand effectively [32].

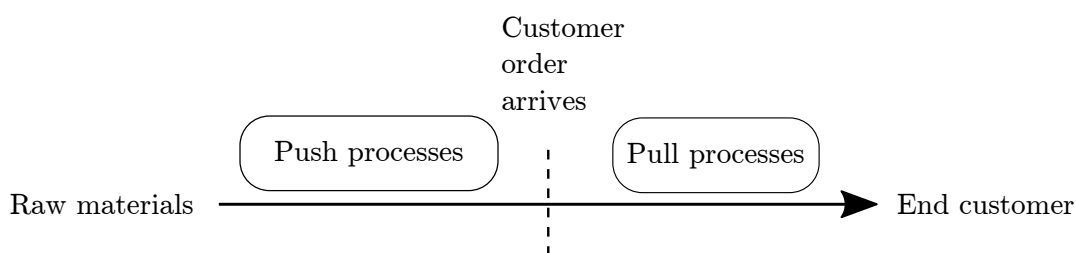


FIGURE 2.2: A schematic representation of a generic push-pull-based supply chain [32].



## 2.3 The bullwhip effect

The so-called *bullwhip effect* manifests itself in a supply chain when fluctuations in order sizes propagate as they move upstream from facility to facility [32, 95, 96]. The consumer goods corporation Procter & Gamble was the first to coin the term ‘bullwhip effect’ after it had observed a disparity between the ordering and demand patterns of Pampers disposable diapers, one of their most popular products. Logistics personnel at Procter & Gamble observed that the demand for Pampers diapers was largely uniform at retail stores. But despite the stable consumption of these diapers, the retailers’ orders to their distributors fluctuated to a small degree. Curiously, the orders placed for raw materials further upstream in the diaper supply chain exhibited even larger swings in variability. Puzzled by this apparent discrepancy, Procter & Gamble realised that each firm independently stockpiled inventory in order to buffer against demand uncertainty and variability. As a result, information about actual market demand became distorted in the form of orders as they moved upstream in the supply chain [96].

The notorious bullwhip effect is, of course, not limited to diaper supply chains and can manifest itself in any type of supply chain. The distortion of demand information induced by the bullwhip effect holds severe repercussions for supply chain performance. Consequences of the bullwhip effect include stock surpluses, inventory shortages, insufficient demand forecasts, inadequate or excessive production and storage capacities or even damaged customer relations due to stock-outs [95].

### 2.3.1 Causes of the bullwhip effect

There are four predominant causes of the bullwhip effect in supply chains [95, 96]. The first of these prevailing sources lies in *demand signal processing*. Inventory managers typically employ forecasting techniques to estimate future demand so that they can plan their local inventory management activities accordingly. Order quantities and desired safety stock levels are, for example, informed by demand forecasts. These estimates about future demand are often based on historical orders received from an organisation’s immediate downstream neighbour. A retailer would, for example, observe end-user demand and base its ordering decisions on this observation. Suppose that this retailer orders from a distributor upstream. The distributor typically does not have visibility of the end-user demand and bases its ordering decisions on the orders received from the retailer. In a serially linked supply chain, implicit information about end-user demand is then relayed from facility to facility in this sequential fashion.

The main problem with this serial ordering pattern is that information about end-user demand becomes distorted as orders move upstream. When a retail inventory manager receives a large order (s)he may, for example, interpret it as a surge in demand and adjust the demand forecast accordingly. When (s)he places a new order to the distributor, the distributor may similarly predict an increase in future demand. Demand information, therefore, becomes increasingly distorted as larger and larger order quantities are placed at different tiers along the supply chain. The outcome of this phenomenon is that upstream manufacturers receive orders that differ significantly from the actual end-user demand. Manufacturers (located even further upstream) are often not equipped to deal adequately with these large fluctuations in demand. The impact of the bullwhip effect is often more pronounced in supply chains with longer lead times. Since order quantities are typically informed by lead times, small changes in customer demand may lead to significant fluctuations in order quantities.

The phenomenon of *order batching* is also referenced as a significant cause of the bullwhip effect [95, 96]. Order batching occurs when firms order stock in large batch sizes so to take

advantage of the economies of scale associated with transportation. Ordering in large batches also allows ordering entities to order less frequently in order to save on ordering costs. With order batching, demand is typically accumulated over a period of time before a new order is placed. Reorder quantities are therefore often larger than actual demand. Suppliers may subsequently receive an erratic stream of orders where a single large order from a customer may, for example, be followed by a significant period without any orders from the same customer. This unpredictable ordering behaviour creates uncertainty for suppliers who cannot anticipate the timing and sizes of future orders. The problem is compounded for a supplier with multiple customers that exhibit this volatile ordering behaviour. As a result, suppliers may often be understocked.

The third major source of the bullwhip effect is *price fluctuations* that lead to variability in customer behaviour [95, 96]. When suppliers run promotions such as price or quantity discounts, ordering firms typically place larger orders so as to take advantage of these potential cost-saving opportunities. This phenomenon where firms purchase stock well in advance of demand is known as *forward buying*. When prices increase again after the discounts, consuming entities stop ordering temporarily and satisfy demand from inventory held in storage. The variation in order quantities during forward buying is typically larger than the variation in consumption rate and this induces the bullwhip effect. The ramifications are particularly severe on manufacturers who ultimately experience large swings in demand. Furthermore, while manufacturers may require costly overtime to fulfil demand during periods of forward buying, their operations may be idle otherwise.

The fourth primary cause of the bullwhip effect is attributed to a phenomenon known as *shortage gaming* [95, 96]. In periods of actual or anticipated supply shortages, retailers often place inflated orders at their suppliers in an attempt to guarantee stock availability. When demand exceeds supply capacity, suppliers would typically ration stock to their customers. A common approach is to ration products to customers based on their respective order quantities. Shortage gaming often materialises when ordering entities anticipate this form of rationing and exaggerate their actual demand in the form of larger orders. This inflation of real demand triggers the bullwhip effect. When the shortage period is over, retailers may cancel their pending orders and stop ordering for a considerable period of time. When these retailers eventually return to their usual order quantities, they distort all previous demand estimates made by a supplier. Shortage gaming is typically promoted in a supply chain environment where buyers are free to place and cancel orders as they like without any repercussions.

### 2.3.2 Preventing the bullwhip effect

Lee *et al.* [95] provided measures for countering the four primary causes of the bullwhip effect discussed in §2.3.1. One of the foremost of these countermeasures is the sharing of market demand information across an entire supply chain [30, 140]. When actual demand information is shared in real time, all supply chain members can develop sound forecasts based on the same information. Effective production schedules may also be established when they are based on actual demand information and not on fluctuating orders. When a supplier carries knowledge of a demand surge in the market it will, for example, not be caught off-guard by larger orders. The bullwhip effect may, however, still persist in an information sharing supply chain where members employ different forecasting methods. Demand information sharing may, therefore, not eliminate the bullwhip effect entirely, but it may reduce its impact considerably [30, 140].

A more radical approach towards eliminating the bullwhip effect is to conduct all ordering and forecasting activities from a centralised point in a supply chain [95]. With centralised

information, a decision-maker may potentially make optimal decisions for the whole supply chain. Since longer lead times may also contribute to the bullwhip effect, another potential corrective strategy is to reduce lead times in a supply chain [96].

A supply chain can arguably thwart order batching only when order processing and transportation costs can be reduced. Today, advanced information technology systems allow organisations to transmit and process order information at faster speeds and at a reduced cost [32]. These rapid and cost-effective ordering methods make it possible for organisations to order in smaller batches and on a more frequent basis. This allows organisations to develop more accurate forecasts and to place orders that better match actual demand. Another driving force behind order batching is the cost benefits associated with full truckload economies. The rise of third party logistics providers has, however, made small batch replenishment a feasible solution for individual retailers. These third party transporters can realise full-truckload economies themselves by consolidating orders from different retailers. In other words, when retailers can outsource transport, they are allowed the freedom of ordering in smaller batches.

The simplest method to negate the bullwhip effect caused by forward buying is to reduce the frequency and magnitude of promotional campaigns [96]. A popular approach adopted by manufacturers is to employ a so-called *everyday low price* (EDLP) strategy. Under an EDLP policy, a manufacturer offers a product at a fixed price throughout the year, without any periodic price promotions. This pricing scheme naturally eliminates the need for traditional forward buying. An EDLP strategy further allows manufacturers to save on costs that were typically associated with marketing schemes during price promotions.

A natural approach towards preventing shortage gaming is to remove incentives for buyers to place inflated orders. With respect to rationing during supply shortages, a simple improvement may be to allocate stock according to retailers' historical sales records, instead of their current orders [95]. When buyers have little knowledge of a manufacturer's supply capacity they may feel pressured to engage in gaming to ensure stock availability. Gaming under such a scenario may be prevented by providing retailers with information on the manufacturer's capacity and inventory levels [95]. Finally, enforcing rules that restrict an ordering entity's flexibility may discourage gaming. Penalising order cancellations and limiting order quantities may, for example, deter buyers from gaming.

## 2.4 Information sharing in supply chains

*Information Technology* (IT) embodies the means that supply chains employ to collect information, to analyse this information and to base decision-making on this analysis with a view to improve supply chain performance. The ability to share information over large distances has made IT an effective vehicle for the integration and coordination of supply chains [32, 142]. According to Chopra and Meindl [32], IT should possess four distinct characteristics for it to support effective decision-making in a supply chain. Firstly, information should be *accurate* so that decision-makers can make the most effective decisions. Next, supply chain information should be *practicable*. Data collection efforts should therefore be aimed at collecting relevant and meaningful data that can provide value to decision-makers [80]. Information should also be easily *accessible* in a timely fashion, because effective decision-making is made possible with real-time information. Finally, and arguably the most critical, information should be *shared* in a supply chain so that decision-makers can coordinate their activities based on the same (real-time) data [142]. When decisions are made based on the same information, the overall performance of a supply chain may be improved [98, 155].

The sharing of information across an entire supply chain is considered an essential practice for improving overall supply chain performance [98]. The outcome of information sharing practices in a supply chain is typically described as *supply chain visibility* or *information visibility* [155]. Supply chain visibility is the extent to which organisations in a supply chain have access to information, or share information with each other, which they consider as fundamental to their own operations, as well as to be of mutual benefit to other supply chain members [9]. Francis [51] proposed a more comprehensive definition of supply chain visibility:

“Supply chain visibility is the identity, location and status of entities transiting the supply chain, captured in timely messages about events, along with the planned and actual dates/times for these events.”

When shared supply chain information relates to both supply-side and demand-side factors of a supply chain and is accurate, timely, complete and in a usable form, a supply chain is said to exhibit a high level of visibility [9, 155, 160].

### 2.4.1 Types of shared information

Greater supply chain visibility holds many benefits for supply chain performance and may, in particular, reduce the impact of the bullwhip effect. When both supply- and demand-side information are shared across a supply chain, it enables all members to increase their responsiveness through faster and improved decision-making processes [39, 110, 155]. This improved responsiveness may, for example, lead to the reduction of cycle times and a decrease in the number of stock-outs [85]. When more information is shared in a supply chain, it may naturally complement forecasting techniques and, in particular, lead to more reliable demand forecasts [121, 141]. This may enable supply chain members to better understand the nature of demand and subsequently enhance their own inventory management processes [80]. Ultimately, information sharing and utilisation typically lead to reduced expenditure and therefore increase profit margins [95].

Lee and Whang [98] described five distinct types of information that may be shared between supply chain partners with the goal of enhancing overall supply chain performance.

The first and most common kind of shared information is *inventory levels*. When inventory level information is shared, the total inventory in a supply chain may be reduced which may, in turn, lead to reduced costs. Consider a simple supply chain where a warehouse provides stock to a retailer. If both of these organisations manage their inventory independently without sharing inventory information, both of them may duplicate safety stock levels or incur stock-outs simultaneously. A standard approach followed to address this potential inefficiency is to share a facility’s inventory level data with its upstream supplier. When a supplier has constant visibility over its customer’s inventory levels, it can proactively increase (or decrease) its own inventory when the customer’s inventory falls below, or exceeds, a pre-specified target level. Sharing inventory level information is expected to reduce upstream order distribution, decrease overall inventories and the number of stock-outs, and also allow manufacturers to develop improved demand forecasts [95].

One of the most popular partnering initiatives where inventory level information is shared is *vendor-managed inventory* (VMI) [154]. According to the principle of VMI, a supplier or manufacturer is responsible for all of the inventory replenishment decisions (reorder points and reorder quantities) at a retailer [32, 120]. This process is facilitated by providing the supplier or manufacturer with access to the retailer’s real-time inventory levels. This level of visibility removes the need for a supplier to maintain excessive inventories and the frequency of replenishment is

also increased, which reduces the need for large levels of safety stock [154]. VMI may also deter shortage gaming and order batching [41].

The second type of information that may be shared in supply chains pertains to *sales information*. Traditionally, information about customer demand is conveyed through orders. An organisation would typically base order decisions on its own interpretation of market demand. As a result, the order information may distort the actual market demand and lead to the bullwhip effect as discussed in §2.3.1. Sharing sales information would, therefore, provide a truer reflection of the real market demand and enable organisations to react to demand fluctuations more effectively [36].

Since multiple independent sales forecasts can promote the bullwhip effect in supply chains, increased effort has been devoted to the sharing of *sales forecast* information. A downstream facility would typically share forecast information with its supplier upstream, because the former is located closer to the market and may therefore develop more accurate and reliable forecasts. Suppliers may, however, have more expertise and access to global market information which may prove useful in forecasting end-user demand. *Collaborative planning, forecasting and replenishment* (CPFR) is a popular business practice embraced by supply chain members where they discuss, share and coordinate demand forecasts in a collaborative spirit [120]. The result of this collaboration is a common demand forecast that may be employed to plan manufacturing and replenishment operations. Benefits associated with CPFR include a potential decrease in inventories, larger order fulfilment levels and improved resource utilisation [4].

Sharing *order status* information can help customers track the movement of their orders along a supply chain. Traditionally, many independent organisations are involved in the delivery of products or services in a supply chain. It is, therefore, often difficult for a customer to enquire about the status of an order, because (s)he is not necessarily familiar with all of the organisations involved in the fulfilment of an order. According to such a scenario, a customer may, for example, repeatedly be referred to different supply chain members when inquiring about order status. Today, an increased emphasis is placed on sharing order status information so that a customer can instantly access the order status, irrespective of its position in the supply chain. A significant benefit associated with this approach is the improvement of the quality of customer service since customer inquiries may be resolved in a single call.

Order status information may include two crucial pieces of information that are also recognised as some of the most significant information types that may be shared in a supply chain [102]. This information pertains to the *location* and *condition* of products during shipment. Knowledge on the location of a shipment may enhance order processing and inventory control activities within a supply chain. Sharing information about the condition of a product, on the other hand, may provide valuable insights into possible damage or even pilferage.

A fifth type of information that may be shared in supply chains involves *production and delivery schedules*. With visibility over production and delivery schedules, organisations can better plan their own production operations. When a manufacturer, for example, has visibility over its supplier's production schedule for its orders, it can better estimate the order fulfilment date.

Lee and Whang [98] also reference *performance metrics* and *capacity* as information types that are often shared in supply chains. Sharing performance metrics, such as lead times and service levels, allow supply chains to identify under-performing stakeholders so that proper corrective action may be taken. When supplier capacity information is shared in a supply chain, it can help to prevent potential shortage gaming behaviour.

Although the types of information that may be shared in a supply chain are well documented, the required granularity of the information is less clear. This observation stems from the fact that

supply chain visibility needs tend to differ between upstream and downstream organisations [80]. Suppliers may, for example, have little use for information about daily market demand. Instead, end-user demand information should rather be aggregated to an appropriate level where suppliers can utilise it effectively in respect of procuring and manufacturing decisions [98].

While mere information sharing can facilitate enhanced coordination in a supply chain, the success thereof is only determined by the effective utilisation of the shared information.

### 2.4.2 Barriers to information sharing

The benefits associated with information sharing are clear, but the practical implementation of information sharing practices in supply chains poses a number of obstacles. One of the most significant barriers to information sharing is that of incentive alignment amongst all supply chain members undertaking an endeavour to share information [98]. Supply chain members may be reluctant to share sensitive information, fearing that other members may exploit this information in order to claim all the benefits of information sharing for themselves. Similarly, supply chain entities may fear that the confidentiality of their information may be compromised when shared with other supply chain members [98]. And even when supply chain members are guaranteed a positive return from information sharing, fears may arise that one organisation will benefit more than the other and this may discourage information sharing entirely [98].

Information sharing may also raise concerns among competing organisations [98]. Consider, for example, two retailers that share the same supplier and both of them share their demand forecast information with this supplier. Suppose furthermore that one of these retailers is planning to run a sales campaign in the near future. This decision may reflect implicitly in its (adapted) demand forecast provided to the supplier. Potential peril may arise if the other retailer would manage to gain access to this information before the planned promotions are implemented. The second retailer may exploit this information and possibly influence the market by adjusting its own prices.

While technology infrastructure provides the means for information sharing, it may also be a limiting factor in an information sharing endeavour [98]. Implementing a synchronised IT system across a whole supply chain may be expensive, time-consuming and even complicated further when partners do not agree on the design of the system. Finally, the timeliness and accuracy of shared information can hinder effective supply chain operation [98]. Complexity is often induced when supply chain members record and share data in an asynchronous fashion. Some organisations, for example, may share information on a monthly basis and others on a weekly basis only. Or in the case of monthly reporting, some organisations may report at the end of every calendar month, while others may define a month as running from the 15th of one month to the 15th of the next, for example.

### 2.4.3 Previous supply chain information sharing studies

The objective in this section is to provide a brief review of some of the most influential work pertaining to the value of information sharing in supply chains. The purpose of this discussion is to strengthen the case for information sharing as a means to improve supply chain performance.

Gavirneni *et al.* [56] produced some of the earliest work towards investigating the value of information sharing in capacitated supply chains. They considered a two-tier supply chain with a single retailer and a single supplier. Three information sharing models were studied: A partial information sharing model, a full information sharing model and a no information sharing

model employed as a base case scenario. According to the partial information sharing model, the supplier has knowledge over the retailer's demand distribution, reorder points and reorder quantities. Under the full information sharing scenario, the supplier additionally observes the retailer's inventory levels on a daily basis. Gavirneni *et al.* [56] proved that supply chain costs tend to decrease as the level of information sharing increases. They concluded that information sharing is always beneficial in respect of supply chain improvement.

Lee, So and Tang [97] also studied information sharing in a two-tier supply chain featuring a manufacturer and a retailer. They proved that information sharing may particularly benefit the manufacturer in respect of cost savings and significantly reduced inventories. In particular, they found that information sharing would benefit the manufacturer most significantly when demand is highly correlated over time, when the demand is highly variable and when lead times are long.

Cachon and Fisher [23] followed a simulation-based approach to compare full information sharing policies with no information sharing policies. For a supply chain with one supplier and multiple identical retailers they showed that supply chain costs can be lowered by 2.2% on average with a full information sharing policy. They reported a maximum savings figure of 12.1%. According to the full information sharing policy, the supplier had access to the retailers' demand and inventory levels. The salient elements of their model included that the supplier is considered perfectly reliable and can fulfil every order successfully after a constant lead time. The demand experienced at the retailers was discrete, independent and identically distributed.

The results obtained by Cachon and Fisher [23] appeared to be counter-intuitive at first. They conjectured that a supplier did not benefit significantly from observing the demand data of retailers. Typically, a supplier would use incoming order information as a guideline for managing its own inventory availability (when order quantities start to increase, the supplier may be prompted to increase its own inventory levels). When a retailer carries a large level of stock, however, the sharing of demand information would be of little value to the supplier. During this period, the supplier would not experience any demand, because the retailer can fulfil demand from its own inventory. Cachon and Fisher [23] posited that a retailer's demand information is most useful to a supplier when the retailer's inventory declines to a level where it would prompt the supplier to order more inventory. This scenario is, however, most likely to occur when the retailer would normally place an order at the supplier. When a retailer places an order, the order quantity would implicitly reflect demand information. Sharing demand information by itself may therefore be considered redundant.

Zhao *et al.* [170] studied the impact of information sharing in a supply chain containing one capacitated supplier and four retailers. They found that demand forecasting methods significantly influence the value of information sharing. As may be expected, the value of information sharing was enhanced when more accurate forecast techniques were implemented. This observation underlines the need for sharing practicable, high-quality data. Another key outcome of their study was the conclusion that it is typically more valuable to share information about future orders as opposed to only sharing information about future demand. They concluded that information sharing may typically lead to considerable cost savings in a supply chain.

Kulp *et al.* [85] studied the effect of information integration between manufacturers and retailers in the food and consumer packaged goods industry. Their results showed that sharing retailer inventory levels was positively correlated with larger profit margins.

More recently, Yu *et al.* [169] explored different information-sharing scenarios in an attempt to identify the most efficient ones. They experimented with different combinations of shared capacity, demand and inventory information. These combinations included a scenario with no information sharing at all as well as a policy with full information sharing. A distinguishing

feature of their analyses include an extensive set of performance measures employed to measure the efficacy of the various information-sharing scenarios. They found that sharing demand information exclusively, proved to be the most beneficial in respect of supply chain performance. Notably, their results suggested that no information sharing policies may yield better supply chain performance than some partial information-sharing practices. Sharing inventory and/or capacity information, without any accompanying demand information, yielded inferior results when compared with the no information sharing policy. They attributed this observation to the fact that demand information is the primary feature driving upstream processes in a supply chain. Therefore, when demand information is absent, it may mislead upstream operations and ultimately amplify the bullwhip effect.

Notably, many of the aforementioned studies involved small supply chain networks.

## 2.5 Demand-driven supply chain management

The traditional definitions of *supply chain management* all tend to emphasise the activities involved in the movement of goods downstream towards the end user [46, 89]. This is primarily achieved by optimising product manufacturing and distribution activities so as to minimise supply chain costs. Although these functions are essential in any supply chain, it may seem as if this particular conceptualisation of supply chain management disregards the final customer to some extent. It may appear as if the downstream movement of commodities are prioritised over the particular needs of the end user — the very party for whom the supply chain exists.

A second shortcoming of traditional supply chain management includes a strong focus on the optimisation of internal business operations [33]. In other words, businesses orientate themselves toward optimising their own performances with little or no regard for their supply chain partners (or the overall performance of the supply chain). A manufacturer may, for example, manufacture products in large batches to realise the economies of scale associated with lower unit costs. The actual consumer demand may, however, not justify the increased production levels. If the actual demand is significantly less, it may lead to excessive stock held in storage. Holding surplus stock may incur significant holding costs and increase the risk of obsolescence [32]. Whereas a manufacturer (or any other organisation) may look inward to optimise its own internal operations, this may be to the detriment of downstream facilities and the supply chain overall.

Supply chains can match supply and demand with considerable ease when demand is fairly stable and predictable [49]. When demand, however, starts to fluctuate, the entire supply chain is compelled to adjust its supply capabilities accordingly. A major problem arises when significant fluctuations in demand are discovered belatedly by facilities upstream in a supply chain. As a result, supply chain members may be unable to fulfil demand on time (incur stock-outs) or, on the other hand, carry inventory surpluses. Such complications evidently arise from a lack of access to real-time information about actual demand. This problem has driven supply chain practitioners towards harnessing the fast-growing power of IT in order to better understand and communicate demand information in real time. This fresh approach led to a departure from traditional supply chain management principles focusing on manufacturing and distribution, with the introduction of the concept of *demand-driven supply chain management* (DDSCM) or DCM where the principal focus is placed on customer demand.

A demand-driven supply chain, or demand chain, prioritises market mediation over its physical efficiency, as discussed in §1.1. In other words, a demand chain places a premium on understanding and fulfilling customer demand successfully. DCM involves a focus on the needs of the customer and subsequently designing the chain to fulfil these particular needs [66]. This focus



on the needs of the customer has led to a reversal of how supply chains are traditionally managed. DCM entails the management and the coordination of the entire demand chain, starting at the end user and working backwards to the suppliers of raw materials [33, 132, 153]. In other words, the customer is considered as the starting point and not as the destination in a demand chain. Indeed, there are two principal objectives of DCM: To coordinate all members in a supply chain and to place an emphasis on customers and their needs as opposed to focusing on local optimisation [22, 153].

One of the primary features of a demand-driven supply chain is that information about demand is made available in real time to all supply chain members. A demand-driven supply chain captures and presents real-time information on current inventory levels and customer demand patterns to all the entities in a supply chain, so that they can react quickly and effectively by updating their forecasts and/or production schedules accordingly [22]. Although the theoretical benefits of DCM have long been established, the dawn of the Internet, coupled with significant improvements in computing power, has made its implementation a real possibility [22, 53].

Since information sharing is pivotal to effective DDSCM, the benefits of information sharing (as discussed in §2.4.1) typically transpire in demand chains. In particular, demand-driven supply chain management may increase supply chain responsiveness, minimise stock-outs and lead to lower inventory levels held in storage [22]. According to research by The Boston Consulting Group [22], organisations with established demand-driven supply chains can carry up to 33% less inventory. When shared information is utilised in an effective manner, it may eliminate the need for overtime and reduce lead times because less emphasis is placed on improving forecast accuracy [47].

A demand-driven supply chain is underpinned by four pillars [22]. The first is *visibility* and involves the sharing of practicable information across the supply chain. Secondly, a resolute *supply chain infrastructure* is required to adapt quickly to sudden changes in supply and demand. Next, careful *coordination* is required amongst supply chain members in order to perform effectively. Finally, a demand-driven supply chain is also focused on the *optimisation* of overall supply chain performance.

There are some significant differences between supply chain management and DCM. Arguably the most prominent difference is that DCM is orientated towards fulfilling demand in the correct market, whereas traditional supply chain management focuses on pushing products and services to undifferentiated markets [46]. The two supply chain philosophies may also be contrasted by their nature of response to market demand. DCM strives to generate revenue through proactively managing demand while supply chain management adopts a reactive approach and only reacts to demand [46]. This difference in foci leads to the progression that DCM places an emphasis on supply chain effectiveness whereas supply chain management prioritises supply chain efficiency [46]. With regards to information sharing, information about customer needs typically pervade an entire chain, but stops at some intermediate stage in a traditional supply chain [46]. In contrast to traditional supply chain management, performance measurement in demand chains furthermore prioritises measuring performance from a customer's perspective [25].

Eagle [47] describes four key characteristics of DDSCM. First, planning in a demand-driven supply chain is separated from the execution phase. In contrast to traditional supply chains, demand forecasts do not drive inventory replenishment decisions in a demand chain. Instead, forecasts are employed to gain forward insights into potential capacity and financial constraints. The second distinctive feature of DDSCM is the strategic positioning of multiple independent inventory locations along the supply chain with the objectives of absorbing supply- and demand-side variability, increasing responsiveness and decoupling replenishment activities. The implementation of additional storage buffers in a supply chain may seem counter-intuitive at first. The strategic

positioning of these buffers, however, often prevent the need for alterations to production schedules which may lead to considerably reduced aggregate inventories. These inventory positions are replenished based on their pre-specified unique and optimal replenishment policies. During each replenishment cycle, inventory is only procured, manufactured or distributed in order to replenish the buffers up to pre-specified stock targets. Finally, these inventory buffers are chosen large enough that they can typically satisfy daily demand and they are only replenished in response to actual demand. Since these buffers are not replenished based on forecasts, they eliminate the propagation of variability so often seen in traditional supply chains [47].

The literature appears divided in its view on whether the notion of DCM should replace supply chain management, or whether it should be considered as an entirely independent, different philosophy [39, 66, 153]. Nonetheless, the introduction of DCM has underlined the need for, and benefits of, demand-driven practices in any supply (or demand) chain.

## 2.6 Supply chain collaboration

As supply chains increased in complexity and expanded in size over time, it became increasingly imperative for supply chain partners to collaborate in order to improve supply chain performance. A *collaborative supply chain* is a supply chain in which two or more independent organisations work together to plan and execute supply chain activities with greater effectiveness than when operating individually [139]. Although collaboration can manifest itself in various forms, its most apparent objective is to create a transparent and visible demand pattern that dictates the entire operation of a supply chain [71]. Hence, the practice of information sharing is vital to the success of a collaborative supply chain.

Collaboration has also been described as the *driving force* behind effective supply chain management [72] and it typically allows supply chains to synchronise supply and demand closely in order to increase overall supply chain performance [139]. Importantly, collaboration demands the buy-in from all partners, irrespective of their role or size in the supply chain [72]. A successfully coordinated supply chain is one in which all decisions are aligned to achieve global supply chain objectives [128]. According to Barratt [8], a true collaborative culture is built on the foundations of trust, mutuality, high-quality information exchange, and transparent communication.

Simatupang and Sridharan [139] described four hindrances to effective supply chain collaboration. The first obstacle is a lack of proper supply chain *performance measures*. Organisations traditionally employ performance measures that only measure their own performances and not those of the overall supply chain. These performance measures are often cost-centric which leads to a focus on the minimisation of individual costs and not on the maximisation of customer value. Hence, organisations often seek to improve their own performances exclusively and this usually comes at the expense of other organisations and overall supply chain profitability [139]. It may therefore be argued that partners can only learn to collaborate effectively when performance measures are integrated across a supply chain so as to measure overall supply chain performance [139].

The issue of localised performance measurement is closely related to the second hindrance of *incentive misalignment* [128, 145]. Typically, supply chain decisions are made based on their localised impacts and not based on their potential (negative) impact on overall supply chain profitability. In other words, if a supply chain partner typically aims to maximise its own performance, this behaviour may have a detrimental effect on other supply chain members. The outcome of such self-focused behaviour may often transpire as an imbalance between supply and demand that hampers the effective flow of products to end users [49].

The third obstacle to supply chain collaboration lies in the presence of *asymmetric information*. Each organisation presides over particular and private supply chain information, such as inventory level and demand data. Organisations are, however, often reluctant to share this information with the rest of the supply chain because they perceive the information to be of particular economic value [139]. As a result, organisations may be forced to make decisions based on their local information and this may lead to sub-optimal outcomes. When supply chain members choose, however, to share the appropriate information, the supply chain may adopt a clearer view of demand and synchronise its operations more effectively. Notably, supply chain performance may still be sub-optimal when each organisation focuses on local optimisation, even under conditions of full supply chain visibility [128]

Finally, *outdated policies* used for day-to-day decision-making may prevent successful supply chain collaboration. Since a supply chain environment is typically dynamic, policies used for practices such as inventory management and demand forecasting may become obsolete rather quickly [139]. A second contributing factor is, furthermore, the fact that these kinds of policies are typically aimed at localised benefits and not at overall supply chain performance. Hence, organisations may aim to exploit outdated policies in order to maximise their own performances [139].

## 2.7 Inventory management

The discipline of inventory management has been widely researched in operations research and management science domains. According to Hillier and Lieberman [69], *inventory* consists of stocks of goods that are held in storage for future use or sale. The practice of inventory management is central to supply chain management and its basic objective is to minimise the costs involved in maintaining inventory, whilst simultaneously meeting customer demand [120, 162]. *Demand* for a product entails the number of product units requested from inventory for some particular use [69]. *Lead time* represents the time duration between the moment at which an order for inventory is placed and the time instant at which the ordered goods are received into inventory [69]. Additional inventory held to buffer against uncertainties or unexpected fluctuations in demand or lead times are classed as *safety stock* [88, 140].

There are five primary functions of holding inventory [120]. The first is that of *decoupling*. Holding inventory at different locations in a supply chain decouples sequential processes and renders them independent from one another. Holding inventory (raw materials or finished products) at intermediate stages in a supply chain may prevent either bottlenecks or stoppages during production [120]. The second function of holding inventory is to *balance* supply and demand, in particular when the time period between product production and consumption is significantly large. This phenomenon is typically observed in environments exhibiting seasonal supply and/or demand. Raw materials for the production of a particular product may, for example, only be available at certain times of year although the finished product is subject to year-round demand. Finished product inventories may therefore be held throughout the year in an attempt to fulfil demand timeously. The third purpose of inventory storage is to *buffer* or *safeguard* against uncertainties about future demand, lead times or supply [38, 120, 140]. Simchi-Levi *et al.* [140] added that firms also choose to carry inventory when it may be more economical than the alternative frequent ordering which incurs large fixed costs.

In large-scale supply chains, inventory is often held to facilitate *geographical specialisation*. The locations of suppliers and manufacturers are often determined by the availability and cost of production components, such as land, power, materials and human resources [120]. The financial benefits associated with this strategic location of facilities are considered to eclipse the increased

inventory and distribution costs that may result from this geographic specialisation [120]. The fifth function of holding inventory according to Pienaar and Vogt [120] is to prevent the cost of a stock-out. A *stock-out* or *shortage* occurs when customer demand is not met on time [162].

According to Pienaar and Vogt [120], there are three possible costs associated with stock-outs. The first is the cost of a *backorder*. Backordering occurs when a customer is willing to have his or her originally unsatisfied order fulfilled at a later date. Backordering costs are the additional costs incurred when processing and expediting the original order. When a customer does, however, decide to fulfil his or her purchase elsewhere, the cost of a *lost sale* is incurred. Finally, the cost of a *lost customer* is incurred in the worst-case scenario where a customer decides to change his or her supplier permanently.

Inventory management policies are employed to determine what inventory level targets should be, when inventory must be replenished and what the order quantities should be [28, 162]. These policies, or inventory models, are typically used to model inventory and demand. Mathematical inventory models are categorised into two main classes based on the predictability of demand [69]. *Deterministic* inventory models are employed when demand is either known or assumed to have been forecast sufficiently accurately. *Stochastic* inventory models, on the other hand, are applicable to inventory systems in which product demand is unknown and cannot be predicted (*i.e.* where demand is a random variable). Although demand may be random, it may follow a known probability distribution with known parameters.

Inventory models may further be classified according to the technique employed to monitor stock levels. *Continuous-review* models are employed when the inventory level is tracked continuously and a replenishment order is placed as soon as the inventory level decreases to a pre-specified *reorder point* [32, 69]. According to a *periodic-review* policy, the inventory level is monitored at fixed intervals (periodically) and order decisions are made only at these review times.

Winston [162] described four prevalent costs associated with inventory models. The first category is classified as *ordering and setup costs* and these costs are independent of order quantities. The ordering cost portion includes the costs associated with the handling and administration of order placement and processing. When a product is manufactured internally, the labour cost and time required for setting up a machine for a production run is included as setup costs. The second cost type is the *unit purchasing cost*, which is simply the variable cost associated with the procurement or production of a single product unit. This cost typically comprises variable labour costs, variable overhead expenses and the cost of raw materials associated with the production or procurement of a single product unit. *Holding or carrying cost* is often a substantial part of inventory costs. Holding cost is the cost of holding one product unit in inventory for a single time period. Components of holding cost include storage expenses, maintenance cost, insurance cost and the costs incurred for potential pilferage or obsolescence. The final cost element is *stock-out cost*, which is the cost of backordering, incurring a lost sale or losing a customer. Stock-out costs are typically harder to quantify than the three preceding cost types.

According to Simchi-Levi *et al.* [140], there are six key elements that influence an inventory policy. The first, and arguably most important, is *customer demand* which may be deterministic or stochastic. Secondly, *replenishment lead times* may be uncertain and therefore influence the timing and quantity of orders. The number of *different products*, the length of the *planning horizon* and various *inventory costs* also affect the complexity of inventory management decisions. Finally, *service-level requirements* are also considered in the formulation of inventory models. It is usually impossible to achieve and maintain a service level of 100%. Therefore, many customers specify and demand an acceptable service level target. According to Hillier and Lieberman [69], any inventory policy should at least provide clear and unambiguous rules for determining when to place an order and how much to order.

### Deterministic continuous-review inventory models

In its most basic form, inventory managers are faced with inventory levels that deplete over time and are replenished once a new batch of ordered goods arrive. In cases where demand is known, deterministic continuous-review inventory models are often suitable for determining appropriate order quantities [69, 162]. Although deterministic continuous-review inventory models are somewhat elementary, they provide the basis for the development of more complicated models. Arguably the most popular deterministic continuous-review inventory model is the celebrated *economic order quantity* (EOQ) model [48].

The basic EOQ model is subject to several assumptions [69]. First, demand is assumed to be deterministic and occurs at a known constant rate of  $d$  units per time. Furthermore, inventory is assumed to be replenished when needed by ordering a fixed batch size of  $Q$  units. The entire batch is considered to arrive at once at the desired time. The lead time for each order is constant and often assumed to have a value of zero in the basic EOQ model. Finally, no planned stock-outs are allowed and demand must therefore be fulfilled from inventory held in storage. Since orders are assumed to be fulfilled instantaneously, orders are placed at the exact time instant that the inventory level depletes to zero. According to this basic EOQ model, orders are placed in a cyclic fashion and the length of time elapsed between successive orders is known as the *cycle length*. Mathematically, the cycle length is expressed as  $Q/d$ .

The inventory level over time in such a basic deterministic continuous-review inventory model is shown in Figure 2.3. The inventory level starts at zero and an order size of  $Q$  units is placed at the start of the period under consideration. The inventory level is depleted at a constant rate  $d$  and a new order is placed as soon as the inventory level reaches zero.

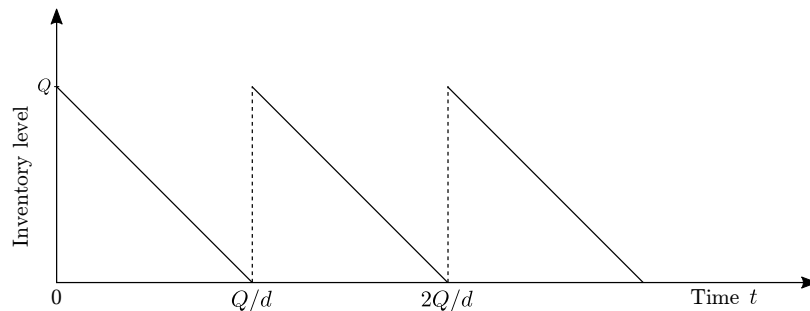


FIGURE 2.3: *Inventory level as a function of time according to the basic EOQ model [69].*

The basic EOQ model only accommodates ordering and holding costs [69]. The setup cost (included in ordering costs) for ordering one batch of products is denoted by  $K$ , while the unit product cost of either purchasing or producing is denoted by  $c$ . The holding cost captures the cost per unit per unit time held in inventory and is denoted by  $h$ . The objective in the EOQ model is to determine the optimal lot size  $Q$  in order to minimise the total cost per unit time,  $T$  [69].

The ordering cost per cycle may be expressed as  $k + cQ$ . Furthermore, the average inventory level during any cycle may be expressed as  $(Q + 0)/2 = Q/2$  and since the cycle length is  $Q/d$ , the holding cost per cycle may be calculated as  $h \times Q/2 \times Q/d = hQ^2/(2d)$ . The total cost per cycle may subsequently be expressed as  $K + cQ + hQ^2/(2d)$ , and therefore the cost per unit time is

$$T = \frac{K + cQ + hQ^2/(2d)}{Q/d} = \frac{dK}{Q} + dc + \frac{hQ}{2}.$$

The value  $Q^*$  of  $Q$  that minimises  $T$  is determined by setting the first derivative of  $T$  with respect to  $Q$  equal to zero and solving for  $Q^*$ . The resulting value is

$$Q^* = \sqrt{\frac{2dK}{h}},$$

and is known as the *EOQ formula*, introduced by Harris in 1913 [48]. Notably, small deviations from the optimal EOQ value  $Q^*$  typically result in only a slight increase in the total cost [162].

The EOQ model may be adapted and extended in several ways in order to accommodate more complicated (and more realistic) scenarios. Backorders may, for example, be allowed in some cases where it makes financial sense to permit temporary periods of planned stock-outs [69, 162]. It is, however, imperative that in this case customers are prepared to tolerate some kind of delay in the fulfilment of their orders. In the basic EOQ model, it is assumed that the ordering cost is independent of the order size. Many suppliers, however, permit *quantity discounts* where they reduce the unit purchasing price for larger orders. Quantity discounts may also be incorporated into the EOQ model [162].

### Deterministic periodic-review inventory models

Deterministic periodic-review inventory models are not based on the assumption of a constant demand rate and, therefore, the EOQ formula does not guarantee a minimum-cost solution [69]. Deterministic periodic-review inventory models are typically considered as multi-period decision problems where the planning in respect of the number of products to be produced (or ordered) at the start of each period needs to be done beforehand. Importantly, the demands for the respective periods are known, but are not necessarily the same. The method of *dynamic programming* may be employed to calculate appropriate order quantities that would minimise the total cost over the entire decision period considered [69]. Dynamic programming is a solution methodology according to which a complex temporal decision problem is subdivided into a series of smaller decision problems that are solved, working backwards in time [162].

### Stochastic continuous-review inventory models

Stochastic inventory models serve as a useful starting point for inventory situations in which there is significant uncertainty about future demands. Owing to the stochastic nature of demand in these models (*i.e.* the demand rate is not constant), the EOQ model does not hold for stochastic inventory situations.

A so-called  $(R, Q)$ -*policy* is typically employed in stochastic continuous-review inventory models [5, 69]. This policy is characterised by two parameters  $R$  and  $Q$ , where  $R$  denotes the reorder point and  $Q$  denotes the order quantity. The order quantity includes two components: The average demand during the replenishment lead time and safety stock safeguarding against possible deviations from average demand during the lead time [140]. Since the inventory level is monitored continuously, an inventory manager can place an order for  $Q$  units at the precise time that the inventory level reaches the reorder point. It may, however, happen that a large number of units is demanded at once when the inventory level is close to  $R$ , in which case the inventory level may decline too far below the reorder point. Stock-outs are often inevitable in such a scenario and the  $(R, Q)$ -*policy* may therefore be insufficient. The so-called  $(s, S)$ -*policy* or *min-max policy* has been proposed as an alternative solution to the aforementioned issue. According to this policy, a new order (with a variable order size) is placed whenever the inventory level is less than or equal to  $s$  [162]. The order quantity is chosen so that it increases the inventory level to

a level of  $S$  units. Occasionally, an inventory manager may choose to order multiple batches of size  $Q$ . In this case, an  $(R, nQ)$ -policy may be adopted, where  $n$  denotes the number of batches ordered [5].

The inventory level over time of a stochastic continuous-review inventory model is shown in Figure 2.4. The demand rate is stochastic and a new order is placed as soon as the inventory level reaches the reorder point. When safety stock and lead time are incorporated, inventory will only be replenished after the lead time has elapsed, as shown in Figure 2.4. Significant increases in the demand rate during the lead time may force a facility to utilise safety stock for the fulfilment of orders placed during the lead time period. It is, however, also possible that the safety stock level may be insufficient and that stock-outs may be incurred before the inventory is replenished.

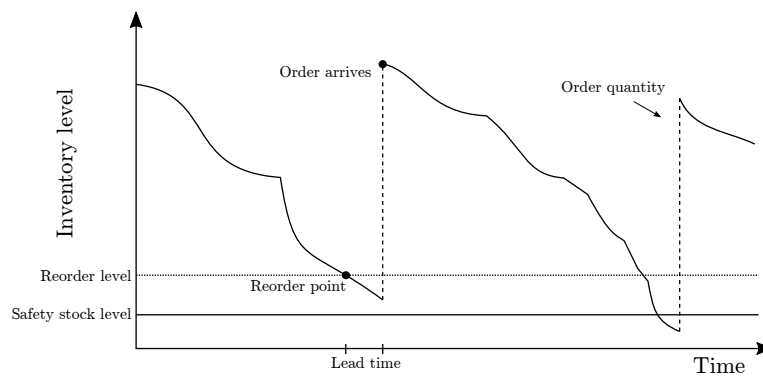


FIGURE 2.4: Inventory level as a function of time according to a stochastic continuous-review inventory model where safety stock and lead times are considered.

### Stochastic periodic-review inventory models

Stochastic periodic-review inventory models typically involve perishable products that can only be held in inventory for a limited period of time after which it can no longer be sold. Because of the perishable nature of the products, only one inventory decision has to be made and therefore models in this category are often called *single-period decision problems*.

The classical *newsvendor problem* is a popular instance of a stochastic single-period inventory model. In this celebrated problem, a vendor selling newspapers has to make a daily decision regarding the appropriate number of newspapers to purchase. If the vendor orders too few newspapers (*i.e.* does not meet daily demand), he or she will forfeit potential profit. If, on the other hand, the vendor orders too many papers (*i.e.* exceeds daily demand), he or she will be left with several redundant newspapers at the end of the day. The objective of the vendor is therefore to choose order quantities that will appropriately balance the costs associated with underordering or overordering [162].

A popular policy adopted in stochastic periodic-review inventory situations (when considering more than one period) is the so-called  $(R, S)$ -policy [162]. The *on-order inventory level* is central to this policy and it is defined as the sum of the available inventory held in storage and the ordered inventory that is still to be received. According to this policy, a review of the on-hand inventory is conducted every  $R$  units of time and an order is placed to increase the on-order inventory level up to  $S$ .

### More complex inventory systems

Organisations, such as suppliers, distributors and retailers, may be responsible for thousands of products and are often not able to develop near-optimal inventory policies for each product. Classification schemes, such as *ABC classification*, are therefore commonly employed to categorise products based on their demand and contributions to total profit [162]. This enables organisations to develop high-quality inventory policies for those products that have the most significant impact on profit.

Whereas many of the aforementioned inventory models were discussed in the context of one facility, a retailer may, for example, choose to manage the inventories of its respective outlet stores from one central point. An important consideration in such a scenario is that the different stores are typically subjected to different demand patterns because of differences in geographical locations, local economic conditions and local culture [1]. This phenomenon may complicate the retailer's decision-making significantly, especially when limited inventories have to be allocated across outlets in an optimal manner.

In most cases, the practice of inventory management spans several groupings of role-players, known as *echelons* [69]. One echelon of a multi-echelon inventory system may, for example, involve a manufacturer, a second echelon may pertain to distribution centres and a third echelon may contain a set of retail stores. A multi-echelon inventory system is typically embedded in a supply chain and the objective of such a system is to minimise the total cost incurred by the entire multi-echelon system [69]. Effective inventory management in a multi-echelon system therefore typically depends on the successful collaboration amongst echelons, which may or may not be managed by independent organisations.

The inventory models discussed in this section are by no means exhaustive and do not consider complexities such as elaborate multi-echelons inventory systems and interactions between products. Efforts to include such considerations do, however, tend to make inventory models unwieldy and intractable [69]. It is important to underline the fact that the inventory management domain of each organisation is unique and that there is no standard model that can accommodate all of the characteristics and limitations of any given inventory situation [171].

## 2.8 Measuring supply chain performance

Measuring the performance of a supply chain is pivotal in determining the effectiveness and efficiency of a supply chain. Chopra and Meindl [32] identified six primary logistics drivers of supply chain performance. The first driver is supply chain *facilities*. Facilities are the physical structures in a supply chain network, such as manufacturing sites and storage facilities. The function, location, capacity and flexibility of each facility influences supply chain performance. *Inventory* is the second driver and refers to all of the raw materials, work-in-progress materials and finished products in a supply chain. The choice of inventory replenishment policy determines the responsiveness, and therefore the performance, of a supply chain to a large extent. Next, *transportation* involves the movement of inventory from facility to facility and holds great significance for both supply chain responsiveness and cost. The fourth driver of supply chain performance is *information*. Sharing information about variables such as inventory levels, transportation, costs and customer demand in real-time can greatly enhance the responsiveness and efficiency of a supply chain. Fifth, *sourcing* involves the contracting of firms to perform particular supply chain functions, such as manufacturing and transportation. Sourcing decisions have significant influences on both strategic and operational functions in a supply chain. Finally, the



*pricing* of goods and services shapes the behaviour of the customer and, therefore, influences supply chain performance.

Supply chain performance measures are typically based on either cost or a combination of cost and customer responsiveness [14]. Costs may include expenses associated with inventory and operations, while lead times and stock-out probabilities are typical indicators of customer responsiveness. Beamon [14] proposed the use of three distinct performance measurement types: *resource measures*, *output measures* and *flexibility measures*. The purpose of resource measurement is to measure levels of efficiency, while output measures are aimed at evaluating the levels of customer service. The capacity of a supply chain to respond to changes in the environment is reflected in flexibility measures.

Resource measures typically involve inventory levels, manpower requirements, equipment utilisation, energy consumption and cost [14]. Specific examples of resource measures include *total supply chain cost*, *distribution costs*, *manufacturing costs*, *inventory costs* and *return on investment* [26]. The total supply chain cost amounts to the total cost of all resources utilised within a supply chain, while distribution costs include the transportation and handling expenses. Physical labour, maintenance work and rework costs are all included in manufacturing costs. Inventory costs are typically associated with work-in-progress inventories, inventory obsolescence and the value of goods held in storage. According to Christopher [33], inventory often constitutes 50% or more of an organisation's current assets. Return on investment is a reflection of a supply chain's profitability and is measured as the ratio of net profit to capital. A supply chain typically pursues the maximisation of return on investment and the minimisation of costs (expenses) [13, 27].

Output measures typically involve customer responsiveness and the quantity and quality of the final product [14]. While many output measures are quantitative, some (such as customer satisfaction and product quality) are of a qualitative nature. Quantitative measures include the total *sales revenue*, *profit*, *order fill rate* and the number of *on-time deliveries*. *Manufacturing lead time* and *customer response time* are two additional examples of output measures and supply chain managers always seek to minimise these times [13, 27]. Further output measures that are afforded considerable attention include *stock-outs* and *back-orders* — these components are often used to describe service levels. The probability that a stock-out will not occur between the time that an order is placed and subsequently fulfilled is a popular measure of service level [69].

*Flexibility* — the ability to adapt to unique customer demand dynamically — is recognised as one the predominant determinants of supply chain competitiveness [17, 61]. Flexibility can be a measure of the ability to readily adapt the number of products manufactured, the capability to change planned delivery dates, the capacity to alter the mix of products manufactured, as well as the ability to adapt existing operations in order to introduce new products [14]. Since flexibility is a measure of potential, some scholars argue that flexibility is a qualitative performance measure while attempts have been made to quantify the different types of flexibility measures [14]. Examples of flexibility measures include the total product development cycle time, machine set-up time and the number of inventory turns [61].

Chan [26] has proposed two measures for measuring the level of supply chain visibility. The first metric, *time*, is a measure of the amount of time required for new information to be transferred to, and processed by, an entire supply chain. Because information can be transferred almost instantaneously *via* a computer, it is imperative that the time metric considers the entire period from when new information is generated up to the point where the implications of the new information is implemented in practice. The second visibility metric described by Chan [26] is *accuracy*. This metric is employed to evaluate whether activities prompted by information sharing have been performed properly. When a new product design is, for example, introduced,

the accuracy metric would reflect the proportion of new products manufactured correctly based on the newly communicated design.

## 2.9 Pharmaceutical supply chains

The discussion in §2.1–2.8 served as an introduction to some of the most salient concepts in supply chain management in general. The aim in this section is, however, to provide a more focused perspective on some of these concepts in respect of pharmaceutical supply chains in particular.

### 2.9.1 Global challenges in pharmaceutical supply chains

A pharmaceutical supply chain is a highly sensitive supply chain with the responsibility of ensuring that the correct pharmaceuticals are delivered to the right people, at the right time and in the condition required to treat disease effectively [151]. Considering the sensitive matter of human health, it may be argued that a pharmaceutical supply chain is obliged to achieve a customer service level target of 100% [151]. Failure to do so will imply that at least one stock-out has occurred and that at least one patient did not receive his or her medication on time. Not only do stock-outs pose significant risks to the health of patients, but they also diminish patients' trust in the ability of the health-care system to serve them appropriately [44].

Although developed nations are not exempted from pharmaceutical supply chain problems, it is the severely constrained resources in developing regions that make the latter's supply chain challenges more pronounced. A recent study by Privett and Gonsalvez [122] identified ten global challenges that are the most prevalent in the pharmaceutical supply chains of developing regions. These challenges are categorised at a system level, at a facility level and at an item level, respectively. There are four system-level challenges and they are described as the most critical because they influence the performance of an entire supply chain. The first system-wide challenge involves a *lack of coordination* amongst supply chain stakeholders because of the typically fragmented structure of pharmaceutical supply chains. This lack of coordination may also be driven by potentially conflicting objectives exhibited by the various supply chain players. Effective coordination amongst supply chain stakeholders is imperative when attempting to minimise total supply chain costs whilst aiming for a service level target of 100% [151]. It is evident, then, that successful supply chain collaboration amongst all relevant stakeholders, as discussed in §2.6, is of particular importance to pharmaceutical supply chains. Secondly, the absence of *demand information* hampers effective procurement and supply decision making across the entire supply chain. Although health-care facilities experience the actual demand for pharmaceuticals on a daily basis, this demand information is seldom shared with the upstream warehouses and suppliers. The value of exploiting demand information according to DDSCM practices was elucidated in §2.5. When demand information is, however, available, it is often aggregated over a large period of time. Hence, historical demand fluctuations are usually obscured, which makes the demand information less useful. A lack of transparent demand information sharing may therefore leave a pharmaceutical supply chain particularly vulnerable to the bullwhip effect — the well-documented phenomenon described in §2.3.

Next, the limited *human resources* typically available in developing nations constrain the overall performance of pharmaceutical supply chains. Health-care facilities are often understaffed and personnel tend to neglect critical duties because of their large workloads. Additionally, clinical staff are also often responsible for stock procurement despite not being trained to make supply

chain decisions. In South Africa, nurses and pharmacists were, for example, reported to be so preoccupied with the rationing of medication amongst patients during stock-outs that they barely afforded attention to the needs of their patients [44]. Finally, *shipment visibility* is the fourth system-wide challenge and tends to be virtually non-existent as a shipment moves from the manufacturer downstream along the supply chain. The problem of shipment visibility is exacerbated by a lack of communication amongst organisations and it is often unknown whether shipments have reached their final destinations successfully. Yadav [166] describes a general lack of information capture and sharing (not necessarily limited to demand and shipment information) as one of the most significant causes of supply chain underperformance. He therefore argues that information flows should be synchronised (and shared in real time) to maximise the performance of supply chains at a minimum cost. A motivation for information sharing in supply chains was provided in §2.4.

The first of five facility-level challenges identified by Privett and Gonsalvez [122] involves practices associated with *inventory management*. A lack of supply chain information and the unique contextual circumstances of each facility make it difficult to manage inventory levels, storage capacity and replenishment policies effectively. A recent study in Zambia revealed, for example, that the inventory policies employed at health-care facilities were responsible for medicine stock-outs despite stock availability at upstream warehouses [99]. These inventory replenishment strategies were fixed without considering the impacts of variables such as delivery lead times and demand seasonality. As a result, safety stock levels and order quantities were not adjusted in accordance with demand fluctuations caused by seasonality. These inventory decisions were furthermore based on historical consumption, causing previous stock-outs to be overlooked [99]. Innovative approaches toward circumventing stock-outs include the ‘borrowing phenomenon,’ where clinical personnel would borrow stock from a nearby health-care facility when faced with shortages [70]. In South Africa, nurses have, however, borrowed stock without reporting a formal stock-out. The term *borrowing* is also ironic because stock is never returned. Only when stock could not be borrowed from a nearby facility, did personnel report an official stock-out [70].

A lack of information (including the absence of shipment visibility) makes *order management* another complex challenge for supply chain managers. When inventory cannot be traced along a supply chain, it makes it virtually impossible to establish whether adequate stock levels are available in the supply chain. Orders may also be delayed, incomplete or incorrect, and there are no means to establish this before a shipment has arrived at its intended recipient. The next facility challenge involves *shortage avoidance*, which involves techniques employed to prevent stock shortages. Stock shortages are often countered when replenishment occurs frequently and large inventories are held in order to maximise service levels. This approach is, however, highly uneconomical because it induces large holding costs as well as lost, damaged, unused or expired stock that leads to wastage [75, 123]. *Warehouse management* is also described as a distinct source of facility-level challenges. Warehouse management problems stem from insufficiently equipped and poorly designed facilities, as well as a lack of proper employee training. When warehouse functions fail, for example, to identify and discard damaged or expired stock, such stock may be included in new shipments. Finally, *shipment visibility* is also classified as a facility-level challenge and, similar to its manifestation on a system level, makes it difficult for inventory managers to manage incoming stock and outgoing stock.

Finally, the two item-level challenges discussed by Privett and Gonsalvez [122] relate to *product expiration* and *temperature control*, respectively. Product expiration is often cited as a significant source of wastage because it resembles financial losses and missed opportunities to provide the stock elsewhere. Overstocking may occur due to improper forecasting, faulty demand quantification, insufficient warehouse management or poorly trained staff. Employees may, for ex-

ample, not adhere to the *first-expired-first-out* principle typically employed in the management of perishable products. Finally, the storage temperatures of many pharmaceuticals need to be monitored and controlled continuously to prevent them from damage. Exposure to extremely hot or extremely cold temperatures may reduce or destroy the efficacy of pharmaceutical products. The main problem experienced in the pharmaceutical supply chains of many developing regions is that storage temperatures are typically monitored and controlled successfully under the manufacturer's ownership only. Temperature control, however, deteriorates as products are handled and repacked by intermediate parties as they are moved downstream.

All ten of the aforementioned challenges influence the performance of a pharmaceutical supply chain in some way or another. By implication, if a pharmaceutical supply chain can work towards resolving these challenges, it can most likely improve its performance. The performance of pharmaceutical supply chains may be measured in respect of the performance measures discussed in §2.8.

### 2.9.2 Inventory management in pharmaceutical supply chains

A general introduction to the field of inventory management was provided in §2.7. The aim in this section is to provide a general background to inventory management in the particular context of pharmaceutical supply chains.

Public health-care facilities cannot carry large amounts of stock due to a lack of space and therefore typically rely on upstream warehouses to resupply them with stock. According to the most prevalent public distribution model in developing nations, pharmaceuticals are distributed to health-care facilities *via Central Medical Stores* (CMSs), regional-level and/or district-level stores [167]. CMSs are commonly employed as the central points for warehousing and distribution, with regional and district stores employed as lower-tier distribution facilities. CMSs may, in turn, receive their inventory from various sources, including manufacturers, importers, distributors and procurement agents [167]. As a result, multiple tiers of storage points and distribution channels exist within a pharmaceutical supply chain which can complicate inventory management. This supply chain complexity is compounded by the fact that procurement and distribution functions are often decoupled from one another with little or no information sharing between them [167].

The two predominant approaches to distribution in pharmaceutical supply chains are the well-known *push* and *pull* systems, described in §2.2. According to a push system, CMSs, regional or district stores decide on the order quantities that are pushed down to lower-tier health-care facilities, based on centrally estimated allocation quantities. In a pull system, on the other hand, each facility manages its own inventories and has to purchase stock from an upstream facility. Push systems are attractive because they are immune to insufficient inventory management practices at the lowest level of the supply chain [167]. Although the effectiveness of push systems depends on proper information systems, they also facilitate impartial rationing decisions in the case of stock shortages. Pull systems, on the other hand, hold the advantage of improved access to local information about demand, but rely on sound decision-making abilities at a decentralised level [167]. Since inventory management capacities at health-care facilities are often inadequate, many pharmaceutical supply chains adopt a hybrid push-pull system for stock distribution [143]. According to this configuration, regional and district stores pull stock from CMSs which, in turn, push stock to health-care facilities [167]. Such a system enables the realisation of the benefits associated with both push and pull systems.

Saedi *et al.* [127] stressed the fact that conventional inventory management policies, such as those discussed in §2.7, cannot be applied *verbatim* to inventory management models in a phar-

maceutical context. According to Saedi *et al.* [127], there are three notable differences that differentiate pharmaceutical inventory models from more general (not health-care-specific) inventory models. Arguably the most prominent difference is that a health-care facility typically aims to maximise its quality of service or care, as opposed to minimise costs. Health-care facilities do not, however, disregard costs entirely, but given the sensitive matter of patient health and well-being, the penalties associated with stock-out and substitute costs are expected to be much more significant than ordering and holding costs [127]. The second distinguishing feature involves the importance of demand satisfaction. Pharmaceutical supply chains tend to place a premium on the prevention of stock-outs since stock shortages may have a detrimental impact on patients' health. In other words, it is crucial to fulfil demand at all times. The impact of stock-outs in commercial or service supply chains, on the contrary, may be less severe because they do not necessarily affect human health directly. Finally, in some cases, pharmaceutical inventory management permits the opportunity to replace an item in shortage with an alternative product — something which is not always possible in commercial supply chains.

With respect to particular inventory management policies, the *min-max* and  $(R, Q)$ -policies (discussed in §2.7), or variations of these policies, are most often employed in public health-care facilities in developing countries [143, 167]. According to Yadav *et al.* [167], however, adherence to these policies remain poor for a large majority of developing countries.

### 2.9.3 A perspective on the South African pharmaceutical supply chain

Medicine stock-outs are pervasive in developing countries all over the world and their consequences are typically severe for patient health, as stated in §1.1. The aim in this section is to provide a brief overview of a number of case studies describing pharmaceutical supply chain problems pertinent to the South African context.

One of the best-documented pharmaceutical supply chain disasters in recent years in South Africa resulted from an unprotected strike at the Mthatha medical supplies depot in the Eastern Cape in 2012. The strike led to the suspension of numerous staff members, leaving the depot with the inadequate number of ten remaining employees [43]. Given that the Mthatha depot was responsible for supplying ARV therapy medication to more than 100 000 patients at the time, doubts were raised about the depot's capability to carry out its daily operations successfully [43]. The consequences of the nearly month-long strike were calamitous. Due to a lack of staff and limited management capacity, orders were not processed at the warehouse, inventory not distributed to clinics and, as a result, inventory levels decreased at both the depot and the health-care facilities that it served. Crucially, medicines could not be dispensed to patients in need during the time period that followed the strike.

Despite interventions from organisations such as Doctors Without Borders and Treatment Action Campaign to support staffing and drug delivery in the ensuing months, many problems at the Mthatha depot remained unresolved. During the period September 2012 to January 2013, for example, 24% of a total of 72 health-care facilities served by the Mthatha depot reported that they had to turn HIV or TB patients away without any medication [43]. And in May 2013, four months after the strike, as many as 40% of the affected health-care facilities still suffered from stock-outs, with a median duration of 45 days reported for these stock-outs [43]. Although persistent staff shortages contributed to the underperformance of the depot, blame was also later attributed to insufficient ordering practices. Some health-care facilities served by the depot reported that the quantities of drugs delivered to them were significantly smaller than the quantities they had ordered. Orders were, furthermore, often fulfilled only after considerable delays. Staff members at the Mthatha depot, on the other hand, claimed that many orders

received from health-care facilities exceeded the perceived need of ordering. As a result, the depot management would make its own adjustments as to the ordered quantities with the goal of preserving stock at the depot. The strike was, however, not the only tragedy to befall the Eastern Cape province. Between May and November 2011, 19 tons of medical drugs stored across two depots and 92 health-care facilities in the Eastern Cape, were destroyed after it had either expired or reported to have been ‘tampered with’ [11].

A study amongst 31 government clinics in the Tshwane Health District furthermore revealed that each facility in this district had experienced stock-outs of eleven different vaccines at least once during the 2013 calendar year [117]. In some cases, the stock-outs lasted for longer than two weeks. And a further 16% of these clinics reported that the delivery lead times for emergency orders exceeded one week [117]. Although many factors conspired to cause these stock-outs, some of the more prominent causes involved insufficient inventory management practices. For example, the majority of the clinics employed stock cards to record inventory levels (as opposed to computerised inventory management systems). Results showed that only 52% of the clinics managed to record inventory levels accurately according to these stock-card systems. Stock cards are, however, outdated and do not allow for the possibility to integrate seamlessly with other supply chain technologies. Moreover, with stock cards it is impossible to track inventory levels continuously.

Another study by Mokheseng *et al.* [114] investigated the management of ARV drug inventories at a district hospital and its peripheral clinics in the QwaQwa district of the Free State. The results showed that the hospital had frequently experienced stock-outs and that these typically lasted between one and three months. This phenomenon was largely attributed to the fact that the district hospital often received either incorrect quantities or stock with short expiry dates from its supplier. The study also found that neither the hospital nor the peripheral clinics employed uniform (or effective) ordering policies.

Interestingly, a 2014 national audit found that only 20% of reported stock-out incidents in South Africa were caused by manufacturing issues. The remaining 80% were attributed to poor inventory management practices (specifically related to order quantities and forecasting practices) at medicine depots and clinics on both provincial and district levels [12]. Although many research studies (such as those described above) have shed light on the problems in the South African pharmaceutical supply chain, it is important to recognise that the statistics do not necessarily provide the complete picture, since many stock-outs and their impacts go by unreported [11].

Arguably one of the most significant challenges faced by the South African public pharmaceutical supply chain is the expansion of stock level visibility across the entire supply chain. The current lack of visibility has also partially been ascribed to the fragmented nature of the country’s ARV therapy supply chain (a common occurrence in many developing countries, as mentioned in §2.9.1) [42]. The most prominent intervention towards enhancing supply chain visibility involves the implementation of the SVS at public health-care clinics, as discussed in §1.1. The SVS mobile application allows health-care facility staff to report the current stock level, the expiry date, the quantity of stock received and the number of units lost due to expiration since the last logged update, for each stock item in the facility [60]. Although implementation of the SVS technology has delivered mixed results since its deployment, it is reported to have helped reduce the number of stock-outs and incidents of overstocking for health-care facilities across the country [60].

Although the SVS system was launched as a pilot project in KwaZulu Natal in 2013, very few formal evaluation studies on the use of the system have been conducted since. An evaluation report, discussing the implementation of the SVS in two districts in KwaZulu Natal, was finally

published in 2018 and highlighted concerns associated with the implementation of the system in these districts [58]. One key concern raised was that SVS users were not sufficiently trained and skilled with respect to using the SVS mobile application successfully. And in the most cases where training was provided, the training sessions reportedly emphasised the physical use of the SVS mobile application and overlooked the importance of the actual use and interpretation of the information collected. And in terms of measuring the success of the SVS intervention, a tendency to emphasise reporting compliance as opposed to stock management was also observed. In other words, users of the SVS were more focused on submitting their reports in time, irrespective of the accuracy contained within these reports. Furthermore, the responsibility for implementing the SVS often fell on nursing staff experiencing an already significant workload. Clinic staff reported that they typically prioritised patient care, which came at the expense of their SVS-reporting responsibilities. Another worrying concern highlighted by the report was that stock level data were also not necessarily captured accurately, which rendered the data unreliable for use further upstream by decision-makers. The use of the SVS was also hampered by poor mobile network connectivity which compromised attempts to report stock levels using the mobile application. The evaluation report solemnly concluded that the SVS data contained multiple errors to such an extent that the impact of the SVS on stock management in this particular case could not be determined with sufficient accuracy.

The discussion above is testament to the fact that the South African pharmaceutical supply chain is pursuing the noble goal of increased supply chain visibility, but is mostly hindered by implementation issues. Since many instances of data captured within the SVS system are inaccurate or incomplete, it is conjectured that the true potential impact of proper supply chain visibility still remains unproven in the South African public health-care context.

## 2.10 Chapter summary

A brief overview of some of the most salient characteristics related to supply chain management was provided in this chapter. The notion of supply chain management, with a particular focus on supply chain strategies, or processes, was first introduced. The infamous bullwhip effect, which is renowned for its potential to wreak havoc in supply chains, was reviewed next. In order to motivate the case for supply chain information sharing, this particular business practice was also afforded considerable attention. This was followed by a brief review of the budding demand-driven supply chain management philosophy. A concise overview of the importance of collaboration in a supply chain was also presented. This was followed by a brief discussion on inventory management — a business practice integral to supply chain management. Next, guidelines and measures for measuring supply chain performance were discussed. Finally, a discussion was provided on pharmaceutical supply chain management in developing countries, with a focus on the South African context.





---



---

## CHAPTER 3

---

# Computer simulation and agent-based modelling

### Contents

3.1	An introduction to computer simulation modelling . . . . .	44
3.2	Basic simulation modelling concepts . . . . .	45
3.3	Prevailing simulation modelling paradigms . . . . .	46
3.3.1	<i>Discrete-event modelling</i> . . . . .	46
3.3.2	<i>System dynamics modelling</i> . . . . .	47
3.3.3	<i>Agent-based modelling</i> . . . . .	47
3.3.4	<i>Dynamic systems modelling</i> . . . . .	47
3.4	Typical steps in a sound simulation study . . . . .	47
3.5	Simulation input modelling . . . . .	49
3.6	Verification and validation of a simulation model . . . . .	50
3.6.1	<i>Model verification</i> . . . . .	50
3.6.2	<i>Model validation</i> . . . . .	51
3.7	Developing an agent-based model . . . . .	52
3.7.1	<i>Definition of an agent</i> . . . . .	52
3.7.2	<i>Designing an agent-based model</i> . . . . .	53
3.7.3	<i>Agent-based modelling of supply chains</i> . . . . .	54
3.8	Chapter summary . . . . .	56

This chapter is devoted to a brief introduction to the broad field of computer simulation modelling, starting with a general description of simulation as well as some benefits and drawbacks of the discipline in §3.1. This is followed by a brief overview of some important simulation modelling concepts in §3.2. Four predominant simulation modelling paradigms are next reviewed in §3.3 and a typical twelve-step procedure for conducting a sound simulation study is briefly reviewed in §3.4. A brief description of the simulation input modelling process is next provided §3.5. The focus shifts in §3.6 to a brief discussion on the critical steps of simulation model verification and validation. Finally, some specific considerations pertaining to the agent-based modelling of supply chains are provided in §3.7. The chapter closes with a brief summary in §3.8.

### 3.1 An introduction to computer simulation modelling

Numerous definitions of *simulation* abound in the literature, but arguably the most popular one is that of Banks [6] who defined simulation so eloquently as the

“...imitation of the operation of a real-world process or system over time.”

The implication of this definition is two-fold. First, simulation involves a replica of some real-world system in a shape or form other than the actual system itself. Secondly, simulation involves the progress of time — a simulation model is a dynamic construct capable of emulating the operation of a real-world process as it evolves over time. A *simulation model* may be described more formally as a set of assumptions that describe the operating characteristics of a real-world system in terms of mathematical, logical and symbolic relationships between the entities of the system [7, 162].

The most basic of simulations can be performed by hand. A conventional die may, for example, be used to generate an artificial sequence of service times if it is assumed that these service times are uniformly distributed between one and six time units. While simulation *via* pen and paper may prove satisfactory in some rudimentary cases, the shortcomings of such an approach are plain to see. Fortunately, the meteoric rise of computing power has made it possible to construct simulation models of much larger and more complex systems with the aid of a computer. Computer simulation entails the use of a computer to both develop a simulation model and to conduct simulation experiments with this model. The overarching purpose of simulation can best be summarised as it being a tool for studying and analysing the behaviour of a system that may or may not exist in practice [6]. This ability to simulate conceptual systems renders simulation a valuable tool for the design of new systems [6, 7]. Essentially, any simulation model predicts the expected behaviour of a particular system for a given set of inputs. With a simulation model at his or her disposal, the simulation user can ask what-if questions so as to evaluate the effect of particular inputs on model outputs [6].

A highly celebrated feature of simulation is its ascendancy over traditional analytic problem-solving techniques. Analytic methods often lack the flexibility to accommodate the intricacies and stochastic elements present in most real-world systems. For this reason, analytic approaches often fall short in their attempts to represent real-world systems. Simulation is, however, free of any such limitations or shortcomings and is therefore often the only viable instrument for modelling complex and stochastic systems at an appropriate level of realism [162].

Simulation may be used to study the effects of certain changes on a system without perturbing the physical system itself [6]. By implication, this also means that simulation can be used to evaluate the effects of any decision extensively, without the need for committing manpower or financial resources to the actual system. Since a simulation model can be operated in a risk-free space, simulation is a suitable vehicle for evaluating the effects of proposed system changes, such as new policies or plant layouts [93]. This freedom to experiment allows simulation to facilitate the rapid design of sound and robust systems [108]. The flexible nature of simulation models also make them ideal platforms for experimenting with systems whose behaviour is only partially known, or perhaps not known at all. Simulation may, therefore, be employed to test a particular hypothesis [108].

Simulation also allows for improved control over experimental conditions, something that is extremely difficult, if not impossible, to control in the real world. Furthermore, the power of the computer makes it possible to compress and expand time in a simulation model almost effortlessly [6, 93]. The compression of time makes it possible to simulate lengthy periods of

simulated activity within a short period of real time. Likewise, almost instantaneous simulated events can be studied in expanded real time.

The process of simulation often provides insight into the particular behaviour of a system in a way that is not necessarily achievable otherwise. With access to a complete system representation in the form of a simulation model, it is often possible to infer reasons for particular system behaviour [6]. Similarly, simulation presents a convenient medium for identifying problems within a given system [6]. Many real-world systems are so complex that it is nearly impossible to comprehend the effects of the interactions amongst system components. Simulation, however, makes it possible to study phenomena that occur almost simultaneously in a sequential fashion. Another prominent benefit of simulation lies within its visualisation capabilities. Animation makes it possible to inspect a system (often containing three-dimensional objects) from a multitude of angles, something that is not possible with two-dimensional drawings [6].

Although simulation is a powerful problem-solving tool with several benefits, it also possesses some drawbacks. A simulation model with random elements can only yield approximate values of simulation output variables [93]. For this reason, many potentially time-consuming simulation runs have to be performed in order to derive appropriate estimates. The very presence of randomness also makes it difficult to distinguish between results caused by randomness and those caused by implicit interrelationships in a system [6]. Apart from lengthy simulation runs, an entire simulation study is also typically costly and time-consuming [7]. Large costs are attributed to the need for skilled model builders. And even when simulation runs can be performed relatively quickly, it is the collection of input data and the analysis of output data that may consume valuable time [136].

A notable trait of simulation is that any given simulation model is typically either fully practical or not at all. Any simulation model that is not a true, valid representation of the system modelled cannot be trusted to deliver reliable results [93]. In other words, even the most sophisticated simulation model cannot compensate for invalid assumptions or inaccurate input data.

## 3.2 Basic simulation modelling concepts

While there are several different modelling paradigms within the realm of simulation, they all share some fundamental concepts that serve as the building blocks of any simulation model. These key components include the *system*, *model*, *system state*, *entities*, *attributes*, *events*, *activities* and *system state variables*. A brief review of these elements is provided in this section.

Any simulation model is a model of some real-world system. Such a *system* encompasses a set of objects, or entities, that interact and cooperate in order to achieve a particular goal [7]. A *model*, then, is an abstraction or representation of a system and expresses the inter-dependencies between a system's components by means of structural, mathematical or logical relationships [7].

*System state variables* represent a collection of all the information required to describe the situation in a system sufficiently accurately at any given time [6]. Collectively, the system state variables define the particular *state* of a system. Any instantaneous occurrence that can potentially change the state of a system is known as an *event* [7].

An *entity* is any object within, or component of, a system that can change the state of a system [76]. Entities are typically characterised by the fact they require 'explicit' definition in a model [6, 7]. People, vehicles and machines are simple examples of entities. Entities have *attributes* that describe information specific to them. While entities of the same type share the same attributes, it is the particular values of these attributes that distinguish entities from one another. Attributes define the behaviour and performance of entities in a simulation model [76].

An *activity* is any process or logic in a simulation that occurs over a period of time for which the duration is known when the activity starts [6, 7]. The duration of an activity may be deterministic, stochastic or even specified as user input. A *delay*, on the other hand, causes the movement of an entity to be delayed for an indefinite period of time [6, 7]. A delay is only terminated once some future condition is satisfied. A *resource* is a special type of entity that has a limited capacity and can provide a service to one or more entities [6, 159].

Simulation models are generally characterised along three dimensions [7, 93]:

**Static or dynamic.** A *static* simulation model is independent of time. A static simulation model either represents a system at a particular time instant or a system on which the passage of time has no influence. The throw of a conventional die is an example of a static simulation. A *dynamic* simulation model, on the other hand, is a representation of a system that changes dynamically with the passage of time. A simulation of vehicles travelling along a highway is an example of a dynamic simulation.

**Deterministic or stochastic.** A *deterministic* simulation model does not contain any random (probabilistic) variables. A specific set of inputs to a deterministic model will always yield the same output if the simulation model were to be re-executed. A simulation model that contains at least one probabilistic element, on the other hand, is classified as a *stochastic* model. Since the inputs to a stochastic model are random, the output of such a model is also random.

**Discrete or continuous.** The manner in which system state changes manifest themselves over time determines whether a model is classified as either discrete or continuous. In a *discrete* simulation model, the state variables change instantaneously at (possibly random) distinct, separate points in time. In a *continuous* system, on the other hand, the state variables change continuously over time. Although many systems today comprise both discrete and continuous elements, one or the other type typically predominates to such a degree that a system is usually classified as either one of the two [7, 93].

### 3.3 Prevailing simulation modelling paradigms

The concept of *abstraction* is often used in a simulation context to refer to the level of detail incorporated into a simulation model. The required level of abstraction in a simulation model typically governs the choice of simulation modelling paradigm. At present, the four predominant simulation modelling paradigms are *discrete-event modelling*, *system dynamics modelling*, *agent-based modelling* and *dynamic systems modelling* [21].

#### 3.3.1 Discrete-event modelling

The most prominent hallmark of discrete-event simulation is that the state of the system changes instantaneously only at discrete, but possibly random, points in time [108, 130]. Discrete-event modelling therefore places a strong emphasis on the notion of *events*, because events are the instantaneous occurrences that change the state of the system [6]. In discrete-event modelling, the state of the system remains unchanged between two consecutive events. A basic example of a discrete-event simulation modelling context is that of a car park, where vehicles enter or exit the system at discrete points in time.

### 3.3.2 System dynamics modelling

Coyle [35] defined *system dynamics* as follows:

“System dynamics deals with the time-dependent behaviour of managed systems with the aim of describing the system and understanding, through qualitative and quantitative models, how information feedback governs its behaviour, and designing robust information feedback structures and control policies through simulation and optimisation.”

The modelling paradigm of system dynamics is oriented towards strategic modelling and, in particular, the design and evaluation of policies within the realm of complex systems [144]. Much of system dynamics modelling revolves around the notion of information feedback loops. Since real-world decision-making processes are based on feedback loops, they are used extensively to model system behaviours. There are two central concepts in system dynamics modelling: *Causal loop diagrams*, and *stocks* and *flows* [144]. Causal loop diagrams capture feedback processes as well as the causal influences of variables on one another. A *stock* represents any object or item that can be accumulated or depleted over time. Stocks can be measured and therefore used to describe the state of a system. A *flow*, on the other hand, is any mechanism that changes the value of a stock over time. Through the use of these modelling structures, system dynamics embraces a higher level of abstraction, focussing on aggregates and not on individual entities and their respective characteristics.

### 3.3.3 Agent-based modelling

Autonomous decision-making entities, called *agents*, form the cornerstone of *agent-based modelling*. In agent-based modelling, the focus is on modelling the individual behaviour of agents as opposed to modelling system behaviour [124]. The individual behaviour of agents, as well as their interactions with other agents, are governed by rule sets specified by the simulation modeller. This approach makes it possible to inspect a system’s emergent behaviour arising from the interactions between multiple agents [106]. Agent-based modelling is often superior to system dynamics and discrete-event modelling in terms of capturing complex real-world phenomena [21].

### 3.3.4 Dynamic systems modelling

Dynamic systems modelling is often described as the precursor to system dynamics modelling and was purposefully developed for design cycles in technical engineering disciplines such as mechanical, electrical and chemical engineering [21]. A dynamic systems model typically contains a number of state variables and algebraic differential equations involving these variables. As opposed to system dynamics, these variables and equations do not represent accumulations of entities or objects, but carry direct physical meaning, such as location or volume [21].

## 3.4 Typical steps in a sound simulation study

Another celebrated concept in the simulation literature is that of the typical steps suggested for conducting a sound simulation study. Many simulation scholars have contributed to this particular stream of the literature on simulation with their own adaptations of, or introduction

of new, methodologies. The twelve-step procedure for carrying out a proper simulation study recommended by Banks [6] forms the basis of the discussion in this section.

1. *Problem formulation.* The foundation of any simulation study rests on a sound problem formulation in which the problem of interest is clearly stated by the decision-maker (or client) [93, 91]. In the case where a problem is not yet fully understood at the start of a simulation study, the problem may be reformulated at a later stage as more information becomes available.
2. *Setting of objectives and an overall project plan.* The objectives of a simulation study are typically determined by the specific questions that should be answered by the simulation. These questions should be specific in order to determine the required level of model detail [91]. The project plan should also outline the scope of the simulation model, key performance indicators that will be employed to measure system performance as well as the time, manpower and monetary resources required.
3. *Model conceptualisation.* The significance of this step is often underestimated [90]. A conceptual model is simply an abstraction of the system under consideration and is typically described in a form that is not software-specific. The proposed operation of the simulation may be documented in either a graphical form (block diagram or process flow chart) or pseudo-code form [136]. The purpose of a conceptual model is to identify and clarify model input, model logic and the simplifying assumptions [125].
4. *Data collection.* Information and data related to the structure and operating characteristics of the system are collected. Data are especially important for model validation and the specification of model parameters and probability distributions [93]. Data collection may be an extremely arduous and time-consuming task and is often executed in parallel with other steps.
5. *Model translation.* The conceptual model of Step 3 is converted to a computerised form, typically with the aid of a dedicated simulation software package.
6. *Model verification.* The purpose of model verification is to evaluate whether the computerised simulation model functions properly. This is often achieved through the process of debugging. It is highly recommended that model verification is performed continuously during the model building phase.
7. *Model validation.* The aim of model validation is to ascertain whether the simulation model is an accurate and reliable representation of the real-world system under consideration [136].
8. *Experimental design.* All the required simulation experiments are stipulated. For each experimental configuration, decisions are made with respect to the length of the warm-up period, model starting conditions, the required length of the simulation runs and the required number of replications [90, 108, 136].
9. *Production runs and analysis.* The simulation experiments, as designed in Step 8, are performed and the results are analysed statistically to compare the model outputs of the respective scenarios modelled.
10. *Additional runs.* Based on the analysis of the results in Step 9, the simulation analyst decides whether any additional or different simulation experiments are required.

11. *Documentation and reporting.* The conceptual model, a thorough description of the computer implementation and the results of the study are documented (usually in a single report) for current and future use [93]. Model documentation is especially important when the simulation model will be used again in the future. Rigorous reporting of a simulation model's features may also help different analysts to familiarise themselves with the working of the model.
12. *Implementation.* In this final step of a simulation study, the simulation analyst provides recommendations for possible improvements to the system of interest. This step is typically accompanied by the handover of the report compiled in Step 11 to the client so that he or she can use it for decision support. It is, however, not a given that the client (or decision-maker) will implement any of these recommendations. The probability of the suggestions being implemented is contingent on the success of the previous eleven steps.

### 3.5 Simulation input modelling

Simulation input modelling involves the process of selecting probability distributions to represent stochastic processes within a simulation model [6, 19]. In a supply chain simulation model, for example, input data may include probability distributions for end-user demand and for delivery lead times (given that they are random variables). Since it is often impossible to derive an exact probability distribution for any given stochastic input, the process of simulation input modelling is aimed at obtaining approximations that reflect the real-world process at least sufficiently accurately [18].

The practice of simulation input modelling can be divided into two broad classes, based on either the availability of, or the absence of, real-world data [18]. When real-world data are available, probability distributions are fitted to the available data in order to obtain approximations. When such data are not available, on the other hand, an input model is constructed based on any other available information. Banks *et al.* [7] described four steps that should be followed in the simulation input modelling process when data are available. During the first step, data are collected from the real system (this is Step 4 in the simulation procedure described in §3.4). Thereafter, a probability distribution is fitted to the data. The type of probability distribution is typically selected based on a frequency distribution, or histogram, of the original data. Today, many commercial input modelling software packages are available to perform this step with relative ease. After the probability distribution has been selected, the parameters of the particular distribution may be estimated from the data. Finally, *goodness-of-fit* tests are performed to establish whether or not the chosen distribution is a good approximation of the data. The *chi-square* test [111] and the *Kolmogorov-Smirnov* test [100] are two popular instances of goodness-of-fit tests.

It is possible, however, that a standard theoretical distribution cannot be fitted to a given data set. This may happen when the data collected are from two or more heterogeneous populations, or when the data values have been rounded significantly [92]. In this case, an empirical distribution may be constructed from the data and used as an approximation. An empirical distribution is, however, based purely on the observed data and, if the sample size is significantly small, the resulting empirical distribution may be a misrepresentation of the actual system. Since extremely unusual events do not occur very frequently in practice, for example, it may happen that they are not appropriately represented in the data sample. An empirical distribution that does not include the probability of such extreme events therefore does not represent the risks of a system sufficiently accurately [19]. Another shortcoming of using an empirical distribution is that it

cannot generate values outside of the range of the observed data values [92]. Furthermore, an empirical distribution also requires more storage space than its compact, theoretical distribution counterpart [93]. For an empirical distribution derived from  $n$  data values,  $2n$  values (the data and their corresponding cumulative probabilities) should be stored in the computer memory.

It is not always possible to collect data when, for example, time is limited, the input process does not yet exist or when the collection of data is prohibited [7]. In such a case, any other relevant information may be used to construct an input model in order to approximate the real process. Examples of such information include the opinions of subject-matter experts, the physical and conventional limits of a process, and the nature of the process itself [18]. Since subject-matter experts often have considerable experience of a process, they can typically provide reliable estimates of the most optimistic and pessimistic values, as well as the most likely values, of the data under consideration [7]. These estimates may then be used to construct an appropriate input model. The uniform, triangular and beta distributions are typically used as input models in the absence of data [7]. The triangular distribution is often preferred in the absence of data because it places the bulk of the probability at the most likely value and much less at the upper and lower bounds, respectively.

## 3.6 Verification and validation of a simulation model

Arguably the most critical tasks in a simulation study are that of simulation model verification and validation. Verification involves the process of confirming whether a conceptual simulation model is implemented correctly in a software environment [7]. The purpose of verification, in other words, is to determine whether a simulation model has been built correctly. The process of validation, on the other hand, is aimed at establishing whether the right model has been built [7]. The ‘right’ model, by implication, is a simulation model that is an accurate and reliable reflection of the real-world system modelled. Verification and validation are iterative processes that are performed concurrently with model building.

### 3.6.1 Model verification

Simulation model verification is a well-researched topic and there are multiple techniques that may be employed to verify a model. Banks [6], for example, recommended a series of seven steps for the verification of a simulation model. The first step involves the application of structured programming principles during the model building phase. This is achieved through proper planning of the simulation model prior to the actual programming phase. The model development phase may also be simplified with the aid of adopting a modular approach (*i.e.* dividing a simulation model into subcomponents) [84]. The second verification technique is also related to programming practice and includes the extensive use of comments in the computer code. Comments facilitate an increased understanding of the code by both the developer and other parties not involved with the original model development. It is important that detailed descriptions of variables and code sections are provided [7]. Next, a simulation model may also be verified by allowing another person to scrutinise the computer code to identify potential errors [7].

The fourth verification technique proposed by Banks [6] is to ensure the correct use of input data values. Units of input variables should, for example, be used consistently throughout the model. Furthermore, a simulation model may also be verified by inspecting the reasonableness of output values when model inputs are varied. The sixth verification technique involves the use of a debugger to detect and rectify programming errors. Finally, the use of animation can



be of great value to identify model or programming flaws that may be more difficult to detect otherwise [84].

Banks *et al.* [7] have also suggested some verification measures in addition to those discussed above. A logic flow diagram may, for example, be used to evaluate the model logic by documenting each logically possible action in the underlying system in this diagram. Another verification technique may be to experiment with a variety of settings of the input parameters and to then evaluate the model output for soundness. It is also recommended that the input parameter values should be printed out at the end of a simulation run to ensure that these parameter values have not been changed accidentally. Kleijnen [84] also proposed calculation of some simulation results by hand and to compare these results with the actual outputs of the simulation model as another means for verification.

### 3.6.2 Model validation

The purpose of validation is to ensure that a model is such an accurate representation of the real-world system that it can effectively substitute the physical system and exhibit the same behaviour as the actual system would have exhibited under the same circumstances [7]. A properly validated model is also likely to instil increased confidence in the credibility of a model [7]. Validation is an iterative process during which differences between the model and underlying system behaviour are continually evaluated in order to improve the model. This process is known as *calibration* and is continued until a model is considered as sufficiently adequate.

For any model to be declared valid, it has to satisfy three validity requirements, namely *conceptual validity*, *operational validity* and *credibility* [93]. Conceptual validation involves the process of establishing whether the conceptual model is an appropriate representation of the real-world system under consideration [129]. The most prominent conceptual validation techniques include *face validation* and *traces*. A model has face validity when its behaviour is considered consistent with the operating characteristics of the real system [91]. The prospective users of a simulation model and people with expertise in respect of the real-world system are typically employed to establish face validity [7]. Traces, on the other hand, are employed to rigorously track the movement of entities through a simulation model in order to establish whether the overall model logic is correct. Sensitivity analyses are also often performed to confirm face validity [7, 84]. When (radical) changes are made to particular input variables, people knowledgeable about the real system should be able to predict the direction of change in model output with a reasonable degree of certainty.

Operational validation, on the other hand, pertains to the comparison of model output data and the actual system's behavioural data to determine whether they are comparable. Law and Kelton [93] calls this process *results validation* and it can only be performed when real-world data are available. Operational validity typically constitutes a large array of statistical tests that may be employed to determine whether model output data differ significantly from the real-world system's data. It is also possible to provide historical data as input with an expectation that the model will yield similar results (within acceptable statistical error) to those observed in the real system [6, 91]. When probability distributions are assumed for input variables, it is imperative to validate these distributions by applying appropriate goodness-of-fit tests. When statistical analyses are not possible, Turing tests may be performed to validate a model [7, 84]. A Turing test involves instances of both real-world system output as well as simulation output. When an expert cannot distinguish between the real and simulated results, a model is deemed valid. Another popular validation technique is an *extreme condition test*, where extreme values for input variables are chosen in order to ascertain whether the model output changes correspondingly as expected.

Finally, the credibility of a simulation model is deemed appropriate when the user or decision-maker unconditionally accepts the model as correct [93]. Model credibility is a reflection of the confidence that (potential) users have in the working and results of a model [91, 129].

## 3.7 Developing an agent-based model

The aim in this section is to elaborate on the paradigm of agent-based modelling (introduced in §3.3.3) by describing the characteristics of an agent as well as guidelines for developing an agent-based model. Finally, a motivation is provided for modelling supply chains within an agent-based modelling paradigm.

### 3.7.1 Definition of an agent

The notion of an *agent* is central to agent-based modelling [103], as discussed in §3.3.3. Although there are several definitions of the concept of an agent available in the literature, arguably the most popular description thereof was provided by Wooldridge and Jennings [163]. They proposed that the term *agent* may be used to identify any object or computer system that possesses the following four characteristics:

1. *Autonomy*. Any agent should have some measure of control over its own actions and work without any explicit human intervention. The ability of an agent to act autonomously and self-directed is arguably its most prominent property [105, 106].
2. *Social ability*. An agent should be able to communicate and interact with other agents (and possibly with humans as well). The dynamic interactions with other agents typically influence aspects of an agent's behaviour, such as protocols for movement and communication [106].
3. *Reactivity*. This property stipulates that agents should be able to perceive their environment and that they should be able to respond to changes in their environment appropriately.
4. *Pro-activeness*. According to this characteristic, an agent is not restricted to acting solely in response to a change in its environment. Instead, an agent should be pro-active — that is, able to seize the initiative as a part of some goal-directed behaviour.

Macal and North [106] also identified four fundamental characteristics of an agent and these include the aforementioned properties of autonomy and sociability. In addition, they described an agent as *self-contained* and *modular*. By implication, an agent is individualised and has a clear boundary that distinguishes it from other agents. An agent also assumes a so-called *state* that captures all of the relevant variables that describe its current situation in its environment. The behaviour of an agent is also influenced significantly by its current state. Macal and North [106] further described three potential, but not compulsory, features of an agent:

1. *Adaptation*. An agent may have the ability to adapt and modify its behaviour based on particular rules. This adaptiveness may be facilitated by a learning ability where an agent can learn new behaviour based on previous experiences [20, 119].

2. *Goal-directedness.* An agent may pursue certain goals in respect of its behaviour. This goal-directed behaviour allows an agent to continually evaluate the outcomes of its actions and to adapt its behaviour according to its goals.
3. *Heterogeneity.* Agent-based modelling provides a unique opportunity to model a population of heterogeneous agents. This capacity makes it possible to endow agents with properties and behaviours of which the degree and complexity may vary among different agents.

Since agents are autonomous and adaptive, they often exhibit self-organising behaviour. When individual agents and their unique behaviours are modelled according to the paradigm of agent-based modelling, patterns and structures which were not explicitly programmed into a model, may often emerge at a higher level over time. The notions of self-organisation and emergence (discussed in §1.1) may, therefore, often materialise within agent-based models [20, 106, 104, 119]. According to Serugendo *et al.* [134], self-organisation may be based on the abilities of agents to adapt their own behaviours dynamically according to some *reinforcement*. A particular behaviour may, for example, be reinforced when rewards are received for actions associated with this behaviour. Undesirable actions, on the other hand, may be met with punishments that will discourage repetition of the same behaviour. Given the properties of an agent, as discussed above, and its ability to learn based on reinforcement, it would seem that agent-based modelling is well-suited to the machine learning paradigm of *reinforcement learning* [106].

### 3.7.2 Designing an agent-based model

Macal and North [106] recommended a series of seven questions that should be answered in order to develop a sound preliminary design of an agent-based model:

1. *What particular problem needs to be solved by the model?* This question involves the specific questions that should be answered by the model. It is also important to motivate the use of agent-based modelling over other modelling paradigms.
2. *Who or what should be the agents in the model?* Agents are typically identified as the entities in a model with decision-making abilities and explicit behaviours.
3. *In what environment do these agents reside?* The nature of agents' interactions with the environment has to be stipulated. It is also important to clarify whether or not the movement through space of agents is of any significance.
4. *What are the relevant agent behaviours?* Agent behaviours involve the decisions that they make, what actions they perform and the particular behaviours that they exhibit.
5. *How do agents interact with one another and their environment?* This question relates to the nature of agent interactions: Are they extensive or localised?
6. *Where can the data for the model be obtained?* The importance of data collection was mentioned briefly in §3.4.
7. *How will the model be validated?* Guidelines for model validation were discussed in §3.6.2. An important consideration for agent-based models is the manner in which agent behaviours will be validated.

### 3.7.3 Agent-based modelling of supply chains

Given the multifaceted nature of supply chain management, it is possible to model various aspects of it within the modelling paradigms of either system dynamics, discrete-event or agent-based modelling. Whereas system dynamics allows a modeller to focus almost exclusively on strategic or high-level supply chain decisions, discrete-event simulation makes it possible to develop more detailed and more realistic models equipped with powerful analytical capabilities [168]. The rise of agent-based modelling has, however, increased interest in modelling supply chains within this paradigm because of the complexity often encountered in supply chains.

Any given supply chain may become so large and complex that it is extremely difficult, if not impossible, to manage and control it effectively [146]. Moreover, due to persistent changes in organisational and market trends, a supply chain should be dynamic, scalable, agile and adaptive in order to deliver the best possible performance [146]. These characteristics are reminiscent of that of an agent, as described in §3.7.1, and suggest that supply chain entities (such as manufacturers, warehouses and retailers) may potentially be modelled as agents within an agent-based modelling paradigm [29]. Agent-based modelling is considered a suitable modelling approach toward capturing the complexity that arises in a supply chain as a result of these interactions between autonomous entities [158]. Considering that supply chains are often decentralised systems where members tend to act independently and in their own interests, the case for an agent-based approach is reinforced [29].

A supply chain may also be seen as an emergent phenomenon resulting from the self-organising behaviour of its constituent entities (or agents) [31, 146]. In a supply chain, there is no single entity that deliberately controls or organises the functioning of an entire supply chain network. Instead, it is only through localised decision-making behaviour that entities give rise to the emergent structure known as a supply chain. For this reason supply chains are often perceived as complex adaptive systems. According to Fox *et al.* [50], an agent-based model of a supply chain may include the following features:

1. *Distributed.* The respective functions of supply chain management are divided among a set of agents.
2. *Dynamic.* Each agent acts in an asynchronous manner, as required.
3. *Intelligent.* Each agent is considered an ‘expert’ in its function and may draw on artificial intelligence and operations research techniques to solve problems.
4. *Integrated.* Agents are aware of one another and may access the functional capabilities of others.
5. *Responsive.* Each agent may ask another agent for information or a decision.
6. *Reactive.* Agent behaviour is malleable and agents can adjust their behaviour based on events in their environment.
7. *Cooperative.* Agents do not necessarily work independently. Instead, agents can cooperate with one another in the pursuit of a solution to a greater problem.
8. *Complete.* The functional capabilities of all the agents should sufficiently capture the entire range of functions needed to manage the supply chain.

A review of the literature revealed a number of studies that have adopted agent-based modelling in attempts to model supply chain management functions. One of the most prominent examples

is that of Swaminathan *et al.* [148], who developed a multi-agent framework for developing a sound model of a supply chain. All of the components in this framework are stratified into two distinct categories, namely *structural elements* and *control elements*. The structural elements, modelled as agents, are involved in the actual manufacturing and transportation of goods. Control elements, on the other hand, are policies that are employed to coordinate the flow of goods in the supply chain.

The structural elements can further be classified as either *production agents* or *transportation agents*. Production agents are directly involved with the management of inventory and common examples include retailers, distribution centres and manufacturing plants. A transportation agent, or rather transportation vehicle, is responsible for the movement of products from one production agent to another. The framework also includes a customer agent that generates the demand for finished products in the supply chain.

The control elements in the framework of Swaminathan *et al.* [148] involve the protocols that facilitate the manufacturing and distribution of products. The first type of control element is *inventory control* and this element prescribes the nature of inventory replenishment in a supply chain. Secondly, the *demand control* element captures the marketing and forecasting strategies employed in a supply chain. The *supply control* component typically involves supply contracts that stipulate the terms and requirements for the delivery of materials once an order has been placed. Next, the *flow control* element defines the nature of loading and unloading operations as well as vehicle routing for the delivery sequence of products. Finally, the *information control* element controls the timing and contents of information flows within the supply chain environment.

Van der Zee and Van der Vorst [152] were, however, critical of the work of Swaminathan *et al.* [148] because their discussion on the control elements is fairly limited. Apart from only introducing the concept of control elements, their framework neither elaborates on which entities are responsible for control, nor on the nature of the relationships between these entities (*i.e.* how they collaborate) as well as on the timing of these control activities.

Julka *et al.* [78] proposed an agent-based management framework for the modelling and monitoring of supply chains. The framework combines several supply chain elements, such as enterprises, their business processes as well as relevant business data and knowledge, with the aim of combining them all in a unified structure. Supply chain entities are modelled as agents while material and information flows are represented by objects. A so-called *enterprise agent* represents any production facility that produces a set of products from raw materials. An enterprise agent may have one or more sub-agents that are responsible for different tasks internal to the entity. A *sales agent* is, for example, responsible for the receipt and processing of orders. When an enterprise agent receives a request for a quotation, it forwards the message to the sales agent. The sales agent, in turn, communicates the order information with the warehouse and production agents in order to establish whether or not the demand can be fulfilled. A *warehouse agent* replenishes raw material inventory through ordering from other suppliers, while a *production agent* controls the production lines of the enterprise. An advantage of this modular approach to supply chain modelling is that the framework can accommodate different supply chain systems with relative ease. The value of the framework as a decision support tool for supply chain management was demonstrated by means of two case studies within a petrochemical context.

A common inventory problem faced by many vendors involves two conflicting objectives: A vendor has to keep his or her inventory costs to a minimum, whilst maintaining enough stock so that a satisfactory customer service level may be achieved. Franklin [52] demonstrated how agent-based modelling may be adopted to solve this multi-objective optimisation problem. He proposed an agent-based simulation model comprising two agents, namely a sales manager and

an inventory manager. Each of these two agents represents one of the aforementioned conflicting objectives. The sales agent aims to maximise the vendor's service level while the inventory manager attempts to minimise inventory costs. Through the inclusion of concepts such as satisfaction indexes, aggression factors and recollection abilities, the two agents are allowed to negotiate with one another in order to find optimal reorder policies for a vendor. Apart from solving the multi-objective inventory problem at hand, Franklin [52] also illustrated a unique ability of agent-based modelling to include cognitive skills and capacities.

### 3.8 Chapter summary

A brief overview of the literature related to computer simulation and agent-based modelling was provided in this chapter. A general introduction to the discipline and certain basic modelling concepts were first presented. Four predominant simulation modelling paradigms were reviewed next and this was followed by a description of a twelve-step approach recommended for conducting a sound simulation study. An overview of the steps included in the simulation input modelling process was next provided. The critical activities of model verification and validation were also described. Some guidelines for designing agent-based models and, in particular, agent-based supply chain models were finally discussed.

---



---

## CHAPTER 4

---

# Reinforcement learning

### Contents

4.1	An introduction to machine learning . . . . .	57
4.2	Reinforcement learning . . . . .	58
4.2.1	<i>Evaluative feedback</i> . . . . .	60
4.2.2	<i>Formulation of the reinforcement learning problem</i> . . . . .	62
4.2.3	<i>Reinforcement learning solution approaches</i> . . . . .	67
4.3	Chapter summary . . . . .	70

This chapter is devoted to a brief overview of reinforcement learning, a subdiscipline of machine learning. In §4.1, a general introduction to the notion of machine learning is provided. This is followed by a more extensive discussion on the field of reinforcement learning in §4.2. The focus in this section is on the nature of evaluative feedback, the reinforcement learning problem in general and on a selection of reinforcement learning solution approaches. The chapter concludes in §5.3 with a brief summary of the material included.

### 4.1 An introduction to machine learning

The field of *machine learning* is concerned with the development of methodologies for enabling computers to learn to adapt or modify their actions in pursuit of more accurate or optimal ones, based on example data or past experience [2, 109]. Machine learning is a diverse discipline and integrates concepts from the fields of biology, statistics, mathematics, physics and neuroscience [109]. In machine learning, a mathematical model is constructed and the main purpose of this model is to draw inferences from a sample [2]. A machine learning model may be either *predictive* (*i.e.* aimed at making predictions) or *descriptive* (*i.e.* aimed at inferring knowledge from data), or both [2].

A computer is said to learn when it improves its performance at some task through experience [113]. Formally, Mitchell [113] states that a machine is considered to *learn* with respect to a particular class of tasks  $T$  and performance measure  $P$ , if its performance at  $T$  improves sufficiently following an increase in experience  $E$ .

When the ability of a computer to learn is considered, two questions arise [109]: How does the computer know whether it is improving or not? And, secondly, how does the computer know how to improve? Since there are different potential answers to these questions, a number of machine learning paradigms have emerged. In some cases, an algorithm can be told the correct

answer and in this manner the hope is that the algorithm will be able to generalise and calculate the correct answers for different input data. In a different scenario, the correct answers may be unknown and the algorithm may try to identify similarities in the input data. Alternatively, an algorithm can be told how good an answer is but not how to improve it. In such a case the algorithm has to search for superior answers. Marsland [109] identifies four distinct machine learning paradigms, based on how computers find answers:

**Supervised learning.** In supervised learning, the objective is to learn a mapping from particular inputs to outputs where the correct answers are provided by a supervisor [2]. More specifically, a training set of examples with the correct answers (targets) are provided to a supervised learning algorithm. Given this training set, the algorithm generalises to ultimately respond to all possible inputs [109]. *Regression* and *classification* are examples of two popular supervised learning methods.

**Unsupervised learning.** In unsupervised learning, a training set containing inputs only (no outputs) are provided. An unsupervised learning algorithm attempts to identify similarities between inputs so as to categorise those inputs with common characteristics together [109, 126]. In statistics, this method is called *density estimation*.

**Reinforcement learning.** Reinforcement learning lies somewhere between supervised and unsupervised learning. A reinforcement learning agent only receives a signal that evaluates how good (or bad) an answer is, but is not instructed how to improve or correct it [109]. Instead, a reinforcement learner has to search for the correct answer by trial-and-error. Whereas supervised learning is learning together with a teacher, reinforcement learning is sometimes also called *learning with a critic* where the critic does not provide any instructions, but simply evaluates the performance of an action [2].

**Evolutionary learning.** Biological evolution may be perceived as a kind of learning process because biological organisms learn to adapt to their environment in order to improve their chances of survival and to produce offspring. An analogy of this biological process is found in evolutionary learning. According to this paradigm, each set of answers is assigned a level of fitness, which corresponds to a measure that indicates how good the current solution is [109].

There are several factors that influence an inventory management policy, as described in §2.7. Given that many of these elements, such as customer demand and lead times, are stochastic and change over time, it is imperative that inventory policies account for all the possible permutations of these factors. Given this complexity and the online nature of the inventory management problem, it is reinforcement learning that has drawn the attention of the author for further analysis and implementation in this thesis. It is anticipated that reinforcement learning may be employed to experiment with different inventory policies based on trial-and-error search so as to ultimately learn near-optimal policies given enough time.

## 4.2 Reinforcement learning

The machine learning paradigm of reinforcement learning is carefully aligned with the primary nature of human learning. When a person is in the process of learning a new action or skill, such as playing chess, he or she relies on feedback from the surrounding environment to evaluate the consequences of his or her chosen actions. The human learner then chooses his or her actions in such a way that it influences the environment, typically in pursuit of a particular goal. Sutton



and Barton [147], who are widely considered as the pioneers of reinforcement learning [149], describe reinforcement learning as a computational approach towards learning from interaction with an environment, with a particular focus on goal-directed learning [137].

Reinforcement learning may be conceptualised in terms of a learning agent that is not instructed as to which actions it should take, but instead should discover for itself which actions yield desirable results by attempting them. The desirability associated with any particular action is measured in terms of a numerical reward signal and the learning agent's objective is to maximise this reward. A learning agent is therefore encouraged to explore the available action space (the set of all possible actions) in an attempt to discover which actions yield the highest reward. In some cases, it is possible that a particular action may influence not only the immediate reward, but also the following action and, therefore, all subsequent actions. This approach to machine learning captures two unique traits, namely *trial-and-error search* and *delayed reward*, which distinguish reinforcement learning from other machine learning paradigms [147]. These two characteristics, however, pose a unique challenge in reinforcement learning in terms of establishing a suitable trade-off between exploitation and exploration. A reinforcement learning agent needs to exploit what it already knows to obtain high rewards, but it can only discover potentially better actions by exploring them [126]. Both exploration and exploitation may have significant effects on the learning time and quality of the learned policies. For this reason, the quest to balance exploration and exploitation effectively is afforded considerable attention in the literature [150].

Beyond the learning agent and the surrounding environment, there are four additional main subelements of a reinforcement learning system, namely a *policy*, a *reward function*, a *value function* and a *model* of the environment [147].

The *policy* captures a mapping from the agent's perceived states of the environment to the actions that should be performed by the agent when in a particular state [2, 147]. In other words, the policy prescribes agent behaviour. Policies may often be stochastic in nature. The *state space* encapsulates all the possible states achievable by the reinforcement learner [109].

The primary objective of any reinforcement learning problem is embodied in the *reward function* which assigns scalar rewards based on the perceived state (or state-action pair) of the environment. The reward measures the intrinsic desirability of any given state. It is therefore the goal of any reinforcement learning agent to maximise its cumulative reward received over time [147]. The reward function is typically unalterable by the learning agent, but it may be used as a cause for adjusting the policy. When some action, for example, yields a low reward, the policy may be altered to select a different action in the same situation in the future. Reward functions are also typically stochastic.

Whereas a reward function specifies what is desirable in the short term, the *value function* defines what is good in the long run. The *value* of a state captures the total reward that an agent may possibly accumulate over time, starting from a particular state onwards. This is achieved by considering the reward of the current state, as well as the states that are likely to follow and their respective rewards [147]. In contrast to rewards, values determine the long-term desirability of residing within a particular state at any given time. This implies that an action with a relatively low initial reward may be preferred over an action with a higher initial reward, when the following states are expected to yield even higher rewards. In some cases, however, the converse of this phenomenon may hold true. Whereas rewards are generally provided directly by the environment, it is much harder to estimate value functions since values are continually estimated based on an agent's observations made over time. For this reason a method for efficiently estimating values is widely recognised as the most important element of a reinforcement learning problem [147].

A *model* of the environment is the fourth and final component of any reinforcement learning problem. This model is any construct that mimics the behaviour of the environment. Given a current state and action, the model has the capacity to predict the resulting next state and its corresponding reward, and is therefore often used as a tool for planning. Notably, not every reinforcement learning agent necessarily uses a model of the environment during the learning process [137].

### 4.2.1 Evaluative feedback

Arguably the main feature that distinguishes reinforcement learning from other machine learning paradigms is that “correct” actions are not provided as instructions, but rather that training information is employed to evaluate the effects of selected actions [147]. This phenomenon calls for techniques to guide active exploration of the action space by means of an extensive trial-and-error search. A drawback of purely evaluative feedback, however, is that it only pronounces on the quality of a chosen action, without providing any indication of whether superior or worse actions exist in the action space. The objective in this section is to review briefly some evaluative feedback methods, as discussed by Sutton and Barto [147].

#### Action-value methods

According to the paradigm of reinforcement learning, an action is evaluated in terms of the value, or cumulative reward, reaped as a result of the particular action. One basic method for estimating the value of an action is to average all of the rewards received when this particular action was chosen in the past. This implies that if at the  $t$ -th play of an iterative game, an action  $a$  was performed  $k_a > 0$  times before, producing rewards  $r_1, r_2, \dots, r_{k_a}$ , then the corresponding action value may be calculated as

$$Q_t(a) = \frac{r_1 + r_2 + \dots + r_{k_a}}{k_a}. \quad (4.1)$$

In the special case where  $k_a = 0$ ,  $Q_t(a)$  is typically assigned a pre-defined value, such as  $Q_t(a) = 0$  [147]. This approach toward action-value estimation is commonly known as the *sample-average* method, since each estimate is simply based on the average of the rewards observed in the sample space. One of the most basic action-selection techniques is to simply choose the action which yielded the largest estimated value up until time  $t$ . This technique is known as the *greedy method*, since it exploits existing knowledge in its quest to maximise short-term rewards [109]. A drawback of the greedy method, however, is that it forsakes exploration which may potentially uncover different actions which may yield higher rewards in the long run. The so-called  *$\epsilon$ -greedy method* has been introduced to attempt to find a balance between exploitation and exploration. According to the  $\epsilon$ -greedy method, a learning agent is allowed to behave greedily most of the time but occasionally the agent is allowed to choose an entirely random (irrespective of its action value) with a small, positive probability  $\epsilon$ . This allows the agent to explore the action space and to possibly find better solutions. By implication, the degree of exploration is dictated by the value of the parameter  $\epsilon$ . When the agent is allowed to explore, the  $\epsilon$ -greedy method chooses an alternative action with uniform probability [109]. Notably, the value of  $\epsilon$  may differ among various learning problems and it is often an extremely time-consuming task to find a suitable value for this parameter [150].

### Softmax action selection

Although the  $\epsilon$ -greedy method encourages exploration, this exploration is random and chooses equally among all available actions. In other words, this action-selection rule is equally likely to choose the worst-performing action as it is to choose the next-best action. The so-called *softmax* action-selection technique is a refinement of the  $\epsilon$ -greedy method and has been proposed as a potential solution to this problem. According to the softmax method, the various action-selection probabilities are assigned as a graded function of the estimated value. The action with the highest estimated value is assigned the highest probability for selection, while all other actions are sorted and weighted according to their respective value estimates [147]. The softmax method typically employs a Boltzmann or Gibbs distribution to determine the action-selection probabilities. At the  $t$ -th play in an iterative game, an action  $a$  is chosen with probability

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

where  $\tau$  is a non-negative parameter known as the *temperature*. A high temperature typically results in all action-selection options to adopt nearly equal probabilities, whereas low temperatures will cause larger differentiation among selection probabilities for actions with different estimated values [150]. Lower temperatures will therefore encourage greedy action selections.

Both the  $\epsilon$ -greedy and softmax methods rely upon maintaining a record of all the historical rewards received up to time step  $t$  in order to estimate the value of any given action. A drawback of such an approach, however, is that its memory and computational requirements grow unbounded as time increases. Sutton and Barto [147] introduced an incremental update formula for calculating averages that may reduce the memory and computational burden. Let  $Q_k$  denote the average rewards achieved for some action  $a$ , performed  $k$  times. When a new reward  $r_{k+1}$  is received, the average of all  $k + 1$  awards may be calculated as

$$Q_{k+1} = Q_k + \frac{1}{k+1}[r_{k+1} - Q_k]. \quad (4.2)$$

This implementation requires memory only for  $Q_k$  and  $k$  and the small computation in (4.2) to determine the new average. A general form of this update rule, as proposed by Sutton and Barto [147], is given by

$$NewEstimate \leftarrow OldEstimate + StepSize[Target - OldEstimate]. \quad (4.3)$$

The quantity  $[Target - OldEstimate]$  reflects an *error* in the original estimate. The size of this error is reduced by taking a step towards the target by means of a step-size parameter — a value presumed to signify a desirable direction in which to move.

### Tracking a nonstationary problem

The aforementioned methods are applicable only to a stationary environment — that is an environment that remains constant over time. Most environments encountered in practice are, however, not stationary, but change over time. Common examples of nonstationary environments include a pharmaceutical supply chain, the stock market, traffic flow on a highway and a game of chess. For nonstationary environments, it is often recommended to weight recent rewards more heavily than rewards received further back in the past [147]. This weighting tech-

nique is often implemented by employing a constant step-size parameter  $\alpha$  to compute a weighted average value

$$\begin{aligned} Q_k &= Q_{k-1} + \alpha[r_k - Q_{k-1}] \\ &= (1 - \alpha)^k Q_0 + \sum_{i=1}^k \alpha(1 - \alpha)^{k-i} r_i, \end{aligned} \quad (4.4)$$

where  $0 < \alpha \leq 1$  is constant. Since the weights satisfy the condition  $(1 - \alpha)^k + \sum_{i=1}^k \alpha(1 - \alpha)^{k-i} = 1$ , this is known as a weighted average. Furthermore, because the weight decays exponentially, this technique is often called an *exponential, recency-weighted average* [147].

### 4.2.2 Formulation of the reinforcement learning problem

The aim in this section is to introduce a generic formulation of the reinforcement learning problem and to discuss some of the key mathematical components of this particular problem class.

#### The agent-environment interface

As discussed in §4.2, reinforcement learning is concerned with a learner learning from interaction with its environment in order to achieve a particular goal or objective. This learner, which also serves as a decision maker, is known as the *agent*. The domain in which the agent resides and with which it interacts, is called the *environment*. The agent learns about this environment [109]. The agent proceeds to perform some action which influences the environment. The environment, in turn, provides a new situation (state) to the agent along with rewards which are observed by the agent. A schematic representation of the agent-environment interaction is shown in Figure 4.1.

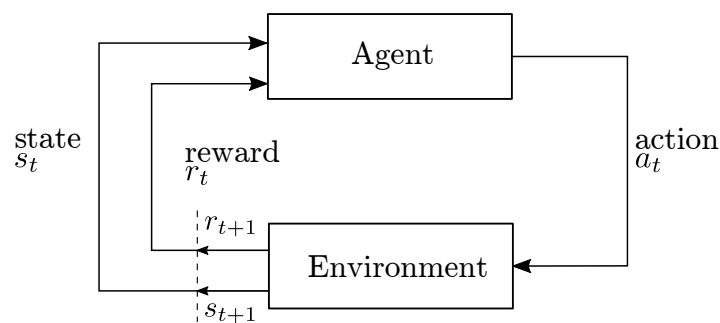


FIGURE 4.1: *The agent-environment interaction in reinforcement learning* [147].

As may be seen in the figure, the agent-environment interaction manifests itself over time at a sequence of discrete time steps  $t = 0, 1, 2, \dots$ . At every time step, the agent is provided with some representation of the environment's state,  $s_t \in \mathcal{S}$ , where  $\mathcal{S}$  is the set of all possible states. An action,  $a_t \in \mathcal{A}(s_t)$  is chosen based on the current state, where  $\mathcal{A}(s_t)$  represents the set of all actions available to the agent when in state  $s_t$ . When the agent has performed some action, it receives a numerical reward at the following time step and transitions to a new state,  $s_{t+1}$ . The reward is denoted by  $r_{t+1} \in \mathcal{R}$ , where  $\mathcal{R}$  encapsulates all possible rewards. At each discrete time step, the agent employs a mapping from states to the different probabilities of choosing

each possible action. This mapping is embodied in the agent's policy and the policy is denoted by  $\pi(s_t, a_t)$ . Reinforcement learning techniques are employed by an agent to adjust its policy as it learns from experience.

### Goals, rewards and returns

In reinforcement learning, the objective of the learning agent is to achieve a certain goal which is formalised in respect of a special reward signal that is transmitted from the environment to the agent. At any time step  $t$ , the reward is a single real number  $r_t \in \mathcal{R}$ . Although the agent may receive immediate rewards, its overall objective is still to maximise its cumulative reward, as alluded to in §4.2. The notion that a goal is described by a reward signal is one of the most prominent characteristics of reinforcement learning [147]. Importantly, the reward signal serves only to communicate to the agent *what* needs to be achieved, instead of *how* it should be achieved. Consider, for example, a reinforcement learning agent playing chess. It is imperative that the agent receives reward only for winning a game and not for achieving secondary goals, such as taking the opponent's pieces, since this may not necessarily lead to a win. In such a case the agent may, for example, learn to take the opponent's pieces at the expense of losing the game. Additionally, it is important to recognise that rewards are calculated in the environment and not in the agent. The motivation behind this phenomenon is that the agent should not be allowed perfect control in the pursuit of its goal [147].

Suppose that the sequence of rewards received after time step  $t$  is denoted by  $r_{t+1}, r_{t+2}, r_{t+3}, \dots$ . An agent's goal is to maximise its *expected (future) return*, denoted by  $R_t$ , and it is often defined by some function of the reward sequence. The expected return can more formally be expressed as

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \dots + r_T, \quad (4.5)$$

where  $T$  represents the final time step. This approach is, of course, only sensible in situations where a final time step is relevant. The notion of a final time step is observed when the agent-environment interaction is segmented into subperiods known as *episodes*, such as plays of a game. Each episode concludes with a so-called *terminal state*, after which a reset to some pre-defined initial state occurs. This starting state may also be sampled from a standard distribution of initial states. Tasks that comprise multiple episodes are called *episodic tasks*. When dealing with episodic tasks, it may be necessary to distinguish between the set of all non-terminal states, denoted by  $\mathcal{S}$ , and the set of all terminal and non-terminal states, denoted by  $\mathcal{S}^+$ .

In many cases, however, a task cannot be segmented into distinct episodes, but rather continues indefinitely without pause. Tasks that fall in this category are known as *continuing tasks* [147]. Since continuing tasks do not have an identifiable final time step (*i.e.* no terminal state), they render the expected return formula described in (4.5) infeasible. In other words, the final time step would be  $T = \infty$  and this implies that the expected return may easily become infinite. To address this problem, Sutton and Barton [147] have introduced the notion of *discounting* during the calculation of the expected return. According to this approach, the goal of the agent is to maximise the discounted return

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (4.6)$$

where  $\gamma$  is the so-called *discount rate*, which is a scalable parameter in the unit interval  $[0,1]$  [147]. The function of the discount rate is to determine the present value of future rewards. By implication, the value of a reward received  $k$  time steps in the future is reduced by a factor

$\gamma^{k-1}$  of its original value. When  $\gamma < 1$ , the reward sequence  $\{r_k\}_{k=1,2,3,\dots}$  is bounded and the discounted return sum in (4.6) is finite. If, however,  $\gamma = 0$ , the agent is described as *myopic* since it is concerned only with the maximisation of immediate rewards. When  $\gamma$  approaches 1, on the other hand, the significance of future rewards increases and the agent is said to be more *far-sighted*. For the sake of simplification, both the episodic and continuing cases described in (4.5) and (4.6) can be combined and the formula for the expected reward may then be written as

$$R_t = \sum_{k=0}^T \gamma^k r_{t+k+1}, \quad (4.7)$$

accommodating either  $T = \infty$  or  $\gamma = 1$  [147].

### The Markov property

As mentioned above, a reinforcement learning agent makes decisions as a function of a signal received from the environment, known as the *state*. A state captures all the information available to an agent at any given time instant and is typically provided by some preprocessing system inherent to the environment. Importantly, a state signal should summarise past sensations succinctly, in a manner such that all pertinent information is retained [147]. A state signal that successfully retains all relevant information is said to possess the *Markov property* [109]. Consider, again, a game of chess as an example. The current configuration of all the pieces on the chess board may be considered a Markov state, since it summarises the complete sequence of situations that led to the current state. Although no record is kept of the exact sequence of moves that led to the current state, all the information required to make decisions in the future is preserved.

For any state signal that exhibits the Markov property in a reinforcement learning problem, the environment's response at time step  $t + 1$  depends only on the state and action representations at time step  $t$ . From this it follows that the dynamics of the environment may be described by specifying only

$$P_r(s_{t+1} = s', r_{t+1} = r \mid s_t, a_t), \quad (4.8)$$

for all  $s' \in \mathcal{S}$ ,  $r \in \mathcal{R}$ ,  $s_t \in \mathcal{S}$  and  $a_t \in \mathcal{A}(s_t)$ . For any environment with the Markov property, the next state and expected reward may be predicted based only on the current state and action. In other words, the expression in (4.8) may be employed iteratively to predict all future states and rewards just as well as would be possible if the entire history up to the current time was known. For this reason, Markov states provide the best basis for selecting actions and the policy is typically established as a function of the Markov states.

### Markov decision processes

The notion of *Markov decision processes* is central to the theory of reinforcement learning. Any reinforcement learning task that satisfies the Markov property is called a Markov decision process (MDP) [137, 147]. In the case where the state and action spaces are finite, the process is known as a *finite* MDP. Any particular finite MDP is defined by its state and action sets, as well as by the one-step dynamics of the environment. Formally, an MDP may be described as a tuple  $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$ , where  $\mathcal{S}$  denotes a finite set of all possible states,  $\mathcal{A}$  denotes a set of possible actions,  $\mathcal{R}$  denotes the reward function and  $\mathcal{P}$  denotes the state transition function [137, 162]. For any state  $s \in \mathcal{S}$  and action  $a \in \mathcal{A}(s)$ , the probability of each possible following state  $s'$  is given by

$$P_{ss'}^a = P_r(s_{t+1} = s' \mid s_t = s, a_t = a). \quad (4.9)$$

These probability values are called *transition probabilities* [138, 162]. Likewise, for any given current state  $s \in \mathcal{S}$  and chosen action  $a \in \mathcal{A}(s)$ , with the next state being  $s' \in \mathcal{S}$ , the expected value of the next reward is formalised as

$$R_{ss'}^a = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\}. \quad (4.10)$$

The quantities  $P_{ss'}^a$  and  $R_{ss'}^a$  described in (4.9)–(4.10) encapsulate the most fundamental characteristics of the dynamics of a finite MDP (only information pertaining to the distribution of rewards around the expected value is lost).

### Value functions

The majority of reinforcement learning algorithms are based on estimating *value functions* — functions of states (or of state-action pairs) that estimate how beneficial it is for an agent to reside in a given state [147]. The degree of benefit (or desirability) of a particular state is defined in terms of the total expected return. The future rewards are, of course, determined by the particular actions that the agent will take. Since the agent's actions are prescribed by a specific policy, value functions are established with respect to particular policies. The value of a state  $s$  under a policy  $\pi$  is the total expected return when starting in state  $s \in \mathcal{S}$  and following  $\pi$  thereafter, and is denoted by  $V^\pi(s)$ . In MDPs, the *state-value function for policy  $\pi$* , denoted by  $V^\pi(s)$ , is formalised as

$$V^\pi(s) = E_\pi\{R_t \mid s_t = s\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\}, \quad (4.11)$$

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

$$Q^\pi(s, a) = E_\pi\{R_t \mid s_t = s, a_t = a\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\}. \quad (4.12)$$

In reinforcement learning, the value functions  $V^\pi$  and  $Q^\pi$  may be learnt from experience. For example, if an agent follows policy  $\pi$ , it obtains a reward for each particular state visited. The average of all the returns received each time that a particular state is encountered converges to the state's true value  $V^\pi(s)$  when the number of visits to that particular state approaches infinity. Similarly, if separate averages are recorded for each action performed in a state, then these averages will converge to the action values,  $Q^\pi(s, a)$ . Estimating values in this fashion is categorised as *Monte Carlo methods* because they involve computing the average over random samples of actual returns and not over expected returns [147].

An essential quality of value functions employed in reinforcement learning is that they satisfy special recursive relationships. Given a policy  $\pi$  and some state  $s$ , the consistency conditions are given by

$$\begin{aligned} V^\pi(s) &= E_\pi\{R_t \mid s_t = s\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\} \\ &= E_\pi\left\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s\right\}. \end{aligned} \quad (4.13)$$

It follows from the transition probabilities that (4.13) may be expanded to

$$\begin{aligned} V^\pi(s) &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a \left[ R_{ss'}^a + \gamma E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s' \right\} \right] \\ &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')]. \end{aligned} \quad (4.14)$$

These conditions hold between the value of  $s$  and the value of its possible following states, where it is implicit that the actions are chosen from the set  $\mathcal{A}(s)$ , and the successor states are taken from the set  $\mathcal{S}$ . The expression in (4.14) is called the *Bellman equation for  $V^\pi$*  and it captures a relationship between the value of a current state and the values of its successor states. The Bellman equation (4.14) captures the average return over all possible states, weighting each state according to its probability of occurring. The equation furthermore states that the value of the initial state must equal the (discounted) value of the successor state, plus the expected reward. For this reason, a value function  $V^\pi$  for a specific policy  $\pi$  is the unique solution to its corresponding Bellman equation for that policy. From this follows that the Bellman equation is a suitable basis from which  $V^\pi$  may be calculated, estimated or learnt [147].

For finite MDPs, it is possible to formulate an optimal policy in the following way. A policy  $\pi$  is said to be superior over or equivalent to a policy  $\pi'$  if its expected return is strictly greater than or equal to that achieved by following  $\pi'$ , for all possible states. By implication,  $\pi$  is at least as good as  $\pi'$  if and only if  $V^\pi(s) \geq V^{\pi'}(s)$  for all  $s \in \mathcal{S}$ . Any policy that is at least as good as any other policy is said to be an *optimal policy*, and is denoted by  $\pi^*$  [147]. Each optimal policy  $\pi^*$  (there may be multiple optimal policies) adopts an *optimal state-value function* which is denoted by  $V^*$  and defined as

$$V^*(s) = \max_{\pi} V^\pi(s), \quad (4.15)$$

for all  $s \in \mathcal{S}$ . As a result, for each optimal policy there also exists a corresponding *optimal action-value function* given by

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a), \quad (4.16)$$

for all  $s \in \mathcal{S}$  and all  $a \in \mathcal{A}(s)$ . For each potential state-action pair, this function value signifies the expected return obtained by performing an action  $a$  while in state  $s$  and thereafter following an optimal policy  $\pi^*$ . The value of an optimal state-action pair may therefore be expressed as

$$Q^*(s, a) = E\{r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a\}. \quad (4.17)$$

## Optimality and approximation

While optimal value functions and optimal policies may exist in theory, they can rarely be formulated in practice due to the high computational cost associated with large state and action spaces [147]. Even in cases where an accurate and exhaustive model of the environment is known, it is typically impossible to compute an optimal policy by solving the Bellman equation, due to time and memory constraints [147]. Alternatively, function approximations may be employed where value functions, policies and models are approximated. In reinforcement learning tasks with a small, finite set of states, it is often possible to establish approximations using tables or arrays containing an entry for each state-action pair. Methods that fall within this domain are called *tabular* methods. For large problems, however, with potentially infinitely many states, tabular methods are often insufficient and scholars often resort to function approximations. In these cases, functions are approximated (typically at the cost of optimality) by means of a more compact parameterised function representation. Functions approximations do, however,



present unique opportunities for formulating effective approximations. Given some reinforcement learning problem, for example, there may be many states that will be reached with such low probability that selecting suboptimal actions for them would have a negligible influence on the amount of reward received by the agent. The online nature of reinforcement learning makes it possible to prioritise function approximations for frequently occurring states, resulting in effective decisions being made when these states are encountered. Similarly, less attention is then afforded to learning good policies for less frequently encountered states. This is one of the hallmarks that distinguishes reinforcement learning from other approximate solution methods to MDPs.

### 4.2.3 Reinforcement learning solution approaches

Sutton and Barto [147] describe three basic classes of solution approaches for solving a reinforcement learning problem. The first is *dynamic programming* and, although elegantly developed mathematically, this approach requires a complete and accurate model of the environment — something that is not always possible. The second approach, *Monte Carlo methods*, do not require a full model of the environment but they do not accommodate stepwise incremental computation. Finally, *temporal-difference learning methods* do not require a model of the environment and allow for step-by-step incremental computation. Temporal-difference methods are, however, significantly more complex to analyse. This section is devoted to a brief review of some basic reinforcement learning algorithms which may be employed to find optimal policies.

#### Policy iteration

In the case where the dynamics of an environment are fully known (*i.e.* the transition and reward functions are known), the Bellman equation in (4.14) represents a system of  $|\mathcal{S}|$  linear equations in  $|\mathcal{S}|$  unknowns, for all  $V^\pi(s)$  and all  $s \in \mathcal{S}$ . To solve this system of equations is, however, often impractical, especially when the state space becomes large. For this reason, iterative solution methods for estimating value functions are often preferred [147]. The estimation of the value function at the  $(k + 1)^{th}$  learning iteration, denoted by  $V_{k+1}^\pi(s)$ , is given by

$$\begin{aligned} V_{k+1}^\pi(s) &= E_\pi\{r_{t+1} + \gamma V_k^\pi(s_t + 1) \mid s_t = s\} \\ &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V_k^\pi(s')], \end{aligned} \quad (4.18)$$

where the value of  $V_s^\pi$  is chosen arbitrarily (terminal states are always assigned a value of zero). It has been shown that  $V_k^\pi$  converges to  $V^\pi$  as  $k \rightarrow \infty$  under the condition that either  $\gamma < 1$  or the events are episodic [147]. Adopting this method whereby value functions are estimated through the repeated implementation of (4.18) is called *policy evaluation*. Formally, iterative policy evaluation converges only in the limit but in practice the process must be terminated short of this. The policy evaluation process can be stopped beyond a certain number of iterations, because the corresponding greedy policy will not change anymore although the value function tends to the optimal value function [138, 147]. One of the most popular stopping criteria involves terminating the iteration process when the quantity  $|V_{k+1}(s) - V_k(s)|$  is sufficiently small.

When  $V_s^\pi$  and  $Q^\pi(s, a)$  are known for all states  $s \in \mathcal{S}$  and actions  $a \in \mathcal{A}(s)$ , it is possible to establish the optimal policy by choosing, in each state, the action with the largest value of

$Q^\pi(s, a)$ . In this manner, a new *greedy* policy  $\pi'$  is formalised as

$$\begin{aligned}\pi'(s) &= \max_a Q^\pi(s, a) \\ &= \max_a E\{r_{t+1} = \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')].\end{aligned}\tag{4.19}$$

This approach towards formulating a new policy greedily (with respect to the value function of the original policy) is called *policy improvement* [147]. The process therefore involves the repeated evaluation and improvement of policies. In other words, once a policy  $\pi$  has been improved based on the value of  $V^\pi$ , a superior policy  $\pi'$  is established. Calculating  $V^{\pi'}$ , in turn, then makes it possible to find an even better policy  $\pi''$ . A sequence of monotonically improving policies and value functions

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

is therefore obtained, where  $\xrightarrow{E}$  denotes a policy evaluation and  $\xrightarrow{I}$  denotes a policy improvement. This process of finding an optimal policy is called *policy iteration* [147]. A pseudo-code description of the policy iteration algorithm is provided in Algorithm 4.1.

---

**Algorithm 4.1:** The policy iteration algorithm [147].

---

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

**Output:** An optimal policy  $\pi^*(s)$ .

```

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

---

### Value iteration

One shortcoming of policy iteration is that each iteration requires policy evaluation, which may result in a lengthy computational process. This is most evident in cases where a large number of

sweeps through the entire state set is required. The method of *value iteration* requires no explicit policy and has been introduced to counter this drawback. According to value iteration, the policy evaluation step of policy iteration is truncated, without the loss of convergence guarantee of policy iteration. This is achieved by performing only one sweep of each state, instead of employing the evaluation process until convergence is reached [147]. Value iteration combines the policy improvement and truncated policy evaluation steps in such a manner that the estimated value of a set is given by

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

A pseudo-code description of the value iteration algorithm is given in Algorithm 4.2.

---

**Algorithm 4.2:** The value iteration algorithm [147].

---

**Input** : An arbitrary initial value  $V(s) \in \mathfrak{R}$  for all  $s \in \mathcal{S}$ .

**Output:** An optimal policy  $\pi^*(s)$ .

```

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

---

## Q-learning

Q-learning is a so-called off-policy temporal difference algorithm first proposed by Watkins [157]. It is a value iteration-based method, does not require a model of the environment and is therefore said to be a *model-free* algorithm [16, 126]. The goal of Q-learning is to learn the optimal action-value function, irrespective of the policy being followed [147]. This is achieved by approximating  $Q(s, a)$  based on comparing the current action-value estimate to a new action-value estimate, which is calculated based on the immediate reward obtained, and the maximum value achievable over all actions in the future state,  $\max_a Q(s_{t+1}, t)$ . At the heart of Q-learning lies the update rule

$$Q_{k+1}(s_t, a_t) \leftarrow Q_k(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_a Q_k(s_{t+1}, a) - Q_k(s_t, a_t) \right], \quad (4.20)$$

where  $\gamma$  denotes the discount rate and  $\alpha \in (0, 1]$  represents the learning rate. The learning rate parameter  $\alpha$  is employed to control the magnitude of the influence that the new estimation of the value has on the current action-value estimate. If  $\alpha = 1$ , for example, the old value will be replaced by the new estimate. Owing to the stochastic nature of MDPs, however, it is necessary to compute the average value based on many independent samples obtained over time in order to converge to  $Q^*$ . Therefore, the learning rate is typically employed in order only to partially update the old values. In other words, the learning rate serves as a means to blend the current estimate with previous estimates in an attempt to converge to  $Q^*(s, a)$  [101]. A pseudo-code description of the Q-learning algorithm is given in Algorithm 4.3.

---

**Algorithm 4.3:** The Q-learning algorithm [147].

---

**Input** : An arbitrary initial value  $Q(s, a)$  for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}(s)$ .

**Output:** A near-optimal policy  $\pi^*(s)$ .

```

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

```

---

Watkins and Dayan [156] proved that the Q-learning algorithm converges to the optimal action-value function  $Q^*(s, a)$  under the condition that an infinite number of visits to and updates of the state-action value are performed for each possible state-action pair. This holds for any policy followed, as referred to in line 4 of Algorithm 4.3. Once the Q-learning algorithm has terminated (close to convergence), an optimal policy may be inferred greedily from the final approximation of state-action values [79]. It is imperative that the implementation of the Q-learning algorithm finds an appropriate balance between exploration and exploitation. Common approaches adopted for this purpose include the  $\epsilon$ -greedy and softmax action-selection techniques, discussed in §4.2.1. Drawbacks of Q-learning include that it does not consider any of the difficulties involved with generalisation over large state or action spaces, and it may potentially converge relatively slowly to a good policy [79].

### 4.3 Chapter summary

An overview of the machine learning paradigm of reinforcement learning was provided in this chapter. The chapter opened with a general introduction to the field of machine learning in order to elucidate how reinforcement learning differs from other machine learning paradigms. This was followed by a more in-depth description of the reinforcement learning problem. In this section, the notion of evaluative feedback was first reviewed because it is such a fundamental component of the reinforcement learning framework. A generic formulation of the reinforcement learning problem in general was provided next. This discussion was complemented with an introduction to some of the critical elements of the mathematical structure of the reinforcement learning problem. Finally, descriptions of three reinforcement learning approaches followed, namely policy iteration, value iteration and Q-learning.

## Part II

# Pharmaceutical supply chain modelling



---



---

## CHAPTER 5

---

# Information sharing in a pharmaceutical supply chain

### Contents

5.1	Investigating the impact of information sharing . . . . .	73
5.2	Five information-sharing scenarios . . . . .	74
5.2.1	<i>Scenario 1: No information sharing</i> . . . . .	74
5.2.2	<i>Scenario 2: Intra-neighbourhood information sharing between clinics</i> . . . . .	75
5.2.3	<i>Scenario 3: Limited inter-tier information sharing</i> . . . . .	77
5.2.4	<i>Scenario 4: Information sharing between warehouses</i> . . . . .	78
5.2.5	<i>Scenario 5: Extended inter-tier information sharing</i> . . . . .	79
5.3	Chapter summary . . . . .	80

The purpose of this chapter is to describe five potential information-sharing scenarios that may be implemented in a pharmaceutical supply chain network. Each scenario possesses a distinct information-sharing configuration that is expected to have a different impact on inventory management in a pharmaceutical supply chain. The five scenarios pertain to the experimental design according to which the impact of information sharing on inventory management is evaluated later in this thesis.

Some general considerations that are relevant to information sharing are discussed in §5.1. This is followed in §5.2 by detailed descriptions of five distinct information-sharing scenarios in a pharmaceutical supply chain context. The chapter finally closes in §5.3 with a brief summary.

### 5.1 Investigating the impact of information sharing

A primary objective in this project is to demonstrate how information sharing can benefit inventory management in a pharmaceutical supply chain. Generally, information sharing involves two key considerations: What types of information should be shared and with whom that information should be shared. To maximise the impact of information sharing, appropriate types of information have to be shared with the right stakeholders. This is evident from findings in the literature, as elucidated in §2.4.3. While it may be argued that full information sharing across a whole supply chain network is desirable, this configuration may be practically infeasible, too expensive to implement, or even result in information overload or redundancy. Subsequently,

this may prompt an investigation into identifying the minimum, but most critical, information that has to be shared in order to achieve the desired impact.

There is a plethora of information available to share in any supply chain that is only amplified by an increase in supply chain size and complexity. This results in a considerable number of potential information-sharing configurations that need to be investigated in order to evaluate the impact of information sharing extensively. The aim in this thesis is, however, not to conduct such a painstaking analysis of every possible permutation, but simply to illustrate the impact of information sharing conceptually. This is done by designing five particular, natural information-sharing scenarios arbitrarily.

## 5.2 Five information-sharing scenarios

The aim in this section is to describe five hypothetical different information-sharing scenarios within a pharmaceutical supply chain context. These scenarios are investigated and analysed later in this thesis in order to ascertain the influence of each scenario on pharmaceutical supply chain performance. The first information-sharing scenario involves no information sharing at all, serving merely as a benchmark, and thereafter the scope of information sharing is increased incrementally for each subsequent scenario.

### 5.2.1 Scenario 1: No information sharing

The most basic information-sharing scenario considered for investigation in this project is one where there is absolutely no information shared between facilities in a pharmaceutical supply chain. Instead, each facility has access to its own local information and can base its inventory replenishment decisions on this information only. According to this scenario, called *Scenario 1*, hospitals and clinics that dispense pharmaceutical products to patients directly, are the only parties with access to end-user demand information. In other words, manufacturers and warehouses upstream can only infer the nature of end-user demand by analysing the frequency and the sizes of incoming replenishment orders from hospitals and clinics. When end-user demand is not shared with entities upstream, the bullwhip effect may most likely materialise, as discussed in §2.3.1. As a result, the no-information sharing scenario is expected to perform relatively poorly in respect of inventory management, especially for supply chains with long lead times and highly variable end-user demand.

The absence of information sharing in Scenario 1 is portrayed visually by the hypothetical pharmaceutical supply chain in Figure 5.1. In this example network, there is one manufacturer who distributes some pharmaceutical product to a warehouse and to a hospital. The warehouse, in turn, ships inventory to three clinics, while the hospital serves as a supplier for two other clinics. Since the warehouses do not have any visibility over their respective customer clinics, they may be susceptible to, and unprepared for, sudden changes in the frequency and sizes of incoming replenishment orders. The same principle applies to the manufacturer, which does not have any information visibility over either the warehouses or the clinics.

Scenario 1 may also be described as the base case of the comparative analysis conducted later in this thesis. Any scenario in which information is shared between at least two facilities, may be compared to the base case in order to appraise the relative value of the shared information. The base case is particularly relevant in the South African public health-care context, since implementation problems associated with the SVS have largely compromised efforts to demonstrate the impact of information sharing, as conjectured in §2.9.3. If the findings in this thesis, how-



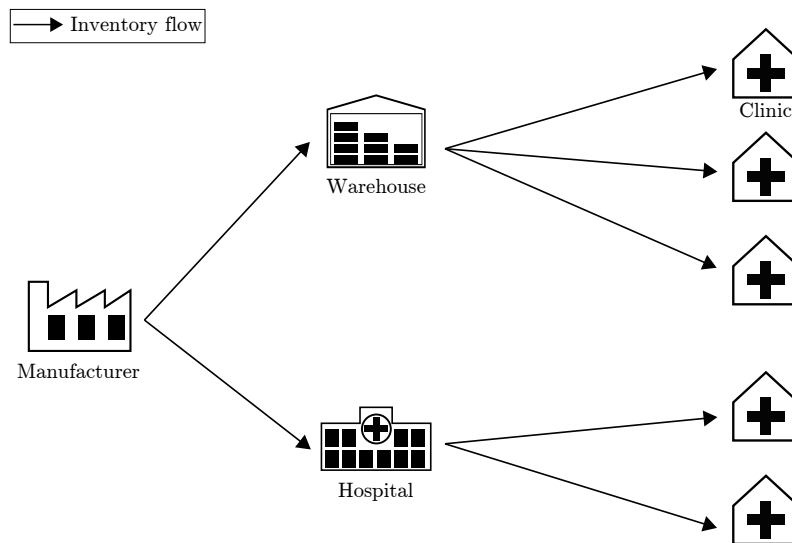


FIGURE 5.1: *Scenario 1: No information sharing between the facilities in a pharmaceutical supply chain.*

ever, proves the value of sound information sharing conceptually, it may serve as a motivation for the relevant stakeholders to rectify and possibly improve the current state of visibility in the South African pharmaceutical supply chain.

### 5.2.2 Scenario 2: Intra-neighbourhood information sharing between clinics

The second information-sharing scenario, called *Scenario 2*, involves the sharing of information within mutually disjoint pockets (called *neighbourhoods*) of health-care facilities, where the facilities in such a neighbourhood are in close geographical proximity. No information sharing, however, occurs between these neighbourhoods. The aim of this type of information sharing is to enable health-care facilities to exchange stock amongst one another locally when confronted with severe stock shortages. The practice of exchanging stock between health-care facilities in this informal manner is known as the *borrowing phenomenon*, described in §2.9.1. The term ‘borrowing’ is recognised as a misnomer since stock is never returned in practice, as mentioned in §2.9.1. The same principle is, however, adopted in this thesis and it is assumed that ‘borrowing’ implies only the once-off and unidirectional movement of stock from one facility to another. The term ‘inventory sharing’ is arguably a more suitable description for this phenomenon. The underlying objective of Scenario 2 is to determine whether or not local information sharing between clinics can facilitate sound inventory-sharing practices, and ultimately help these facilities to mitigate stock-outs.

Scenario 2 retains the information-sharing configuration of Scenario 1 (*i.e.* otherwise no information sharing between facilities), but uniquely includes the addition of clinic neighbourhoods and two instances of shared information. The first type of shared information pertains to the arrival time of shipments delivered to health-care facilities. In Scenario 2, every health-care facility is assumed to know on any given day, with certainty, the amount of stock that it will receive on the following day from its supplier(s). In practice, delivery lead times are often stochastic and a supplier cannot always guarantee on-time delivery, especially when the travel time is significantly long. For any in-transit shipment it is, however, typically easier to predict the delivery time with increased confidence as the shipment gets closer to its destination. Based on this argument, it is assumed in Scenario 2 that a health-care facility will know the exact arrival time of any incoming delivery twenty-four hours in advance. The second information type shared in

Scenario 2 is the total amount of inventory available for exchange within a clinic neighbourhood on any given day. Access to these two information types may help a clinic decide whether or not it should, and will be able to, exchange stock with other clinics.

Assuming that a health-care facility experiences end-user demand at least once a day, it is evident that the facility cannot share all of its stock with clinics in close proximity. In other words, a health-care facility must withhold at least a portion of its stock in order to fulfil its own demand, before it can make the remainder available for exchange. In Scenario 2, it is the total amount of inventory specifically earmarked for exchange that is shared between health-care facilities residing in the same neighbourhood. An example of a pharmaceutical supply chain implementing Scenario 2 is shown in Figure 5.2. There are two neighbourhoods present in this network. The first one comprises three clinics (enclosed by a rectangular shape called the *neighbourhood boundary*), while the remaining two clinics are members of the second neighbourhood (also enclosed by a neighbourhood boundary). The boundaries demarcating the neighbourhoods are also indicative of the information shared between members of the same neighbourhood. Finally, the neighbourhood boundaries also partially overlap the inventory-flow arrows between the warehouses and the clinics to signal the (limited) visibility over incoming shipments.

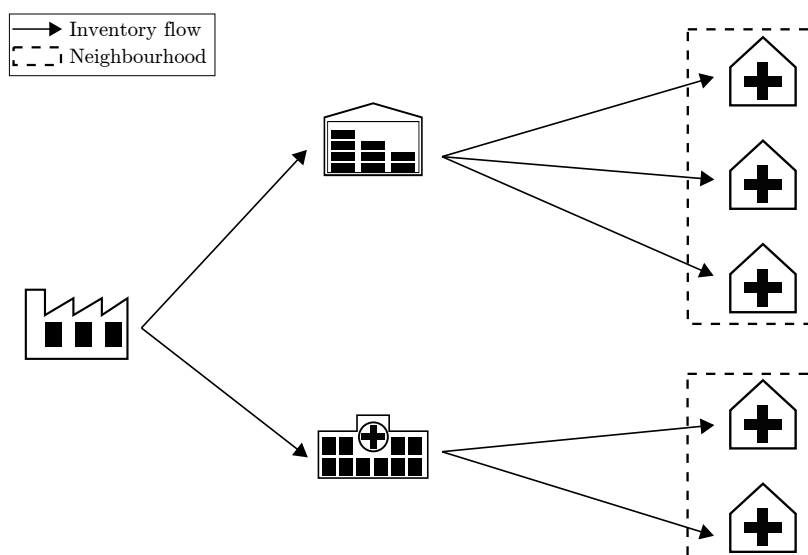


FIGURE 5.2: *Scenario 2: Information sharing between clinics in the same neighbourhood.*

Compared with Scenario 1, Scenario 2 introduces not only an additional layer of information sharing, but also a new business practice in the form of informal stock sharing between health-care facilities residing in the same neighbourhood. Intuitively, it is expected that a pharmaceutical supply chain operating under Scenario 2 would experience fewer stock-outs than one implementing the no-information sharing option of Scenario 1. This is motivated by the fact that clinics have the option of choosing an alternative supplier (a neighbouring clinic), so to speak, when their primary supplier cannot supply them with stock in a timely manner according to Scenario 2. The intention of Scenario 2 is that health-care facilities seek only to obtain stock from within their neighbourhood when they run the risk of incurring stock-outs in the short-term. An unfavourable outcome of the inventory-sharing practice would, for example, materialise when clinics essentially deplete each other's stock, leaving an entire neighbourhood with an insufficient amount of stock. Inventory sharing between clinics in the same neighbourhood may potentially be successful only when there is a sufficient amount of stock available for

exchange. Conversely, when each clinic carries a significantly small amount of inventory (less than what is demanded by patients), inventory sharing will not be possible and stock-outs may occur on a large scale.

### 5.2.3 Scenario 3: Limited inter-tier information sharing

As Scenario 2 was an extension of Scenario 1, Scenario 3 is again an expansion of Scenario 2. Whereas Scenario 2 focused on information sharing between entities of the same supply chain tier (*i.e.* clinics), Scenario 3 additionally includes the sharing of information between facilities of different supply chain tiers. According to this scenario, the scope of information sharing is increased incrementally by granting distribution facilities (warehouses and hospitals) visibility over some information pertaining to their customers (health-care facilities). On any given day, a distributor has visibility over the aggregate inventory level held by their customers, as well as over the end-user demand experienced by those health-care facilities.

A hypothetical implementation of Scenario 3 in a pharmaceutical supply chain is shown in Figure 5.3. Clinics are still allowed to share both information and inventory between them in their respective neighbourhoods, as proposed in Scenario 2. The information-flow arrows reaching from the clinic neighbourhoods to the respective warehouses indicate that information is shared by these neighbourhoods with their suppliers.

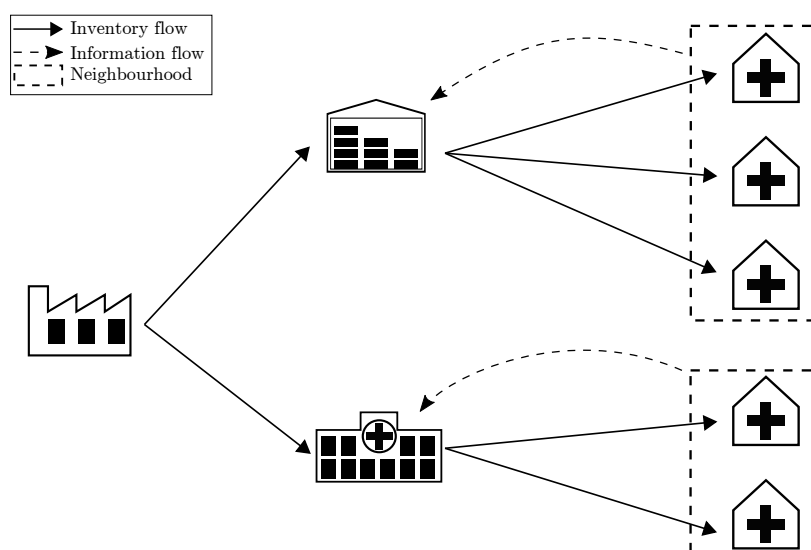


FIGURE 5.3: Scenario 3: Clinics share information in their neighbourhoods and with their direct suppliers.

Of the first three information-sharing scenarios introduced thus far, it is only in Scenario 3 that information is shared between facilities of different supply chain tiers. According to Scenario 3, warehouses and hospitals have instant access to real-time data pertaining to the inventory held at, and the demand experienced by, their customer clinics. This is significant, because it enables warehouses and hospitals to respond to changes in end-user demand much faster than would be possible without the information sharing. A sudden increase in end-user demand may, for example, prompt a warehouse to order stock from its own supplier proactively, in anticipation of larger incoming replenishment orders placed by clinics. Inter-tier information sharing is therefore particularly powerful because it effectively reduces the information-transfer lead time in a supply chain.

A warehouse in a pharmaceutical supply chain operating under Scenario 3 may be expected to offer a higher service level than in Scenario 2, based on the increased scope of information sharing. This would, however, most likely be true only for warehouses that have reliable suppliers and experience predictable demand from their customers. Nevertheless, in the case where a warehouse has insufficient stock available, clinics would still enjoy the ability to share inventory between themselves according to the informal exchange scheme. The supposition in Scenario 3 is that clinics may potentially be less dependent on intra-neighbourhood inventory sharing, because warehouses and hospitals are expected to have higher service levels. The major drawback of Scenario 3 is, however, that manufacturers — the very parties that inject finished products into a supply chain — are still excluded from information sharing.

#### 5.2.4 Scenario 4: Information sharing between warehouses

The fourth information-sharing scenario, called *Scenario 4*, is an expansion of Scenario 3 and introduces the concept of intra-neighbourhood inventory sharing between warehouses and hospitals. Similar to the arrangement of health-care facilities into neighbourhoods, as proposed in Scenario 2, warehouses may now also be segmented into neighbourhoods of their own. The aim with this particular configuration is to investigate whether warehouses and hospitals can exchange stock amongst themselves when their supplier(s) cannot provide them with stock in a timely manner. Suppose, for example, that a manufacturing facility experiences a temporary shutdown and cannot supply a number of warehouses with sufficient stock. In such a case, an affected warehouse with a low level of stock may not be able to fulfil the demand of its customers. According to Scenario 4, such a warehouse may then seek a short-term change in supplier and order stock from a neighbouring warehouse instead.

A schematic exhibiting the scope of information sharing in Scenario 4 is shown in Figure 5.4. In this example, the warehouse and the hospital now form a neighbourhood of their own. In the same manner as information is shared between clinics in their respective neighbourhoods, the warehouse and the hospital share information with each other in respect of how much inventory they have available for exchange. The scope of supply chain visibility is enlarged even further in Scenario 4 by making some manufacturing-related information available to the customers of manufacturers (*i.e.* warehouses and hospitals). In particular, each warehouse and hospital is provided with visibility over the inventory level, the inventory in production and backlogged inventory of its direct supplier or suppliers. Compared with Scenarios 1–3, the information-sharing architecture in Scenario 4 is particularly unique because information is shared with warehouses from both upstream and downstream entities. Because warehouses serve as the middlemen between manufacturing entities and health-care facilities, they may be able to make the best possible decision at the time for the given circumstances. When manufacturers have sufficient stock available, warehouses may continue to order from the former as usual. If manufacturers cannot provide adequate supply, on the other hand, warehouses immediately learn that their only plausible option is intra-neighbourhood inventory exchange.

The impact of this bidirectional information sharing may, however, be limited since manufacturers still do not possess any supply chain visibility under Scenario 4. Suppose, for example, that end-user demand for a pharmaceutical product has increased dramatically over time to such an extent that the entire supply chain is left vulnerable to stock-outs. Suppose furthermore that, due to a lack of both supply chain visibility and manufacturing capacity, manufacturers are unable to recover the resulting backlog. In such a case, information sharing will not help to prevent stock-outs at all, because the fundamental problem is that there is not sufficient stock available in the supply chain as a whole.

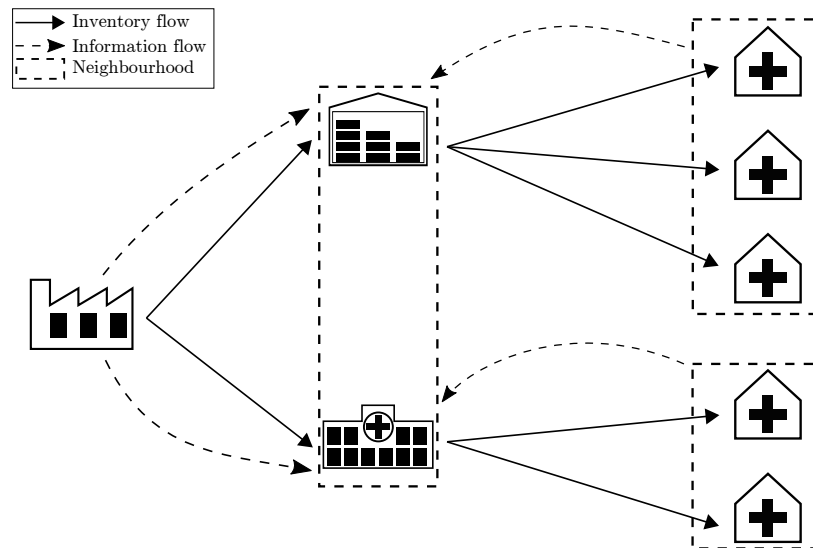


FIGURE 5.4: *Scenario 4: Intra-neighbourhood information sharing between warehouses and clinics, respectively.*

### 5.2.5 Scenario 5: Extended inter-tier information sharing

The fifth and final information-sharing scenario considered for analysis in this thesis involves information that is shared across the whole length of the supply chain. This scenario, called *Scenario 5*, is an expansion of the information-sharing arrangement in Scenario 4. In Scenario 5, all manufacturers have access to some information about the dealings of both their immediate customers (hospitals or warehouses) and their customers' customers (health-care facilities). In particular, each manufacturer is given access to inventory-level information of warehouses, hospitals and health-care facilities downstream. Furthermore, manufacturers are also granted visibility over the end-user demand experienced by those health-care facilities.

A schematic showing the structure of information sharing according to Scenario 5 may be seen in Figure 5.5. The information-flow arrow that extends from the warehouse neighbourhood to the manufacturer indicates that neighbourhood information is shared with the manufacturer. Additionally, there are information-flow arrows reaching from each of the clinic neighbourhoods to the manufacturer. These arrows are representative of the inventory level and end-user demand information that are shared with the manufacturer.

A hallmark of Scenario 5 is that it is the only scenario (of the five proposed in this chapter) in which end-user demand is made available to the manufacturing entities in a pharmaceutical supply chain. An example of a potential situation in a supply chain where manufactures do not have access to end-user demand was described in §5.2.4. According to Scenario 5, information about changes in end-user demand will be made available instantaneously to manufacturers. This is in stark contrast to the arrangement in Scenario 1 where demand information is relayed only implicitly between facilities as orders move upstream in a supply chain. Without any information sharing, the demand signal is often distorted in this fashion and may lead to the bullwhip effect, as described in §2.3.1. The supposition with the information sharing in Scenario 5 is that it may allow manufacturers to respond proactively to fluctuations in demand by adjusting their manufacturing operations accordingly. Manufacturers may, for example, be able to increase their manufacturing capacities in a timely manner when a sudden increase in end-user demand is observed. Failure to do so will, however, still present facilities downstream with the opportunity to share inventory where possible, as explained in §5.2.4.

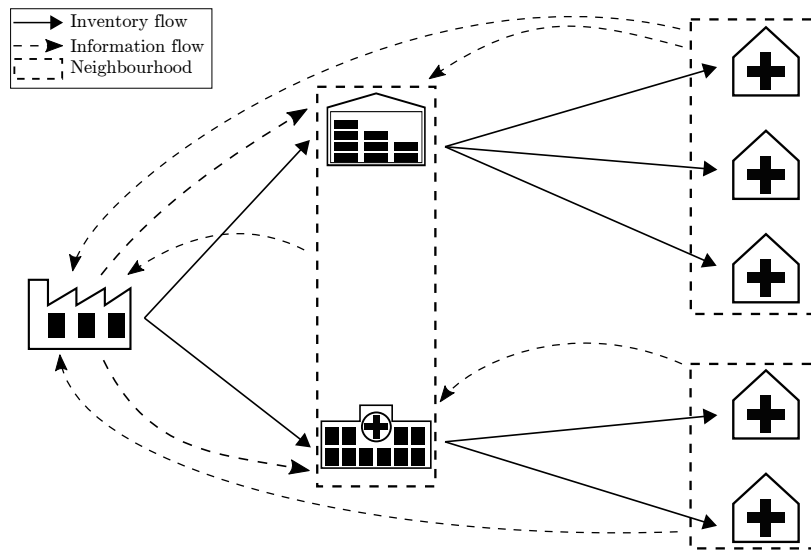


FIGURE 5.5: Scenario 5: Extended inter-tier information sharing.

### 5.3 Chapter summary

Careful descriptions were provided in this chapter of five distinct information-sharing configurations that may be implemented in a pharmaceutical supply chain. These five information-sharing scenarios are analysed later in this thesis. Some considerations relevant to analysing the impact of information sharing on pharmaceutical supply chain performance were first discussed. This was followed by descriptions of the five arbitrarily chosen hypothetical information-sharing arrangements. The first information-sharing scenario does not involve any information sharing between supply chain entities. The scope of information sharing was further increased incrementally for each of the following four scenarios. The fifth information-sharing scenario is the only scenario that included the sharing of end-user demand with manufacturing entities.

---



---

## CHAPTER 6

---

# An agent-based pharmaceutical supply chain simulation model

### Contents

6.1	Model framework . . . . .	81
	6.1.1 <i>Model input</i> . . . . .	84
	6.1.2 <i>Inventory replenishment orders</i> . . . . .	89
	6.1.3 <i>The prescriptive paradigm</i> . . . . .	90
	6.1.4 <i>The graphical user interface</i> . . . . .	91
	6.1.5 <i>Model output</i> . . . . .	92
6.2	Model verification and validation . . . . .	93
	6.2.1 <i>Verification of the simulation model</i> . . . . .	93
	6.2.2 <i>Validation of the simulation model</i> . . . . .	94
6.3	Chapter summary . . . . .	95

This chapter is devoted to a detailed description of the agent-based pharmaceutical supply chain simulation model that lies at the heart of this thesis. In §6.1, the general framework of the simulation model is discussed. This discussion comprises the nature of the model input data, the model output data and how a selection of supply chain processes are modelled. The techniques employed to verify and validate the simulation model are reviewed next in §6.2. Finally, the chapter closes in §6.3 with a brief summary of the material included in this chapter.

### 6.1 Model framework

An agent-based simulation model of a pharmaceutical supply chain was designed and implemented for use as a tool to discover and evaluate the effectiveness of different inventory replenishment strategies within a pharmaceutical supply chain. This simulation model embraces two modelling paradigms, namely a *descriptive* paradigm and a *prescriptive* paradigm. According to the descriptive paradigm, the simulation model is employed simply as a test bed for evaluating the effectiveness of traditional, user-specified inventory replenishment policies in a given pharmaceutical supply chain network. These conventional policies include the continuous-review and periodic-review inventory replenishment policies, described in §2.7. The *prescriptive* paradigm, on the other hand, facilitates the use of a reinforcement learning algorithm to *discover* effective inventory replenishment policies, specifically informed by information sharing, for a given

pharmaceutical supply chain network. This is achieved by implementing the five information-sharing scenarios of §5.2 in the prescriptive paradigm. In other words, a reinforcement learning algorithm is employed to learn effective inventory control policies based on each of the five information-sharing scenarios. Similar to the descriptive paradigm, the prescriptive paradigm also pronounces on the effectiveness of the inventory replenishment policies provided by the reinforcement learning method.

Any pharmaceutical supply chain is a complex system comprising many entities, processes, intricacies and interdependencies which renders comprehensive modelling of the entire supply chain a non-trivial task. Examples of such complexities that may be encountered in a supply chain include the particular nature of decision-making processes, the influence of resource availability on the execution of daily operations, and the flow of information or money between entities. Since the principal focus of this thesis is on information sharing and inventory management in pharmaceutical supply chains, the proposed simulation model does not incorporate all of the potential supply chain complexities such as those referred to above, exclusively. Instead, the simulation model serves the purpose of demonstrating how shared information may be utilised specifically to benefit inventory management in a pharmaceutical supply chain. For this reason, the simulation model introduced in this chapter is appropriately called a *concept demonstrator*. The proposed simulation model adopts a particular level of abstraction that is deemed sufficient for it to fulfil its purpose as concept demonstrator successfully and effectively.

The simulation model concept demonstrator was implemented in the AnyLogic 8.3.2 [3] software suite, and has been designed in such a manner that any user-specified supply chain network may be modelled. The simulation model replicates the high-level operation of a pharmaceutical supply chain over time, with a particular focus on the flow of inventory along the supply chain. Designed according to the paradigm of agent-based modelling, the simulation model includes two distinct agent types. The most prominent agent type is the so-called **Facility** agent. This agent represents any type of facility within a pharmaceutical supply chain and it is responsible for all inventory-related decisions and activities involving the particular facility. Any given supply chain network modelled therefore comprises several **Facility** agents. An advantage of modelling facilities as agents according to the agent-based modelling paradigm, is that it makes it possible to assign unique attribute values to each agent. Attributes of the **Facility** agent include its name, the facility type, its connections with other facilities and its inventory level. A second agent type embedded in the simulation model is the so-called **Order** agent, which is used to model any inventory replenishment order. Attributes of the **Order** agent include the type of product ordered, the order quantity and the facility that placed the order. Since all inventory replenishment orders share the same attributes (although with different values), they are well-suited to be modelled as agents.

Any pharmaceutical supply chain comprises different facility types, such as the manufacturers of pharmaceutical products, warehouses used for storage and distribution, and hospitals and clinics that dispense pharmaceutical products to patients. The concept demonstrator accommodates the ability to model four different facility types by means of the **Facility** agent. The first type is a *manufacturer* and this facility uniquely possesses the capability to manufacture pharmaceutical products. A manufacturer is classified as a tier-1 facility in the concept demonstrator. The second facility class is a *warehouse* and it is classified as a tier-2 facility. A warehouse orders inventory from an upstream facility and stores it temporarily, before distributing it to downstream facilities. In a health-care context, some facilities may be responsible for both dispensing pharmaceuticals to patients and for supplying stock to peripheral health-care facilities. Any facility that fulfils this role in the supply chain modelled is classified as a *hospital*. A hospital may order stock from manufacturers or warehouses upstream. Since a hospital is typically located



downstream from a warehouse, it is classified as a tier-3 facility. Finally, a *clinic* functions as a service point that dispenses pharmaceutical products to patients directly. A clinic may order stock from a hospital, a warehouse or a manufacturer, and is typically located the farthest downstream in a pharmaceutical supply chain and is therefore called a tier-4 facility. Although there are some functional differences between these facility types as defined above, there are many similarities between them that make it possible to employ the same model logic for modelling a selection of the same supply chain activities. The model logic for reviewing inventory levels, placing a purchase order and receiving a shipment are, for example, the same for any warehouse, hospital or clinic. For this reason, all of these facility types are modelled as `Facility` agents, but are distinguished from one another by a parameter called `tier`. This parameter value is used to specify which portions of the model logic are applicable to the respective facility types (based on their respective tiers). This generic modelling framework therefore makes it possible to model different supply chain network layouts with considerable ease.

The time unit of the simulation model concept demonstrator is chosen as days. For the purposes of this concept demonstrator, any time unit larger than days is presumed to be too coarse, while a more finely-grained choice such as hours may induce an intractable level of complexity. Consider the following scenario in a pharmaceutical supply chain. A patient visits a clinic at 09:00 on any given day and finds that the clinic does not have stock available of the product that he or she needs. In this case, the clinic would incur a stock-out and the patient's demand will remain unfulfilled. Suppose, however, that a delivery of the product in question is scheduled to arrive at the same clinic at 12:00 on the same day. Assuming that the shipment arrives on time and is processed in a timely manner, it may well be possible that the stock is available on the clinic's shelves later in the afternoon. If the same patient had rather arrived at 17:00 (and not at 09:00), for example, the clinic may have been able to fulfil his or her demand at the time. This example illustrates how the particular timing of events can influence the performance of a supply chain.

In an attempt to avoid the kind of complexity described above, a number of simplifying assumptions, in the form of business rules, are made in the concept demonstrator. These business rules pertain to inventory-related activities and are assumed to be applied uniformly by each facility, on a daily basis. The daily activities related to inventory management in any facility are executed according to the following sequence:

1. *Receive stock.* The first event to occur on any given day is the receipt of incoming deliveries. It is assumed that all of the in-transit shipments due to arrive on the current day are simultaneously received and processed at the start of the day. The newly received stock is considered to be immediately available for use on the same day that it was received.
2. *Fulfil demand.* After all of the incoming deliveries for the current day have been processed, manufacturers, warehouses and hospitals send away shipments in response to replenishment orders received from their downstream facilities. In the case of a hospital or a clinic, the daily patient demand is satisfied from the stock on hand. Unmet demand is recorded as stock-outs.
3. *Discard expired products.* The remaining shelf-lives of all stock on hand is evaluated and expired stock is discarded.
4. *Place order or manufacture.* After all incoming deliveries have been received and demand has been observed for the day, a facility decides whether or not to place a new order according to its replenishment policy. In the case of a manufacturer, the manufacturer decides whether or not to initiate a new production run based on its manufacturing policy.

The implementation of these business rules ensures that the maximum amount of stock is available at the exact time when demand needs to be fulfilled. In other words, a facility is given the best chance possible to satisfy demand, on a daily basis. The uniform application of these business rules may also illuminate the impact of information sharing on inventory management more clearly.

### 6.1.1 Model input

As mentioned, the simulation model concept demonstrator is generic in the sense that it accommodates any user-specified pharmaceutical supply chain configuration as model input. This is achieved by means of an extensive input data structure called the *input framework* that captures the information required to model the operation of a supply chain over time sufficiently accurately. This input framework should specify, amongst others, the supply chain topology, the flow direction of inventory between facilities and the nature of end-user demand. This input framework builds on the work of Du Plessis *et al.* [45], who proposed a preliminary design of such an input framework for the purpose of modelling a pharmaceutical supply chain network.

AnyLogic provides a built-in database function that allows a simulation model implemented in the software environment to read data from, and write data to, database tables during any simulation run. The input framework introduced in this section relies on this database function to store the model input data as provided by the user. The input framework comprises a total of nine database tables and the architecture of these tables are described in this section.

The first of these nine tables is called `table_facilities` and it captures some facility-specific information related to the facilities modelled as part of a pharmaceutical supply chain. This table contains a list of all the facilities modelled as `Facility` agents in the simulation model. A unique number (that serves as a primary key<sup>1</sup>) is assigned to each facility in the database table and this number also corresponds with the index of the agent in the `Facility` agent population. The index is a sequential number and starts at zero. A facility's name, its relevant supply chain tier and its location information are also stored in this table. A facility's location is not specified geographically, but simply as a position in the two-dimensional user interface of the simulation model. The convention used for classifying supply chain tiers in this thesis was described above. According to this table-based approach, the user is free to model a supply chain network of virtually any size (a database table in AnyLogic can store more than 50 000 records). The user can therefore decide whether he or she wants to model an entire supply chain network or only a particular portion of a supply chain. A summary of the attributes found in the `table_facilities` table is shown in Table 6.1.

Column name	Data type	Short description
<code>index</code>	integer	A unique identifier (primary key) to distinguish agents
<code>name</code>	string	The name of the facility
<code>tier</code>	integer	Specification of the relevant echelon
<code>x</code>	integer	$x$ -coordinate of the facility in a two-dimensional space
<code>y</code>	integer	$y$ -coordinate of the facility in a two-dimensional space

TABLE 6.1: *The structure of the `table_facilities` database table.*

A second table, called `table_products`, contains a list of all the different product types considered in the supply chain network modelled. Similar to how the respective facilities are indexed,

<sup>1</sup>A *primary key* is used to uniquely identify a record in a database table [82].

a unique index code (*i.e.* primary key) is also assigned to each product type. The product name and its expected shelf-life (in days) are stored in this table. These attributes and their respective data types are tabulated in Table 6.2. Although information pertaining to multiple product types may be included in the input framework, only one product type (selected by the user) is considered during a simulation run. In the concept demonstrator, the shelf-life of a product is considered to be the maximum number of days remaining before a product has to be sold, otherwise it is discarded. This is based on the argument that a health-care facility may choose not to dispense pharmaceutical products that are relatively close to expiration. If a product is, for example, discarded 60 days before actual expiry, the user may specify the shelf-life in the `table_products` table as *absolute shelf-life* – 60 days.

Column name	Data type	Short description
type	integer	A unique identifier to distinguish among products
name	string	The name of the product
shelflife	integer	The expected lifetime of the product

TABLE 6.2: *The structure of the table\_products database table.*

The dynamics of inventory flow in the supply chain are captured in the `table_connections` database table of the input framework. Each record in this table represents a connection between two facilities where one facility is classified as the ordering facility (the entity that places an order) and the other as the supplying facility (the entity that fulfils the order). These connections are established within the table using the unique facility indices defined in `table_facilities`, as foreign keys<sup>2</sup>. In the concept demonstrator, it is assumed that any supplier sends order shipments to its respective customers at fixed time intervals. This is achieved by recording a specific shipping policy for each supplier-customer pair defined in `table_connections` in the following way. For each supplier-customer pair, an initial shipping date is provided by the user — a value that specifies the first available date at which a supplier can ship to a customer. Additionally, the user is also required to specify a so-called *shipping interval* (measured in number of days) that specifies the number of days between two consecutive shipments from a supplier to a customer for a given supplier-customer pair. A shipping interval of size one, for example, implies that a supplier can ship to a customer on a daily basis. Once the current shipping date has lapsed during simulation execution, a new shipping date is generated by incrementing the old shipping date by the number of days specified as the shipping interval.

Since order picking and packing activities occur over time, a supplier may not necessarily have an order ready for shipment at the time of the next scheduled shipping date if, for example, the order was received too close to the shipping date. For instance, a supplier who receives a large order one hour in advance of the next scheduled shipping date, may most likely fail to prepare the order in time. To address this issue in the concept demonstrator, it is assumed that a supplier will initiate a new delivery only at the next immediate shipping date, provided that the corresponding order was received before a so-called *cut-off date* — a date value also specified in the `table_connections` table. It is therefore assumed that the necessary order picking and packing activities are completed in a timely manner between the cut-off date and the corresponding, subsequent shipping date. Similar to the process responsible for updating the shipping date value, the cut-off date is updated by incrementing the old value by the shipping interval. This implies that the time interval between the cut-off date and its corresponding shipping date remains constant throughout. When a facility can choose between multiple sup-

<sup>2</sup>A *foreign key* is non-key in one database table and refers to the primary key in another table. A foreign key is used to link two database tables together [82].

pliers, it is assumed that the facility will issue a new replenishment order to the supplier with the earliest shipping date. The shipping date attribute only specifies *when* a delivery between a supplier-customer pair is initiated, while the corresponding delivery lead time is determined separately. The length of the delivery lead time is modelled stochastically according to a triangular probability distribution. The minimum, maximum and mode parameter values of this distribution are specified in terms of whole-numbered days in three further columns. When a delivery is initiated during model execution, the delivery lead time value is sampled from a triangular distribution using the parameter values specified. It is assumed that the entire journey time and the time associated with the unloading of stock at the customer is included in the delivery lead time. In other words, when an order arrives at the customer (*i.e.* the lead time has lapsed), the inventory is immediately made available for use. It is also assumed that transport capacity is sufficient at all times in the supply chain. Finally, a boolean attribute called *neighbours* is employed to specify whether or not a particular connection exists between two neighbouring warehouses and/or hospitals. This is applicable to Scenarios 4–5 of §5.2.4 and §5.2.5, where a warehouse or hospital can choose to order from an alternative supplier in its own neighbourhood. The attributes of the `table_connections` table is further described in Table 6.3.

Column name	Data type	Short description
<code>ordering_facility</code>	integer	Customer index from <code>table_facilities</code>
<code>supplying_facility</code>	integer	Customer index from <code>table_facilities</code>
<code>cutoff_date</code>	date	The order deadline before the next shipment
<code>shipping_date</code>	date	The next shipping date for a supplier-customer pair
<code>shipping_interval</code>	integer	The number of days between two consecutive shipments
<code>min_leadtime</code>	integer	The minimum value of a triangular probability distribution
<code>max_leadtime</code>	integer	The maximum value of a triangular probability distribution
<code>mode_leadtime</code>	integer	The mode value of a triangular probability distribution
<code>neighbours</code>	boolean	True if the connection exists between two neighbours

TABLE 6.3: *The structure of the table\_connections database table.*

Another table that forms part of the input framework is the `table_inventory` database table. This table captures data pertaining to conventional inventory replenishment policies for each facility-product pair. A facility-product pair is specified in each row of this table using the respective primary keys of the `table_facilities` and `table_products` tables as foreign keys. The parameters of two inventory replenishment policies, namely the continuous-review and periodic-review policies, are further specified in this table. A reorder point and a reorder quantity may be specified under a continuous-review policy, while a reorder point, an order-up-to level and a review interval are provided for a periodic-review policy. A separate attribute called *replenishment\_policy* is used to specify which policy should be implemented during simulation execution for a particular facility-product pair. A continuous-review policy is indicated by 0, while a periodic-review protocol is specified by a value of 1. A summary of the `table_inventory` database table and its attributes is provided in Table 6.4.

The `table_manufacturers` database table is included in the input framework to record information pertaining to the production capabilities of manufacturers. Unlike a customer that places an order for goods to a supplier upstream, a manufacturer needs to engage in the activity of production when running relatively low on the finished product that it provides to its customers. For each manufacturer-product pair, a stock level that triggers a production run, called the *manufacturing point*, is captured in this table. This manufacturing point is analogous to the reorder point of a continuous-review inventory replenishment policy. Furthermore, the size of a single production batch (in number of product units), as well as the corresponding production

Column name	Data type	Short description
facility_index	integer	Foreign key from <code>table_facilities</code>
product_type	integer	Foreign key from <code>table_products</code>
replenishment_policy	integer	Determines which type of policy is implemented
reorder_point_r	integer	The reorder point under a continuous-review policy
reorder_point_q	integer	The reorder quantity under a continuous-review policy
review_interval	integer	The time interval applicable to a periodic-review policy
reorder_point_s	integer	The reorder point under a periodic-review policy
order_up_to_s	integer	The corresponding order-up-to inventory level

TABLE 6.4: *The structure of the table\_inventory database table.*

lead time (in days) associated with the production run are included in this table. It is assumed that a manufacturer always has a sufficient amount of raw material available for production. Additionally, it is assumed that the manufactured goods are available for use immediately after the production lead time has elapsed. The attributes of the `table_manufacturers` table are further described in Table 6.5.

Column name	Data type	Short description
facility_index	integer	Foreign key from <code>table_facilities</code>
product_type	integer	Foreign key from <code>table_products</code>
manufacturing_point	integer	Stock level that triggers a production run
batch_quantity	integer	The number of products included in a single batch
manufacturing_time	integer	The manufacturing lead time, specified in days

TABLE 6.5: *The structure of the table\_manufacturers database table.*

The `table_starting_inventory` database table captures the initial inventory levels with which the simulation model is initialised. Stock-quantity and stock-age information are recorded for each facility-product pair in this table. Stock-age information is specified in terms of its remaining shelf-life, measured in days. The amount of stock and its corresponding shelf-life are captured in the `inventory_level` and `remaining_life` columns, respectively. A two-dimensional array called `inventoryMatrix` is embedded in each `Facility` agent and this structure is used to store stock-level and stock-age data during simulation execution. When the simulation model is first initialised, the values of the `table_starting_inventory` table are transferred to the corresponding `inventoryMatrix` arrays. The values of these arrays are updated during model execution when stock is received into inventory, when stock is withdrawn from inventory and when expired stock is discarded from inventory. A summary of the attributes found in the `table_starting_inventory` table is shown in Table 6.6.

Column name	Data type	Short description
facility_index	integer	Foreign key from <code>table_facilities</code>
product_type	integer	Foreign key from <code>table_products</code>
remaining_life	integer	Remaining shelf-life measured in days
inventory_level	integer	The corresponding amount of stock

TABLE 6.6: *The structure of the table\_starting\_inventory database table.*

The concept demonstrator grants the user considerable freedom in respect of modelling patient demand as it occurs over time. Unique demand patterns may be specified per product, for

each hospital and clinic, by means of the `table_demand` database table. The user may choose any one of three probability distributions to model the daily patient demand as experienced by a clinic or hospital. These distributions are the gamma probability distribution, the uniform discrete probability distribution and the triangular probability distribution. These distributions are typically employed in the absence of empirical data, as mentioned in §3.5. AnyLogic has a built-in function that can sample a value from any of these three probability distributions. Although the gamma and triangular probability distributions are continuous, the sampled values are rounded to integer values in the model logic.

A total of eight columns are devoted to capturing the parameters of the respective probability distributions in the `table_demand` table. The first two of these seven columns are reserved for recording the values of the gamma probability distribution's shape parameter  $\alpha$  and rate parameter  $\beta$ . An additional column is employed to store a minimum value that is imposed on the gamma distribution. When a value is sampled from the gamma distribution during a simulation run, AnyLogic ensures that this value is strictly larger than, or equal to, this minimum value. The minimum value may, for example, be set to zero to ensure that demand is always non-negative. For the uniform discrete distribution, the user is required to specify the minimum and maximum values, respectively. Finally, in the case of the triangular distribution, the user may specify the values of the minimum, mode and maximum parameters. The user is free to implement any one of these distributions during simulation execution. While parameter values for all three distributions may be recorded in the table, an additional attribute is used to specify exactly which distribution should be employed during simulation run time. An integer value of 0 in the *distribution* column indicates that the daily demand for a facility-product pair should be sampled from the the gamma distribution. A value of 1 indicates that the uniform distribution should be used instead, while the triangular distribution is associated with a value of 2.

Given the variable nature of end-user demand, the user may specify different demand probability distributions (or the same probability distributions with different parameter values) for different periods of simulated time, called *demand periods*. This capability may be used to model demand fluctuations that may, for example, occur as a result of seasonality. Demand periods are numbered sequentially, starting from one, and the duration of each demand period (in number of days) is specified by the user. When the simulation model is initialised, the daily demand for each facility-product pair is sampled from the probability distribution associated with Demand Period 1. Once the first demand period's duration has lapsed, the active demand period becomes Demand Period 2, and so on. The attributes of the `table_demand` table and their corresponding data types are recorded in Table 6.7.

The eighth component of the input framework is the `table_neighbourhoods` database table. The purpose of this table is to define clinic neighbourhoods and their respective members based on the inventory-sharing scheme described in Scenarios 2–5 of §5.2. Each row in the database table corresponds to a neighbourhood-facility pair. Each neighbourhood is uniquely identified by its own neighbourhood number defined in the *neighbourhood* column. A facility is added to the corresponding neighbourhood by including its facility index in the *member* column. A summary of the attributes found in the `table_neighbourhoods` table is shown in Table 6.8.

The ninth and final element of the input framework is the `table_events` database table. This table is devoted to capturing information that may be used to model so-called special-case *events*. In the concept demonstrator, a special-case event is defined as any occurrence where a facility cannot place an order (or initiate a production run in the case of a manufacturer), for a specified period of time. This type of event may, for example, occur in practice due to equipment failure, a shortage of raw materials or even strike action. Failure to carry a sufficient amount of stock in a pharmaceutical supply chain may, of course, hold significant implications for patient health.

Column name	Data type	Short description
facility_index	integer	Foreign key from <code>table_facilities</code>
product_type	integer	Foreign key from <code>table_products</code>
demand_period	integer	Identifier associated with a period of demand
alpha	double	Shape parameter of the gamma distribution
beta	double	Rate parameter of the gamma distribution
minimum	double	Minimum value imposed on the gamma distribution
unif_min	integer	Lower bound on the uniform distribution
unif_max	integer	Upper bound on the uniform distribution
triangular_min	integer	Minimum value of the triangular distribution
triangular_mode	integer	Mode value of the triangular distribution
triangular_max	integer	Maximum value of the triangular distribution
distribution	integer	Distribution used during the relevant demand period
length	integer	The duration of the demand period in number of days

TABLE 6.7: The structure of the `table_demand` database table.

Column name	Data type	Short description
neighbourhood	integer	Neighbourhood identifier
member	integer	Foreign key from <code>table_facilities</code>

TABLE 6.8: The structure of the `table_neighbourhoods` database table.

The implementation of these special-case events makes it possible to evaluate the robustness of the relevant inventory management policies. The contents of the database table is summarised in Table 6.9.

Column name	Data type	Short description
event	integer	Unique scenario identifier
facility_index	integer	Foreign key from <code>table_facilities</code>
start_on_day	integer	Event starting time (in terms of simulated time)
length	integer	Duration of the event (in days)

TABLE 6.9: The structure of the `table_events` database table.

Each event is distinguished by a sequential number stored in the `event` column. The `facility_index` column is used to specify the index of the facility on which the event is imposed. Furthermore, the user is required to specify the day on which the event is initiated, as well as the duration of the particular event. An event for a warehouse may, for example, start on day 55 and last for five days. This means that the warehouse is unable to order stock from any supplier on simulated days 55–59.

### 6.1.2 Inventory replenishment orders

Inventory replenishment orders are modelled as `Order` agents, as described in §6.1. When any facility places a new replenishment order according to its inventory replenishment policy, a new `Order` agent is generated within the modelling environment. This agent object is sent instantaneously to the corresponding supplier using AnyLogic’s built-in messaging function. The `Order` agent contains, for example, information about the customer facility as well as the order quantity. Once the supplying facility has prepared the order for distribution to the customer, the

same **Order** agent is sent back to the ordering facility. In this case, however, the agent is not sent as a message but instead travels towards the customer over a period of time that corresponds with the associated delivery lead time. Only when the **Order** agent arrives at the customer, can the customer proceed to process the replenishment order. In the descriptive paradigm, it is assumed that a new replenishment order cannot be placed when at least one other order for the same product is already pending.

In some instances, it may happen that a supplying facility does not carry enough stock to fulfil the demand from downstream facilities in full. A warehouse may, for example, only have 500 product units in stock at a given time instant, when a total of 700 units is demanded from clinics in the form of replenishment orders. When the demand associated with an order cannot be met in full, a *backorder* is generated for the portion of unmet demand. For each newly generated backorder, a new **Order** agent is created and a boolean variable **backorder** is used to identify it as a backorder. In the concept demonstrator, it is assumed that the demand associated with backorders is fulfilled as soon as possible by a supplier. The shipment of a backorder is therefore not restricted to the shipping policy as specified in **table\_connections** of the input framework, as discussed in §6.1.1. The adoption of this business rule is based on the argument that it is the responsibility of a supplier to fulfil customer demand in full, and in a timely manner. The timely fulfilment of demand is especially important in a health-care context, given the potential adverse impact of stock-outs on patient health.

It may also happen that a supplier is faced with such a large number of backorders from several customers, that it cannot satisfy the total backorder demand with its current stock level. In such a scenario, the supplier may choose to ration stock amongst the customers to ensure that at least every customer is served to some extent. Alternatively, a supplier may prioritise the demand of some customers over others according to some predefined priority scheme. In the context of a pharmaceutical supply chain, it may be argued that any given facility cannot be left without stock for too long. In this thesis, it is therefore assumed that, in the case of stock shortages, the available stock is divided equally amongst customers.

The final type of order embedded in the concept demonstrator is called an *informal* order. An informal order is employed to model the informal exchange of stock between clinics in the same neighbourhood, as proposed in Scenario 2 of §5.2.2. When a clinic, for example, requests a certain number of products from a neighbouring facility, it is modelled as an informal order issued by the former to the latter. A boolean variable is employed to classify an **Order** agent as an informal order. It is assumed that any informal order can be fulfilled within one day of simulated time. If a clinic, for example, issues an informal order on any given day, the order will be fulfilled on the next day before that day's demand is incurred. This short lead time is motivated by the fact that clinics in the same neighbourhood are so close to one another that the impact of delivery lead time may be considered negligible. When a facility places an informal order, it does so in a greedy manner by ordering from the clinic (or clinics) with the most inventory available for exchange during the given time step.

It is assumed that all stock is dispensed according to the *first-expired-first-out* principle. Furthermore, any stock with a remaining shelf-life of less than the expected delivery lead time, will not be included in a new shipment. This is done to ensure that stock does not expire during transport when the delivery lead time is longer than one day.

### 6.1.3 The prescriptive paradigm

As mentioned, the prescriptive paradigm of the concept demonstrator is employed to discover effective inventory management policies for a pharmaceutical supply chain, by means of a rein-



forcement learning method. The reinforcement learning algorithm chosen for implementation in this thesis is Q-learning and the working of this algorithm was discussed in §4.2.3. More specifically, Q-learning is employed by every `Facility` agent modelled in the concept demonstrator in order to learn an inventory control policy. The learnt policy involves the following two aspects of inventory management: When to place an order and how much to order. The prescriptive paradigm furthermore facilitates the implementation of any one of the five information-sharing scenarios proposed in §5.2. Thus, Q-learning may be applied to learn inventory control policies based on a particular information-sharing arrangement.

The use of the prescriptive paradigm comprises three distinct phases. The first phase involves the population of the input framework, as described in §6.1.1. The `table_inventory` database table is the only table that does not have to be populated in order for the prescriptive paradigm to work successfully. During the second phase, the relevant information-sharing configuration is selected before the simulation model is executed. During the execution phase, Q-learning is applied to discover effective inventory management policies for the respective facilities. Finally, after the termination of the Q-learning process, the user may choose to implement the learnt policies in the concept demonstrator in order to evaluate their efficacy.

#### 6.1.4 The graphical user interface

The *graphical user interface* (GUI) designed in the concept demonstrator of this project allows a simulation operator or analyst to interact with a simulation model during its execution. Before a simulation experiment is initiated in the simulation model, the operator may use the *GUI* to define a selection of parameter values. The first of these parameters is the *model paradigm* — the operator may select either the *descriptive* paradigm or the *prescriptive* paradigm by means of a set of radio buttons. For the prescriptive paradigm, the simulation operator is furthermore required to specify whether a control policy should be learned or whether a learnt policy should be implemented for evaluation. Since the input framework accommodates more than one product type, the user has to select a specific product type for implementation in the simulation from a drop-down list. Thereafter, the user may specify the applicable information-sharing scenario by selecting the corresponding scenario number (1, 2, 3, 4 or 5) from a drop-down list. Finally, the length of the simulation run is specified by entering the desired length (in number of simulated days) in a designated text box.

During the execution of a simulation run, the *GUI* provides a visual representation of the pharmaceutical supply chain modelled. A screenshot of the *GUI* for the simulation model is shown in Figure 6.1. Each supply chain facility is represented by an icon of a building. A manufacturing entity is coloured yellow (denoted A in Figure 6.1), while a warehouse has a green colour (denoted B in Figure 6.1). Any hospital is represented in red (denoted C in Figure 6.1) and every clinic is coloured blue (denoted D–G in Figure 6.1). The black lines between facilities represent the supplier-customer connections as defined in `table_connections` of the input framework of §6.1.1. Furthermore, all the facilities that form part of the same neighbourhood are connected to each other by means of blue, dashed lines. In Figure 6.1, facilities D and E, and facilities F and G, reside within the same neighbourhood, respectively. The *GUI* also makes it possible to track the movement of orders during simulation run time. Any in-transit order is represented by a box icon that moves from a supplier to a customer over time. Any perfect order (an order that is delivered in full and on time) is denoted by a green colour (denoted H in Figure 6.1), while a partially fulfilled order is coloured yellow. Finally, a red box is used to indicate a backorder (denoted I in Figure 6.1). In Figure 6.1, a perfect order is moving from facility A towards facility B, and two backorders are shipped from facility C to facilities F and G, respectively.

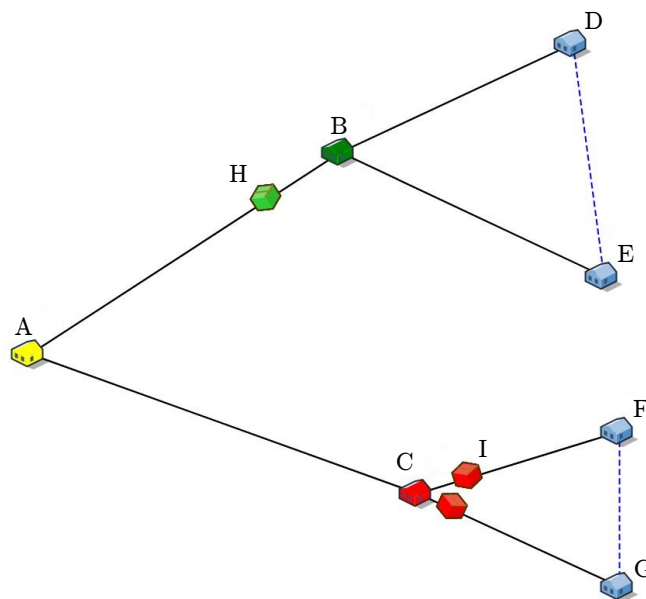


FIGURE 6.1: A screenshot of the animation portion of the simulation model GUI.

### 6.1.5 Model output

Supply chain performance data are saved and written to a Microsoft Excel file at the end of each simulation run. These data are the KPIs according to which the effectiveness of the inventory control policies, as determined by the reinforcement learning algorithm in the prescriptive paradigm, may be evaluated. These KPIs are employed to measure the effectiveness of the respective control policies for each individual facility, as well as for the supply chain as a whole.

Two KPIs are reported as model output. The first of these KPIs is the *total number of stock-outs*, which is simply the sum total of all the stock-outs incurred by a facility. Each unit of unmet demand is classified as a stock-out. A clinic that, for example, turns fifteen patients away without stock on any given day, would incur fifteen stock-outs. For manufacturers, warehouses and hospitals which distribute stock to downstream facilities, a stock-out is recorded for each product unit that had to be shipped on a given day, but was not shipped due to a shortage of stock. If a warehouse, for example, has to ship 500 units to a customer on any given day, but can only ship 300 units due to a lack of stock,  $500 - 300 = 200$  stock-outs are incurred on that day by the warehouse. The total number of stock-outs in the whole system is also provided as output. The second KPI is the *total number of expiries*, which is simply the sum total of product units that had expired before they could reach patients. Again, this KPI is broken down into the number of expiries per facility and the total number of expiries in the entire supply chain.

The performance of both the individual facilities and of the system as a whole are measured to ensure that the global KPI values do not conceal significant performance differences between the respective facilities. Consider, for example, the case where the overall number of stock-outs and the total number of expiries in the system are relatively large. This does not necessarily imply that each facility suffered from large numbers of stock-outs and expiries. Instead, the global KPI values could have been heavily influenced by the severe underperformance of only some facilities, while others performed relatively well.

With respect to stock-outs, it may also be of value, in particular, to evaluate the total number of end-user stock-outs incurred only by health-care facilities (*i.e.* hospitals and clinics), as opposed

to system-wide stock-outs. In a pharmaceutical supply chain context, it may be argued that stock-outs upstream in the supply chain (*i.e.* missed shipments) are tolerable provided that they do not compromise the service level of the health-care facilities downstream. Since inventory may be shared among facilities according to the various information-sharing scenarios described in §5.2, it is possible that stock-outs upstream do not necessarily lead to stock-outs at health-care facilities downstream. In such a case, a decision maker may be more interested in the performance of health-care facilities than the performance of the entire supply chain.

## 6.2 Model verification and validation

The aim in this section is to describe how some of the verification and validation techniques reviewed in §3.6 were applied to the simulation model introduced in §6.1. These techniques were employed continually throughout the model building process.

### 6.2.1 Verification of the simulation model

The objective of simulation model verification is to ascertain that the model is free of programmed and logical errors. Although there are several verification techniques available in practice, only the most prominent methods employed in this study are described in this section.

AnyLogic has a built-in debugger that was used extensively to verify the soundness of the programming code underlying the simulation model. Every time that the simulation model is built (prior to model execution), the debugger scans the code for logical and syntax errors. When such errors are detected, the output console provides information about the causes of these errors and points to their locations in the computer code. As part of the incremental model-building approach followed in this study, the debugger was employed after every addition of a new code section in order to verify the accuracy of the code. When the model has been compiled successfully, errors may, however, still manifest themselves during model execution. A common example of such an error is division by zero. When errors occurred during run time, the simulation model was suspended and the debugger again provided information pertaining to the cause and location of the error. The user-friendly nature of the debugger facilitated relatively quick and comfortable rectification of logical and syntax errors during the model-building process. Furthermore, all variables and code sections were supplemented with descriptive comments that outlined the function of the particular code.

Significant attention was also afforded to the process of ensuring that the input data recorded in the input framework of §6.1.1 were correctly translated to the simulation model during model start-up. This was achieved mainly through the use of animation and print statements. Pharmaceutical supply chain configurations of varying sizes were, for example, provided as input and the resulting network layouts were verified for correctness at the hand of the visual representation displayed in the *GUI*. This process was simplified by using colour to distinguish among the different facility types. If a supply chain with five clinics was, for example, specified in the input framework, it was expected that five blue-coloured facilities were shown in the *GUI*. Parameter and variable values read from the input framework were also printed out during model execution for verification purposes. The delivery lead times were, for example, verified in the following way. When a delivery lead time value was sampled from the relevant triangular distribution, the value was printed out to the model console. Since the movement of an order shipment was depicted visually in the *GUI*, the lead time was verified by comparing the sampled value with the length of simulated time that lapsed when the order had travelled from the supplier

to the customer. At the same time, animation proved valuable to verify whether orders were sent from the correct suppliers to the right customers. Perfect orders and backorders were also distinguished from one another with different colours.

Print statements were also frequently utilised to verify the values of variables that did not form part of the input framework. Examples of these variables include any given facility's stock level, the number of stock-outs incurred and the total amount of inventory ordered at any given time instant. AnyLogic also supports the capability to set variables to "visible" during simulation execution, allowing their values to be displayed visually in the *GUI*. This feature (known as variable tracing) made it possible to monitor variable values easily during simulation run time, instead of having multiple print statements in the computer code. Variable tracing and print statements finally also played a major role in the verification of the implementation of the Q-learning reinforcement learning algorithm. In particular, the learning rate, the exploration rate, the reward size and the newly calculated Q-value were printed out to enable manual verification. These values were verified by means of separate calculations performed by hand.

### 6.2.2 Validation of the simulation model

Model validation is the process followed to determine whether the model represents the real-world system sufficiently accurately. A selection of the validation methods discussed in §3.6 were employed in this study to ensure a valid simulation model.

Face validation was the principal validation technique employed during the model-building process. Face validation was specifically aimed at the nature of supply chain business processes, such as the picking and packing of orders, shipping policies, lead times and the management of backorders. Although the simulation model serves as a concept demonstrator, it is imperative that the model is still a reasonable representation of an actual pharmaceutical supply chain network and its operations.

A second validation technique involved performing a sensitivity analysis during which critical input parameters were varied, after which the model output was evaluated to establish whether the output had changed as expected. An example of this was varying the demand rate observed by a single clinic (or clinics) and observing the effect this had on the number of stock-outs incurred by the clinic(s). In the case where the demand rate is increased significantly and all other parameters are kept constant, it is expected that the number of stock-outs will increase. Conversely, if the demand rate is decreased, the number of stock-outs is expected to decrease. Similarly, the service level of any facility is expected to decrease when delivery lead times are made significantly longer, and *vice versa*. Multiple simulation runs (with random seeds) were performed to verify the validity of these expectations.

The final major validation method employed in this study was the analysis and interpretation of the simulation model results in order to validate the consistency of the model. This was achieved by comparing the output of several simulation runs for similarity, given that the input parameters were kept constant throughout. Because of the stochastic nature of lead times and customer demand, the model output is not expected to be identical, but relatively similar for comparable simulation runs. The aim in this particular validation process was to ascertain that the variance observed in the output values during these runs were not extreme. After a number of simulation runs had been performed, it was found that the model yielded consistently similar results in respect of various output statistics including the number of stock-outs incurred by a facility, the service level of each facility, the total amount of inventory ordered by a facility, and the total amount of inventory in the system.

### 6.3 Chapter summary

Various aspects of the simulation model concept demonstrator at the heart of this thesis were discussed in this chapter. The discussion opened with a detailed description of the concept demonstrator's input framework — an extensive construct used to capture the model input data that are required to model the operation of a user-specified pharmaceutical supply chain. Thereafter, the different types of inventory replenishment orders incorporated in the concept demonstrator were reviewed, before some important elements of the prescriptive paradigm were briefly highlighted. This was followed by descriptions of the simulation model *GUI* and model output, respectively. In the second part of the chapter, a selection of the model verification and validation techniques employed in this thesis were reviewed.



---



---

## CHAPTER 7

---

# Reinforcement learning in a pharmaceutical supply chain

### Contents

7.1	The state space . . . . .	97
7.2	The action space . . . . .	102
7.3	The reward function . . . . .	103
7.4	Learning rate . . . . .	106
7.5	Exploration rate and action selection . . . . .	106
7.6	Chapter summary . . . . .	109

The purpose of this chapter is to describe the formulation of the inventory management problem addressed in this thesis as a reinforcement learning problem. This problem pertains to a number of learning agents in a pharmaceutical supply chain that should learn effective inventory replenishment policies. Reinforcement learning for inventory management is implemented in the concept demonstrator of §6 and, specifically, in respect of the five information-sharing scenarios of §5.2. The inventory control problem is therefore formulated as a reinforcement learning problem for each of these scenarios in this chapter.

In §7.1, the state space design is delineated for each agent and for each of the five scenarios. This is followed in §7.2 by a description of each agent’s action space. The focus shifts in §7.3 to the reward function employed during the implementation of a reinforcement learning algorithm. The learning rate and exploration rate parameters found to be suitable for the problem instances are next described in §7.4 and §7.5, respectively. The chapter finally closes in §7.6 with of summary of the chapter contents.

### 7.1 The state space

The state space of an agent represents all the possible states that the agent can encounter, as mentioned in §4.2. At any given time instant, an agent’s state is therefore not necessarily representative of the absolute state of the environment, because the agent may have limited information about the environment. An agent’s state may consequently be described as a “perceived” state of the environment [147]. More specifically, the state in which each agent finds itself is a combination of a number of state variables, and its state space encapsulates the set of

all possible state variable combinations. Since the state of an agent is limited to its own perception of the environment, different agents in the same environment may perceive different states at the same time. In the context of the inventory management reinforcement learning problem, each agent should learn which action it should take when it encounters a particular state, at any given time. The state space of a particular agent is therefore representative of all possible information potentially available to the agent when it has to make a decision at any point in time. The size of such a state space of course correlates with the amount of information available to an agent. In terms of information sharing between agents in a pharmaceutical supply chain, the size of the state space of an agent involved in information sharing will increase as it is given access to more information (by means of the incorporation of new state variables).

The concept demonstrator of §6 accommodates four different facility types, namely a manufacturer, a warehouse, a hospital and a clinic. These entities are the decision makers and are therefore considered as the learning agents in the reinforcement learning problem. Based on the operational differences between these agent types, each agent perceives the environment differently, and is afforded its own state space formulation. The state space design for each of these four agents is described in this section for each of the five information-sharing scenarios of §5.2. The state space of each agent is carefully designed by capturing the most critical information in the least number of state variables possible, in order to simplify the reinforcement learning process.

### Scenario 1

According to Scenario 1 of §5.2.1, no information sharing takes place between any of the agents in the pharmaceutical supply chain. The state space of each agent, therefore, includes only local information pertaining to the agent itself. Each of the four agent types considered in the concept demonstrator of §6 has its own unique state space design, although some state variables are shared by more than one agent (although such variables may have different values for different agents).

Two state variables are shared by all four agent types in Scenario 1. The first of these two variables is the *own inventory level* and it represents the amount of inventory on hand at an agent at any given time instant. This state variable is selected because it provides the agent with an indication of how much stock is available to fulfil customer demand in the short-term future. The second state variable shared by all agents is the *number of own product expiries during the lead time*. This state variable represents the portion of the agent's current inventory level that will expire during the lead time. For manufacturing entities, the lead time in this context is the time that elapses between the initiation and completion of a single production run. For entities that have to order stock from upstream suppliers (*i.e.* warehouses, hospitals and clinics), the lead time is the expected supplier lead time. This state variable is included because it provides the agent with information pertaining to the age of the stock on hand. If the number of products expected to expire during the lead time is relatively large, it may increase the risk of stock-outs in the short-term.

Warehouse, hospital and clinic agents each has the state variable *amount of own inventory on order*. This state variable is included because it supplies the agent with information pertaining to the amount of stock already ordered, but not yet received, from its suppliers. A manufacturer has an analogous state variable, called *own inventory-in-production*, that reflects the number of units (if any) in production at the current time step. A further state variable is the *own inventory backlog* of each manufacturer, warehouse and hospital agent, respectively. This state variable is chosen because it provides the agent with information about unfulfilled demand (in the form of



orders) that still need to be recovered. The final state variable incorporated in the state space design of Scenario 1 is *own customer demand*, which is partitioned into two distinct components, based on facility type. Manufacturers, warehouses and hospitals experience demand in the form of orders from their customers, while hospitals and clinics experience demand in the form of end-user demand. For tier 1–3 facilities, the *own demand* state variable is the mean total customer demand (sum total of order quantities) experienced during the lead time, over the past two lead-time periods. A window size of two lead-time periods, as opposed to one period, is chosen because it may provide a smoother representation of the mean demand. A longer window size, on the other hand, is not considered because it may conceal valuable information about significant fluctuations in recent demand. Customer demand, in terms of end-user demand as experienced by hospitals and clinics, is captured in the *own end-user demand* state variable and is measured as a five-day moving average of daily demand. A five-day window size is selected because it provides a reasonable estimate of the current demand trend and it filters out noise from demand (which may be more pronounced for end-user demand than for demand from orders). Since a hospital distributes stock to both clinics and patients, both of these customer demand state variables are included in the state spaces of the hospital agents. The demand state variables are selected because they provide the agent with information on the rate at which inventory is depleted.

A summary of the state space design adopted in Scenario 1 is shown in Table 7.1. Each manufacturer agent has a total of five state variables and each warehouse agent also has five state variables. Each hospital agent has the largest number of state variables, namely six, while each clinic agent only has four state variables. All of these state variables are preserved in Scenarios 2–5.

State variable	Manufacturer	Warehouse	Hospital	Clinic
Own inventory level	Yes	Yes	Yes	Yes
Own expiries during lead time	Yes	Yes	Yes	Yes
Own backlog	Yes	Yes	Yes	No
Own inventory in production	Yes	No	No	No
Own inventory on order	No	Yes	Yes	Yes
Own demand (orders)	Yes	Yes	Yes	No
Own end-user demand	No	No	Yes	Yes

TABLE 7.1: The state space design of each agent according to Scenario 1. The inclusion of a state variable in an agent’s state space is indicated by a ‘Yes.’

## Scenario 2

The notion of inventory sharing between clinics in a neighbourhood was introduced in Scenario 2, as described in §5.2.2. According to this information-sharing arrangement, each clinic has visibility over all incoming shipments that are scheduled to arrive within the next 24 hours at any point in time. Furthermore, each clinic shares information involving the amount of inventory it has available for exchange with its neighbouring facilities. As a result, there are two new instances of information (in the form of state variables) that should be incorporated into the state spaces of each clinic agent according to Scenario 2. The respective state space designs of the manufacturer, warehouse and hospital agents remain unchanged from Scenario 1.

The *own effective-inventory-level* state variable is the sum of a clinic’s current inventory level and the amount of stock scheduled to arrive within the next 24 hours, less the amount of inventory reserved already for sharing with its neighbours at the current time step. This state

variable is included because it provides more comprehensive information about the inventory that will be available to a clinic agent on the following day, so as to satisfy that day's expected demand. Access to this information may help each clinic agent to decide whether or not it needs to engage in inventory sharing at the current time instant. Since the effective inventory level includes information about each agent's current inventory level implicitly, it renders the original inventory-level state variable somewhat redundant and the latter is therefore discarded. The second new state variable is the *own effective neighbourhood inventory* and it reflects the total, effective amount of stock available in an agent's neighbourhood for sharing. A clinic that experiences demand on a daily basis should typically not be allowed to share all of its inventory with neighbouring clinics. For this reason, it is assumed that each clinic can only make inventory available for sharing if the sum of its current inventory level and incoming stock exceeds a predefined threshold. The threshold value is chosen as the maximum expected daily demand for the current demand period. This expected value is inferred from the demand information provided in the model input framework of §6.1.1. If a clinic's inventory exceeds this threshold, the balance is made available for sharing. The effective neighbourhood inventory state variable is selected because it may help each agent to decide how much (if any) inventory it should request from neighbouring facilities. An outline of the state space design adopted in Scenario 2 is shown in Table 7.2.

State variable	Manufacturer	Warehouse	Hospital	Clinic
Own inventory level	Yes	Yes	Yes	No
Own effective inventory level	No	No	No	Yes
Own expiries during lead time	Yes	Yes	Yes	Yes
Own backlog	Yes	Yes	Yes	No
Own inventory in production	Yes	No	No	No
Own inventory on order	No	Yes	Yes	Yes
Own demand (incoming orders)	Yes	Yes	Yes	No
Own end-user demand	No	No	Yes	Yes
Own effective neighbourhood inventory	No	No	No	Yes

TABLE 7.2: The state space design of each agent according to Scenario 2. The inclusion of a state variable in an agent's state space is indicated by a 'Yes.'

### Scenario 3

In Scenario 3, distributors (*i.e.* warehouses and hospitals) have visibility over the inventory levels of their customers and over the end-user demand experienced by those customers, as discussed in §5.2.3. A warehouse that, for example, supplies inventory to five clinics will have access to both the aggregate inventory level of, and the end-user demand experienced by, those five clinics. Compared with Scenario 2, it is the respective state spaces of the hospital and warehouse agents that are expanded in this scenario.

The total amount of inventory held by each distributor's customers at any given time instant is captured in the *customer clinics' inventory levels* state variable. This state variable is selected because it provides each distributor with an indication of the total amount of inventory available at its customers at any given time. If this aggregate inventory level is relatively low, the distributor may anticipate a number of incoming orders from its customer clinics in the short term. The second new state variable is *customer clinics' demands* and it indicates the mean daily demand (expressed as a five-day moving average) experienced by an agent's customer clinics. The inclusion of this state variable is motivated by the fact that it provides information pertaining to

the magnitude of demand experienced by the clinics. When patient demand is relatively high, a warehouse or a hospital may expect larger and/or more frequent orders from its customers. An outline of the state space design for each agent adopted in Scenario 3 is shown in Table 7.3.

State variable	Manufacturer	Warehouse	Hospital	Clinic
Own inventory level	Yes	Yes	Yes	Yes
Own effective inventory level	No	No	No	Yes
Own expiries during lead time	Yes	Yes	Yes	Yes
Own backlog	Yes	Yes	Yes	No
Own inventory in production	Yes	No	No	No
Own inventory on order	No	Yes	Yes	Yes
Own demand (incoming orders)	Yes	Yes	Yes	No
Own end-user demand	No	No	Yes	Yes
Own effective neighbourhood inventory	No	No	No	Yes
Customer clinics' inventory levels	No	Yes	Yes	No
Customer clinics' demands	No	Yes	Yes	No

TABLE 7.3: The state space design of each agent according to Scenario 3. The inclusion of a state variable in an agent's state space is indicated by a 'Yes.'

#### Scenario 4

The fourth information-sharing scenario considered for analysis in this thesis was discussed in §5.2.4. In Scenario 4, intra-neighbourhood inventory sharing is now possible between hospitals and warehouses, and manufacturers share their inventory levels with their direct customers. The state spaces of the warehouse and hospital agents are therefore enlarged in this scenario.

The first new state variable for each hospital and warehouse agent is the *own effective neighbourhood inventory*. This state variable is a measure of the availability of inventory for sharing within an agent's particular neighbourhood. At any given time instant, the effective neighbourhood inventory is the sum of all the inventory held at an agent's neighbours, and the neighbours' total inventory on order, less the amount of inventory carried in backlog by the neighbouring facilities. This state variable is chosen so as to help an agent decide whether or not intra-neighbourhood inventory sharing is a feasible option for its particular situation, at the given time instant. If the amount of backlogged inventory is sufficiently large, the effective neighbourhood inventory value may be negative.

The second additional state variable is the *manufacturer's effective inventory level* which is the sum of the amount of inventory held by a manufacturer and the amount of inventory in production, minus the inventory in backlog. This state variable is included because it conveys information about the potential ability of a manufacturer to supply inventory to a warehouse or a hospital in the short term. If a warehouse agent, for example, sees that its manufacturer's inventory level is relatively low, it may choose to rather order stock from a neighbouring warehouse or hospital (with sufficient inventory). A summary of the state space adopted in Scenario 4 is provided in Table 7.4.

#### Scenario 5

Scenario 5 is the fifth and final information-sharing scenario considered in this thesis and its architecture was described in §5.2.5. Out of all five information-sharing scenarios, Scenario 5 is

the only scenario where information from other facilities is shared with manufacturing agents. The state space of the manufacturer is expanded in this scenario, while the state spaces of the other three agent types remain unchanged.

State variable	Manufacturer	Warehouse	Hospital	Clinic
Own inventory level	Yes	Yes	Yes	Yes
Own effective inventory level	No	No	No	Yes
Own expiries during lead time	Yes	Yes	Yes	Yes
Own backlog	Yes	Yes	Yes	No
Own inventory in production	Yes	No	No	No
Own inventory on order	No	Yes	Yes	Yes
Own demand (incoming orders)	Yes	Yes	Yes	No
Own end-user demand	No	No	Yes	Yes
Own effective neighbourhood inventory	No	Yes	Yes	Yes
Customer clinics' inventory levels	No	Yes	Yes	No
Customer clinics' demands	No	Yes	Yes	No
Manufacturer's inventory level	No	Yes	Yes	No

TABLE 7.4: The state space design of each agent according to Scenario 4. The inclusion of a state variable in an agent's state space is indicated by a 'Yes.'

According to this particular configuration, clinics share their inventory levels and their end-user demand information with their upstream manufacturer(s). These two instances of shared information are captured in the *customer clinics' inventory levels* and *customer clinics' demands* state variables, respectively. The customer clinics' inventory levels state variable comprises the sum total of inventory held at all of the clinics to whom the particular manufacturer's primary customers supply. The clinics' demand are measured as a five-day moving average of the daily demand experienced by those same clinics. These state variables are included because they may help a manufacturing agent to act proactively in the case of extreme fluctuations in end-user demand. Furthermore, warehouse inventory level information is shared with manufacturers in the *customer warehouses' inventory levels* state variable. This value is the aggregate total of inventory held by a manufacturer's primary customers (*i.e.* warehouses and/or hospitals). This state variable is chosen because it provides the manufacturer with information pertaining to the potential ability of its primary customers to fulfil the demand of those clinics located further downstream. An outline of the state space design adopted in Scenario 5 is shown in Table 7.5.

## 7.2 The action space

The action space for each agent comprises a set of actions  $\mathcal{A}$ , where each action corresponds to a specific order quantity or production batch size, depending on the facility type. Since it is assumed that a manufacturer always has sufficient raw materials available for production (as mentioned in §6.1.1), a manufacturing agent's only actions are to choose the number of units to manufacture during a new production run. A warehouse, a hospital and a clinic, on the other hand, must each decide how much inventory it should order from its primary supplier at any given time step. Consider, for example, the set of actions  $\mathcal{A} = \{0, 100, 200\}$ . The first action,  $a_1 = 0$ , denotes a decision not to place an order or, in the case of a manufacturer, not to initiate a new production run. The second and third actions are associated with order or manufacturing quantities of 100 and 200, respectively. In the reinforcement learning problem considered in this thesis, each agent type (*i.e.* manufacturer, warehouse, hospital and clinic) has its own unique action space.

State variable	Manufacturer	Warehouse	Hospital	Clinic
Own inventory level	Yes	Yes	Yes	Yes
Own effective inventory level	No	No	No	Yes
Own expiries during lead time	Yes	Yes	Yes	Yes
Own backlog	Yes	Yes	Yes	No
Own inventory in production	Yes	No	No	No
Own inventory on order	No	Yes	Yes	Yes
Own demand (incoming orders)	Yes	Yes	Yes	No
Own end-user demand	No	No	Yes	Yes
Own effective neighbourhood inventory	No	Yes	Yes	Yes
Customer clinics' inventory levels	Yes	Yes	Yes	No
Customer clinics' demands	Yes	Yes	Yes	No
Manufacturer's inventory level	No	Yes	Yes	No
Customer warehouses' inventory levels	Yes	No	No	No

TABLE 7.5: The state space design of each agent according to Scenario 5. The inclusion of a state variable in an agent's state space is indicated by a 'Yes.'

The exact structure of an agent's action space is determined by the particular information-sharing scenario implemented in the concept demonstrator of §6. For agents that are eligible for inventory sharing in their neighbourhoods, their action spaces have to be adapted to account for smaller-than-usual order quantities. Considering that inventory-sharing clinics choose their actions on a daily basis, it would be illogical for a clinic to request a large amount of stock from a neighbour when the daily demand is considerably lower in comparison. It is therefore argued that an inventory-sharing facility would most probably issue smaller replenishment orders to a neighbour, compared with the order quantities typically issued to its primary supplier(s). A distinction is therefore made between *formal* replenishment orders (ordering from primary suppliers) and *informal* replenishment orders (ordering from a neighbour). Since a manufacturing agent does not engage in any inventory sharing according to the five information-sharing scenarios of §5.2, it is the only agent with an invariant action space over all five scenarios.

For inventory-sharing facilities, such as warehouses, hospitals and clinics, the action space may effectively be split into two sets,  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . The actions in  $\mathcal{A}_1$  are reserved for formal replenishment orders issued to its primary, default supplier(s). The action set  $\mathcal{A}_2$ , on the other hand, contains the actions associated with the order quantities of informal replenishment orders issued to neighbours.

### 7.3 The reward function

In commercial supply chain management, performance is often measured in terms of financial cost, as mentioned in §2.8. In the context of pharmaceutical supply chains, it may, however, be argued in some cases that the successful fulfilment of demand supersedes monetary savings. The latter perspective is adopted in this thesis and the performance of a pharmaceutical supply chain is measured only in terms of the number of stock-outs and product expiries that occur during a particular time period. The manner in which supply chain performance is measured, holds significant implications for the design of the reward function considered in this reinforcement learning problem.

The objective of each reinforcement learning agent considered in this problem is to minimise stock-outs and product expiries locally. This is incorporated in the reward function of an agent

by assigning a relatively large negative reward (*i.e.* punishment) to each unit stock-out and unit expiry. Since an agent aims to maximise its cumulative reward, this will encourage the agent to order sufficient inventory at each time step in order to avoid stock-outs and expiries.

Although monetary cost is not considered as a performance measure indicator, its impact cannot be disregarded by the reinforcement learning algorithm. If an agent is rewarded (punished) only for stock-outs and expiries, the agent may learn to maximise its inventory level (especially in the case of significantly long product shelf-lives) at all times. This could be achieved by ordering excessive amounts of inventory at regular intervals. In order to ensure that the Q-learning algorithm learns a practical policy, it is necessary to consider inventory costs to some extent. A *practical* inventory policy is considered one for which inventory levels are never excessively high, and order quantities are in accordance with the current demand. Furthermore, orders should typically not be placed too frequently, because it may lead to excessive order and transportation costs. In order to encourage an agent to learn a practical inventory policy, a negative reward is awarded for each product unit held in inventory at each discrete time step. Assigning such a holding cost may deter an agent from placing excessively large orders and also from placing new orders when the current inventory level is relatively high. To ensure that the holding cost punishment does not overshadow the stock-out and expiry punishments, it is suggested that this punishment is made relatively small.

An intriguing characteristic of the inventory management problem is that the full effect of an ordering decision is not always immediate. Typically, the ordered inventory will arrive only a number of time steps after the order was placed (because of the supplier lead time). This implies that the action of ordering a set number of units may lead to a significant increase in the holding cost on the day that the inventory is received (provided that the demand is relatively low). In other words, the agent's reward is delayed and it is imperative that the reinforcement learning algorithm recognises this phenomenon. A reinforcement learning algorithm that takes delayed reward into account, should be able to learn that larger order quantities may lead to larger holding costs in the future. If the stock-out and expiry punishments are considerably larger than the holding cost punishment, an agent is expected to learn to avoid unnecessarily large order quantities, whilst still minimising stock-outs and expiries. It may, however, happen that an agent learns to order in relatively small (albeit sufficient) quantities, but too frequently. In practical terms, this may be extremely uneconomical in terms of ordering and distribution costs. It is therefore considered more desirable for an agent to order relatively less frequently. In order to allow an agent to learn this type of behaviour, it is punished for placing a new replenishment order if it had already placed at least one other order that has not been received yet. In other words, the agent is punished if it chooses to order when the amount of inventory already on order is greater than zero.

In order to accelerate learning, an agent may be punished even more severely if it chooses to order when the amount of inventory already on order is relatively large. Additionally, an agent may also be punished if it issues a new replenishment order when its current inventory level is relatively high. Likewise, it may be unnecessary for a manufacturer to initiate a new production run if its current inventory level is excessively high and/or the amount of inventory in production is relatively high. Although the holding cost and ordering punishments described above may be sufficient for learning to avoid the above-mentioned type of behaviour, this process may be accelerated considerably by awarding a relatively large punishment when an agent chooses to order in these states. This may allow the agent to learn the most effective actions for these states much quicker, whilst ensuring that it does not spend too much time in the excessively high inventory states needlessly.

Table-based reinforcement learning algorithms (such as Q-learning) rely on discretised state spaces. In other words, each state variable has to be discretised into several intervals. As a result, the magnitude of each reward and each state variable interval may have a profound impact on the performance of a reinforcement learning algorithm. Suppose, for example, that a reward of  $-1$  is assigned for each product unit held in inventory at any given time instant. Furthermore, assume that the inventory-level state variable is discretised in equally-sized integer intervals of magnitude 50. If the inventory level of an agent is, for example, 20 (this value falls in the interval 0–49), a reward of  $20 \times (-1) = -20$  is assigned to the agent at the current time step. Depending on the specific inventory level, however, the reward for holding cost may range from  $-49$  to 0 for the same state variable. This variability may have a significant influence on the performance of the reinforcement learning algorithm if the magnitudes of the other state variables' intervals and rewards are not chosen appropriately.

The author found that the Q-learning algorithm struggled to differentiate sufficiently between different actions (*i.e.* different order quantities) when the state space discretisation was too coarse. For instance, if it was known (theoretically) that the optimal action for a given state was to place a replenishment order, the agent occasionally learnt not to place an order. In most cases where this phenomenon was observed, it led to future stock-outs (because the agent failed to order in a timely fashion). Closer inspection revealed that there is relatively little difference between the  $Q$ -values for the respective state-action pairs for those states. Based on empirical experimentation, the author found that the best method for mitigating this problem was to adopt a finer discretisation of the inventory-level state variable, and to assign much larger punishments for stock-outs.

The principal aim of the inventory-sharing schemes investigated in this thesis is to provide an alternative ordering method for facilities experiencing critically low inventory levels. In other words, an informal order should be issued only when a formal replenishment order (the primary option) may not suffice, as explained in §5.2.2. Informal order quantities are typically smaller than formal order quantities, and this implies that punishment based on holding cost (as discussed above) may be much smaller for informal orders than for formal orders. The lead times associated with informal orders are also shorter than the lead times of formal replenishment orders, as discussed in §6.1.2. This may encourage an agent to prefer informal orders over formal replenishment orders, even when its current inventory level is relatively high. This is undesirable because it may deplete the inventory levels of the supplying neighbours unnecessarily over a sustained period of time. It is therefore recommended that the reward function assigns a relatively large negative reward to any informal ordering action. This punishment should typically be larger than the punishment associated with a formal replenishment order so as to encourage the agent to choose informal ordering only when absolutely necessary, but smaller than the stock-out punishment in order to encourage the agent not to neglect the possibility of an informal ordering action in the face of an impending stock-out.

The reward awarded to each learning agent is calculated in this thesis as

$$r(t) = -1h - 600s - 600e - p(j)(y) - k(1 - y) - z(m), \quad (7.1)$$

where  $h$  denotes the number of product units held in inventory at the end of time step  $t$ ,  $s$  denotes the number of stock-outs incurred during time step  $t$ , and  $e$  is the number of expiries during time step  $t$ . A reward of  $-1$  is therefore awarded for each unit held in inventory during any given time step. Each unit expiry and unit stock-out, on the other hand, is awarded a reward of  $-600$ . This punishment is 600 times larger than the punishment for one unit held in inventory. Based on empirical observations, a ratio of this magnitude was found suitable for learning behaviour that minimises stock-outs while not disregarding the costs typically associated with holding

large amounts of inventory. Furthermore,  $p$  denotes the reward awarded to an agent for placing a new replenishment order when at least one other replenishment order is already pending. In (7.1),  $j$  is a binary variable taking a value of 1 when inventory is already on order at the start of time step  $t$ , or 0 otherwise. A second binary variable,  $y$ , takes a value of 1 when the action chosen at time step  $t$  is either a formal order or a decision not to order, or 0 in the case of an informal order. The variable  $k$  is the fixed reward awarded to an agent for placing an informal order at time step  $t$ . Finally,  $z$  denotes the fixed reward awarded to an agent if it chooses to order (or manufacture) when its current inventory level and/or amount of inventory on order (or in production) is considered excessively high at time step  $t$ . The binary variable,  $m$ , has a value of 1 if the agent should be given reward  $z$  at time  $t$ , or 0 otherwise. The decision maker should specify the particular states in which  $z$  is to be awarded explicitly beforehand. The values of  $p$ ,  $k$  and  $z$  in (7.1) differ for each agent type and are determined empirically based on the experiments conducted later in this thesis.

## 7.4 Learning rate

Watkins and Dayan [156] demonstrated that the Q-learning algorithm converges to optimal  $Q$ -values if a suitably decreasing learning rate is employed, as long as the sum

$$\sum_{i=1}^{\infty} \alpha_{n^i(s,a)}, \quad (7.2)$$

diverges, irrespective of whether or not the sum

$$\sum_{i=1}^{\infty} (\alpha_{n^i(s,a)})^2, \quad (7.3)$$

diverges, for all state-action pairs. The index of the  $i$ -th time at which the state-action pair  $(s, a)$  is visited, is denoted by  $n^i(s, a)$ . The method according to which the learning rate is determined in this thesis, is given by

$$\alpha_n^i(s, a) = \left( \frac{1}{1 + i(1 - \gamma)} \right)^{0.95}, \quad (7.4)$$

where  $i$  denotes the index of the  $i$ -th visit to the specific state-action pair  $(s, a)$  and  $\gamma$  denotes the discount factor (described in §4.2.2). In this thesis, the discount factor is set to 0.99. This relatively large discount factor is chosen due to the fact that the inventory control reinforcement learning problem involves both immediate and delayed rewards that are all considered equally significant.

## 7.5 Exploration rate and action selection

Establishing a suitable trade-off between exploration and exploitation during action selection is critical for the performance of reinforcement learning algorithms, as stated in §4.2. In order to achieve this delicate balance, the  $\epsilon$ -greedy method, described in §4.2.1, is implemented in this thesis. The exploration rate is determined as

$$\epsilon(s) = \max \left\{ 0.03, \left[ \frac{1}{1 + \frac{1}{15} \frac{1}{N_a(s)} \sum_{i=1}^a i(s)} \right] \right\}, \quad (7.5)$$



where  $N_a(s)$  denotes the number of actions  $a$  available to the agent when it perceives the system to be in state  $s$ , and  $i(s)$  denotes its total number of visits to state  $s$ . This state-dependent exploration rate encourages exploration in the case where a state has not been visited many times, but encourages exploitation as the number of visits to the state increases. A popular strategy involves preventing the exploration rate from decaying all the way to zero, but instead to a relatively small (yet greater than zero) value. This allows an agent to explore sporadically in any state that has been visited a large number of times already. This occasional exploration may provide the agent with new information that may enhance its learning performance. A minimum exploration rate of 0.03 is therefore set for all experiments. When an agent explores, it chooses uniformly between all of the available actions.

Preliminary experiments involving the Q-learning algorithm, however, revealed the following problem. When an agent is exploring, it is more likely to choose an action that involves ordering than choosing the action of not ordering. This is because the action space contains only one action that corresponds with not ordering (order quantity of 0), whereas the other actions all have a order quantity greater than zero. Hence, when the agent is exploring the state space, it tends to place replenishment orders at consecutive time steps. This initially leads to a dramatic increase in the agent's inventory level and the inventory level very quickly becomes excessively high. The agent therefore typically proceeds to spend a large amount of time in states that are associated with relatively high inventory levels. Over time, the agent eventually learns that it is carrying too much inventory (based on the negative holding cost reward) and that it is more effective to refrain from ordering when perceiving high-inventory states. When the agent does not order for a number of consecutive time steps, the inventory level decreases gradually over time due to user demand. In this case, the agent may visit a new state where the inventory level is slightly lower than before, but still relatively high. Since the agent has not visited this new state many times before (if at all), it will start to explore and, again, trigger a sustained period of ordering. Once more, the agent will eventually learn that the best action for these new, still relatively high inventory level state(s), is to refrain from ordering. Subsequently, the inventory level declines over time until a new, lower-inventory state is encountered and the agent starts to explore. This phenomenon of sustained ordering followed by a period of no ordering and a decrease in inventory level often continues iteratively during the learning phase.

A major problem with such excessive ordering exploration prevents the agent from visiting the lower inventory level states sufficiently many times (if at all) during learning. The inventory level tends to decrease gradually over time, as discussed above, and eventually the agent encounters an intermediate state where the inventory level is neither extremely low nor excessively high. In the vicinity of this intermediate inventory level, the agent may incur stock-outs if it does not order in a timely manner. If the agent does not order when perceiving such state(s), the inventory level may become extremely low and even reach zero. In this new, low-inventory state, the agent explores again and the inventory level increases swiftly to the familiar (already-visited) intermediate inventory level states. As a result, the agent barely (if at all) encounters the significantly low inventory level states, and therefore does not learn the appropriate actions for those states. In most cases, the agent tends to choose irrational actions when the inventory level is extremely low during policy implementation. In the worst case, it may happen that the agent mistakenly learns not to order when the inventory level is low (or zero). When this phenomenon is observed, the agent is effectively trapped in an infinite, self-reinforcing loop during which it persistently chooses not to order despite incurring frequent stock-outs.

Not only is the implementation of the Q-learning algorithm, as discussed above, insufficient for learning an optimal (or near-optimal) policy, it is also extremely time-consuming. In order to address these problems, the following mechanisms are adopted during the action-selection pro-

cess. The first countermeasure is to discourage an agent from ordering under certain conditions. This is achieved by awarding a relatively large punishment when an agent decides to order when its inventory level and/or amount of inventory on order is excessively high, as mentioned in §7.3. In the context of the experimental design employed later in this thesis, the onus rests on the analyst to decide which values of the inventory-level and inventory-on-order (or inventory-in-production) state variables are considered as ‘too high.’ In this thesis, a state variable value is considered excessively high for the perceived state(s) corresponding to the state space interval with the highest value. If the inventory-level state is, for example, discretised into five equally-sized integer intervals, each of cardinality 50 (starting at 0–49), the agent is awarded a large punishment for ordering when its inventory level is between 200 and 249 (*i.e.* in the fifth interval). It is important that the discretisation of these relevant states is performed appropriately (to contain sufficiently many intervals), so as to ensure that this reward structure does not compromise the performance of the Q-learning algorithm. Therefore, the last inventory-level (or inventory-on-order) interval should involve an inventory amount so high, that the agent should never be expected to order in that particular state.

The second countermeasure is aimed at ensuring that the agent visits the lower inventory-level states sufficiently many times during learning. Arguably the most natural method for allowing the inventory level to decrease is to refrain from ordering for a sustained period of time. Since this is extremely unlikely to happen in the lower inventory level states (as explained above) during exploration, it is imposed onto the agent during learning. This is done by forcing the agent to choose the action of not ordering ( $a_1 = 0$ ) for a successive number of time steps during a period called a *no-ordering streak*. Once such a streak has lapsed, the agent may find itself in a new, low-inventory state that it may not have visited otherwise. During the learning phase, the agent is exposed to many of these no-ordering streaks. These streaks occur at random times and the length of each streak is also stochastic. The timing and the length of these streaks should, however, be chosen carefully in order to ensure that it does not compromise the performance of the reinforcement learning algorithm.

During a sufficiently long no-ordering streak it may, for example, happen that an agent is stuck in the exact same state for a number of consecutive time steps. This state is most likely the state corresponding to an inventory level of zero. Importantly, each time step spent in any perceived state counts as a state visit. If the no-order streaks occur too frequently, and are too long, the agent’s visits to that particular state may very quickly increase dramatically. Although the number of state visits increases considerably, it may not be representative of an equal exploration of the action space. It is most likely that the bulk of the state visits involve only one state-action pair — the not-ordering state-action pair. This has significant implications for a state-dependent exploration rate which encourages exploitation as the number of visits to a state increases. The implementation of no-ordering streaks may therefore inflate the number of state visits unnaturally, leading the Q-learning algorithm to exploit long before all the available actions could be chosen sufficiently many times.

Considering the implications discussed above, it is suggested that a relatively slowly-decreasing exploration rate is employed when no-ordering streaks are implemented. This will encourage a more even exploration of the action space and ensure that the agent does not converge on exploitation too quickly for any given state. Furthermore, it is suggested that the number of, and the length of, no-ordering streaks imposed on an agent be limited so as to ensure that the artificially-induced action selection does not continue indefinitely. The final consideration involves determining the relative timing of the implementation of a set of no-ordering streaks. For instance, no-ordering streaks may be implemented at the very start of the learning process, or only after a large number of iterations. If no-ordering streaks are implemented too early, and

too frequently, it may lead to ineffective state space exploration in some cases, as mentioned. Therefore, it is considered safer to impose no-ordering streaks only after a large number of iterations had been completed, as it will allow the Q-learning algorithm to learn more “naturally” for longer. The implementation of no-ordering streaks enhances the robustness of the Q-learning algorithm since it provides a technique for exploring parts of the state space that may have not been explored otherwise.

Criticism may, however, be levelled at the implementation of no-ordering streaks, because it appears to infringe upon the spirit of reinforcement learning. By forcing an agent to choose a specific action, it deprives the reinforcement learner from an opportunity to learn ‘naturally.’ Subsequently, the question arises as to whether or not reinforcement learning (without no-ordering streaks) may be sufficient for learning inventory management policies in general. Answering this question falls, however, outside the scope of the work covered in this thesis. Based on empirical observations, the author found that, in the context of the experiments conducted in thesis, no-ordering streaks may be disregarded if an agent’s action space and state space is relatively small. For a large number of actions and states, on the other hand, an agent may find it considerably more difficult to visit low inventory states sufficiently many times.

## 7.6 Chapter summary

The inventory management problem considered in this thesis was formulated as a reinforcement learning problem in this chapter. This formulation was presented in respect of the five information-sharing scenarios introduced in §5.2. The state space designs of the four different agent types were first described for each information-sharing scenario and this was followed by an outline of the agents’ action spaces. The design of the reward function was described next and special mention was made of the challenges encountered during the development of this function. The expression employed for determining the learning rate in this thesis was provided next. Finally, the method for determining the exploration rate, and the action-selection policy adopted in this thesis were described. This discussion included the amendments made to the traditional  $\epsilon$ -greedy method in order to enhance the performance of the Q-learning algorithm for the inventory management reinforcement learning problem.



---



---

## CHAPTER 8

---

# Experimental design

### Contents

8.1	Experimental design overview . . . . .	111
8.2	The pharmaceutical supply chain network . . . . .	113
8.2.1	<i>Facilities</i> . . . . .	114
8.2.2	<i>Connections</i> . . . . .	114
8.2.3	<i>Neighbourhoods</i> . . . . .	115
8.2.4	<i>Inventory</i> . . . . .	116
8.2.5	<i>Demand</i> . . . . .	117
8.3	Q-learning algorithmic implementation . . . . .	118
8.3.1	<i>The state space</i> . . . . .	118
8.3.2	<i>The action space</i> . . . . .	123
8.3.3	<i>The reward function</i> . . . . .	124
8.3.4	<i>Learning rate and action selection</i> . . . . .	125
8.4	Experimental procedure . . . . .	125
8.5	Statistical analysis . . . . .	126
8.6	Chapter summary . . . . .	129

The purpose of this chapter is to delineate the experimental design employed in this thesis. A set of experiments is designed to evaluate the relative effectiveness of the five information-sharing scenarios of §5.2 in the context of an hypothetical pharmaceutical supply chain.

A motivation for the design of the hypothetical pharmaceutical supply chain employed in all experiments of this thesis is provided in §8.1, and the architecture of this supply chain is discussed next in §8.2. This is followed by a description of the exact algorithmic implementation of the Q-learning algorithm in §8.3. A brief outline of the experimental procedure is presented next, before the statistical tests employed to analyse the results of the experimental design are finally described in §8.5. The chapter closes with a brief summary of the material in §8.6.

### 8.1 Experimental design overview

An experimental design process is followed in order to demonstrate how the simulation model concept demonstrator of §6 may be employed to investigate the problem described in §1.2. The main objective of the experimental design is therefore to elucidate conceptually how information

sharing may benefit inventory management in a pharmaceutical supply chain. The impact of information sharing is analysed in respect of the five information-sharing scenarios of §5.2. The reinforcement learning algorithm Q-learning (described in §4.2.3) is employed by agents to learn inventory management policies based on the information made available to them according to the respective information-sharing scenarios. The effectiveness of the policies learnt (and by implication the relative effectiveness of information sharing) is evaluated by implementing each set of policies under the same set of end-user demand conditions.

During the reinforcement learning procedure, each agent learns independently and therefore learns its own unique inventory control policy. The objective of each agent is to learn a policy that minimises local stock-outs and expiries, as outlined in §7.3. Each agent may therefore be considered as self-organising, since it acts autonomously, within a greater network, in its pursuit of a particular goal. Self-organisation may lead to the notion of emergence, as described in §1.1. A second avenue pursued as part of the experimental design is to investigate whether the agents' self-organising behaviour may lead to a form of emergence in the pharmaceutical supply chain network where stock-outs and expiries are minimised globally. It is expected that this phenomenon may be most pronounced in the case of upstream stock shortages and when facilities have to resort to informal inventory sharing between them in order to avoid stock-outs.

In order to evaluate the five information-sharing scenarios considered in this thesis effectively, it is required to do so with respect to a fixed pharmaceutical supply chain network with fixed supply chain variables. In other words, each of the five scenarios should be evaluated separately, but in the same supply chain environment with the same end-user demand conditions. The experimental supply chain should not be too small so as to ensure that the concept demonstrator fulfils its purpose effectively and efficiently.

The application of reinforcement learning plays a prominent role in establishing an appropriate size for the experimental supply chain network. Reinforcement learning is extremely computationally expensive and it is expected that experiments involving large networks may require a considerable amount of training time. This is attributed to the fact that each agent in the network is trained individually during the reinforcement learning process. This is done in order to adhere to the principles of self-organisation, and also because each agent holds a unique position in the supply chain network. It is therefore not desirable to train one clinic agent and apply its learnt policy to all other clinics uniformly. The required learning time therefore increases with every addition of a new agent to the network. Furthermore, the size of each agent's state space increases as the scope of information sharing increases over the five information-sharing scenarios. The sizes of each agent's state space in Scenario 5 is, for example, significantly larger than the size of its respective state space in Scenario 1. By implication, the time required for a reinforcement learning algorithm to solve the problem successfully will vary across the five scenarios, even for the same supply chain network. This problem is compounded by the fact that the increase in state space size across the five scenarios is exponential. Hence, the required learning time also increases exponentially as the scope of information sharing increases. Taking all of these considerations into account, a single hypothetical pharmaceutical supply chain was carefully designed and employed in the experimental design of this thesis.

The experimental supply chain should be large enough so that the impact of each information-sharing scenario can be analysed sufficiently. This implies that the network should comprise at least one facility of each of the four available facility types, at least one warehouse/hospital neighbourhood and at least one neighbourhood of clinics. This consideration led the author to experiment empirically with networks of different sizes in order to identify a suitable candidate network. The network first considered comprised a total of sixteen facilities — one manufacturer, one hospital, two warehouses and twelve clinics. The hospital and both warehouses formed a

neighbourhood, while the clinics were segmented into neighbourhoods of sizes three, four and five, respectively. Preliminary experiments involving this network revealed that at least 48 hours were required (given the available computing power<sup>1</sup>) for the Q-learning algorithm to solve the problem for Scenario 1 successfully. In these preliminary experiments, the hospital agent had 5 376 possible states in Scenario 1 and a total of 1 032 192 states for Scenarios 4 and 5. Considering that each hospital agent has ten state variables for Scenarios 4 and 5, it is evident that the size of the state space may become intractable quite rapidly. Coupled with larger state spaces for the other agents as well, the required learning time for Scenario 5 (in the 16-facility network) was estimated as 25 days. Limited time and limited computing resources consequently compelled the author to devise a smaller network with smaller state spaces in order to reduce the required training time without compromising the quality of the concept demonstrator.

The computational power required to learn each instance of the four agent types (*i.e.* manufacturer, warehouse, hospital and clinic) varies considerably because of the different sizes of their respective state spaces. The state spaces of warehouse and hospital agents are, for example, significantly larger than those of the manufacturer and clinic agents, specifically for Scenarios 3, 4 and 5. As a result, it is significantly more expensive to train one warehouse or hospital agent than a single clinic agent. Considerable training time can, therefore, be saved by reducing the number of warehouses and/or hospitals in a network. Since the first experimental network considered above contained only one hospital and two warehouses, it was decided to remove one warehouse. A further six clinics were also removed from the original network. This reduced the number of clinic neighbourhoods to two — each neighbourhood comprising three clinics. Furthermore, the remaining warehouse and the hospital formed an inventory-sharing neighbourhood. Empirical experiments with this smaller network (nine facilities) and smaller agent state spaces, revealed that the required training time may be reduced by up to 70% for Scenario 5 when compared with the network of sixteen facilities considered originally. This enhanced the feasibility of the experimental design considerably, and it was therefore decided to adopt and implement the smaller network during the experimental design.

The experimental design process comprises the following phases. First, the structure and the properties of the experimental supply chain network are established. This involves specifying the constituent facilities, the connections between these facilities, the relevant shipping policies and the relevant inventory-sharing neighbourhoods. In particular, a single set of end-user demand conditions is established for implementation in all experiments. Thereafter, the Q-learning algorithm is employed to learn effective inventory control policies for each agent in the supply chain for each of the five information-sharing scenarios. Once this training procedure has been completed, a selection of experiments involving the learnt policies is carried out in order to evaluate the effectiveness of the respective information-sharing scenarios. The results of these experiments are analysed statistically in order to determine the relative effectiveness of information sharing. The effectiveness is measured with respect to the two KPIs introduced in §6.1.5 (*i.e.* the total number of stock-outs and the total number of expiries).

## 8.2 The pharmaceutical supply chain network

The architecture of the hypothetical pharmaceutical supply chain network chosen for implementation in the experiments conducted in this thesis is discussed in detail in this section. The bulk

---

<sup>1</sup>Three personal computers were employed during all experiments performed in this thesis. Two of these computers had the following specifications: An Intel® Core™ i7-4790 CPU with 8 GB of RAM operating at 3.60 GHz within a 64-bit Windows 7 operating system. The specifications of the third computer were as follows: An Intel® Core™ i7-4770 CPU with 8 GB of RAM operating at 3.40 GHz, also within a 64-bit Windows 7 operating system.

of the content in this section represents the model input data that are required in the input framework of the concept demonstrator, as described in §6.1.1.

### 8.2.1 Facilities

The experimental pharmaceutical supply chain selected for implementation in all experiments comprises a total of nine facilities, as established in §8.1. The network contains one manufacturing entity that supplies inventory to a hospital and to a warehouse, respectively. The hospital, in turn, distributes stock to a cluster of three clinics, and the warehouse also serves a further three clinics. The pharmaceutical supply chain contains at least one instance of each of the four facility types embedded in the concept demonstrator so as to demonstrate the ability of each agent type to learn an effective inventory management policy by means of reinforcement learning. The supply chain comprises nine `Facility` agents and the population index, the facility type, the supply chain tier and the name of each agent is shown in Table 8.1. This information is captured in the `table_facilities` table of the input framework of §6.1.1.

Agent index	Type	Tier	Name
0	Manufacturer	1	Manufacturer M
1	Hospital	3	Hospital H
2	Warehouse	2	Warehouse W
3	Clinic	4	Clinic A
4	Clinic	4	Clinic B
5	Clinic	4	Clinic C
6	Clinic	4	Clinic D
7	Clinic	4	Clinic E
8	Clinic	4	Clinic F

TABLE 8.1: *The population index, facility type, supply chain tier and name of each Facility agent included in the experimental pharmaceutical supply chain network.*

### 8.2.2 Connections

The structure of the connections between the nine facilities in the experimental pharmaceutical supply chain is tabulated in Table 8.2. The shipping interval, as well as the triangular probability distribution parameter values associated with the delivery lead times between each supplier-customer connection are also shown in this table. The `Facility` agent with index 1 (the hospital) serves the `Facility` agents indexed 3–5, and `Facility` agent 2 (the warehouse) serves the `Facility` agents indexed 6–8. A shipping interval of one day is chosen for each supplier-customer pair in order to limit excessive randomness in the delivery lead times during simulation. Since the average length of the lead time plays a prominent role in the learning process of each agent, it is important that an agent’s estimate of the expected lead time is relatively accurate. A shipping interval of five days, for example, would imply that a customer can place an order one, two, three, four or five days in advance of the next shipping date. Coupled with an expected delivery lead time of three days, for example, the total expected lead time may vary between four and nine days. Such a large variance in the lead time may complicate the learning process for an agent whose state space does not explicitly include the expected lead-time duration. Since this is the case for the learning agents considered in the concept demonstrator of this thesis, a shipping interval of 1 is chosen for implementation. Not only does this maximise supplier responsiveness, but ensures that the expected supplier lead time remains constant throughout.



Supplier index	Customer index	Shipping interval (days)	Minimum lead time (days)	Mode lead time (days)	Maximum lead time (days)
1	0	1	6	7	9
2	0	1	6	7	9
3	1	1	4	5	7
4	1	1	4	5	7
5	1	1	4	5	7
6	2	1	4	5	7
7	2	1	4	5	7
8	2	1	4	5	7

TABLE 8.2: The primary supplier-customer connections in the experimental pharmaceutical supply chain as well as their corresponding shipping intervals and delivery lead times.

It is assumed that the manufacturer is located a significant distance from each of its two customers and the delivery lead times between each manufacturer-customer pair are therefore relatively long (the mode lead time is seven days). The hospital and the warehouse, in turn, are considered to be located relatively close to their customers and therefore the delivery lead times between suppliers and clinics are considerably shorter (a mode lead time of five days). For each primary supplier-customer connection, the delivery lead time distribution is skewed to the right. This is done to introduce more variation in the supply chain and to evaluate the robustness of the policies learnt for relatively unreliable supply. Furthermore, the production lead time associated with any given batch size is also modelled by means of a triangular probability distribution. The minimum, the mode and the maximum parameter values are chosen as 6, 7 and 10, respectively. The simulation model starting date is set as Monday 7 January 2019. The first cut-off date for any incoming order between any supplier-customer pair is set as 7 January 2019, and the first available shipping date for each supplier is selected as 8 January 2019. In other words, each supplier is first available to ship a new delivery on the second day of simulated time, provided that the corresponding order was placed on the first day of simulated time. The information pertaining to the supplier-customer connections, cut-off and shipping dates, as well as delivery lead times as discussed above, are captured in the *table\_connections* database table of §6.1.1.

### 8.2.3 Neighbourhoods

The concept of inventory-sharing neighbourhoods was first introduced in Scenario 2 of §5.2.2. According to Scenarios 2–5 of §5.2.2–5.2.5, clinics residing within the same neighbourhood can choose to share inventory between themselves. Hospitals and warehouses, on the other hand, can also form inventory-sharing neighbourhoods according to Scenarios 4 and 5 of §5.2.4 and §5.2.5, respectively. In the experimental pharmaceutical supply chain considered in this thesis, each cluster of clinics is served by the same primary supplier form a distinct neighbourhood. This implies that the three clinic agents with indices 3–5 are in Neighbourhood 1, while Neighbourhood 2 comprises the three clinic agents with indices 6–8. Finally, the hospital (index 1) and the warehouse (index 2) constitute Neighbourhood 3, where they are eligible for inventory sharing according to Scenarios 4–5. Each neighbourhood and its members are specified accordingly in the *table\_neighbourhoods* database table of §6.1.1.

The shipping interval and the corresponding delivery lead times for each supplier-customer pair of Neighbourhood 3 are shown in Table 8.3. This information is also captured in the *table\_connections* table of the input framework and the *neighbours* attribute is set to *true*. It is assumed that the facilities in Neighbourhood 3 are relatively close to one another in comparison

with their respective distances to the manufacturer. The delivery lead times between the two neighbours are therefore shorter than the manufacturer's delivery lead times.

Supplier index	Customer index	Shipping interval (days)	Minimum lead time (days)	Mode lead time (days)	Maximum lead time (days)
2	1	1	3	4	5
1	2	1	3	4	5

TABLE 8.3: Specification of the delivery lead times between supplier-customer pairs in Neighbourhood 3.

A schematic representation of the layout of the experimental pharmaceutical supply chain network is shown in Figure 8.1. The black lines between facilities indicate primary supplier-customer connections, while the blue, dashed lines specify connections between neighbours. Each facility index in the Facility agent population is also indicated in the figure.

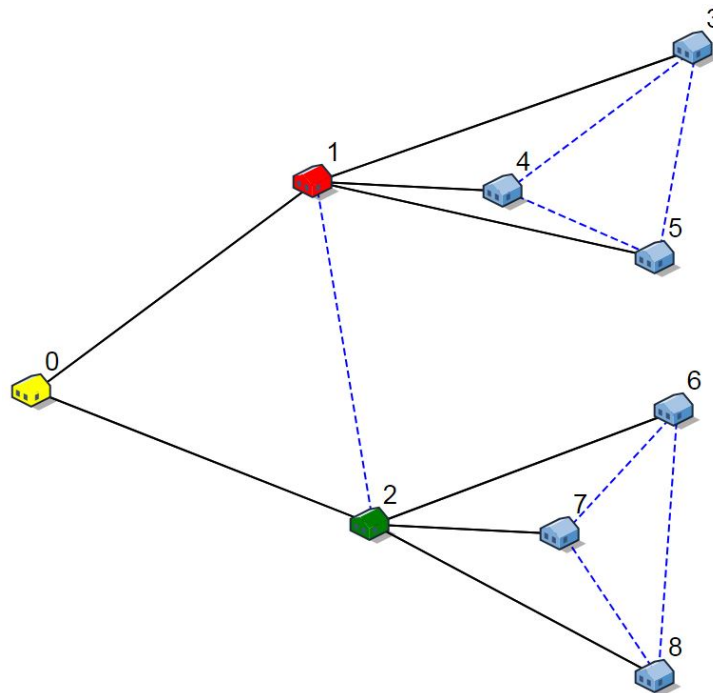


FIGURE 8.1: The layout of the experimental pharmaceutical supply chain network. The black lines specify the connections between suppliers and customers, while the blue, dashed lines indicate connections between neighbours.

#### 8.2.4 Inventory

The only product-specific information that is critical during simulation execution is the product shelf-life. Given that the influence of temperature on product quality is not considered in the concept demonstrator, it is possible to model virtually any product in the pharmaceutical supply chain. The product chosen for consideration in all experiments in this thesis is considered to be any essential pharmaceutical product with a shelf-life of two years. This shelf-life was chosen based on the fact that the average shelf-life of ARV products is typically between 24 and 36 months [123], and the South African pharmaceutical supply chain has experienced significant stock-outs of ARV medicines in recent years, as mentioned in §2.9.3. Incorporating a relatively

short shelf-life of 24 months (as opposed to 36 months) may also better illuminate the ability of the reinforcement learning algorithm to learn a policy that minimises expiries (provided that the simulation period is sufficiently long).

The `table_starting_inventory` database table of the input framework is used to capture the starting inventory levels of each facility, as described in §6.1.1. Manipulating the starting inventory conditions provides an additional method for evaluating the robustness of the replenishment policies learnt. By altering the starting inventory conditions, the decision maker can inspect the impact of the policies under consideration within only a few days of simulated time.

The simulation model does not, however, accommodate the possibility of including pending inventory replenishment orders or in-transit shipments as part of the starting conditions. Additionally, the notion of a simulation warm-up period is disregarded in all experiments. As a result, the starting inventory level of each facility should be chosen carefully so as to ensure that the starting conditions are reasonable. Consider, for example, the case where a clinic's starting inventory level is chosen as 20 units, and the daily end-user demand is 50 units. The clinic will consequently incur stock-outs on the first day (and possibly on a number of subsequent days due to inescapable delivery lead times). Such a low initial inventory level (with no pending order(s)) may not necessarily be representative of the clinic's inventory policy (an effective policy may have dictated that a replenishment order be placed long before the inventory level reached 20 units). It is therefore recommended that the starting inventory levels should be relatively high. By the same token, however, the decision maker may deliberately choose relatively low starting inventory levels so as to inspect how quickly the affected facilities can recover to adequate inventory levels (a clinic may potentially recover relatively quickly if its neighbours have a sufficient amount of inventory available for exchange). The omission of a warm-up period is therefore not necessarily a shortcoming — the onus rests on the decision maker to be cognisant of the impact that either low or high starting inventory levels may have on supply chain performance. It is, however, suggested as future work to include pending orders and deliveries in the model starting conditions.

### 8.2.5 Demand

End-user demand may be modelled according to three different probability distributions, as described in §6.1.1. According to the experimental design implemented in this thesis, the daily end-user demand at each facility is modelled in respect of the triangular probability distribution. In order to differentiate between demand of different magnitudes, two so-called *demand classes* are employed in the experimental design, namely *low* demand and *high* demand. The minimum, the mode and the maximum triangular distribution parameter values associated with each of these demand classes are chosen arbitrarily and are shown in Table 8.4. The expected daily demand for any facility with low demand is 35 units, while the expected daily demand for a facility with high demand is 90. The range for each demand class is 20 units.

Demand class	Minimum	Mode	Maximum
Low	25	35	45
High	80	90	100

TABLE 8.4: The triangular distribution parameter values of the two demand classes employed during the experimental design.

One set of end-user demand conditions is considered in all experiments performed in this thesis. According to this demand set, the hospital and all six clinics in the supply chain experience low

demand for a period of 180 days (approximately six months). After the first six months has lapsed, the demand increases to high at all the facilities for a further six months to complete a one-year demand cycle. It is assumed that this one-year cycle repeats iteratively. A fluctuating demand profile is chosen because the impact of information sharing is expected to be relatively limited during stable demand. This demand information is captured in the *table\_demand* database table of the input framework of §6.1.1. A condensed summary of the demand pattern is shown in Table 8.5.

Demand period	Length (days)	Demand class	Facilities
1	180	Low	All
2	180	High	All

TABLE 8.5: A summary of the end-user demand conditions considered in all experiments.

### 8.3 Q-learning algorithmic implementation

Q-learning is a table-based solution approach that relies on discrete state and action spaces to solve the particular reinforcement learning problem on hand. The Q-values  $Q(s, a)$  associated with each state-action pair are stored in a look-up table called a Q-table. The number of rows in an agent's Q-table is equal to the number of states, and there are as many columns as there are discrete actions. This table-based approach requires the discretisation of continuous state variables into a number of integer intervals (or bins). The number of intervals and the cardinality of each interval, however, hold significant implications for the performance of the reinforcement learning algorithm. Employing a coarser representation of state variables (*i.e.* fewer intervals with large magnitudes) may lead to a single state being associated with functionally different situations and this may lead to poor action selection [55]. Implementing a larger number of intervals, on the other hand, increases the state space which implies an increase in the size of the Q-table. As a result, more training data are required and this is only possible with more or longer simulation runs.

The objective in this section is to describe the exact algorithmic implementation of the Q-learning algorithm employed during the simulation experiments conducted in this thesis. The implementation of the algorithm is based on the inventory management reinforcement learning problem described in §7. This discussion includes the discretisation of each agent's state space and action space, the final design of the reward function, and the learning rate and action-selection technique employed by each agent.

#### 8.3.1 The state space

The state space and the relevant state variables of each agent for each of the five information-sharing scenarios considered in this experimental design were described in §7.1. Since all of the state variables are continuous, each variable should be discretised into discrete intervals for implementation in the simulation model concept demonstrator. In order to achieve a balance between a too fine and a too coarse discretisation, the number of intervals employed for each state variable was determined empirically for each agent. This proved to be an extremely challenging and time-consuming task. For instance, the author found that the inventory level state variable of each agent required a relatively fine discretisation (*i.e.* many intervals) compared with other state variables. In order to restrict the size of each agent's state space, not all state variables

could, however, be discretised as finely. A large number of experiments were performed in order to determine the appropriate size of each interval so as to ensure that functionally different situations are not captured in a single state.

The end-user demand considered in the experimental design is homogeneous for all facilities at any given point in time (*i.e.* either low or high), as described in §8.2.5. By implication, each neighbourhood of clinics may exhibit approximately the same ordering behaviour during a single demand period. As a result, the manufacturer, the warehouse and the hospital may most likely experience relatively stable (albeit either relatively low or relatively high) demand over the course of any given demand period. This phenomenon made it possible to discretise a selection of state variables into two (as opposed to many) intervals — each corresponding to either relatively low or relatively high demand. For each of these state variables, there is an inflection point that distinguishes the two demand instances from one another, and the cardinality of each interval is based on this critical value. Many experiments were performed in order to uncover the inflection points for these state variables so as to adapt the state space discretisation of each agent accordingly. It is acknowledged that this approach may not necessarily suffice for heterogeneous demand, but it was found to simplify the reinforcement learning process considerably in the experimental design of this thesis. Nevertheless, it remains crucial to consider the impact of these inflection points when dealing with discretised state spaces. If the relevant intervals are chosen too large, the agent may fail to recognise these critical values.

Furthermore, the number of intervals and the corresponding magnitudes for each state variable remain constant throughout, irrespective of the information-sharing scenario analysed. This is done to ensure that the performance of the Q-learning algorithm may be evaluated fairly across all five information-sharing scenarios.

The manufacturing agent has five state variables according to the first four information-sharing scenarios, and a total of eight state variables in Scenario 5, as described in §7.1. The manufacturer's own inventory-level state variable is discretised into fifteen equally-sized integer intervals, each of cardinality 2 500. Since the lowest possible inventory level is zero, the first of these fifteen intervals is defined as 0–2 499. It would, however, be impractical to specify the fifteenth interval as 35 000–37 499, because the agent's inventory level may possibly exceed 37 499 during simulation run-time. The final interval of the inventory-level state variable is therefore specified as 35 000 or more in order to accommodate this possibility. This convention (where the last interval is unbounded) is adopted for a selection of other state variables as well where the maximum state variable value may be arbitrarily large. The own expiries during lead time state variable is implemented as a variable capturing the percentage of the inventory on hand that is due to expire during the upcoming lead-time period. This state variable is discretised into two intervals for the manufacturing agent. The first interval indicates that up to a third of the current inventory on hand is due to expire during the expected lead-time period. The second interval, on the other hand, indicates that more than a third of the inventory on hand is due to expire in the short-term. This relatively coarse discretisation is motivated by the expectation that stock should rarely expire in the manufacturer's storage space if the inventory is managed effectively.

For the own inventory-in-backlog state variable, it may be important to distinguish between whether any, or no, inventory is in backlog, at any given time instant. This is especially relevant for a small number of intervals. Consider, for example, a backlog interval of 0–200. If the actual backlog is larger than zero and falls in this interval, the agent may be prompted to take corrective action as soon as possible (*e.g.* place an order). The same interval may, however, specify an actual backlog of zero, in which case the agent may not need to place a new order at the current point in time (especially if the current inventory level is sufficiently high). This is an example

of functionally different situations captured in the same interval, and this may compromise the performance of the Q-learning algorithm. The manufacturer's inventory backlog state variable is therefore discretised into five integer intervals and the first interval signifies that there is no inventory in backlog (*i.e.* the interval is defined as 0–0). This is followed by two equally-sized intervals of magnitude 7 500, namely 1–7 500 and 7 501–15 000. The fourth interval is specified as 15 001–30 000 and, in order to accommodate an arbitrarily large backlog, the fifth interval is defined as 30 001 or more.

The manufacturer's own inventory-in-production state variable is discretised into five integer intervals and these intervals are the same as for the backlogged inventory state variable. In other words, the first interval is defined as 0–0 and indicates that the agent is not engaged in any production run during the current time step. With respect to the manufacturer's own demand state variable, it is discretised into four integer intervals. The first interval is defined as 0–4 000 and the second interval as 4 001–4 999. This is followed by an interval of magnitude of 10 001 (*i.e.* 5 000–15 000) and the fourth interval is reserved for any mean demand value greater than 15 000 (*i.e.* interval is defined as 15 001 or more). This particular discretisation is based on the observation that a mean demand of less than 4 000 is typically associated with low end-user demand, while the manufacturing agent typically experiences a mean demand greater than 5 000 during periods with high end-user demand. The second interval (4 001–4 999) is defined as such because the agent may occasionally experience a mean demand in this region during either low or high patient demand.

Three new state variables are added to the manufacturer's state space in Scenario 5. Since end-user demand is modelled by means of the two demand classes described in §8.2.5, the customer clinics' demand state variable is discretised into two corresponding integer intervals. The first interval is defined as 1–50 (low demand) and the second interval as 51–100 (high demand). The customer clinics' inventory-levels state variable is discretised into two integer intervals and they are indicated as 0–1 000 and 1 001 or more, respectively. This is based on the empirical observation that an aggregate inventory level of less than 1 000 (across all six clinics) may be considered critically low and exposes at least one clinic to a serious risk of incurring stock-outs. Similarly, the customer warehouses' inventory levels state variable is also discretised into two intervals. The first is 0–4 000 and the second 4 001 or more. Again, the author observed empirically that an aggregate inventory level of less than 4 000 may be considered as critically low. For Scenarios 1–4, the manufacturing agent has a total of  $15 \times 2 \times 5 \times 4 \times 5 = 3\,000$  states, whereas the state space comprises 24 000 states in Scenario 5. A summary of the discretisation of the manufacturing agent's state space is shown in Table 8.6.

State	Number of intervals	Equally-sized intervals?	Interval magnitude
Own inventory level	15	Yes	2 500
Own expiries during lead time	2	No	—
Own backlog	5	No	—
Own inventory in production	5	No	—
Own demand (incoming orders)	4	No	—
Customer clinics' inventory levels	2	No	—
Customer clinics' demands	2	Yes	50
Customer warehouses' inventory levels	2	No	—

TABLE 8.6: *The discretisation of the manufacturer agent's state space.*

The warehouse agent has five state variables involving local information, as described in §7.1. The own inventory-level state variable is discretised into twenty equally-sized intervals of cardi-

nality 350 (the last interval is unbounded), and the discretisation of the expiries state variable is carried out in the same manner as for the manufacturer. The warehouse's own backlog state variable is discretised into five intervals and the widths of these intervals are based on the warehouse's available ordering actions. The first interval has a cardinality of 0 (*i.e.* no backlogged inventory), and the remaining intervals are defined as 1–1 500, 1 501–3 000, 3 001–4 500, and 4 501 or more, respectively. Furthermore, the first four of six intervals of the warehouse agent's own inventory-on-order state variable is discretised into equally-sized intervals of magnitude 1 500 (starting with 0–1 499). The fifth interval is specified as 6 000–12 000 and the sixth interval is reserved for any inventory amount on order greater than 12 000. Similar to the manufacturing agent's own demand state, the warehouse's own demand state is also discretised into two intervals. The first interval of this state variable is defined as 0–2 400 and the final interval is indicated as 2 401 or more. Based on empirical experimentation, it was found that a mean demand of less than 2 400 is typically associated with periods of low end-user demand.

The warehouse agent's state space is enlarged in Scenario 3 when customer-related information is included. The state variable capturing the customer clinics' demands is discretised according to the two predefined end-user demand classes (*i.e.* the same as for the manufacturing agent). In an attempt to limit the size of the agent's state space, the customer clinics' inventory levels state variable is also discretised into two intervals. It was found that an aggregate clinic inventory level greater than 500 is typically sufficient for clinics to satisfy short-term demand. The two intervals are consequently defined as 0–500 and 501 or more, respectively.

For a warehouse agent to decide whether or not it should order from a neighbouring facility during Scenarios 4 and 5, it is perhaps more important to know whether or not the neighbourhood has sufficient inventory available for sharing, as opposed to knowing the exact inventory amount available. Based on this argument, and in a further attempt to limit the size of the state space, the effective neighbourhood inventory state variable is discretised into two intervals only. The first interval is defined as integer values in the real interval  $(-\infty, 0]$  (the effective neighbourhood inventory level can be negative, as alluded to in §7.1) and the second interval is specified as integer values in the real interval  $[1, \infty)$ . The supposition is that any effective neighbourhood inventory level greater than zero may imply that a neighbour has sufficient inventory available for sharing. Finally, the manufacturer's inventory level state variable is discretised in an identical fashion. In other words, there are two intervals and they are defined as integer values in the real intervals  $(-\infty, 0]$  and  $[1, \infty)$ , respectively. In Scenarios 4 and 5, the warehouse agent's state space comprises 38 400 states. A summary of the discretisation of the warehouse agent's state space is shown in Table 8.7.

State	Number of intervals	Equally-sized intervals?	Interval magnitude
Own inventory level	20	Yes	350
Own expiries during lead time	2	No	—
Own backlog	5	No	—
Own inventory on order	6	Yes	1 500
Own demand (incoming orders)	2	No	—
Own effective neighbourhood inventory	2	Yes	—
Customer clinics' inventory levels	2	No	—
Customer clinics' demands	2	Yes	50
Manufacturer's inventory level	2	No	—

TABLE 8.7: *The discretisation of the warehouse agent's state space.*

The hospital agent has a maximum of ten state variables, and nine of these variables are shared with the warehouse agent. Since the warehouse and the hospital occupy similar positions and perform similar functions in the experimental supply chain, the discretisation of the nine mutual state variables is done in exactly the same fashion for the hospital. The tenth state variable is the hospital's own patient demand and this variable is discretised based on the two end-user demand classes as explained previously. Because of the two additional intervals, the hospital agent has a total of  $38\,400 \times 2 = 76\,800$  states in Scenarios 4 and 5, compared with the 38 400 states of the warehouse. A summary of the discretisation of the hospital's state space is shown in Table 8.8.

State	Number of intervals	Equally-sized intervals?	Interval magnitude
Own inventory level	20	Yes	350
Own expiries during lead time	2	No	—
Own backlog	5	No	—
Own inventory on order	6	No	—
Own demand (incoming orders)	2	No	—
Own end-user demand	2	Yes	50
Own effective neighbourhood inventory	2	No	—
Customer clinics' inventory levels	2	No	—
Customer clinics' demands	2	Yes	50
Manufacturer's inventory level	2	No	—

TABLE 8.8: *The discretisation of the hospital agent's state space.*

The clinic agent has fewer state variables than the other three agent types and this provides an opportunity for a relatively more finely-grained state space representation. The clinic's own inventory level is discretised into 126 equally-sized integer intervals of cardinality 4. Since a clinic is the entity closest to patient demand, it is imperative that this agent has detailed knowledge of its current inventory level. A sufficiently fine representation of the inventory-level state variable may allow the agent to better distinguish between sufficiently high and critically low inventory levels. The effective inventory-level state variable (which replaces the original inventory-level state variable in Scenario 2, as explained in §7.1) adopts the same discretisation as the original state variable (*i.e.* 126 integer intervals, each of magnitude 4). It is critical that each clinic agent also has detailed knowledge about the remaining shelf-life of its inventory. Unanticipated expiries may, of course, lead to substantial stock-outs in the short term. As a result, the clinic's own expiries during lead time is discretised into three intervals. The first interval is defined as 0–0 and indicates that 0% of the current stock on hand will expire during the expected lead-time period. The second interval specifies that 1–33% of the inventory is expected to expire in the short-term and the final interval indicates that more than a third of the current stock will expire during the upcoming lead time. Based on the clinic's action space, the inventory-on-order state variable is discretised into six intervals and they are specified as 0–0, 1–210, 211–375, 376–540, 541–1 100, and 1 101 or more, respectively.

The end-user demand state variable is also discretised into the two demand classes as explained previously. Finally, the effective neighbourhood inventory state variable is discretised into four distinct intervals. The first interval is defined as 0–0 and indicates that there is no inventory available for sharing in the neighbourhood. The following two intervals each has a magnitude of 100 and the fourth interval is chosen as 201 or more. This configuration allows the agent to distinguish between the cases where no, or at least some, inventory is available for sharing. There are 4 536 states in the agent's state space in Scenario 1, and when inventory sharing is introduced



in Scenario 2, the state space increases to 18 144 states. A summary of the discretisation of the clinic's state space is shown in Table 8.9.

State	Number of intervals	Equally-sized intervals?	Interval magnitude
Own inventory level	126	Yes	4
Own effective inventory level	126	Yes	4
Own expiries during lead time	3	No	—
Own inventory on order	6	No	—
Own end-user demand	2	Yes	50
Effective neighbourhood inventory	4	No	—

TABLE 8.9: *The discretisation of the clinic agent's state space.*

The number of states in each agent's state space according to each of the five information-sharing scenarios is shown in Table 8.10. It is evident that the increase in the size of each agent's state space is exponential as more information sharing takes place. Subsequently, the required amount of learning time is expected to differ across the five information-sharing scenarios.

Scenario	Manufacturer	Warehouse	Hospital	Clinic
1	3 000	2 400	4 800	4 536
2	3 000	2 400	4 800	18 144
3	3 000	9 600	19 200	18 144
4	3 000	38 400	76 800	18 144
5	24 000	38 400	76 800	18 144

TABLE 8.10: *The size of each agent's state space according to the five scenarios.*

### 8.3.2 The action space

At each discrete time step, an agent must decide on either a quantity to order or a quantity to manufacture, as described in §7.2. Since the number of columns in a Q-table is equivalent to the number of actions available to an agent, the required training time will increase as the number of available actions increases. For the purposes of the concept demonstrator of this thesis, the number of available actions included in each agent's action space is relatively small.

For all the simulation experiments conducted in this thesis, the manufacturing agent has an invariant action space  $\mathcal{A}_1 = \{0, 7\,500, 15\,000\}$ . At each discrete time step, the agent may therefore choose either not to initiate a new production run, or to manufacture a new batch comprising either 7 500 or 15 000 units. During the first three information-sharing scenarios where inventory sharing is prohibited between warehouses and hospitals, these agents can only choose between formal ordering actions (as explained in §7.2). The set of formal ordering actions available to each warehouse and hospital agent in Scenarios 1–3 is specified as  $\mathcal{A}_2 = \{0, 1\,500, 3\,000, 4\,500\}$ . When inventory sharing is allowed according to Scenarios 4 and 5, the warehouse and the hospital agent may each order 1 500 units of inventory from a neighbour at any given time. Subsequently, the original action space is expanded to  $\mathcal{A}_2' = \{0, 1\,500, 3\,000, 4\,500, 1\,500\}$ , where the last action in the set indicates the informal order. Finally, the clinic agent can choose between any one of four actions during Scenario 1 and this set of actions is defined as  $\mathcal{A}_3 = \{0, 210, 375, 540\}$ . When a clinic is allowed to place an informal order to a neighbouring clinic, the agent can choose to order 90 units from a neighbour. The resulting action space for each clinic agent during Scenarios

2–5 is therefore  $\mathcal{A}_3 = \{0, 210, 375, 540, 90\}$ , where the final action indicates the informal order. These particular quantities are chosen based on a clinic's expected demand during the lead time. During periods of low demand, the expected demand for any clinic is  $35 \times 6 = 210$  (the mean daily demand is 35 and the mode lead time is six days). Similarly, the expected demand during the lead time when demand is classified as high is calculated as  $90 \times 6 = 540$ . Given that the expected daily demand is 90 units during high demand, the informal order quantity is chosen as 90.

### 8.3.3 The reward function

A general form of the reward function employed by each agent in this thesis was provided in (7.1) of §7.3. In this function there are three variables,  $p$ ,  $k$  and  $z$ , whose values should be determined empirically for each agent type, as explained in §7.3. The fixed reward for placing an order when the inventory amount on order is already greater than zero is  $p$ , while  $k$  denotes the reward assigned to any informal ordering action. Finally,  $z$  denotes the special-case punishment awarded to an agent whose inventory level and/or amount of inventory on order (or in production) is considered excessively high (as mentioned in §7.3). In this experimental design, an agent is awarded this punishment if the value of either its inventory level or inventory-on-order (or inventory-in-production) state variable is in the highest corresponding interval. The final design of the reward function employed by each agent type in the experimental design is presented in this section.

The reward function assigning reward to the manufacturing agent during time step  $t$  is given by

$$r(t) = -1h - 600s - 600e - 50\,000j - 5\,000\,000m. \quad (8.1)$$

By implication, the manufacturing agent is awarded a reward of  $-1$  for each unit held in inventory during any given time step, and a reward of  $-600$  for each unit stock-out and unit expiry. A reward of  $-50\,000$  is assigned when the agent starts a new production run if it is engaged in at least one other ongoing production run at the current point in time. Finally, the agent is awarded a large punishment of  $-5\,000\,000$  when it starts a new production in any state involving the following two conditions: Its inventory level is 35 000 or more (*i.e.* the fifteenth interval of its inventory level state variable), or its inventory in production is 30 001 or more (*i.e.* the fifth interval of its inventory-in-production state variable). These large punishments ensure that the agent learns the most effective action for these states relatively quickly, because their respective magnitudes have such a prominent influence on the computation of the  $Q(s, a)$ -values.

The warehouse and hospital agents each employ the same reward function, and this function is given by

$$r(t) = -1h - 600s - 600e - 15\,000(j)(y) - 50\,000(1 - y) - 500\,000m. \quad (8.2)$$

Each hospital and warehouse agent is consequently assigned a reward of  $-50\,000$  for each informal replenishment order issued. This large punishment is chosen to ensure that the agent learns to place an informal order only when absolutely necessary (*i.e.* when inventory is relatively low). If the informal order punishment is not sufficiently large, the agent may prefer informal orders over formal replenishment orders. In such a case, the warehouse and hospital may order from one another continually until both of them incur stock-outs. It is important to remember that the missed shipment of a single inventory unit is registered as a stock-out (as mentioned in §6.1.5). Given the scale of demand experienced by the warehouse and hospital, the total punishment associated with a large number of stock-outs very quickly overshadows the punishment for a single informal order. The punishment for an informal order, however, needs to be larger than the punishment for a formal replenishment order — a reward of  $-50\,000$  was subsequently

identified empirically as adequate. Each warehouse and hospital agent is given a punishment of  $-500\,000$  each time it places a new order when either its current inventory level is  $6\,650$  or more, or when its inventory on order is  $12\,001$  or more.

Finally, the reward function employed by each of the six clinic agents is given by

$$r(t) = -1h - 600s - 600e - 2\,000(j)(y) - 6\,000(1 - y) - 5\,000m. \quad (8.3)$$

Each clinic agent is therefore given a reward of  $-2\,000$  if it issues a formal replenishment order when the inventory-on-order state variable value is already greater than zero. A reward of  $-6\,000$  is awarded for each informal replenishment order. Empirical observations revealed that a punishment smaller than  $6\,000$  led agents to favour informal orders over formal orders, even when their inventory levels were sufficiently high. A punishment of  $-6\,000$  was, however, found to be adequate for allowing a clinic agent to learn rational behaviour in terms of when to place an informal order.

### 8.3.4 Learning rate and action selection

The learning rate and the exploration rate employed during all reinforcement learning simulation runs were described in §7.4 and §7.5, respectively. With respect to action selection, the *epsilon*-greedy method is employed by all agents. Based on the critical appraisal of no-ordering streaks in §7.5, it was decided not to incorporate these streaks in the experiments conducted in this thesis.

## 8.4 Experimental procedure

The experimental procedure followed in this thesis is described in more detail in this section. The first step involves the implementation of the Q-learning algorithm in order to learn inventory replenishment policies for each agent, for each of the five information-sharing scenarios. Five so-called *training simulation runs* are therefore performed, one for each scenario. Since the demand conditions described in §8.2.5 are cyclic in nature, a single continuous training simulation run is performed for each scenario. A training simulation run may be terminated once the Q-learning algorithm has converged. The Q-learning algorithm may have converged when the most effective action for each state has been learnt, and each agent's policy does not change over a sustained period of time. In this thesis, convergence is determined by evaluating the mean daily amount of inventory held in the supply chain, over a one-year demand period. When the Q-learning algorithm has converged, each agent is expected to make similar decisions over the course of the year-long demand cycle. As a result, it may be expected that the profile of the inventory held daily by each agent (and by implication the entire supply chain) will repeat itself annually once the Q-learning algorithm had reached convergence. A training run is consequently terminated after the deviation in the mean daily amount of inventory held in the supply chain, per year, has become relatively small for a sustained period of time.

Once the training phase has been completed, the effectiveness of the policies learnt may be evaluated. A simulated time window of five years is considered during each experiment. In other words, a single experiment comprises five consecutive one-year demand periods. A five-year period is chosen so as to gain an improved understanding of the effectiveness of the policies learnt with respect to product expiries (the product considered has a shelf-life of two years). Additionally, the five-year period also contains five instances where end-user demand increases from low to high, thus providing more opportunity for investigating the impact of information

sharing when demand increases dramatically. An experiment is performed for each of the five information-sharing scenarios. The experiment involving Scenario 1 is called Experiment 1, the second experiment involving Scenario 2 is called Experiment 2, and so forth. Each experiment comprises 30 replication runs and subsequently produces 30 observations of each KPI. These KPIs are employed in statistical tests to ultimately measure the relative effectiveness of the five information-sharing scenarios.

The starting inventory levels and corresponding remaining shelf-life of each facility at the start of each experiment are shown in Table 8.11. These values also serve as the starting conditions for each of the five training simulation runs. Since each training run is performed over a continuous time window (*i.e.* the system is never manually reset to the original starting conditions), the exact starting conditions for any training run is not as critical for the performance of the Q-learning algorithm. Based on the discussion in §8.2.4, the initial inventory levels are chosen sufficiently high so as to ensure that the performance of the policies learnt is evaluated fairly.

Facility	Inventory amount	Remaining life (days)
0	25 146	619
1	5 436	510
2	4 876	414
3	389	60
4	367	18
5	412	67
6	287	47
7	369	23
8	313	60

TABLE 8.11: *The starting inventory levels and corresponding remaining shelf-lives of each facility for each simulation experiment.*

## 8.5 Statistical analysis

Statistical analyses are performed on the KPIs reported as model output data for all simulation experiments conducted in this thesis. In this analysis, significant differences in the data are reported at a 5% level of significance. The first step towards analysing the respective KPIs, is to perform an *analysis of variance* (ANOVA) [116] in order to ascertain whether or not there is a significant difference between at least two means in a collection of samples. The null-hypothesis states that the means of all the data samples investigated are equal. The alternative hypothesis is that there is a significant difference between at least two of the means. An ANOVA test does, however, only reveal *whether* there is a significant difference between at least two means, but does not indicate *where* this difference occurs. Therefore, *post hoc* tests are employed to establish where the difference occurs. In this thesis, *Fisher's Least Significant Difference* test [161] and the *Games-Howell* test [54] are the two *post hoc* tests employed to determine where significant differences occur.

After an ANOVA has been carried out and has revealed that there are significant differences in the data, a *Levene* test [131] is performed in order to determine whether or not the corresponding variances differ significantly from one another. If the variances are found not to differ statistically from one another at a 95% confidence level, the LSD test (which requires homogeneity of sample variances) is carried out to identify where the differences lie in the data. Alternatively, the Games-Howell test is employed if the Levene test indicates that the variances are statistically

different. The working of the ANOVA, Levene, LSD and Games-Howell tests are described briefly in this section.

### The ANOVA test

During the ANOVA procedure, the sum of squares between sets of data and the sum of squares within sets of data are calculated to test the null-hypothesis. The *sum of squares within groups* ( $SS_W$ ) is given by

$$SS_W = \sum_{i=1}^n \sum_{j=1}^m (x_j - \bar{x}_i)^2, \quad (8.4)$$

where  $n$  denotes the number of samples,  $m$  indicates the number of observations in each sample and  $\bar{x}_i$  denotes the mean value for sample  $i$ . Likewise, the *sum of squares between groups* ( $SS_B$ ) is given by

$$SS_B = m \sum_{i=1}^n (\bar{x}_i - \bar{x})^2, \quad (8.5)$$

where  $\bar{x}$  denotes the average mean for the  $n$  samples (sometimes called the *grand mean*). The *mean square* (MS) is next calculated for the  $SS_W$  and the  $SS_B$  values, respectively. This is done by dividing each value by the number of degrees of freedom. Subsequently, the *mean square within groups* ( $MS_W$ ) and the *mean square between groups* ( $MS_B$ ) are given by

$$MS_W = \frac{SS_W}{mn - n} \quad (8.6)$$

and

$$MS_B = \frac{SS_B}{n - 1}, \quad (8.7)$$

respectively. The test statistic, denoted by  $f$ , is calculated as the ratio between  $MS_W$  and  $MS_B$ . This test statistic is then compared at a significance level of 5% with the critical value  $F(n - 1, mn - n)$  of the  $F$ -distribution. If  $MS_W/MS_B > F(n - 1, mn - n)$ , the null hypothesis is rejected, in which case there is a statistically significant difference between the means of at least two samples at a 95% level of confidence. Alternatively, if the null-hypothesis cannot be rejected at a 5% level of significance, a *post hoc* test needs to be employed to identify where the difference lies.

### The Levene test

Before a *post hoc* test can be selected, a Levene test has to be employed to verify the assumption of homogeneity of variance. The null-hypothesis in the Levene test is that there are no statistically significant differences between the variances of two or more sets of data. The alternative hypothesis is that there are significant differences between the variances of two or more samples. The first step towards conducting the Levene test is to calculate the test statistic  $F_L$  as

$$F_L = \frac{(mn - n) \sum_{i=1}^n N_i (\bar{x}_i - \bar{x})^2}{(n - 1) \sum_{i=1}^n \sum_{j=1}^{N_i} (|x_{ij} - \bar{x}_i| - \bar{x}_i)^2}, \quad (8.8)$$

where  $N_i$  is the number of data points in sample  $i$  and  $x_{ij}$  is data point  $i$  from sample  $j$ . The test statistic is compared with a critical value  $F(n - 1, mn - n)$  from the  $F$ -distribution at a 5% level of significance. If

$$F_L \geq F(n - 1, mn - n), \quad (8.9)$$

it indicates that there is a statistical difference (at a 95% confidence level) between the variances of at least two samples in the original data set and the null-hypothesis is therefore rejected. In such a case, the Games-Howell test is performed in respect of each pair of samples. If, however, the null-hypothesis is not rejected, the LSD test is carried out.

### The Fisher LSD *post hoc* test

Although Fisher's LSD test is a popular parametric statistical test, it has been criticised due to a belief that it does not provide sufficient protection against inflated Type 1 error<sup>2</sup> rates, specifically when more than three data sets are being compared [65]. According to Kidd [83], however, Fisher LSD's test is appropriate for multiple *post hoc* comparisons, provided that the results are reported rigorously.

Consider two different data sets,  $A$  and  $B$ , that are being compared. The test statistic of the LSD test at a 5% level of significance is given by

$$LSD_{A,B} = t_{0.05, mn-n} \sqrt{MS_W(1/m_A + 1/m_B)}, \quad (8.10)$$

where  $m_A$  and  $m_B$  denote the number of data points in sample  $A$  and sample  $B$ , respectively. The means of the two samples are declared significantly different at a 5% level of significance if

$$|\bar{x}_A - \bar{x}_B| \geq LSD_{A,B} \quad (8.11)$$

This procedure has to be performed for all  $\binom{n}{2}$  pairs of data sets.

### The Games-Howell *post hoc* test

The Games-Howell *post hoc* test [74, 73] is a non-parametric test employed in the case where the variances between at least two samples are statistically different at a  $(1 - \alpha)$ -level of confidence. The test employs Welch's degrees of freedom (from Welch's  $t$ -test), and the studentised range distribution, denoted by  $q$ , and is given by

$$|\bar{x}_A - \bar{x}_B| > q_{\sigma, n, df}, \quad (8.12)$$

where

$$\sigma = \sqrt{\frac{1}{2} \left( \frac{s_A^2}{m_A} + \frac{s_B^2}{m_B} \right)} \quad (8.13)$$

and

$$df = \frac{\left( \frac{s_A^2}{m_A} + \frac{s_B^2}{m_B} \right)^2}{\frac{\left( \frac{s_A^2}{m_A} \right)^2}{m_A-1} + \frac{\left( \frac{s_B^2}{m_B} \right)^2}{m_B-1}} \quad (8.14)$$

In (8.13) and (8.14),  $s_A$  and  $s_B$  are the standard deviations of data sets  $A$  and  $B$ , respectively. If (8.12) holds, there is a significant difference between the means of the two data sets at a 5% level of significance. Conversely, if the inequality does not hold, the two means do not differ statistically from one another.

<sup>2</sup>A Type 1 error occurs when the null-hypothesis is rejected while it is, in fact, true.

### ***p*-Values in hypothesis testing**

A popular method that may be used to pronounce on the results of a hypothesis test is known as *fixed significance level* testing. According to this method, the results of a hypothesis test are reported by stating whether or not a null-hypothesis should be rejected at a specified level of significance [116]. This test returns a so-called *p-value* that conveys information about the weight of evidence against the null-hypothesis. More specifically, the *p-value* is the probability that the test statistic assumes a value that is at least as extreme as the observed value, provided that the null-hypothesis is true. In other words, the *p-value* is the smallest level of significance that would lead to rejection of the null-hypothesis based on the available data [116]. In practice, the calculated *p-value* is compared with the level of significance  $\alpha$ . If the *p-value* is smaller than  $\alpha$ , the null-hypothesis is rejected. Alternatively, the null-hypothesis is not rejected if the *p-value* exceeds  $\alpha$ . An advantage of this approach is that the *p-value* is computed independently of a predefined significance level. This allows the decision maker the freedom to interpret the statistical significance of the data without having to impose a predefined level of significance. In other words, it provides a convenient method for comparing the results of two or more tests even when employing different significance levels.

## **8.6 Chapter summary**

This chapter was devoted to a description of the experimental design employed in this thesis. This design was performed in the context of a hypothetical nine-facility pharmaceutical supply chain network. The chapter opened with a discussion on the difficulties (in terms of computational power) encountered during the design of the hypothetical supply chain. This was followed by a detailed description of the supply chain network, its underlying structure, as well as the nature of end-user demand considered in the experiments. The algorithmic implementation of the Q-learning algorithm with respect to the state space, the action space, the reward function, and action selection was presented next. This was followed by brief outline of the experimental procedure followed in this thesis, before a brief review of some statistical tests was provided.





---



---

## CHAPTER 9

---

# Results

### Contents

9.1	Analysis of results . . . . .	131
9.2	Statistical analysis of the impact of fluctuating demand . . . . .	132
9.2.1	<i>Experiment 1</i> . . . . .	132
9.2.2	<i>Experiment 2</i> . . . . .	138
9.2.3	<i>Experiment 3</i> . . . . .	143
9.2.4	<i>Experiment 4</i> . . . . .	148
9.2.5	<i>Experiment 5</i> . . . . .	150
9.2.6	<i>Synopsis of the relative effectiveness of information sharing</i> . . . . .	152
9.3	Chapter summary . . . . .	155

The purpose of this chapter is to present and analyse the results obtained during the information sharing effectiveness comparison analysis performed in this thesis. The format in which these results are presented is described in §9.1. This is followed in §9.2 by a detailed analysis of the results obtained during the experiments involving the five information-sharing scenarios. The chapter finally closes in §9.3 with a summary of the chapter content, as well as an outline of the most significant findings.

### 9.1 Analysis of results

The KPI values obtained during the experiments conducted in pursuit of the experimental design described in §8 are presented in the form of box plots in this chapter. The number of stock-outs observed during the experiments (or information-sharing scenarios) are presented alongside one another, in separate box plots. In these plots, median values are denoted by horizontal lines, while mean values are indicated by diamond symbols. Finally, outlier values are denoted by crosses. The amount of inventory over time, on the other hand, is presented in the form of line graphs. In these graphs, each data point is the total amount of inventory held by the facility (or a selection of facilities) at the end of the corresponding day.

The discussion of the results involves references to the relative effectiveness observed for the respective information-sharing scenarios, as well as the outcomes of the statistical tests applied to compare information-sharing effectiveness at a 95% level of confidence.

## 9.2 Statistical analysis of the impact of fluctuating demand

The results obtained during the five experiments (*i.e.* Experiments 1–5) involving the fluctuating demand pattern of §8.2.5 are discussed in this section. This discussion focusses on the observed KPI values as well as the inventory levels observed over time, for each of the five scenarios. These KPIs are evaluated in terms of overall supply chain performance, but also with respect to the comparative performance of individual facilities. Notably, no expiries were observed during any of the five experiments. This may be attributed to the effectiveness of the policies learnt and the fact that all inventory is managed on a *first-expired first-out* principle (as mentioned in §6.1.2). Subsequently, the remainder of the discussion focusses on the remaining KPI, namely the number of stock-outs. More specifically, a distinction is made between the number of end-user stock-outs and the number of order stock-outs when evaluating the effectiveness of the information-sharing scenarios. This is done based on the argument provided in §6.1.5 that order stock-outs upstream may be tolerated given that they do not compromise the service levels of health-care facilities downstream.

The blueprint for the discussion of the results in the remainder of this section is as follows: A systematic analysis of the results obtained during each of the five experiments is first presented. An analysis is concluded thereafter, and a synoptic outline is provided of the relative effectiveness of the five information-sharing scenarios in respect of the KPIs.

### 9.2.1 Experiment 1

The first information-sharing scenario considered in the experimental design did not involve any information sharing between entities and served as a benchmark. The computation time required for training the agents by means of the Q-learning algorithm for Scenario 1 was fifteen hours and comprised a total of 11 950 200 iterations (*i.e.* simulated days). The convergence of the Q-learning algorithm was evaluated with respect to the mean daily amount of inventory in the system over the course of each year, as explained in §8.4. A graphic illustrating how this metric decreased over time during learning in Scenario 1 is shown in Figure 9.1. Each observation in this figure is a forty-year moving average of the mean daily amount of inventory held in the supply chain per year. This figure is representative of the learning behaviour described in §7.5 where, at the start of learning, each agent tended to explore often which led to excessive ordering and manufacturing (resulting in high inventory levels). Each agent consequently spent considerable time in its high inventory states at the start of learning and very quickly learned that it should not order (or manufacture) in these states. As a result, the mean daily amount of inventory in the supply chain decreased considerably over a short period of time, as may be seen in Figure 9.1, after which it decreased continually although at a progressively slower rate. After the point of this dramatic decrease, the amount of inventory in the system remained relatively large (albeit smaller than before) and it took a considerable number of learning iterations before each agent found a balance between a too low and a too high inventory level. The Q-learning algorithm was terminated after more than 33 000 years of simulated time had lapsed during the training simulation run.

Scenario 1 did not involve any information sharing (or inventory sharing) between facilities and this was evident from the results obtained during Experiment 1. The mean number of end-user stock-outs incurred across the entire supply chain was 5 733. This is a relatively small number considering that the total expected demand in the supply chain was 787 500 product units over the entire five-year period. The number of end-user stock-outs experienced by each of the seven health-care facilities (*i.e.* the hospital (Hospital H) and the six clinics (Clinics A–F)) is shown in

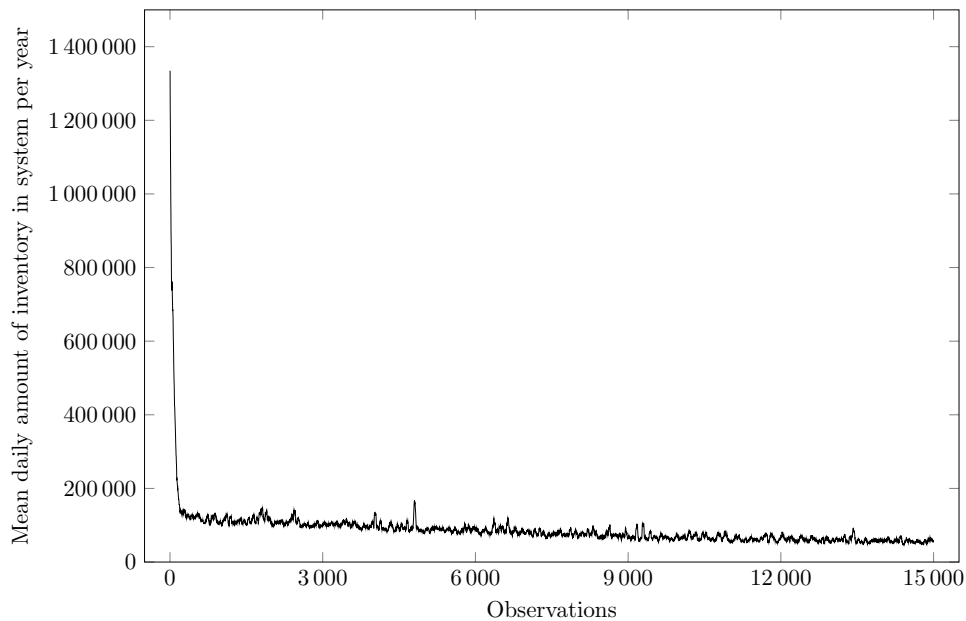


FIGURE 9.1: The learning progression of the Q-learning algorithm over the course of the first 15 000 years of simulated time during Scenario 1. In order to filter out some simulation noise, a moving average over 40 years (of simulated time) is shown.

Figure 9.2. The ANOVA test revealed that there are statistical differences between the means returned by at least one pair of health-care facilities at a 5% level of significance (a  $p$ -value of less than  $1 \times 10^{-17}$ ). The Levene test was performed thereafter to evaluate the variances of the seven data sets and it was found that there is a statistical difference between the variances of at least two samples (a  $p$ -value of less than  $8.2771 \times 10^{-7}$ ). Subsequently, the Games-Howell test was employed to determine between which facilities the differences occur in respect of unfulfilled patient demand.

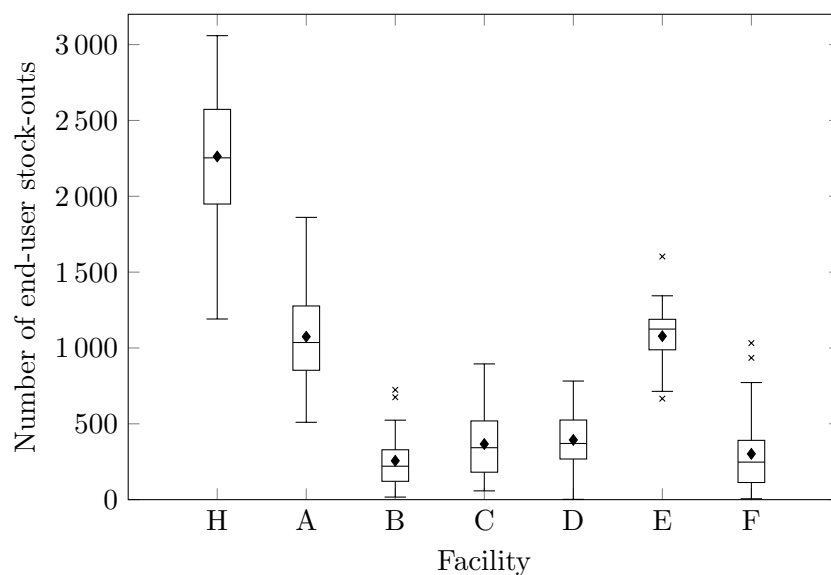


FIGURE 9.2: The number of end-user stock-outs observed at each of the seven health-care facilities during Experiment 1. Facility H is the hospital, while Facilities A–F are the six clinics as per the naming convention provided in §8.2.1.

The hospital incurred an average of 2 262 end-user stock-outs over the five years, which is significantly larger than the number of stock-outs incurred by any clinic during the corresponding time in Scenario 1. Notably, each clinic neighbourhood had one clinic that performed relatively poorly when compared with the other two clinics in the neighbourhood. These under-performing facilities were Clinic A and Clinic E, and they incurred mean numbers of stock-outs of 1 075 and 1 078, respectively. The means returned by these two facilities are statistically indistinguishable at a 95% confidence level (a  $p$ -value of 1). While these two facilities experienced significantly more stock-outs than their peers, there is no statistical difference between the performances of Clinics B, C, D and F in respect of unfulfilled demand at a 5% significance level. Closer inspection of the policies learnt revealed that the reorder points of Clinics A and E tended to be lower than those of their peers. Additionally, these two facilities were often also incapacitated by lead times longer than expected and this led to a large number of stock-outs, even during periods of established high demand (*i.e.* not during demand transitions).

The bulk of the end-user stock-outs observed during Experiment 1 occurred at, or directly after, the points in time where end-user demand increased from low to high. During each five-year (1 800 day) period, the end-user demand transitioned from low to high on days 181, 541, 901, 1 261 and 1 621, respectively. Considerable emphasis is therefore placed in this analysis on the impact of information sharing during these demand transitions.

A graph showing the inventory level of Clinic E over the course of the first 120 days of an arbitrarily chosen replication run during Experiment 1 is shown in Figure 9.3. This figure highlights the significance of the starting inventory level values for any facility, as mentioned in §8.2.4. Clinic E started the first day of the run with 369 units in inventory without any pending replenishment orders or in-transit shipments. As a result, the inventory level decreased continually during the first few days of simulated time. During the first ten days, Clinic E placed three replenishment orders within the space of six days (order quantities of 210, 375 and 540, respectively). The third of these orders, for instance, was placed when the inventory level was 72 units and with the agent in a state it had only visited 975 times during learning, which is relatively few. This is an example of how the agent had not learnt sufficiently for these low inventory states. Considering the nature of the reward function, it may be expected that a clinic would rather place large orders when its inventory is low as opposed to a larger number of consecutive, smaller orders. Nonetheless, the clinic had still managed to avoid stock-outs during this initial phase, which was its primary aim. This would most likely, however, not have been the case if its initial inventory level had been much lower. This phenomenon — where an agent had not spent a sufficient amount of time in its lower-inventory states — led to the proposition of the no-ordering streaks described in §7.5. After day 20, Clinic E largely persisted with order quantities of 540 at relatively evenly-spaced intervals.

All six clinics performed relatively well during the periods of low end-user demand in Experiment 1. The inventory level of Clinic A is shown in Figure 9.4 over the course of the first 180 days of an arbitrarily chosen replication run. As may be seen in this graph, Clinic A exhibited largely consistent ordering behaviour over the course of this half-year period. In most cases, Clinic A placed an order for 540 units at a time, with approximately fifteen days between two consecutive orders. This illustrates the ability of the agent to have learned a policy of not ordering too frequently. In practical terms, this ordering behaviour may translate into reduced ordering and distribution costs. The prioritisation of the avoidance of stock-outs is also clearly visible considering that the inventory level never decreased below 150 during this period. When comparing Figure 9.3 with Figure 9.4, it is clear that Clinic E generally maintained a higher level of inventory during low demand when compared with Clinic A. Despite both of these clinics typically ordering in quantities of 540 at a time, Clinic E maintained a comparatively higher

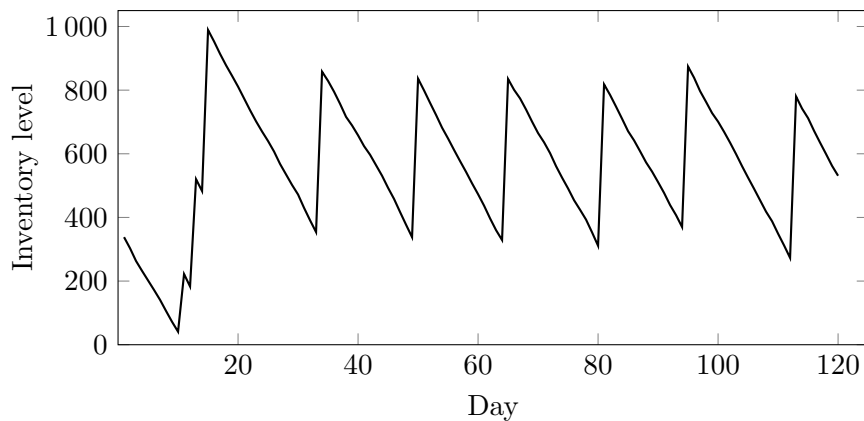


FIGURE 9.3: The inventory level of Clinic E during the first 120 days of low demand during a replication run in Scenario 1.

reorder point than Clinic A. This suggests that the holding cost punishment may not have played an equally prominent role for all clinic agents. Instead, it was overshadowed by the large punishments awarded for stock-outs.

A deviation from Clinic A's mainly consistent ordering profile may, however, be seen right before and after day 160 in Figure 9.4. On day 154, Clinic A placed an order for 210 units and this was followed by an order for 375 units only four days later (before the first order had been fulfilled). Closer inspection of the Q-values associated with the state-action pairs of the states visited on days 154 and 158 revealed relatively small differences for the respective states. These states had also not been visited as many times during learning when compared with the number of visits to other states. It is conjectured that the agent may have learnt to continue with its typical order quantity of 540 on day 154 and not to order on day 158 if it had spent more time in the corresponding states during learning. Nonetheless, the agent's policy still proved effective in terms of avoiding stock-outs during periods of sustained low demand, which was the principal aim of the implementation of reinforcement learning in this thesis.

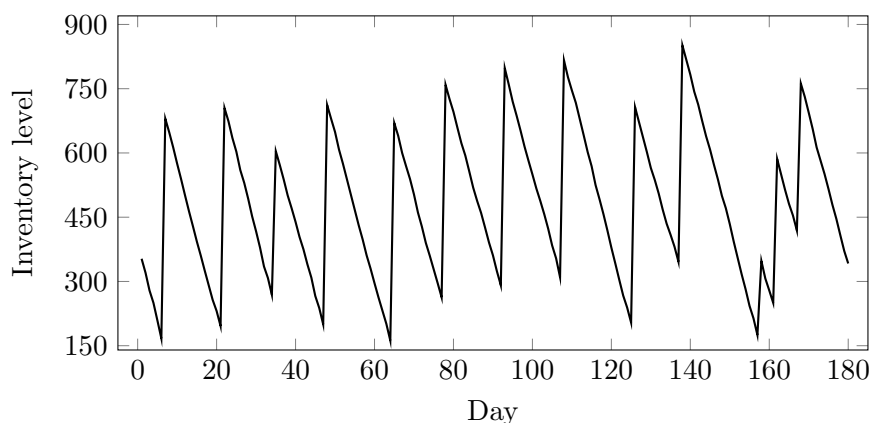


FIGURE 9.4: The inventory level of Clinic A during a period of low demand in Scenario 1.

An example of the effect of a demand increase on the inventory levels of the clinics in Neighbourhood 1 during Scenario 1 is shown in Figure 9.5. The end-user demand increased from low to high on day 1621 at each clinic in the neighbourhood and this is reflected by the changes in the respective slopes in the graph. As may be seen in the figure, each of the three clinics

incurred stock-outs in the aftermath of this demand increase. All three clinics' stock were sufficient for the first three days of high demand. Although the clinics issued new replenishment orders soon after the demand increase was observed, stock-outs were still incurred because of the supplier lead times. Clinic C was, however, affected only for one day because it received a new batch of inventory on day 1625 (the corresponding order was placed before the demand increase). An interesting observation here is that Clinic C concluded day 1626 with 898 units in inventory, while Clinic A and Clinic B incurred stock-outs on the same day as well as on the next day (only Clinic B). This implies that Clinic C had, in fact, sufficient inventory available to satisfy the entire neighbourhood's demand during those two days. This is a striking example of how information sharing (and by implication inventory sharing) could have helped to mitigate the risk of stock-outs in the neighbourhood. After the clinics had reacted appropriately to the change in demand, they tended to continue with order quantities of 540, albeit more frequently than in times of low demand.

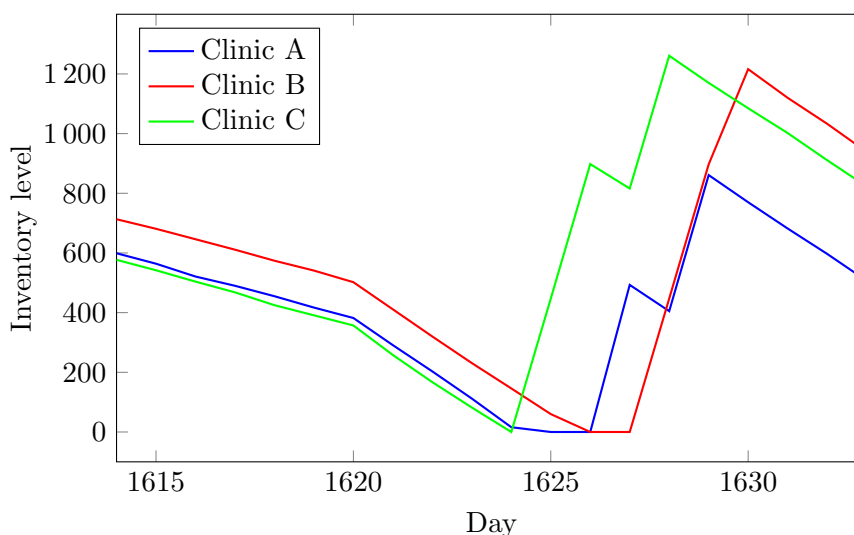


FIGURE 9.5: *The inventory levels of the three clinics in Neighbourhood 1 during a demand increase observed in Experiment 1.*

In any supply chain it is typically the end-user demand downstream that dictates the operations of suppliers and manufacturers upstream. In terms of inventory management, if a neighbourhood of clinics experiences relatively stable demand, their supplier may most likely experience predictable demand in the form of replenishment orders. As a result, the supplier may not always be able to respond sufficiently when a sudden change in demand occurs, and this phenomenon was observed many times during Experiment 1. The inventory level of the hospital observed over five years during a replication run of Experiment 1 is shown in Figure 9.6. Apart from the first demand increase on day 181, the hospital's inventory level was reduced to zero at each of the following four demand transitions from low to high, as well as on days 1394–1396. When the end-user demand increased from low to high, the clinics responded by placing a barrage of orders to the hospital in a very short period of time. Given that the hospital had 'acclimatised' to the relatively low demand up to that point in time, it did not carry enough inventory to deal with the increased demand in some cases, as may be seen in Figure 9.6. Although the hospital could supply the clinics with as much inventory as possible, it did not retain any stock for satisfying its own patient demand and this led to the large number of stock-outs incurred by the hospital (refer to Figure 9.2).

As may be expected, the hospital's response during stock-out periods was to order large amounts of inventory, although this ordering may be considered overly excessive given the inventory levels in excess of 12 000, as indicated in Figure 9.6. This phenomenon — where the inventory level rocketed so dramatically — was observed multiple times during Experiment 1. It is conjectured that the reason for this unrestrained ordering was that the agent had not successfully learnt how much inventory on order may be considered as 'enough.' During stock-outs the agent's natural response was to order persistently because ordering provided the quickest resolve for the stock-out crisis. This led the agent to learn a policy of ordering almost exclusively during stock-outs, irrespective of the amount of inventory already on order. Furthermore, some instances of this unconstrained ordering behaviour took place when the agent encountered states it had not visited sufficiently many times during learning. It may well be that the agent would have learned a more effective policy had it more evenly explored the state space. A thorough exploration of the state space during the training run of Scenario 1 was, however, limited because of the limitations described in §7.5.

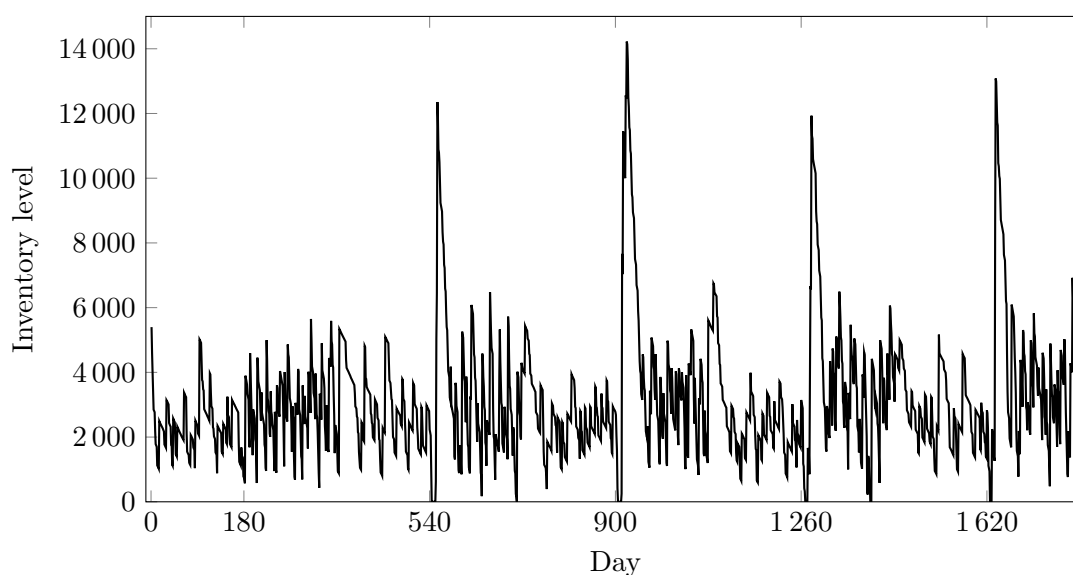


FIGURE 9.6: *The inventory level of the hospital over the course of five years during a replication run of Experiment 1.*

In contrast with the hospital, the warehouse typically returned a more discernible inventory profile over the course of the five-year period. The inventory level of the warehouse during the first 720 days of an arbitrarily chosen replication run (of Experiment 1) is shown in Figure 9.7. From this figure, it is clear that the warehouse's average inventory level is much higher during periods of low end-user demand than during periods of higher demand. During periods of low end-user demand, the warehouse ordered almost exclusively in quantities of 1 500 and did so very infrequently. During periods of high demand, on the other hand, the hospital persisted largely with order quantities of 1 500, albeit it at more regular intervals, and ordered quantities of 3 000 or more somewhat sporadically. In other words, the agent had learned rather to order in smaller quantities at regular intervals, as opposed to ordering more, but less frequently. The rationale behind this behaviour may be explained by the reward function: The punishment for holding inventory in storage was so large that the agent preferred to order in smaller, yet sufficient, quantities because the reward function did not punish 'too frequent' orders. Owing to the fact that the warehouse does not dispense to any patients as the hospital does, it often incurred fewer order stock-outs than the hospital. Similar to the hospital, on the other hand,

the warehouse had also exhibited several instances of unrestrained ordering when its inventory was critically low during Experiment 1.

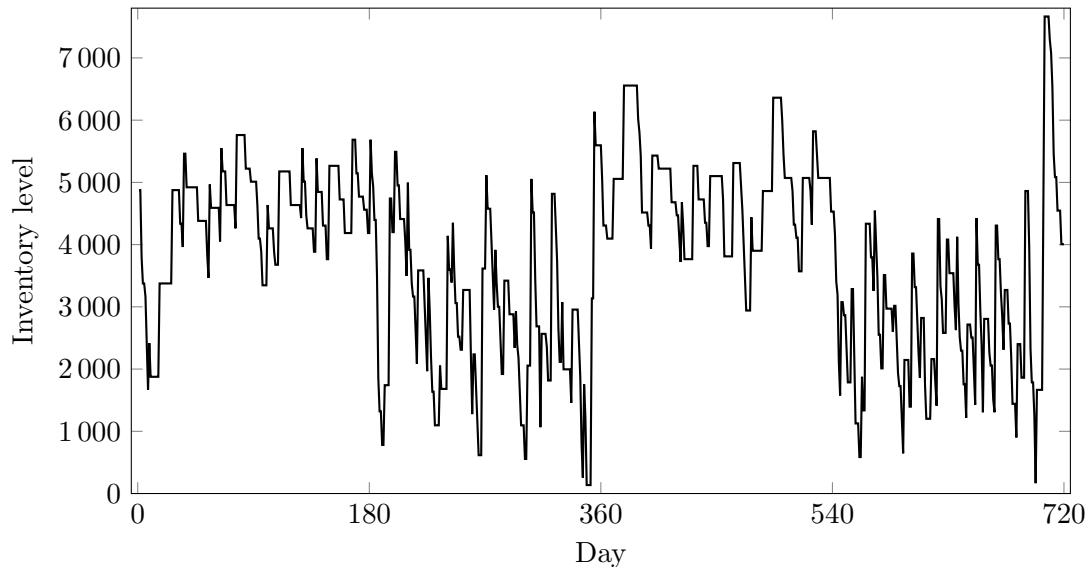


FIGURE 9.7: The inventory level of the warehouse over the course of two years during a replication run of Experiment 1.

The inventory level of the manufacturer over the course of the first four years of a randomly selected replication run of Experiment 1 is shown in Figure 9.8. As may be expected, the manufacturer also experienced a considerable increase in demand when the hospital and warehouse placed large orders as a direct result of the demand increase they experienced from the clinics. Similar to the warehouse, the manufacturer also preferred smaller production quantities over larger production batches, irrespective of the demand. Notably, the manufacturer's inventory level remained above 10000 for the vast majority of days depicted in Figure 9.8. If a decision maker considers this too high, the holding cost punishment may be increased in the future so as to encourage the agent to initiate a production run only when its inventory is much lower. It is, however, important to be cognisant of the fact that the manufacturer still incurred order stock-outs despite these relatively high inventory levels.

The manufacturer also exhibited instances of unrestrained manufacturing albeit more moderate than the warehouse and the hospital. The unconstrained manufacturing often took place as a direct result of complete order stock-outs (*i.e.* inventory level of zero). The manufacturer had, on the other hand, managed to recover sufficiently without excessive manufacturing after the respective demand changes of days 541 and 721, as may be seen in the figure. This is again an indicator that the manufacturing agent had potentially not learnt sufficiently in the states involving critically low inventory levels (because of too few visits).

### 9.2.2 Experiment 2

In Scenario 2, each clinic had visibility over the inventory in its neighbourhood available for exchange at the current point in time. Clinics also enjoyed visibility over incoming orders that were guaranteed to arrive in their neighbourhood within the next 24 hours (*i.e.* the following discrete time step). Coupled with this information visibility, each clinic could also opt to issue an informal order to one (or more) of its neighbours when necessary (most likely when its own inventory was relatively low). In some instances during Experiment 1, it was observed



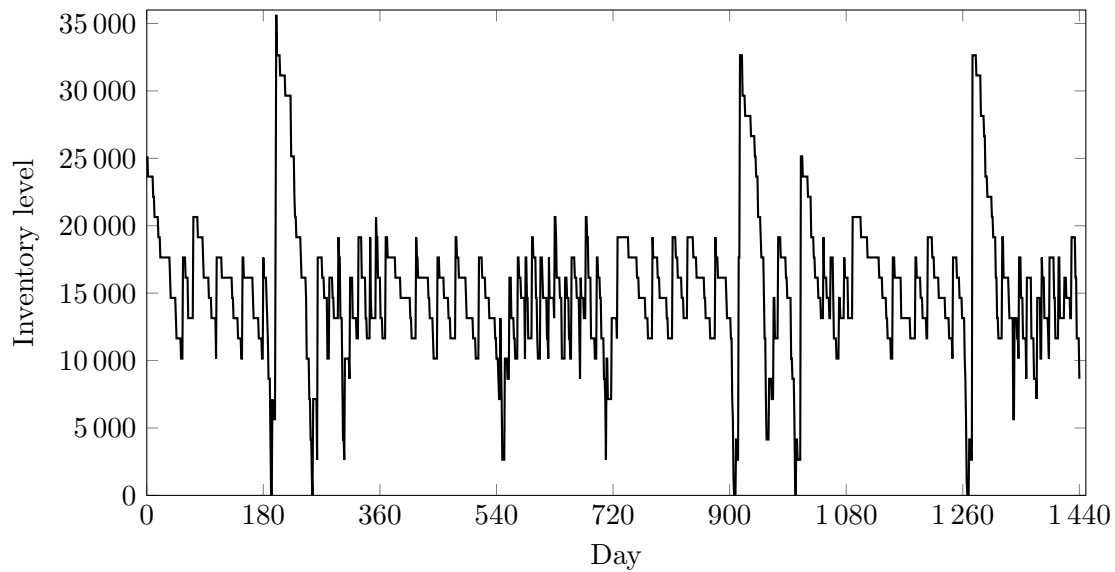


FIGURE 9.8: The inventory level of the manufacturer over the course of four years during a replication run of Experiment 1.

that one or two clinics in a neighbourhood incurred stock-outs despite the aggregate inventory available in the entire neighbourhood being sufficient for the total neighbourhood demand (refer to Figure 9.5). As a result, it may be expected that inventory sharing according to Scenario 2 may lead to a decrease in the number of stock-outs when compared with Scenario 1. Stock-outs may, however, not be eradicated entirely since all of the clinics in any given neighbourhood may potentially experience stock-outs simultaneously.

The mean total number of end-user stock-outs observed during Experiment 2 was 3045 — an average decrease of 46.9% in respect of the benchmark scenario. The number of end-user stock-outs observed at each of the seven health-care facilities during Experiment 2 is shown in Figure 9.9. The mean number of end-user stock-outs incurred by the hospital was 795 in Scenario 2 (compared with 2262 in Scenario 1) and the mean number of stock-outs observed at each of the six clinics varied between 340 and 403, respectively. Arguably the most notable observation here is that all of the clinics seemed to have performed relatively similarly when compared to the differences observed between them in Experiment 1.

The ANOVA test revealed that there are statistical differences at a 5% level of significance between the means of the end-user stock-outs incurred by the seven health-care facilities during Experiment 2 (a  $p$ -value of less than  $1.1 \times 10^{-16}$ ). The Levene test further revealed that there is a statistical difference between the variances of at least one pair of facilities (a  $p$ -value of  $1.6900 \times 10^{-4}$ ) and so the Games-Howell test was performed to determine where the differences in respect of the number of stock-outs occur. The  $p$ -values returned by the Games-Howell *post hoc* test are shown in Table 9.1. These results show that there is a statistical difference between the performances of each hospital-clinic pair, while all six clinics are statistically indistinguishable from one another at a 5% level of significance.

While it is clear that the overall number of stock-outs decreased in Scenario 2, it is worthwhile to inspect the performance of the health-care facilities individually. Comparative box plots showing the differences between the six clinics (the only facilities that were allowed to share inventory) in respect of the number of end-user stock-outs observed during Experiments 1 and 2 are shown in Figure 9.10. Statistical analyses of these results revealed that Clinic B was the

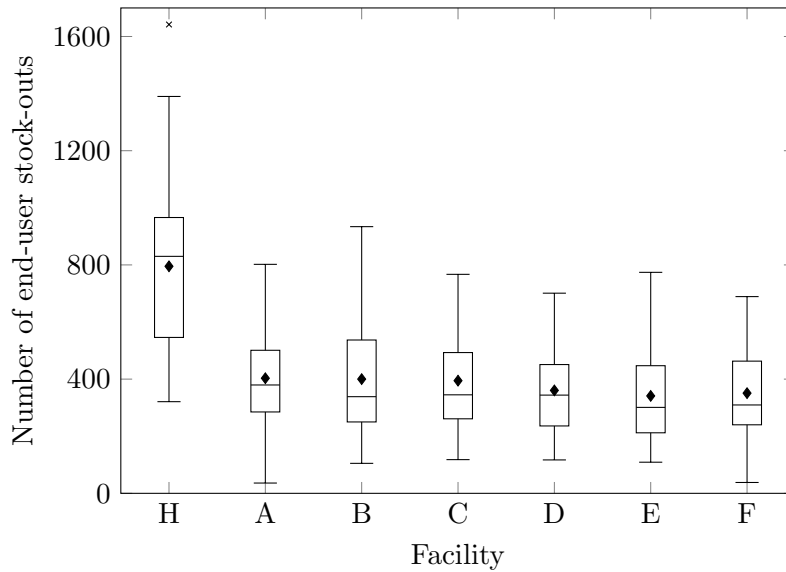


FIGURE 9.9: The number of end-user stock-outs observed at each health-care facility during Experiment 2.

Facility	A	B	C	D	E	F
H	$2.26 \times 10^{-5}$	$2.51 \times 10^{-5}$	$1.12 \times 10^{-5}$	$1.52 \times 10^{-6}$	$8.12 \times 10^{-7}$	$1.08 \times 10^{-6}$
A	—	$9.99 \times 10^{-1}$	$9.99 \times 10^{-1}$	$9.60 \times 10^{-1}$	$8.52 \times 10^{-1}$	$9.14 \times 10^{-1}$
B	—	—	$9.99 \times 10^{-1}$	$9.78 \times 10^{-1}$	$8.99 \times 10^{-1}$	$9.46 \times 10^{-1}$
C	—	—	—	$9.84 \times 10^{-1}$	$9.06 \times 10^{-1}$	$9.54 \times 10^{-1}$
D	—	—	—	—	$9.99 \times 10^{-1}$	$9.99 \times 10^{-1}$
E	—	—	—	—	—	$9.99 \times 10^{-1}$

TABLE 9.1: The Games-Howell test  $p$ -values indicating where significant differences occur between the health-care facilities in Scenario 2 in respect of the number of end-user stock-outs observed. Table entries smaller than 0.05 are typeset in red and indicate a statistical difference between the pair of facilities at a 5% significance level.

only clinic that performed worse during Scenario 2 than during the first scenario at a 5% level of significance (a  $p$ -value of 0.0065). Clinics A and E, on the other hand, produced marked statistical improvements over their Scenario 1 performances (both  $p$ -values less than  $1 \times 10^{-15}$ ). The number of stock-outs incurred by Clinics C, D and F during Scenario 2 are, however, statistically indistinguishable from their Scenario 1 performances at a 5% level of significance (a  $p$ -value of 0.6082 for Clinic C, a  $p$ -value of 0.4840 for Clinic D and a  $p$ -value of 0.3630 for Clinic F). A notable observation is that the respective Scenario 2 box plots of these three facilities are all shorter than their corresponding box plots of Experiment 1. This implies that the introduction of informal inventory sharing in Scenario 2 led to improved consistency in the performances of these facilities. This finding is explained by the fact that each clinic had improved control over its own inventory position when compared with Scenario 1.

A section of the policy learnt by Clinic A during the training run of Scenario 2 is shown in Table 9.2. This table shows a selection of states for which the agent learnt action  $a_4$  to be the most effective — that is the informal ordering action (the action space of the clinic agent was described in §8.3.2). As may be seen in this table, it is evident that the agent had learnt that it should issue an informal order when its effective inventory level is critically low during periods of high demand, and the neighbourhood has a sufficient amount of inventory available

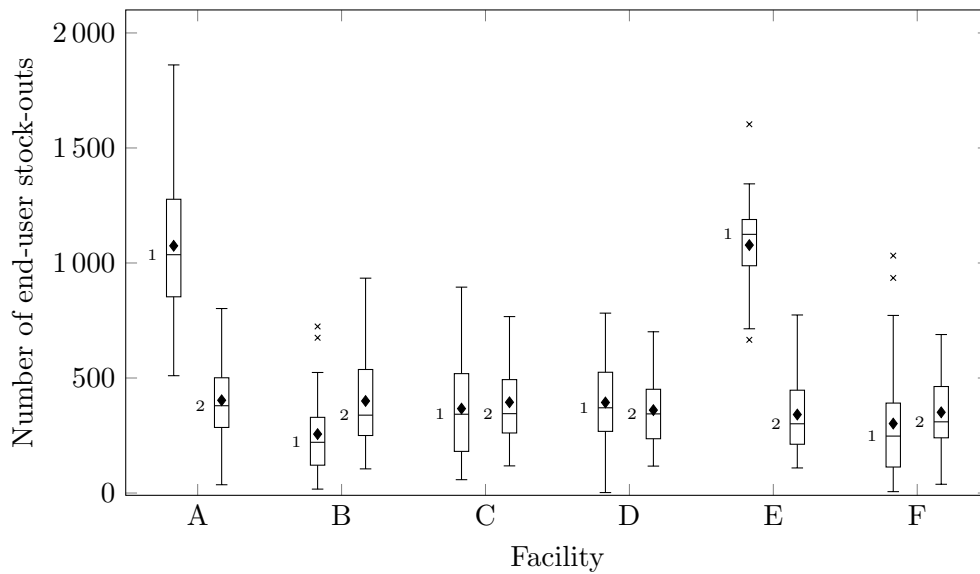


FIGURE 9.10: The number of end-user stock-outs observed at Clinics A–F during Scenarios 1 and 2, respectively. The relevant scenario is indicated to the left of the median of the corresponding box plot.

for exchange. In all of the states shown in this table, the clinic’s amount of inventory on order was between 541 and 1 100. By implication, the agent had already ordered more than the single largest order quantity available to it by the time it encountered any of these states. But if none of this ordered inventory was scheduled to arrive within the next day, the clinic had to resort to informal orders in the interim so as to satisfy the following day’s demand. The ability of each agent to have learned a policy that makes effective use of the informal ordering option is what led to the increased effectiveness observed in Scenario 2.

State	Own inventory	Expiries during lead time	Inventory on order	Demand	Neighbourhood inventory	Greedy action
17 388	0–4	None	541–1 100	High	201 or more	$a_4$
17 389	5–9	None	541–1 100	High	201 or more	$a_4$
17 390	10–14	None	541–1 100	High	201 or more	$a_4$
17 391	15–19	None	541–1 100	High	201 or more	$a_4$
17 392	20–24	None	541–1 100	High	201 or more	$a_4$
17 401	65–69	None	541–1 100	High	201 or more	$a_4$
17 402	70–74	None	541–1 100	High	201 or more	$a_4$
17 403	75–79	None	541–1 100	High	201 or more	$a_4$
17 404	80–84	None	541–1 100	High	201 or more	$a_4$
17 405	85–89	None	541–1 100	High	201 or more	$a_4$

TABLE 9.2: A portion of the policy learnt by Clinic A during the training run of Scenario 2.

An interesting example of how inventory sharing between clinics may help to mitigate stock-outs was observed during Experiment 2 and this particular instance is explained with reference to Table 9.3. This table contains information pertaining to the inventory held by Clinics D, E and F, and their actions taken at the end of each day for a period of nine days following a demand increase during a replication run. At the end of day 181 (the day of the demand transition), each of the three clinics had sufficient inventory to fulfil demand during the following day (the maximum daily demand is 100 units). At the end of day 182, however, Clinic E held only 19 units

of inventory and consequently placed an informal order to Clinic A (whose inventory level was relatively high) so as to avoid stock-outs on day 183. At the end of the following day (day 183), Clinic E issued a formal replenishment order despite holding only 19 units in storage, because it had visibility over an incoming delivery scheduled to arrive at the start of day 184. In other words, it could afford not to place an informal order again because it knew its inventory level would be sufficient at the start of the next day to satisfy end-user demand. Similarly, Clinic F ordered from its neighbours on days 185 and 186 when its own inventory level was critically low. At the end of day 186, Clinic F ordered inventory from both neighbours, since neither of them had a sufficient inventory level (each clinic must retain at least 100 units when making inventory available for sharing as explained in §7.1). As a direct result of this inventory-sharing business rule, all three clinics managed to avoid stock-outs successfully during day 187. At the end of day 187, Clinics D and E could, in turn, order from Clinic F because the latter was guaranteed to receive a large delivery at the start of the next day. This example serves as a demonstration of how the clinics managed to self-organise in order to minimise stock-outs across their entire neighbourhood.

Day	Clinic D		Clinic E		Clinic F	
	Inventory	Action	Inventory	Action	Inventory	Action
181	741	0	107	0	423	0
182	648	0	19	4	331	3
183	472	0	19	3	246	3
184	382	3	308	0	152	0
185	294	0	220	3	61	4
186	110	2	129	0	55	4
187	13	4	11	4	1	3
188	11	0	3	0	276	0
189	462	0	458	0	195	0

TABLE 9.3: A sequence of the actions taken by the clinics in Neighbourhood 2 during a replication run of Experiment 2.

Given the randomness involved in the simulation, it may be expected that clinic neighbourhoods cannot eliminate stock-outs exclusively such as in the instance described above. It may, of course, happen that neither clinic in a neighbourhood has inventory available for sharing at some point in time. As a result, stock-outs may be observed at each neighbourhood member simultaneously. Instances of this phenomenon were noted during Experiment 2, although very seldom. The total amount of inventory available in Neighbourhood 1 (*i.e.* Clinics A, B and C) over time during a replication run of Experiment 2 is shown in Figure 9.11. As may be seen in the figure, the aggregate neighbourhood inventory rarely reached zero. In fact, over the course of the entire five-year period, the aggregate neighbourhood inventory reached a level of zero on four days only (days 547, 548, 1 267 and 1 627, respectively). Stock-outs were observed at all three clinics during, and only during, these four days. On all other days during which stock-outs occurred in the neighbourhood, at least one clinic managed to avoid stock-outs entirely as a result of the inventory sharing.

Similar to Scenario 1, the clinics responded to demand increases during Scenario 2 by placing large orders with their respective suppliers. Unlike in Scenario 1, however, the clinics could mitigate stock-outs in the interim (while waiting for the orders) by means of informal ordering. Due to the clinics alternating between formal and informal orders, the hospital and the warehouse experienced comparatively less demand than in Experiment 1. The clinics had, nevertheless, still placed relatively large orders at the start of a stock-out period, which placed the warehouse and

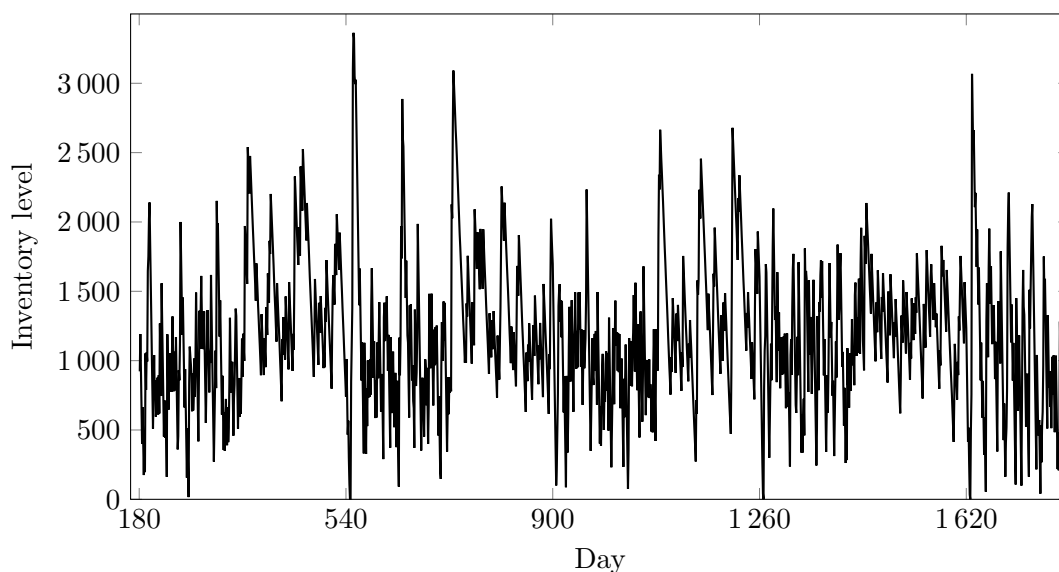


FIGURE 9.11: *The inventory level of Neighbourhood 1 over time during a replication run of Experiment 2.*

hospital under pressure once more. In response to the heightened demand, these two suppliers, in turn, placed large orders with the manufacturer and the latter also struggled to adapt to this demand fluctuation in a timely fashion. Subsequently, the inventory level profiles of these three supplying facilities were relatively similar to those observed during Scenario 1. Instances of the unrestrained ordering and manufacturing described in §9.2.1 were once more observed during Experiment 2. By implication, these agents had again not learnt the most effective actions for the states involving relatively small amounts of inventory.

### 9.2.3 Experiment 3

Compared with Scenario 2, an additional layer of information sharing was added in Scenario 3. In particular, each warehouse and hospital had visibility over the aggregate inventory levels of their customer clinics as well as over the mean demand experienced by those clinics. It may be expected that this added visibility would enhance the respective capabilities of the warehouse and hospital to respond more effectively to changes in end-user demand when compared with Scenario 2.

The results obtained during Experiment 3, however, failed to provide sufficient evidence in support of this conjecture. The supply chain, in fact, delivered worse performance in respect of unfulfilled end-user demand during Scenario 3 than in Scenario 2. The total number of end-user stock-outs observed during Experiment 3 was 4 241 — an average increase of 39.2% when compared with Scenario 2. The results obtained during the third experiment are, however, statistically superior to the effectiveness observed during Experiment 1. There are two primary factors that conspired towards the realisation of the poor performance during Scenario 3. The first is that the warehouse and the hospital typically carried less inventory than during the first two scenarios, and this increased their likelihood for incurring stock-outs substantially. The second motivating factor was that the warehouse and the hospital were also too slow in detecting changes in end-user demand in a timely fashion. As a result, both these facilities incurred substantial order stock-outs which, in turn, led to multiple stock-outs at the clinics downstream.

Each replication run contained five periods of low end-user demand and five periods of high end-user demand (*i.e.* ten demand periods). In order to gain insight into the mean amount of inventory carried by a facility over the course of the five years, the mean daily amount of inventory held by any facility was calculated for each of the ten demand periods. In other words, ten observations were produced and each observation was the mean daily amount of inventory held by a facility over the course of a 180-day demand period (over the 30 replication runs). A graphic depicting the mean daily amount of inventory held per demand period by the warehouse and the hospital during the first three Scenarios is shown in Figure 9.12. In each graph, Observations 1, 3, 5, 7 and 9 correspond with the low end-user demand periods while the remaining observations were made during periods of high demand. As may be seen in both Figures 9.12(a) and 9.12(b), the hospital and the warehouse each carried much less inventory on average during Scenario 3 than during Scenarios 1 and 2.

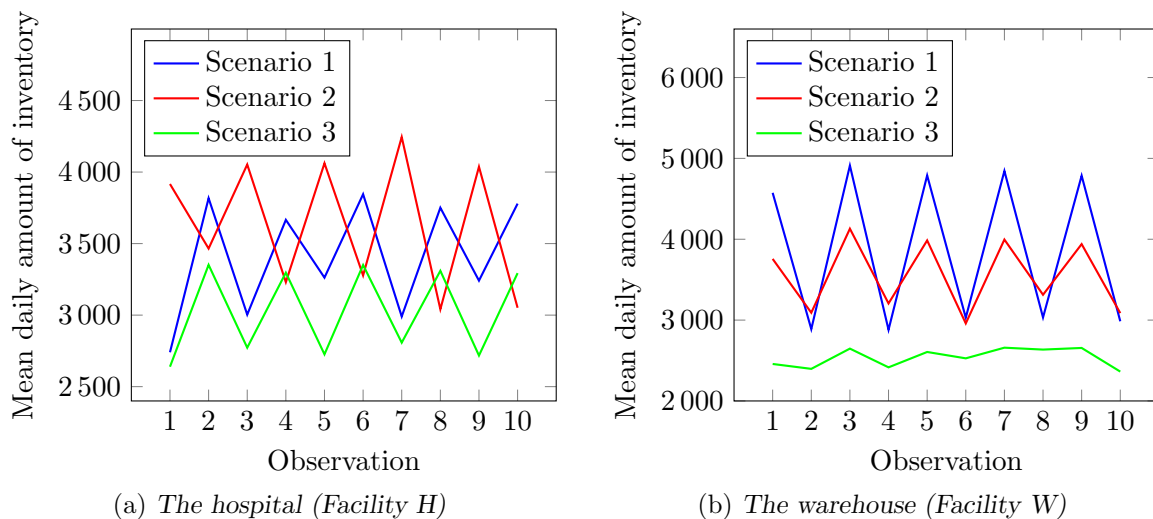


FIGURE 9.12: The mean daily amount of inventory held by the hospital and the warehouse during the ten demand periods for each of the first three scenarios.

For the warehouse and hospital agents, the customer clinics' inventory levels state variable was discretised into two integer intervals, as mentioned in §8.3.1. Subsequently, these two agents could only distinguish between an aggregate customer neighbourhood inventory level of less than 500, or greater than 500. Closer inspection of the results revealed that a neighbourhood that carried more than 500 units of inventory at any given time tended to avoid stock-outs in the short-term. During periods of low end-user demand, the clinic neighbourhoods carried relatively high levels of inventory (much higher than 500) and therefore the clinics did not order as frequently. Newly equipped with visibility over the clinics' high inventory levels during Scenario 3, the warehouse and hospital could afford the luxury of carrying less inventory than during the first two scenarios. Although the lower inventory levels may be desirable in terms of holding cost, this posed a significant problem during substantial demand fluctuations.

During Experiment 3 it was observed that, when the aggregate clinic neighbourhood inventory decreased below 500, it typically happened only two or more days after the initial demand increase (*i.e.* the first day of a new high-demand period). Furthermore, end-user demand was measured as a five-day moving average of the daily demand, as mentioned in §7.1. As a result, the warehouse and hospital occasionally perceived end-user demand as 'low' during the first day or two of a high-demand period when it was, in fact, high. The implications of these phenomena were that the two suppliers were often too slow in recognising the demand transition. Coupled

with their already low inventory levels, this led to substantial order stock-outs which ultimately came to the detriment of the clinics. Although the clinics could still share inventory between them directly after the demand increase (as observed in Scenario 2), they often incurred stock-outs at a later stage because of the suppliers' delayed responses and low inventory levels.

The aggregate inventory level of Neighbourhood 2 observed over a period of forty days during five separate replication runs of Experiment 3 is shown in Figure 9.13. As may be seen in this figure, the total amount of inventory in the neighbourhood often exceeded a value of 500 during either low or high demand (before day 180 and after day 190). It was only in the direct aftermath of the demand increase on day 181 that the aggregate inventory level decreased more rapidly than before. The earliest instances of the aggregate inventory level decreasing below 500 were observed on day 183 for replication runs 1, 2 and 4, respectively. In other words, the warehouse perceived these clinics to have 'sufficient' inventory available on days 181 and 182 during these three replication runs.

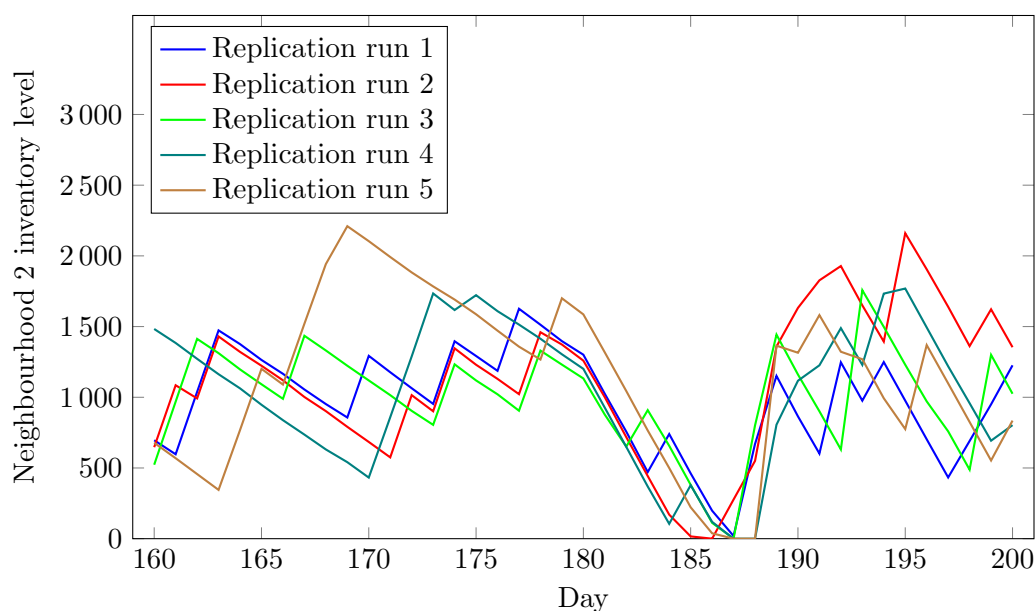


FIGURE 9.13: The total amount of inventory in Neighbourhood 2 during five separate replication runs of Experiment 3.

A selection of the states perceived by, and actions taken by, the warehouse in replication run 2 of Figure 9.13 is shown in Table 9.4. This table contains a selection of the warehouse's state variables perceived over the course of days 179–190. As shown in this table, the warehouse perceived end-user demand as 'low' on day 181 (when it was, in fact, high) and the aggregate neighbourhood inventory level was 1000 units at the same time. It was only on day 182, however, that the warehouse detected the demand as high and responded accordingly by placing a replenishment order for 3000 units. This action, however, proved to be too late and led to substantial order stock-outs on days 186–189. Since the warehouse could not fulfil the heightened demand in a timely fashion, all three clinics in Neighbourhood 2 incurred stock-outs on day 186.

Comparative box plots showing the unfulfilled demand observed at each of the seven health-care facilities during Scenario 3 is shown in Figure 9.14. Similar to the results obtained during Experiment 2, the performances of the clinics were statistically indistinguishable at a 5% level of significance (a  $p$ -value of greater than 0.05 for all clinic pairs). This may be attributed to the fact that all the clinics in a neighbourhood were affected equally when stock-outs incurred upstream. For instance, when either the hospital or the warehouse was out of stock at any given

Day	State number	Own inventory	Clinics' inventory levels	Perceived end-user demand	Order quantity
179	2 445	2 056	1 373	Low	0
180	2 445	2 056	1 259	Low	0
181	2 410	3 556	1 000	Low	0
182	7 208	3 016	718	High	3 000
183	6 085	1 936	443	High	0
184	6 083	1 396	170	High	0
185	6 081	436	16	High	0
186	6 320	0	0	High	0
187	6 320	0	277	High	0
188	8 720	0	550	High	0
189	8 720	0	1 359	High	0
190	8 405	1 986	1 632	High	3 000

TABLE 9.4: A selection of the states perceived by, and actions taken by, the warehouse during a replication run of Experiment 3.

point in time, they could not fulfil the demand of any one of their customer clinics at that time. Given the homogeneous end-user demand profile, all clinics experienced stock-outs of more or less the same magnitude (specifically during stock shortages upstream).

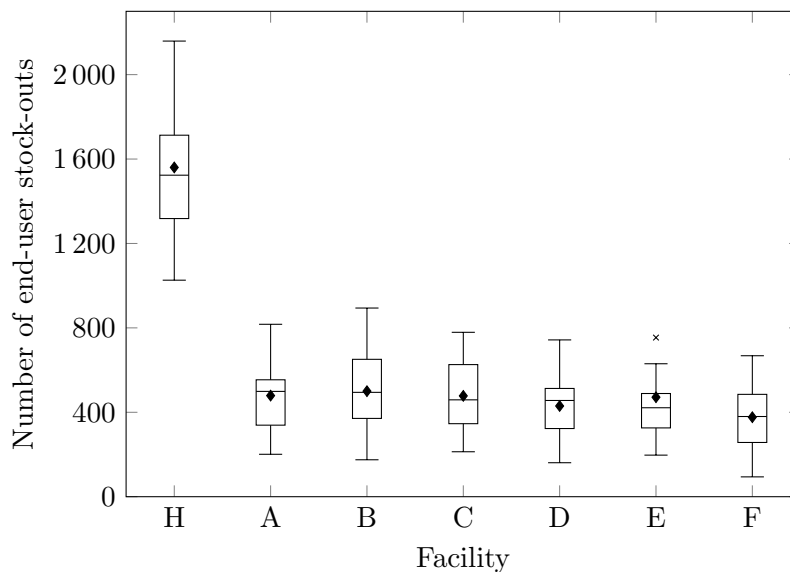


FIGURE 9.14: The number of end-user stock-outs observed at each health-care facility during Experiment 3.

It is evident that the state spaces of the warehouse and hospital agents played defining roles in shaping the policies learnt by these agents during the training run of Scenario 3. More specifically, the discretisation of the clinics' inventory levels and end-user demand state variables was arguably performed too coarsely and this limited the performance of the Q-learning algorithm. A potential approach towards refining the policies learnt by warehouses and hospitals in the future may be to provide a more finely-grained representation of the clinics' inventory level state variable. Although this will increase the size of each agent's state space, access to more detailed information may allow these agents to learn more effectively. A potentially more suitable



approach may, however, be to rather employ a state variable capturing the rate of change in the clinics' inventory levels as opposed to an instantaneous value not linked to time. A second potential intervention involves the manner in which end-user demand is estimated by facilities upstream. In the context of Experiment 3, it was shown that estimating the end-user demand as a moving average over a sample window of five days was inadequate. It is therefore expected that a shorter sample window may provide improved effectiveness than a five-day window. This may, however, not be desirable since a too small window size may lead to an inaccurate estimation of actual demand, specifically when demand is volatile. The end-user demand state variable may further also be discretised into more intervals (of smaller magnitudes) in order to estimate demand more accurately. Since the actual demand distribution is typically unknown in practice this may, however, not be a trivial task.

In order to gain a better understanding of the influence of the sample window size as discussed above, Experiments 1, 2 and 3 were repeated with every facility allowed to estimate end-user demand (as either low or high) based on a sample window of one day. This enabled each clinic to correctly identify the actual demand class on any given day, for each of the three scenarios. Furthermore, there was no delay in the classification of end-user demand by the warehouse and the hospital during the third scenario. Given that the warehouse and the hospital did not have visibility over end-user demand during the first two information-sharing scenarios, the sample window size only took effect for them from Scenario 3 onwards. The repeat of Experiment 1 with a sample window of one day is called Experiment 1(a) and the remaining two are called Experiment 2(a) and Experiment 3(a), respectively. Comparative box plots showing the results obtained in respect of unfulfilled end-user demand during Experiments 1–3 for sample window sizes of one and five, respectively, are shown in Figure 9.15.

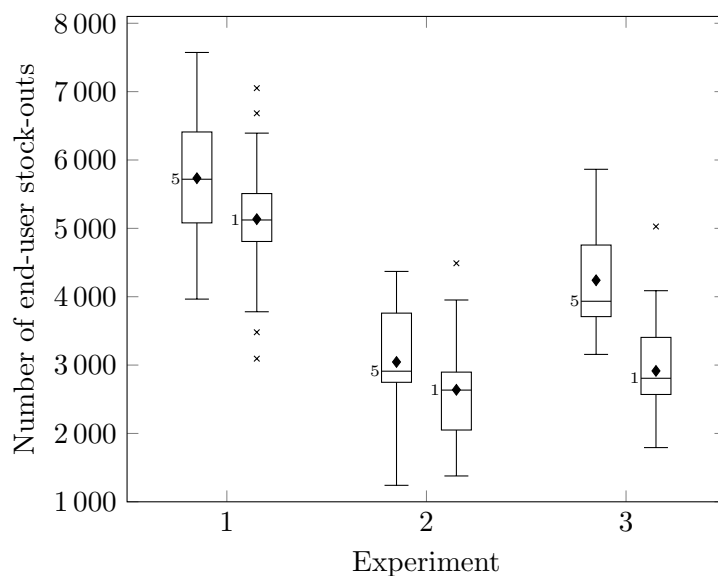


FIGURE 9.15: The number of end-user stock-outs observed during Experiments 1, 2, 3 with sample window sizes of one and five, respectively.

Although each training simulation run was originally performed based on a five-day moving average estimation of end-user demand by all agents, it is clear that the policies learnt proved more effective when they were implemented in the context of a smaller sample window (of one day). For each experiment, end-user demand estimated based on a one-day moving average of end-user demand yielded statistically superior results over its five-day counterpart at a 5% level of significance (a  $p$ -value of  $8.2778 \times 10^{-3}$  for Experiments 1 and 1(a), a  $p$ -value of  $3.3527 \times 10^{-2}$

for Experiments 2 and 2(a), and a  $p$ -value of  $8.5072 \times 10^{-9}$  for the third experiment pair). The respective improvements observed during the first two information-sharing scenarios may be attributed to the clinics placing orders comparatively sooner directly after a demand increase. During Experiment 3(a) it was observed that the warehouse and hospital still carried relatively low inventory levels as was the case in the original experiment. But since the hospital and the warehouse could detect demand changes immediately and effectively, they managed to respond much quicker (by placing orders in a timely fashion) when compared with their performances in the original Experiment 3. Notably, Experiments 2(a) and 3(a) are statistically indistinguishable at a 5% level of significance (a  $p$ -value of 0.1429). This finding suggests that the generally larger amounts of inventory carried by the warehouse and the hospital coupled with their lack of visibility over end-user demand during Experiment 2(a) were neutralised by their comparatively lower inventory levels and quicker response times (based on the visibility over the most recent end-user demand) during Experiment 3(a).

Compared with the benchmark scenario, the mean number of stock-outs decreased by 26.0% during Experiment 3. In view of the discussion above, it may be concluded that it was the informal inventory sharing between clinics, as opposed to the added visibility granted to the warehouse and the hospital, that had a more prominent influence on this improved performance.

#### 9.2.4 Experiment 4

The fourth information-sharing scenario was investigated during Experiment 4. According to this scenario, the warehouse and the hospital could share inventory between them and they also had some visibility over the manufacturer's inventory position. The mean number of end-user stock-outs observed during Experiment 4 was 3456, an average decrease of 18.5% when compared with Experiment 3. This result suggests that, despite the five-day end-user demand sample window, the added information-sharing capabilities of Scenario 4 managed to provide a considerable improvement over the performance over Experiment 3.

Careful inspection of the policies learnt during the training run of Scenario 4, however, revealed that the warehouse and the hospital had not necessarily learnt to make effective use of their informal ordering action. This may be attributed to a relatively poor exploration of their respective state spaces. These agents did not manage to explore their lower inventory states sufficiently during learning, despite the Q-learning algorithm reaching convergence. For instance, out of 13 440 possible states involving an inventory level of 2 499 or less, and 0–33% expiries during the upcoming lead time, the hospital agent had only visited 295 of these states a 100 times or more. Out of the warehouse agent's 6 720 possible states corresponding with the same two state variables, it had only visited 278 of these states more than a 100 times. This occurred despite the total number of learning iterations involved in the training run of Scenario 4 being 43 250 000. This is yet another clear indication of the limitations involved in the particular implementation of reinforcement learning in this context and raises the question of whether or not no-ordering streaks could have enhanced learning.

Despite this apparent shortcoming in the performance of the Q-learning algorithm, the warehouse and the hospital rarely encountered critically-low inventory level states during Experiment 4. This may be attributed to more regular instances of unrestrained ordering (as discussed in the previous sections) which led to generally higher inventory levels for both of these facilities. A graph showing the daily amount of inventory held by the hospital over the course of four years during three separate replication runs of Scenario 4 is shown in Figure 9.16. In contrast with the previous experiments, the hospital ordered extremely large amounts of inventory much more frequently during Scenario 4. The warehouse, on the other hand, did not exhibit the same

erratic ordering pattern although some dramatic spikes in the amount of inventory carried were observed sporadically. The hospital agent had 76 800 states during Scenario 4 — twice as many states as the warehouse agent. In view of this difference, it is conjectured that the hospital agent struggled more to learn an effective policy because its state space was more fragmented. It may have happened that the Q-learning algorithm converged prematurely during learning and therefore deprived the agent of an opportunity to learn more effectively. Despite the relatively higher inventory levels, the warehouse and hospital still incurred order stock-outs somewhat sporadically, resulting in unfulfilled patient demand.

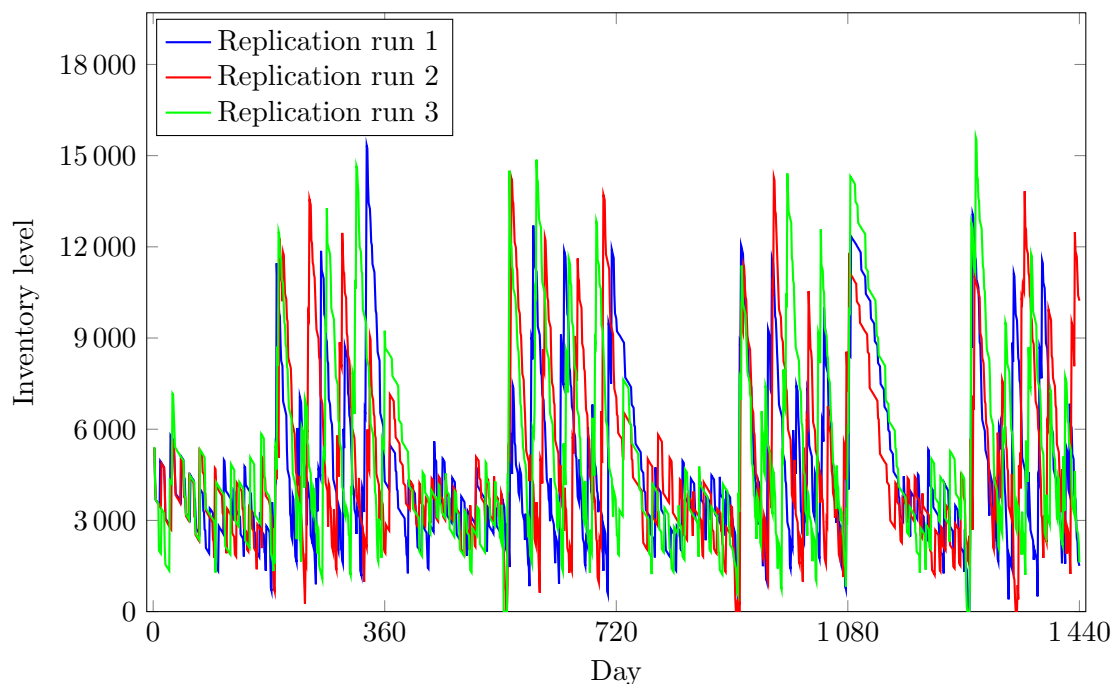


FIGURE 9.16: *The inventory level of the hospital during three replication runs of Experiment 4.*

A relatively small number of informal ordering actions was observed at either the warehouse or the hospital during the respective replication runs of Experiment 4 (even when inventory levels were relatively low). An analysis of the results obtained in Experiment 4 revealed that these facilities typically preferred formal replenishment orders over informal replenishment orders. During the twenty-first replication run of Experiment 4, for example, it was observed that the warehouse placed three consecutive formal replenishment orders on days 185, 186 and 187 (demand increased to high on day 181). The warehouse performed these actions as a direct result of its decreasing inventory level and its realisation of the change in end-user demand. The warehouse, however, resorted to informal ordering only when it did not carry any inventory at all. This was observed during days 188, 189 and 190.

The potential reasons for this behaviour are twofold. The first is that the agent may have learnt to better utilise the informal ordering option had it spent sufficiently more time in the low inventory states. Another motivation may be that the punishment awarded for an informal ordering action was too large. Furthermore, the expected lead time for a delivery between the warehouse and the hospital was four days. A combination of this lead time window and the large penalty may have conspired in such a way that it encouraged the agent to prefer formal orders over informal orders. In the example discussed above, it was interesting to note that the warehouse issued informal orders only at times when its inventory level was zero. This implies that the agent had, in fact, recognised that informal ordering was a solution towards replenishing

stock and the informal order punishment may have not been too large after all. In other words, there were encouraging signs showing that the agent had learnt effective actions although this occurred only in isolated instances. Due to time limitations, it was not possible to experiment further with different values of the informal ordering punishment.

The most significant improvement in respect of the number of end-user stock-outs in the context of Scenario 4 was observed at the hospital. Given that the hospital often held more inventory than during Scenario 3, it enhanced its ability for satisfying its own end-user demand. The hospital and the warehouse had, however, still incurred order stock-outs sporadically during Scenario 4 as a direct result of the fluctuations in end-user demand.

### 9.2.5 Experiment 5

Scenario 5 was the only information-sharing arrangement during which the manufacturer had access to information other than its own local information. The warehouse had visibility over the aggregate inventory levels of its primary customers as well as over its secondary customers (the clinics). Additionally, the manufacturer could also observe the end-user demand observed at the respective health-care clinics. The computation time required for training all the agents during Scenario 5 was 169 hours and comprised 52 400 000 learning iterations.

The mean number of stock-outs observed in the entire supply chain during Experiment 5 was 3 589. This was a 3.8% increase from the number of stock-outs observed during Experiment 4. The two means returned by these two experiments are, however, statistically indistinguishable at 5% level of significance. This implies that the added visibility afforded to the manufacturing entity during this scenario was ineffective in terms of reducing the total number of end-user stock-outs. By the same token, supply chain performance did not worsen in respect of the unfulfilled demand KPI when comparing Scenario 4 and Scenario 5. During Scenario 5 the warehouse and the hospital exhibited considerably fewer instances of unrestrained ordering than in the previous experiments.

Given that the fundamental difference between Scenarios 4 and 5 lies in the manufacturer's supply chain visibility, the analysis in this section is focussed on the performance of the manufacturing entity during Experiment 5. The mean daily amount of inventory held by the manufacturer per demand period during each of the five experiments is shown in Figure 9.17. The highest mean inventory levels were observed during Scenario 4. This is explained by the occasionally erratic ordering behaviour of the warehouse and the hospital which compelled the manufacturer to carry large amounts of inventory. The mean inventory levels observed during the first three experiments were comparatively lower because of the more evenly distributed demand (in the form of orders from the warehouse and the hospital). A graph illustrating how the manufacturer generally carried smaller amounts of inventory during periods of low demand in Experiment 5 than in Experiment 4 is shown in Figure 9.18.

Arguably the most significant observation was that the mean daily amount of inventory held by the manufacturer during Scenario 5 appears to be the inverse of the profiles observed during the first four scenarios. In Experiments 1–4 the manufacturer generally carried more inventory during periods of low demand than during periods of high demand. The results obtained from Experiment 5, however, revealed that the manufacturer carried lower levels of inventory during periods of low demand and larger amounts of inventory during periods of high-user demand. During periods of low end-user demand, Scenario 5 was the scenario that yielded the lowest mean daily amount of inventory carried. By implication, the visibility over the amount of inventory available downstream had allowed the manufacturer to carry comparatively less inventory. This

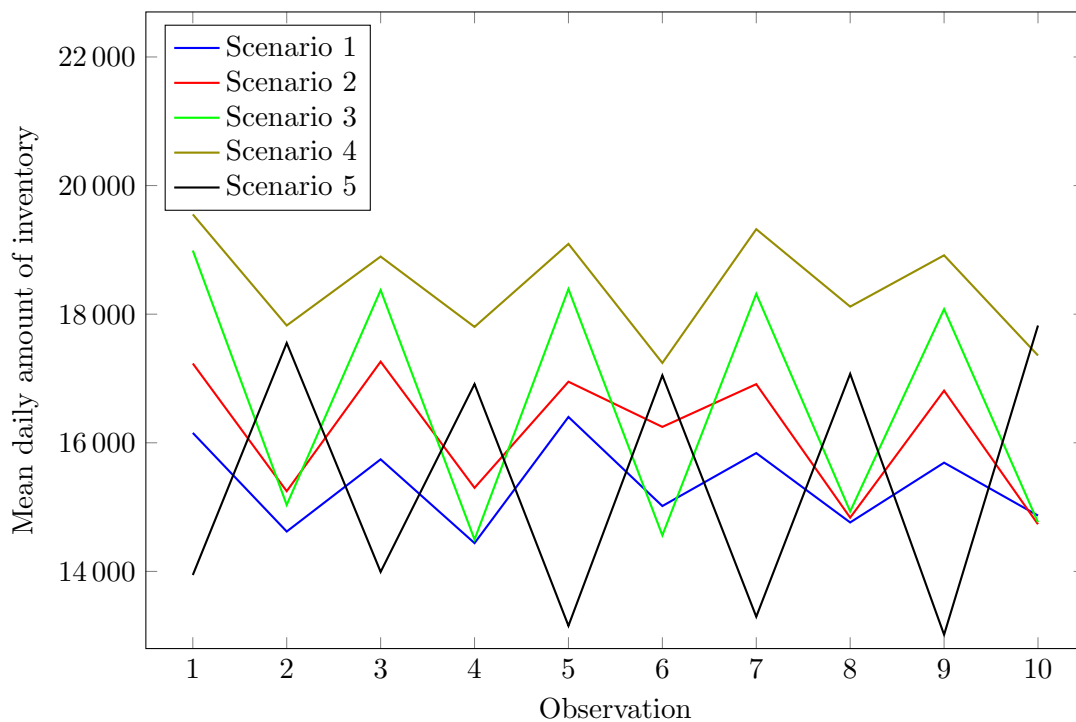


FIGURE 9.17: The mean daily amount of inventory held by the manufacturer during the ten demand periods for each of the five scenarios.

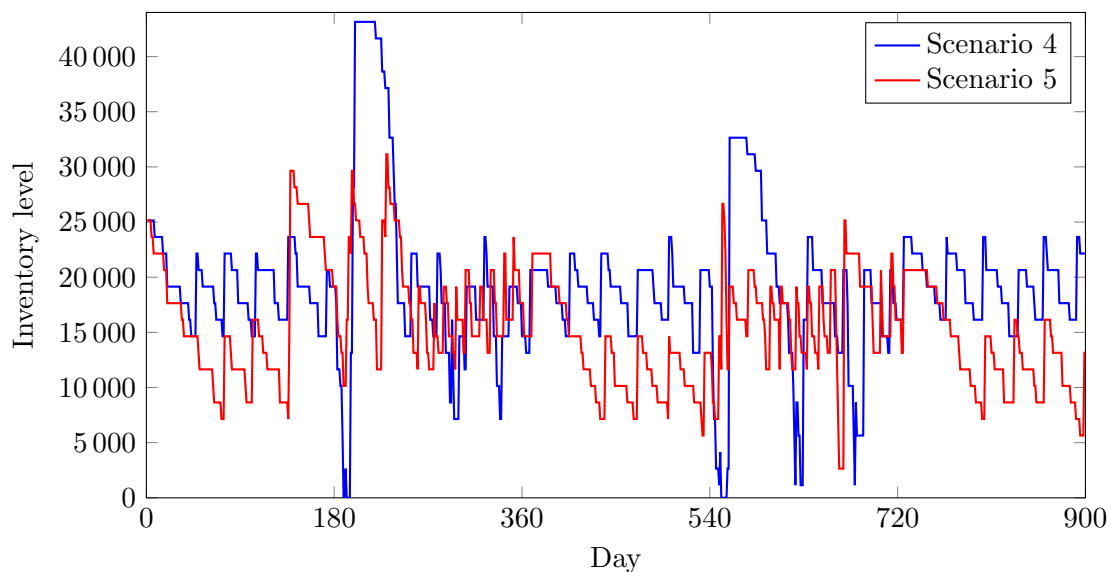


FIGURE 9.18: The amount of inventory carried by the manufacturer during two separate replication runs of Experiments 4 and 5, respectively.

was observed specifically during periods of established low demand because the risk of order stock-outs was relatively small.

Similar to the observation during Scenario 3, the discretisation of the customers' inventory levels and end-user demand for the manufacturer was arguably performed too coarsely. As a result, the manufacturer could not make more effective use of the information types it had at its disposal. The customers' aggregate inventory levels decreased below the respective critical state variable

values far too late for the manufacturer to react in a timely fashion. Because of the five-day moving average employed in respect of quantifying end-user demand, the manufacturer could also only perceive a change in the demand period a day or two after the first day of the new high-demand period.

### 9.2.6 Synopsis of the relative effectiveness of information sharing

The objective in this section is first to provide a condensed summary of some of the most salient observations made during each of the five experiments. Thereafter, the relative effectiveness of the five information-sharing scenarios is evaluated statistically in respect of the unfulfilled demand KPI. The discussion closes with a critical appraisal emphasising the fact that the results should be interpreted conceptually.

The results obtained during Experiment 1 revealed that the policies learnt by the respective agents not always manifested in consistent ordering behaviour. The instances of occasional unrestrained ordering or manufacturing observed showed that the agents had not always explored their respective state spaces sufficiently during learning. It may also be that the discretisation of some state variables was performed too coarsely and that this potentially led agents to perceive functionally different situations as identical. A key outcome of Experiment 1 was a demonstration of instances where clinics incurred stock-outs despite a sufficient amount of inventory being available at neighbouring clinics at the same time instant.

The favourable impact of informal inventory sharing between clinics was clearly illustrated during Experiment 2. The results showed that all the clinic agents had successfully learnt to order from neighbours when their own inventory levels were critically low and those of their neighbours sufficiently high. Due to a greater degree of control over their own inventory levels, the variance in unfulfilled demand decreased considerably when compared with Experiment 1. It is important to stress the influence that the visibility over incoming shipments had in the improved effectiveness observed during Experiment 2. If, for instance, a clinic did not have this level of visibility it would have been less clear when and with whom it should place an informal order at any given point in time. It is therefore a combination of the information shared and the practical ability to share inventory that led to the decrease in unfulfilled demand observed during Experiment 2.

The results obtained during Experiment 3 were exemplary of a case where, although information was shared, it was not sufficiently meaningful to improve overall supply chain performance in respect of unfulfilled demand. Although the warehouse and the hospital had visibility over the aggregate inventory levels of their customers, this information was not sufficiently detailed. It did, however, grant the two suppliers the opportunity to carry smaller amounts of inventory during low end-user demand, when compared with the previous two scenarios. This phenomenon consequently came to the detriment of the end-users because the information made available to the suppliers did not convey the news of a demand transition in a timely fashion.

Instances of unrestrained ordering by the warehouse and the hospital were observed during all five experiments, but it appeared to be more erratic during Scenario 4. This behaviour may again be attributed to an inadequate exploration of the agents' respective state spaces, especially when considering that the size of the warehouse and hospital agents' state spaces quadrupled from Scenario 3. This contributed to the observation that these two agents did not learn how to make best use of their ability to share inventory. The results obtained during Experiment 4 were therefore not as compelling in respect of demonstrating the impact of inventory sharing between hospitals and warehouses.

The results obtained during Experiment 5 revealed that visibility over supply chain events downstream may improve the ability of a manufacturer to manage its own inventory more effectively. It was, for example, observed that the manufacturer could afford to carry comparatively smaller amounts of inventory during Scenario 5 than during scenarios void of supply chain visibility. Similar to the results observed during Experiment 3, the information made available to the manufacturer was not sufficient to reveal substantial changes in end-user demand in a timely fashion. As a result, the manufacturer could not respond effectively to suddenly heightened demand. Additionally, it appeared as if inadequate inventory management observed at the warehouse and the hospital, as opposed to at the manufacturer, contributed largely to unfulfilled end-user demand. In other words, the value of the information shared with the manufacturer was limited because the manufacturer alone could not influence the fulfilment of end-user demand significantly when compared with Scenario 4.

The total number of end-user stock-outs observed during each of the five experiments is shown in Figure 9.19. The ANOVA test revealed that there is a statistical difference in respect of unfulfilled end-user demand between the five experiments at a 5% level of significance (a  $p$ -value of less than  $1 \times 10^{-17}$ ). The Levene test was subsequently employed to compare the variances of the five samples. The test returned a  $p$ -value of 0.8399 indicating that the variances are statistically indistinguishable at a 95% confidence level. The Fisher LSD was therefore finally employed to determine where the differences between the effectiveness of the five information-sharing scenarios occur in respect of the number of end-user stock-outs observed.

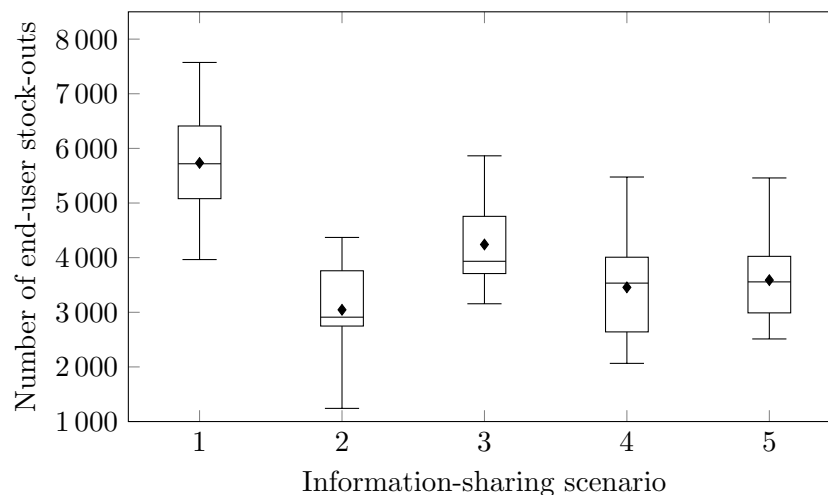


FIGURE 9.19: The number of end-user stock-outs observed for each information-sharing scenario during Experiments 1–5.

The  $p$ -values returned by the Fisher LSD test are presented in Table 9.5. From the results of the Fisher LSD test it follows that there is a statistical difference between the results of every experiment pair apart from the pair involving Scenarios 4 and 5. In terms of the number of end-user stock-outs KPI, Scenario 2 was shown to be the most effective. By implication, it was better to allow informal inventory sharing between clinics, while not affording any form of supply chain visibility to at least one other facility. Nonetheless, the effectiveness of Scenarios 3, 4 and 5 were all statistically superior over the benchmark scenario at a 5% level of significance. It is conjectured that this outcome may be attributed largely to the abilities of clinics to share inventory between themselves, as opposed to the supply chain visibility afforded to suppliers upstream. Contrary to expectations, Experiment 3 revealed that the third information-sharing scenario was, in fact, statistically less effective than Scenario 2 in respect of unfulfilled demand.

	Fisher LSD test $p$ -values: Number of stock-outs				
	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Scenario 1	—	$< 1 \times 10^{-17}$	$8.4884 \times 10^{-12}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Scenario 2	—	—	$1.9010 \times 10^{-8}$	$4.2606 \times 10^{-2}$	$7.6646 \times 10^{-3}$
Scenario 3	—	—	—	$1.4291 \times 10^{-4}$	$1.4472 \times 10^{-3}$
Scenario 4	—	—	—	—	$5.1058 \times 10^{-1}$
Scenario 5	—	—	—	—	—
Mean	5 733.67	3 045.40	4 240.90	3 456.23	3 588.70

TABLE 9.5: Differences in respect of the total number of end-user stock-outs observed during Experiments 1–5. Table entries smaller than 0.05 are typeset in red and indicate a statistical difference between the pair of facilities at a 5% significance level.

This finding was attributed to the inherent limitations involved in the types of information shared with the warehouse and the hospital, as discussed above. Furthermore, the results of Scenarios 4 and 5 were statistically indistinguishable in respect of unfulfilled demand at a 5% level of significance. This implies that the supply chain visibility provided to the manufacturer could not affect a statistical improvement in respect of unfulfilled end-user demand. In practical terms, this finding suggests that the installation of the infrastructure required to share the relevant types of information with the manufacturer is not warranted at a 5% level of significance.

An analysis of the number of order stock-outs observed at the manufacturer, the warehouse and the hospital during each of the five experiments did not reveal valuable insights into the effectiveness of information sharing. Based on the occasional erratic ordering and manufacturing patterns exhibited by these facilities, the variances observed in the number of order stock-outs were large. At times when these facilities carried large amounts of inventory they were able to almost always respond to high demand in a timely fashion. Conversely, during the occasional periods when a supplier did not have any stock on hand, it incurred substantial order stock-outs when these periods coincided with heightened demand from its customers. As a result, the number of order stock-outs varied from relatively low to relatively high with no apparent trend observed over the five experiments.

It is imperative to stress that the results obtained in the information-sharing effectiveness comparison analysis should be interpreted in the context of a concept demonstration. The experimental design was carried out in terms of a hypothetical supply chain network and the results obtained may therefore not be directly transferable to a real-world context. It was also shown that reinforcement learning is a suitable solution approach towards learning largely (although not entirely) effective inventory management policies. It is, however, acknowledged that the performance of the Q-learning algorithm was limited by some of the challenges involved in the learning procedure as described in this thesis. The implementation of reinforcement learning in the context of the five experiments may therefore be refined so as to potentially improve its effectiveness in the future. Nevertheless, the results obtained in this thesis still succeeded in pronouncing conceptually on the potential impact of information sharing in a pharmaceutical supply chain context. It was, for example, shown that informal inventory sharing between clinics is indeed a feasible approach towards mitigating stock-outs in the short-term. The results also emphasised the importance of sharing specific and meaningful data so as to improve overall supply chain performance. Finally, the results provided an indication that self-organising inventory management is a viable approach if the entities involved are provided with sufficient information.



### 9.3 Chapter summary

The results obtained in the information-sharing effectiveness comparison analysis conducted in this thesis were presented in this chapter. The chapter opened with a general introduction explaining the format in which these results would be presented and analysed. This was followed by a systematic discussion on the results obtained during each of the five simulation experiments. As expected, the benchmark scenario (Scenario 1) performed significantly worse in respect of the number of end-user stock-outs KPI, when compared with each of the other four information-sharing arrangements. The results further revealed that Scenario 2 was statistically the most effective information-sharing configuration in respect of minimising unfulfilled demand. In other words, neither of the layers of visibility added during Scenarios 3, 4 and 5 yielded any statistically significant improvement over the effectiveness observed during Scenario 2. The analysis also underlined the importance of sharing disaggregated information as opposed to too-coarsely aggregated information (*i.e.* information should be specific and meaningful). Notably, no product expiries were observed during any experiment as a result of the inventory management policies being learnt on a *first-expire first-out* principle.



# Part III

# Conclusion



---

---

## CHAPTER 10

---

# Conclusion and summary

### Contents

10.1 Thesis summary . . . . .	159
10.2 Appraisal of contributions . . . . .	161

The purpose of this chapter is to provide the reader with a brief overview of the work reported in this thesis. A summary of the thesis contents is provided in §10.1 and this is followed by a critical assessment of the research contributions in §10.2.

### 10.1 Thesis summary

Including the current and following chapters, this thesis comprises a total of eleven chapters. Apart from Chapter 1, these chapters have been partitioned into three parts. The purpose of the only stand-alone chapter, Chapter 1, was to provide the reader with a general introduction to the problem considered in this thesis. That introductory chapter opened with background information pertaining to the problem under consideration in §1.1. This discussion highlighted how a general lack of information sharing across the various tiers of a pharmaceutical supply chain may lead to substantial stock-outs and expiries. This was followed by a formal outline of the problem investigated in this thesis in §1.2. Thereafter, ten research objectives to be pursued in this thesis were formulated and presented in §1.3. A delimitation of the thesis scope was provided next in §1.4, and this was followed by an exposition in §1.5 of the research methodology to be followed in this thesis. The chapter finally closed with a description of the organisation of the contents in this thesis in §1.6.

A review of the academic literature involving the topics identified in Objective I of §1.3 was provided in three further chapters in the first part of this thesis, Part 1, entitled *Literature review*. The topic of supply chain management was researched in fulfilment of Objective I(a) and subsequently discussed in Chapter 2. In §2.1, a general introduction was provided to the notion of supply chain management and its constituent activities. This was followed in §2.2 by a discussion on the predominant strategies typically employed in a commercial supply chain. The third topic of review in that chapter involved the notorious bullwhip effect. Typical causes of the bullwhip effect, as well as methods for countering this phenomenon, were described in §2.3. The focus shifted in §2.4 to the importance of information sharing in supply chains, and this was complemented with a description of demand-driven supply chain management in §2.5. A brief review followed in §2.6 of the factors that typically hinder collaboration between supply chain entities. Inventory management was a central theme in this thesis and a number of well-known

inventory models were described next in §2.7. This was followed in §2.8 by a description of some of the most popular measures employed to measure supply chain performance. The chapter closed in §2.9 with a specific focus on pharmaceutical supply chains. That section included a review of some of the most prominent challenges facing pharmaceutical supply chains not only globally, but specifically also in the South African context.

The second chapter of Part I, Chapter 3, was devoted to a review of some of the most salient characteristics of computer simulation and agent-based modelling, in fulfilment of Objective I(b). A general introduction to the sub-discipline of computer simulation was provided in §3.1, and this was followed by a review of some elementary simulation modelling concepts in §3.2. The four predominant simulation modelling paradigms, namely discrete-event modelling, system dynamics modelling, agent-based modelling and dynamic systems modelling, were briefly discussed in §3.3. A review was provided in §3.4 of a well-known twelve-step procedure for carrying out a sound simulation study. This was followed in §3.5 by a brief review of some important considerations involving simulation input data, before generally accepted techniques for simulation model verification and validation were described in §3.6. Guidelines for developing an agent-based model, with a specific focus on agent-based supply chain models, were finally provided in §3.7.

Part I concluded in Chapter 4 with a brief overview of reinforcement learning, a sub-discipline of machine learning, in fulfilment of Objective I(c). In order to better understand the relative position of reinforcement learning within the realm of machine learning, a general introduction to the latter field was provided in §4.1. The focus shifted in §4.2 to more specific characteristics of reinforcement learning. This included a review of the notion of evaluative feedback, a formulation of the reinforcement learning problem in general, and a description of a selection of reinforcement learning approaches. The working of the reinforcement learning algorithm employed in this thesis, Q-learning, was also described in that final section. The Q-learning algorithm was selected for application in pursuit of Objective III. The literature review of Part I stands in fulfilment of Objective I.

The focus in Part II of this thesis, entitled *Pharmaceutical supply chain modelling*, was on investigating the impact of information-sharing in a pharmaceutical supply chain within a simulation modelling environment. In pursuit of Objective II, five hypothetical information-sharing scenarios were designed and described in the first chapter of Part II, Chapter 5. Certain general considerations involving information sharing were first provided in §5.1, before the architecture of each of the five proposed scenarios was described in §5.2. The relative effectiveness of information sharing in a pharmaceutical supply chain was investigated in respect of these five scenarios in this thesis.

A novel agent-based pharmaceutical supply chain simulation model was designed and developed in fulfilment of Objective V and this model was discussed in Chapter 6. The model framework was reviewed in §6.1, with a particular focus on the model input data, the model output data and how a selection of supply chain processes was modelled. This was followed in §6.2 by a description of the most prominent verification and validation techniques employed in respect of the simulation model, in pursuit of Objective VI.

The inventory management problem considered in this thesis was formulated in Chapter 7 as a reinforcement learning problem, in fulfilment of Objective VII. The state space design for each agent and for each information-sharing scenario was first discussed in §7.1. In §7.2, the nature of each agent's action space was put forward. The derivation of a general form of the reward function employed by each agent in this thesis was discussed next in §7.3. This was followed by an overview of the learning rate and exploration rate parameters found to be suitable for the relevant problem instances in §7.4 and §7.5, respectively.

An experimental design approach towards investigating the five information-sharing scenarios in the context of the agent-based simulation model of Chapter 6 was presented in Chapter 8, in pursuit of Objective VIII. The chapter opened in §8.1 with insight into some of the most prominent considerations involved in the design of the experiments. The architecture of the hypothetical nine-facility pharmaceutical supply chain considered in all experiments was discussed next in §8.2. The exact algorithmic implementation of the Q-learning algorithm in all experiments was described in §8.3. This discussion included the discretisation of each agent's action space and state space, and the final design of the reward function. A brief review of the statistical tests employed to evaluate the results with respect to the KPIs identified in pursuit of Objective IV was finally provided in §8.5.

The final chapter of Part II, Chapter 9, was devoted to a detailed analysis of the results obtained during the experimental design of Chapter 8. The blueprint for the discussion of the results was first presented in §9.1. This was followed by a statistical analysis of the results obtained during each of the five experiments of the experimental design in §9.2, in fulfilment of Objective IX. The results revealed that any information-sharing arrangement allowing inventory sharing between clinics proved statistically superior over the no information-sharing benchmark scenario. It was also shown that added layers of information sharing may not necessarily benefit supply chain effectiveness, specifically when information is not sufficiently granular.

## 10.2 Appraisal of contributions

The objective in this section is to provide a brief summary and appraisal of the main research contributions.

**Contribution 1** *The development of a novel pharmaceutical supply chain simulation modelling framework in AnyLogic.*

An agent-based computerised simulation model of a pharmaceutical supply chain was proposed in Chapter 6. This simulation model is capable of modelling the high-level operation of a pharmaceutical supply chain over time, with a particular focus on the flow of inventory and information shared between entities. An extensive input structure is employed in the model to capture as input a user-specified supply chain network with several user-specified attributes. The simulation model accommodates the modelling of manufacturing facilities, warehouses, hospitals as well as clinics and their respective operations (albeit at a relatively high level of model abstraction). The user may specify the connections between facilities, the corresponding delivery lead time distributions as well as the nature of end-user demand. The simulation model animation makes it possible to visually inspect the flow of inventory during a simulation run. Several performance measures such as inventory levels and the number of stock-outs observed may be reported as model output data.

**Contribution 2** *The successful implementation of reinforcement learning in a simulated environment for solving instances of the inventory management problem.*

Reinforcement learning was embedded in the proposed simulation model and subsequently employed in order to enable agents to learn suitable inventory replenishment policies in the context of a pharmaceutical supply chain modelled. Although reinforcement learning has been applied previously to several instances of the inventory management problem in the literature, none of these implementations are directly comparable with the work reported in this thesis. To the author's best knowledge, the work presented in this thesis is the first successful implementation of reinforcement learning in AnyLogic with a particular view to investigate the impact of in-

formation sharing on inventory management in pharmaceutical supply chains. The nine-facility supply chain considered in this thesis is also relatively large when compared with the networks typically employed in other studies involving reinforcement learning in the context of supply chain management. The implementation of reinforcement learning in this thesis revealed interesting phenomena that may provide valuable insight into solving a reinforcement learning inventory problem in general.

**Contribution 3** *Five hypothetical information-sharing scenarios that may serve as a point of departure for future research.*

Five hypothetical information-sharing scenarios were designed and subsequently investigated in this thesis. The first of these five scenarios did not involve any information sharing and was considered as a benchmark. The scope of information sharing progressively increased further over the remaining four scenarios. Although these information-sharing scenarios are hypothetical, they provide a suitable platform for conducting further research in terms of the relative effectiveness of different information-sharing arrangements.

**Contribution 4** *A demonstration of how inventory may be shared effectively between supply chain peers in order to minimise short-term stock-outs.*

An inventory-sharing scheme is proposed that allows supply chain facilities in close proximity to one another to share inventory between them. The results obtained from the experiments carried out in this thesis showed that inventory sharing between clinics is indeed a viable approach towards mitigating the risk of short-term stock-outs. Although this ‘borrowing’ phenomenon is already observed in practice in the South African context, it appears to be extremely ineffective due to inadequate information sharing and poor regulation of the flow of inventory. The work reported in this thesis presents a formal demonstration of how inventory sharing may be performed effectively and efficiently when the right types of information are made available at the right time.

**Contribution 5** *Evidence that information shared in a supply chain should be meaningful and practicable in order to improve supply chain effectiveness.*

The implementation of reinforcement learning in this thesis revealed that access to more information does not necessarily always translate into improved supply chain performance. It was shown that access to additional information may not always be beneficial, especially when the information is not of the right type or of sufficient granularity.

**Contribution 6** *A successful demonstration of self-organising inventory management in a pharmaceutical supply chain context.*

Each reinforcement learning agent considered in this thesis behaved autonomously and no explicit control was imposed on these agents during any stage. The agents may therefore be said to be self-organising. The policies learnt by the agents as part of the experimental design carried out in this thesis revealed that these agents were able to learn relatively effective inventory management policies independently from one another. The most lucid demonstration of effective self-organising inventory management was found in the exchange of inventory between clinics during the final four information-sharing scenarios considered in this thesis. It was observed that clinics in the same neighbourhood are able to mitigate the risk of incurring stock-outs through self-organising behaviour where decisions were made based only on information provided to them. Although information was shared between agents, no explicit communication or coordination were allowed between them.



---



---

## CHAPTER 11

---

# Suggestions for future work

### Contents

11.1 Suggestions involving the simulation model . . . . .	163
11.2 Scope enlargement of information sharing . . . . .	164
11.3 Solution approach suggestions . . . . .	165

In fulfilment of Objective X of §1.3, a number of suggestions for possible future work involving extensions to, or improvements of, the work documented in this thesis are mentioned in this final chapter. Given the multi-faceted nature of the work reported in this thesis, there are numerous avenues that may be pursued in future work and these avenues are stratified broadly into three classes. Suggestions for future work involving the simulation model of Chapter 6 are provided in §11.1. This is followed in §11.2 by a number of suggestions pertaining to the expansion of the scope of information sharing as investigated in this thesis. Potential techniques for improving the solution approach adopted in this thesis are finally provided in §11.3.

### 11.1 Suggestions involving the simulation model

The proposed simulation model concept demonstrator in its current state may be employed extensively for further experimentation. In this thesis, only one set of end-user demand conditions was considered during the experimental design. It is suggested that further experiments be conducted involving a larger variety of demand conditions. Given that a homogeneous end-user demand pattern was considered in this thesis, it would be more appropriate to investigate the impact of heterogeneous demand (*e.g.* some clinics experience high demand while others experience low demand simultaneously). This may provide improved insight into the impact of information sharing during different end-user demand conditions. Furthermore, experiments involving shorter (or longer) lead times and/or products with shorter (or longer) shelf-lives may be performed so as to establish the relative effectiveness of information sharing when these variables are varied. Despite the computational burden associated with the implementation of the Q-learning algorithm, it would also be appropriate to explore the impact of information sharing on supply chain networks of different sizes or configurations. Finally, it is also suggested that the simulation model's capability to model special-case events (as mentioned in §6.1.1) should be explored in order to investigate the robustness of an information-sharing arrangement.

There are also numerous opportunities available for expanding and refining the simulation model concept demonstrator. Restricted transportation capabilities as well as limited storage capacities

may, for example, be incorporated into the working of the model so as to attain an improved level of realism in the model. In its current state, it is also assumed that each manufacturing agent has a sufficient amount of raw materials available at all times. Since this assumption is fairly idealistic, it would be more suitable to relax this assumption when developing the model further. Another major factor limiting the simulation model in its current state is the modelling of time as discrete (daily) time steps. Supply chain processes, such as the receipt of shipments and the fulfilment of demand, are modelled as instantaneous events occurring in a fixed sequence — a phenomenon that rarely (if at all) occurs in practice. In order to elevate the simulation model towards a higher level of realism, it is suggested that a more finely-grained time unit, such as hours (if not minutes), should be adopted. This may provide a more natural representation of the timing and sequence of most supply chain events.

If the improvements described above are made to the simulation model, it may be possible to calibrate the simulation model with real-world data. Although this is expected to be a relatively difficult endeavour, it may further enhance the potential value of the concept demonstrator considerably. If this were to be carried out successfully, it may even be possible to perform real-world case studies in the context of the simulation model. The agent-based architecture underlying the simulation model also provides an exciting opportunity for modelling patients as agents. This would provide the ability to model the behaviour of individual (or clusters of) patients explicitly. For instance, some patients visit their local health-care clinics more frequently than others and/or on specific days only, thereby influencing demand implicitly. Furthermore, a patient confronted with a stock-out at his or her local health-care facility may proceed to seek medicine elsewhere, again inflating the demand at other clinics. Modelling patients as agents may, in turn, create a further opportunity to model the spread of disease so as to evaluate the impact of this phenomenon on end-user demand at different health-care facilities. Consequently, the relative effectiveness of information sharing in terms of detecting changes in demand that occur as a result of the spread of disease may also be analysed.

## 11.2 Scope enlargement of information sharing

The notion of information sharing within a supply chain has numerous dimensions which provides a large scope for future work. Apart from the no-information sharing scenario, four arbitrarily designed information sharing arrangements were considered in this thesis. It would be appropriate to experiment with alterations made to those four scenarios as well as to add entirely new information-sharing arrangements. Suggestions for extensions in this regard include improved visibility over deliveries (*i.e.* order tracking) and the sharing of facility-level information (as opposed to aggregated information such as the total amount of inventory available in a cluster).

Performing a more extensive analysis in terms of information sharing may, for example, reveal which types of shared information are more critical than others. Suppose, for example, that sharing demand information as opposed to inventory level information is found to be more valuable in the context of a particular objective. In practical terms, this may imply that the technology infrastructure required to capture and share such data only needs to register demand information (as opposed to inventory level information). In other words, fewer resources (in the form of infrastructure, money or manpower) are required to achieve a particular objective. Considering that the practical implementation of accurate information-capturing and sharing systems may be extremely resource-intensive, it provides an added incentive for identifying the most critical types of information that should be shared. It would also be appropriate to investigate the granularity required for information shared to be the most effective.

### 11.3 Solution approach suggestions

Although the implementation of the Q-learning algorithm proved extremely challenging, it may be possible to improve on the performance of the algorithm in the context of the experiments conducted in this thesis. The reward function, the learning rate parameters and the action-selection techniques employed in this thesis may, for instance, be refined in order to enhance learning. The results presented in Chapter 9 have shown that the agents appeared not to visit their lower inventory states sufficiently many times during learning. It would be interesting to investigate methods for improving on this learning performance in the future. It is also suggested that the feasibility (and legitimacy) of the implementation of the no-ordering streaks proposed in §7.5 be explored. Based on the results obtained in this thesis, it is also recommended to adopt a more finely-grained representation of the agents' state spaces so as to better evaluate the impact of information sharing on their learning behaviour. The research conducted in this study did, however, reaffirm a belief that the Q-learning algorithm appears to be inadequate for large problem instances (which have continuous state spaces).

A more suitable reinforcement learning solution approach would be to implement a *general function approximator* that provides a more compact representation of an agent's value function (a value function is represented as a single parameterised function instead of a table). Unlike Q-learning, a general function approximator provides a direct representation of continuous state spaces and also has the ability to generalise across different states and actions. General function approximators have been shown to be far superior over most conventional reinforcement learning solution approaches in terms of learning performance. It is therefore recommended that effort be invested into the implementation of a general function approximator as opposed to perfecting the application of the Q-learning algorithm. It is expected that the use of general function approximators will make it significantly more feasible to study larger supply chain networks.

Finally, it is also suggested that the applicability of solution approaches other than reinforcement learning be investigated when seeking to learn effective policies based on information sharing. There are some indications in the literature that *evolution strategies*, a sub-class of evolutionary algorithms, may provide superior performance over conventional reinforcement learning solution approaches such as Q-learning. It is, however, unknown whether this supposed superiority would manifest itself in the context of the inventory management problem considered in this thesis.



---

## References

- [1] AGRAWAL N & SMITH SA, 2013, *Optimal inventory management for a retail chain with diverse store demands*, European Journal of Operational Research, **225(3)**, pp. 393–403.
- [2] ALPAYDIN E, 2010, *Introduction to machine learning*, 2<sup>nd</sup> Edition, MIT Press, Cambridge (MA).
- [3] ANYLOGIC, 2019, *Multimethod simulation software*, [Online], [Cited March 2019], Available from <https://www.anylogic.com>.
- [4] AVIV Y, 2007, *On the benefits of collaborative forecasting partnerships between retailers and manufacturers*, Management Science, **53(5)**, pp. 777–794.
- [5] AXSÄTER S, 2000, *Inventory control*, Kluwer Academic Publishers, Norwell (MA).
- [6] BANKS J, 1998, *Handbook of simulation: Principles, methodology, advances, applications, and practice*, John Wiley & Sons, New York (NY).
- [7] BANKS J, CARSON J, NELSON B & NICOL D, 2001, *Discrete-event system simulation*, 3<sup>rd</sup> Edition, Prentice-Hall, Upper Saddle River (NJ).
- [8] BARRATT M, 2004, *Understanding the meaning of collaboration in the supply chain*, Supply Chain Management, **9(1)**, pp. 30–42.
- [9] BARRATT M & OKE A, 2007, *Antecedents of supply chain visibility in retail supply chains: A resource-based theory perspective*, Journal of Operations Management, **25(6)**, pp. 1217–1233.
- [10] BARRINGTON J, WEREKO-BROBBY O, WARD P, MWAFONGO W & KUNGULWE S, 2010, *SMS for Life: A pilot project to improve anti-malarial drug supply management in rural Tanzania using standard technology*, Malaria Journal, **9(298)**, pp. 1–9.
- [11] BATEMAN C, 2013, *Drug stock-outs: Inept supply-chain management and corruption*, South African Medical Journal, **103(9)**, pp. 600–602.
- [12] BATEMAN C, 2015, *Inept drug supply management causing stock-outs*, South African Medical Journal, **105(9)**, pp. 706–707.
- [13] BEAMON BM, 1998, *Supply chain design and analysis: Models and methods*, International Journal of Production Economics, **55(3)**, pp. 281–294.
- [14] BEAMON BM, 1999, *Measuring supply chain performance*, International Journal of Operations and Production Management, **19(3)**, pp. 275–292.
- [15] BEKKER LG, VENTER F, COHEN K, GOEMARE E, VAN CUTSEM G, BOULLE A & WOOD R, 2014, *Provision of antiretroviral therapy in South Africa: The nuts and bolts*, Antiviral Therapy, **19(3)**, pp. 105–116.
- [16] BERTSEKAS DP, 2012, *Dynamic programming and optimal control*, 3<sup>rd</sup> Edition, Athena Scientific, Belmont (CA).

- [17] BHAGWAT R & SHARMA MK, 2007, *Performance measurement of supply chain management: A balanced scorecard approach*, Computers and Industrial Engineering, **53(1)**, pp. 43–62.
- [18] BILLER B & GUNES C, 2010, *Introduction to simulation input modeling*, Proceedings of the 2010 Winter Simulation Conference, Baltimore (MD), pp. 49–58.
- [19] BILLER B & NELSON BL, 2002, *Answers to the top ten input modeling questions*, Proceedings of the 2002 Winter Simulation Conference, San Diego (CA), pp. 35–40.
- [20] BONABEAU E, 2002, *Agent-based modeling: Methods and techniques for simulating human systems*, Proceedings of the National Academy of Sciences, **99(3)**, pp. 7280–7287.
- [21] BORSHCHEV A & FILIPPOV A, 2004, *From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools*, Proceedings of the 22<sup>nd</sup> International Conference of the System Dynamics Society, Oxford, pp. 959–966.
- [22] BUDD J, KNIZEK C & TEVELSON B, 2013, *The demand-driven supply chain*, pp. 189–194 in DEIMLER M, LESSER R, RHODES D & SINHA J (EDS), *Own the future: 50 Ways to win from the Boston Consulting Group*, John Wiley & Sons, Hoboken (NJ).
- [23] CACHON GP & FISHER M, 2000, *Supply chain inventory management and the value of shared information*, Management Science, **46(8)**, pp. 1032–1048.
- [24] CAMERON A, EWEN M, ROSS-DEGNAN D, BALL D & LAING R, 2009, *Medicine prices, availability, and affordability in 36 developing and middle-income countries: A secondary analysis*, The Lancet, **373**, pp. 240–249.
- [25] CECERE L, O'MARAH K & PRESLAN L, 2004, *Driven by demand*, Supply Chain Management Review, **8(8)**, pp. 15–16.
- [26] CHAN FTS, 2003, *Performance measurement in a supply chain*, International Journal of Advanced Manufacturing Technology, **21(7)**, pp. 534–548.
- [27] CHAN FTS, QI HJ, CHAN HK, LAU HCW & IP RWL, 2003, *A conceptual model of performance measurement for supply chains*, Management Decision, **41(7)**, pp. 635–642.
- [28] CHASE RB, AQUILANO NJ & JACOBS FR, 1998, *Production and operations management: Manufacturing and services*, 8<sup>th</sup> Edition, Irwin/McGraw-Hill, Boston (MA).
- [29] CHATFIELD DC, HAYYA JC & HARRISON TP, 2007, *A multi-formalism architecture for agent-based, order-centric supply chain simulation*, Simulation Modelling Practice and Theory, **15(2)**, pp. 153–174.
- [30] CHEN F, DREZNER Z, RYAN JK & SIMCHI-LEVI D, 2000, *Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information*, Management Science, **46(3)**, pp. 436–443.
- [31] CHOI TY, DOOLEY KJ & RUNGTUSANATHAM M, 2001, *Supply networks and complex adaptive systems: Control versus emergence*, Journal of Operations Management, **19(3)**, pp. 351–366.
- [32] CHOPRA S & MEINDL P, 2013, *Supply chain management: Strategy, planning and operation*, 5<sup>th</sup> Edition, Pearson Education Limited, Essex.
- [33] CHRISTOPHER M, 2011, *Logistics and supply chain management*, 4<sup>th</sup> Edition, Financial Times Prentice Hall, Harlow.
- [34] COYLE JJ, LANGLEY CJ, NOVACK RA & GIBSON BJ, 2013, *Managing supply chains: A logistics approach*, 9<sup>th</sup> Edition, South-Western Cengage Learning, Boston (MA).

- [35] COYLE R, 1996, *System dynamics modelling: A practical approach*, Chapman and Hall, London.
- [36] CROSON R & DONOHUE K, 2003, *Impact of POS data sharing on supply chain management: An experimental study*, *Production and Operations Management*, **12(1)**, pp. 1–11.
- [37] DAFF BM, SECK C, BELKHAYAT H & SUTTON P, 2014, *Informed push distribution of contraceptives in Senegal reduces stockouts and improves quality of family planning services*, *Global Health: Science and Practice*, **2(2)**, pp. 245–252.
- [38] DAVIS T, 1993, *Effective supply chain management*, *Sloan Management Review*, **34(4)**, pp. 35–46.
- [39] DE TREVILLE S, SHAPIRO RD & HAMERI AP, 2004, *From supply chain to demand chain: The role of lead time reduction in improving demand chain performance*, *Journal of Operations Management*, **21(6)**, pp. 613–627.
- [40] DE WOLF T & HOLVOET T, 2005, *Emergence versus self-organisation: Different concepts but promising when combined*, pp. 1–15 in BRUECKNER SA, SERUGENDO GDM, KARAGEORGOS A & NAGPAL R (EDS), *Engineering self-organising systems: Methodologies and applications*, Springer, Berlin.
- [41] DISNEY SM & TOWILL DR, 2003, *Vendor-managed inventory and bullwhip reduction in a two-level supply chain*, *International Journal of Operations and Production Management*, **23(6)**, pp. 625–651.
- [42] DOCTORS WITHOUT BORDERS, 2015, *Empty shelves: Come back tomorrow*, (Unpublished) Technical Report, Doctors Without Borders, Cape Town.
- [43] DOCTORS WITHOUT BORDERS, RURAL DOCTORS ASSOCIATION OF SOUTHERN AFRICA, RURAL HEALTH ADVOCACY PROJECT, SECTION27, SOUTHERN AFRICA HIV CLINICIANS SOCIETY & TREATMENT ACTION CAMPAIGN, 2013, *The chronic crisis: Essential drug stock-outs risk unnecessary death and drug resistance in South Africa*, (Unpublished) Technical Report, Doctors Without Borders, Cape Town.
- [44] DOCTORS WITHOUT BORDERS, RURAL DOCTORS ASSOCIATION OF SOUTHERN AFRICA, RURAL HEALTH ADVOCACY PROJECT, SECTION27, SOUTHERN AFRICA HIV CLINICIANS SOCIETY & TREATMENT ACTION CAMPAIGN, 2016, *2015 Stock outs National Survey: Third annual report – South Africa*, (Unpublished) Technical Report, Doctors Without Borders, Cape Town.
- [45] DU PLESSIS M, VAN VUUREN JH & VAN EEDEN J, 2018, *A concept demonstrator for self-organising demand-driven inventory management in pharmaceutical supply chains*, *Proceedings of the 2018 South African Institute for Industrial Engineering Conference*, Spier, pp. 435–444.
- [46] DUARTE CANEVER M, VAN TRIJP HCM & BEERS G, 2008, *The emergent demand chain management: Key features and illustration from the beef business*, *Supply Chain Management*, **13(2)**, pp. 104–115.
- [47] EAGLE S, 2017, *Demand-driven supply chain management: Transformational performance improvement*, Kogan Page Limited, London.
- [48] ERLINKOTTER D, 1990, *Ford Whitman Harris and the economic order quantity model*, *Operations Research*, **38(6)**, pp. 937–946.
- [49] FISHER M, 1997, *What is the right supply chain for your product?*, *Harvard Business Review*, **75(2)**, pp. 105–116.

- [50] FOX MS, BARBUCEANU M & TEIGEN R, 2000, *Agent-oriented supply chain management*, International Journal of Flexible Manufacturing Systems, **12(2–3)**, pp. 165–188.
- [51] FRANCIS V, 2008, *Supply chain visibility: Lost in translation?*, Supply Chain Management, **13(3)**, pp. 180–184.
- [52] FRANKLIN C, 2012, *Multi-objective optimisation using agent-based modelling*, Master's Thesis, Stellenbosch University, Stellenbosch.
- [53] FROHLICH MT & WESTBROOK R, 2002, *Demand chain management in manufacturing and services: Web-based integration, drivers and performance*, Journal of Operations Management, **20(6)**, pp. 729–745.
- [54] 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.
- [55] GASKETT C, WETTERGREEN D & ZELINSKY A, 1999, *Q-learning in continuous state and action spaces*, Proceedings of the Australasian Joint Conference on Artificial Intelligence, Berlin, pp. 417–428.
- [56] GAVIRNENI S, KAPUSCINSKI R & TAYUR S, 1999, *Value of information in capacitated supply chains*, Management Science, **45(1)**, pp. 16–24.
- [57] GITHINJI S, KIGEN S, MEMUSI D, NYANDIGISI A, MBITHI AM, WAMARI A, MUTURI AN, JAGOE G, BARRINGTON J & SNOW RW, 2013, *Reducing stock-outs of life saving malaria commodities using mobile phone text-messaging: SMS for life study in Kenya*, PloS One, **8(1)**, pp. 1–8.
- [58] GOVENDER J, SKITI V & MORAR G, 2018, *Evaluation of Stock Visibility Solution implementation in Umzinyathi and Amajuba districts in Kwa-Zulu Natal, 2017*, (Unpublished) Technical Report, National Department of Health, Pretoria.
- [59] GRIFFIN PM, NEMBHARD HB, DEFLITCH CJ, BASTIAN ND, KANG H & MUÑOZ DA, 2016, *Healthcare systems engineering*, John Wiley & Sons, Hoboken (NJ).
- [60] GSM ASSOCIATION, 2018, *Mezzanine's Stock Visibility Solution: A mobile solution driving increased access to medicines*, (Unpublished) Technical Report, GSM Association, London.
- [61] GUNASEKARAN A, PATEL C & MCGAUGHEY RE, 2004, *A framework for supply chain performance measurement*, International Journal of Production Economics, **87(3)**, pp. 333–347.
- [62] HANDFIELD RB & NICHOLS EL, 1999, *Introduction to supply chain management*, Prentice-Hall, Upper Saddle River (NJ).
- [63] HARRIES AD, SCHOUTEN EJ, MAKOMBE SD, LIBAMBA E, NEUFVILLE HN, SOME E, KADEWERE G & LUNGU D, 2007, *Ensuring uninterrupted supplies of antiretroviral drugs in resource-poor settings: An example from Malawi*, Bulletin of the World Health Organization, **85(2)**, pp. 152–155.
- [64] HARTNETT, K, 2018, *The simple algorithm that ants use to build bridges*, [Online], [Cited June 2018], Available from <https://www.quantamagazine.org/the-simple-algorithm-that-ants-use-to-build-bridges-20180226/>.
- [65] 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.
- [66] HEIKKILÄ J, 2002, *From supply to demand chain management: Efficiency and customer satisfaction*, Journal of Operations Management, **20(6)**, pp. 747–767.



- [67] HEYLIGHEN F, 2001, *The science of self-organization and adaptivity*, The Encyclopedia of Life Support Systems, **5(3)**, pp. 253–280.
- [68] HEYLIGHEN F, 2008, *Complexity and self-organization*, Encyclopedia of Library and Information Sciences, **3**, pp. 1215–1224.
- [69] HILLIER FS & LIEBERMAN GJ, 2010, *Introduction to operations research*, 9<sup>th</sup> Edition, McGraw-Hill, New York (NY).
- [70] HODES R, PRICE I, BUNGANE N, TOSKA E & CLUVER L, 2017, *How front-line healthcare workers respond to stock-outs of essential medicines in the Eastern Cape Province of South Africa*, South African Medical Journal, **107(9)**, pp. 738–740.
- [71] HOLWEG M, DISNEY S, HOLMSTRÖM J & SMÅROS J, 2005, *Supply chain collaboration: Making sense of the strategy continuum*, European Management Journal, **23(2)**, pp. 170–181.
- [72] HORVATH L, 2001, *Collaboration: The key to value creation in supply chain management*, Supply Chain Management, **6(5)**, pp. 205–207.
- [73] 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.
- [74] 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.
- [75] HUMPHREYS G, 2011, *Vaccination: Rattling the supply chain*, Bulletin of the World Health Organization, **89(5)**, pp. 324–325.
- [76] INGALLS RG, 2008, *Introduction to simulation*, Proceedings of the 2008 Winter Simulation Conference, Miami (FL), pp. 17–26.
- [77] JACOBS L, 2016, *Vodacom shines as the National Department of Health implements Stock Visibility Solution*, [Online], [Cited March 2018], Available from <http://www.mezzanineware.com/vodacom-shines-as-the-national-department-of-health-implements-stock-visibility-solution/>.
- [78] JULKA N, SRINIVASAN R & KARIMI I, 2002, *Agent-based supply chain management — 1: Framework*, Computers and Chemical Engineering, **26(12)**, pp. 1755–1769.
- [79] KAELBLING LP, LITTMAN ML & MOORE AW, 1996, *Reinforcement learning: A survey*, Artificial Intelligence Research, **4**, pp. 237–285.
- [80] KAIPIA R & HARTIALA H, 2006, *Information-sharing in supply chains: Five proposals on how to proceed*, International Journal of Logistics Management, **17(3)**, pp. 377–393.
- [81] KANGWANA BB, NJOGU J, WASUNNA B, KEDENGE SV, MEMUSI DN, GOODMAN CA, ZUROVAC D & SNOW RW, 2009, *Malaria drug shortages in Kenya: A major failure to provide access to effective treatment*, American Journal of Tropical Medicine and Hygiene, **80(5)**, pp. 737–738.
- [82] KENDALL KE & KENDALL JE, 2014, *Systems analysis and design*, 9<sup>th</sup> Edition, Pearson Education Limited, Essex.
- [83] KIDD M, 2019, Statistician in the centre for statistic consultation at Stellenbosch University, [Personal Communication], Contactable at [mkidd@sun.ac.za](mailto:mkidd@sun.ac.za).
- [84] KLEIJNEN JPC, 1995, *Verification and validation of simulation models*, European Journal of Operational Research, **82(1)**, pp. 145–162.

- [85] KULP SC, LEE HL & OFEK E, 2004, *Manufacturer benefits from information integration with retail customers*, *Management Science*, **50(4)**, pp. 431–444.
- [86] LAMBERT DM, 2004, *The eight essential supply chain management processes*, *Supply Chain Management Review*, **8(6)**, pp. 18–26.
- [87] LAMBERT DM, COOPER MC & PAGH JD, 1998, *Supply chain management: Implementation issues and research opportunities*, *International Journal of Logistics Management*, **9(2)**, pp. 1–19.
- [88] LAMBERT DM, STOCK JR & ELLRAM LM, 1998, *Fundamentals of logistics management*, McGraw-Hill/Irwin, New York (NY).
- [89] LANGABEER JR & ROSE J, 2002, *Creating demand driven supply chains: How to profit from demand chain management*, Spiro Press, London.
- [90] LAW AM, 2003, *How to conduct a successful simulation study*, *Proceedings of the 2003 Winter Simulation Conference*, New Orleans (LA), pp. 66–70.
- [91] LAW AM, 2008, *How to build valid and credible simulation models*, *Proceedings of the 2008 Winter Simulation Conference*, Miami (FL), pp. 39–47.
- [92] LAW AM, 2011, *How to select simulation input probability distributions*, *Proceedings of the 2011 Winter Simulation Conference*, Phoenix (AZ), pp. 1394–1407.
- [93] LAW AM & KELTON WD, 2000, *Simulation modeling and analysis*, 3<sup>rd</sup> Edition, McGraw-Hill, Boston (MA).
- [94] LEE HL, 2002, *Aligning supply chain strategies with product uncertainties*, *California Management Review*, **44(3)**, pp. 105–119.
- [95] LEE HL, PADMANABHAN V & WHANG S, 1997, *The bullwhip effect in supply chains*, *Sloan Management Review*, **38(3)**, pp. 93–102.
- [96] LEE HL, PADMANABHAN V & WHANG S, 1997, *Information distortion in a supply chain: The bullwhip effect*, *Management Science*, **43(4)**, pp. 546–558.
- [97] LEE HL, SO KC & TANG CS, 2000, *The value of information sharing in a two-level supply chain*, *Management Science*, **46(5)**, pp. 626–643.
- [98] LEE HL & WHANG S, 2000, *Information sharing in a supply chain*, *International Journal of Manufacturing Technology and Management*, **1(1)**, pp. 79–93.
- [99] LEUNG NZ, CHEN A, YADAV P & GALLIEN J, 2016, *The impact of inventory management on stock-outs of essential drugs in Sub-Saharan Africa: Secondary analysis of a field experiment in Zambia*, *PloS One*, **11(5)**, pp. 1–18.
- [100] LILLIEFORS HW, 1967, *On the Kolmogorov-Smirnov test for normality with mean and variance unknown*, *Journal of the American Statistical Association*, **62(318)**, pp. 399–402.
- [101] LITTMAN ML, 2001, *Value-function reinforcement learning in Markov games*, *Cognitive Systems Research*, **2(1)**, pp. 55–66.
- [102] LUMSDEN K & MIRZABEIKI V, 2008, *Determining the value of information for different partners in the supply chain*, *International Journal of Physical Distribution and Logistics Management*, **38(9)**, pp. 659–673.
- [103] MACAL CM, 2016, *Everything you need to know about agent-based modelling and simulation*, *Journal of Simulation*, **10(2)**, pp. 144–156.

- [104] MACAL CM & NORTH MJ, 2014, *Introductory tutorial: Agent-based modeling and simulation*, Proceedings of the 2014 Winter Simulation Conference, Savannah (GA), pp. 6–20.
- [105] MACAL CM & NORTH MJ, 2005, *Tutorial on agent-based modeling and simulation*, Proceedings of the 2005 Winter Simulation Conference, Orlando (FL), pp. 2–15.
- [106] MACAL CM & NORTH MJ, 2010, *Tutorial on agent-based modelling and simulation*, Journal of Simulation, **4(3)**, pp. 151–162.
- [107] MANOHAR, V, 2018, *Unity is strength*, [Online], [Cited June 2018], Available from <https://www.shutterstock.com/image-photo/unity-strength1011406435?src=BbM2Y7v-2FfNGqgiTyr5TKw-1-74>.
- [108] MARIA A, 1997, *Introduction to modeling and simulation*, Proceedings of the 1997 Winter Simulation Conference, Atlanta (GA), pp. 7–13.
- [109] MARSLAND S, 2009, *Machine learning: An algorithmic perspective*, CRC Press, Boca Raton (FL).
- [110] MASON-JONES R & TOWILL DR, 1997, *Information enrichment: Designing the supply chain for competitive advantage*, Supply Chain Management, **2(4)**, pp. 137–148.
- [111] MCHUGH ML, 2013, *The chi-square test of independence*, Biochemia Medica, **23(2)**, pp. 143–149.
- [112] MENTZER JT, DEWITT W, KEEBLER JS, MIN S, NIX NW, SMITH CD & ZACHARIA ZG, 2001, *Defining supply chain management*, Journal of Business Logistics, **22(2)**, pp. 1–25.
- [113] MITCHELL TM, 1997, *Machine learning*, McGraw-Hill, New York (NY).
- [114] MOKHESENG M, HORN GS & KLOPPER AG, 2017, *Supply chain solutions to improve the distribution of antiretroviral drugs (ARVs) to clinics in rural areas: A case study of the QwaQwa district*, Health SA Gesondheid, **22(1)**, pp. 93–104.
- [115] MONCZKA RM, HANDFIELD RB, GIUNIPERO LC & PATTERSON JL, 2009, *Purchasing and supply chain management*, 4<sup>th</sup> Edition, Cengage Learning, Mason (OH).
- [116] MONTGOMERY DC & RUNGER GC, 2014, *Applied statistics and probability for engineers*, 6<sup>th</sup> Edition, John Wiley & Sons, Hoboken (NJ).
- [117] NGCOBO NJ & KAMUPIRA MG, 2017, *The status of vaccine availability and associated factors in Tshwane government clinics*, South African Medical Journal, **107(6)**, pp. 535–538.
- [118] NICHOLSON A, ENGLISH RA, GUENTHER RS & CLAIBORNE AB, 2013, *Developing and strengthening the global supply chain for second-line drugs for multidrug-resistant tuberculosis: Workshop summary*, The National Academic Press, Washington (DC).
- [119] ODELL J, 2002, *Agents and complex systems*, Journal of Object Technology, **1(2)**, pp. 35–45.
- [120] PIENAAR WJ & VOGT JJ, 2009, *Business logistics management: A supply chain perspective*, 3<sup>rd</sup> Edition, Oxford University Press Southern Africa, Cape Town.
- [121] PRAJOGO D & OLHAGER J, 2012, *Supply chain integration and performance: The effects of long-term relationships, information technology and sharing, and logistics integration*, International Journal of Production Economics, **135(1)**, pp. 514–522.
- [122] PRIVETT N & GONSALVEZ D, 2014, *The top ten global health supply chain issues: Perspectives from the field*, Operations Research for Health Care, **3(4)**, pp. 226–230.

- [123] RIPIN DJ, JAMIESON D, MEYERS A, WARTY U, DAIN M & KHAMSI C, 2014, *Antiretroviral procurement and supply chain management*, *Antiviral Therapy*, **19(3)**, pp. 79–89.
- [124] ROBERTS SD & PEGDEN D, 2017, *The history of simulation modeling*, Proceedings of the 2017 Winter Simulation Conference, Las Vegas (NV), pp. 308–323.
- [125] ROBINSON S, 2004, *Simulation: The practice of model development and use*, John Wiley & Sons, Chichester.
- [126] RUSSEL SJ & NORVIG P, 2003, *Artificial intelligence: A modern approach*, 2<sup>nd</sup> Edition, Pearson Education, Upper Saddle River (NJ).
- [127] SAEDI S, KUNDAKCIOGLU OE & HENRY AC, 2016, *Mitigating the impact of drug shortages for a healthcare facility: An inventory management approach*, *European Journal of Operational Research*, **251(1)**, pp. 107–123.
- [128] SAHIN F & ROBINSON EP, 2002, *Flow coordination and information sharing in supply chains: Review, implications, and directions for future research*, *Decision Sciences*, **33(4)**, pp. 505–536.
- [129] SARGENT RG, 2005, *Verification and validation of simulation models*, Proceedings of the 2005 Winter Simulation Conference, Orlando (FL), pp. 130–143.
- [130] SCHRIBER T, BRUNNER D & SMITH J, 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.
- [131] SCHULTZ BB, 1985, *Levene's test for relative variation*, *Systematic Zoology*, **34(4)**, pp. 449–456.
- [132] SELEN W & SOLIMAN F, 2002, *Operations in today's demand chain management framework*, *Journal of Operations Management*, **20(6)**, pp. 667–673.
- [133] SERUGENDO GDM, GLEIZES MP & KARAGEORGOS A, 2005, *Self-organization in multi-agent systems*, *The Knowledge Engineering Review*, **20(2)**, pp. 165–189.
- [134] SERUGENDO GDM, GLEIZES MP & KARAGEORGOS A, 2006, *Self-organisation and emergence in multi-agent systems: An overview*, *Informatica*, **30(1)**, pp. 45–54.
- [135] SHALIZI CR, 2001, *Causal architecture, complexity and self-organization in time series and cellular automata*, PhD thesis, University of Wisconsin, Madison (WI).
- [136] SHANNON RE, 1998, *Introduction to the art and science of simulation*, Proceedings of the 1998 Winter Simulation Conference, Washington (DC), pp. 7–14.
- [137] SHI Z, 2011, *Advanced artificial intelligence*, World Scientific, Singapore.
- [138] SIGAUD O & BUFFET O, 2010, *Markov decision processes in artificial intelligence*, International Society for Technology in Education, London.
- [139] SIMATUPANG TM & SRIDHARAN R, 2002, *The collaborative supply chain*, *International Journal of Logistics Management*, **13(1)**, pp. 15–30.
- [140] SIMCHI-LEVI D, KAMINSKY P & SIMCHI-LEVI E, 2004, *Managing the supply chain: The definitive guide for the business professional*, McGraw-Hill, New York (NY).
- [141] SIMCHI-LEVI D, KAMINSKY P, SIMCHI-LEVI E & SHANKAR R, 2000, *Designing and managing the supply chain: Concepts, strategies and case studies*, McGraw-Hill, New York (NY).
- [142] SKJOTT-LARSEN T, SCHARY PB, KOTZAB H & MIKKOLA JH, 2001, *Managing the global supply chain*, 2<sup>nd</sup> Edition, Copenhagen Business School Press, Aarhus.

- [143] SNOW J, 2017, *The supply chain manager's handbook: A practical guide to the management of health commodities*, John Snow, Inc, Arlington (VA).
- [144] STERMAN J, 2000, *Business dynamics: Systems thinking and modeling for a complex world*, Irwin/McGraw Hill, Boston (MA).
- [145] STEVENS GC, 1989, *Integrating the supply chain*, International Journal of Physical Distribution and Materials Management, **19(8)**, pp. 3–8.
- [146] SURANA A, KUMARA S, GREAVES M & RAGHAVAN UN, 2005, *Supply-chain networks: A complex adaptive systems perspective*, International Journal of Production Research, **43(20)**, pp. 4235–4265.
- [147] SUTTON RS & BARTO AG, 1998, *Reinforcement learning: An introduction*, MIT Press, Cambridge (MA).
- [148] SWAMINATHAN JM, SMITH SF & SADEH NM, 1998, *Modeling supply chain dynamics: A multiagent approach*, Decision Sciences, **29(3)**, pp. 607–632.
- [149] THRUN S & LITTMAN ML, 2000, *A review of reinforcement learning*, AI Magazine, **21(1)**, pp. 103–103.
- [150] TOKIC M & PALM G, 2011, *Value-difference based exploration: Adaptive control between epsilon-greedy and softmax*, Proceedings of the 34<sup>th</sup> Annual German Conference on AI, Berlin, pp. 335–346.
- [151] UTHAYAKUMAR R & PRIYAN S, 2013, *Pharmaceutical supply chain and inventory management strategies: Optimization for a pharmaceutical company and a hospital*, Operations Research for Health Care, **2(3)**, pp. 52–64.
- [152] VAN DER ZEE DJ & VAN DER VORST JG, 2005, *A modeling framework for supply chain simulation: Opportunities for improved decision making*, Decision Sciences, **36(1)**, pp. 65–95.
- [153] VOLLMANN TE, CORDON C & HEIKKILA J, 2000, *Teaching supply chain management to business executives*, Production and Operations Management, **9(1)**, pp. 81–90.
- [154] WALLER M, JOHNSON ME & DAVIS T, 1999, *Vendor-managed inventory in the retail supply chain*, Journal of Business Logistics, **20(1)**, pp. 183–204.
- [155] WANG ETG & WEI HL, 2007, *Interorganizational governance value creation: Coordinating for information visibility and flexibility in supply chains*, Decision Sciences, **38(4)**, pp. 647–674.
- [156] WATKINS CJCH & DAYAN P, 1992, *Q-learning*, Machine Learning, **8(3–4)**, pp. 279–292.
- [157] WATKINS CJCH, 1989, *Learning from delayed rewards*, PhD thesis, University of Cambridge, Cambridge.
- [158] WEIMER CW, MILLER JO & HILL RR, 2016, *Agent-based modeling: An introduction and primer*, Proceedings of the 2016 Winter Simulation Conference, Arlington (VA), pp. 65–79.
- [159] WHITE KP & INGALLS RG, 2016, *The basics of simulation*, Proceedings of the 2016 Winter Simulation Conference, Arlington (VA), pp. 38–52.
- [160] WILLIAMS BD, ROH J, TOKAR T & SWINK M, 2013, *Leveraging supply chain visibility for responsiveness: The moderating role of internal integration*, Journal of Operations Management, **31(7-8)**, pp. 543–554.
- [161] WILLIAMS LJ & ABDI H, 2010, *Fisher's least significant difference (LSD) test*, Encyclopedia of Research Design, **218**, pp. 840–853.

- [162] WINSTON WL, 2004, *Operations research: Applications and algorithms*, 4<sup>th</sup> Edition, Brooks/Cole, Belmont (CA).
- [163] WOOLDRIDGE M & JENNINGS NR, 1995, *Intelligent agents: Theory and practice*, The Knowledge Engineering Review, **10(2)**, pp. 115–152.
- [164] WORLD HEALTH ORGANISATION, 2014, *Global update on the health sector response to HIV, 2014*, (Unpublished) Technical Report, World Health Organisation, Geneva.
- [165] WORLD HEALTH ORGANISATION, 2015, *Global health sector response to HIV, 2000–2015: Focus on innovations in Africa: Progress report*, (Unpublished) Technical Report, World Health Organisation, Geneva.
- [166] YADAV P, 2015, *Health product supply chains in developing countries: Diagnosis of the root causes of underperformance and an agenda for reform*, Health Systems and Reform, **1(2)**, pp. 142–154.
- [167] YADAV P, LEGA TATA H & BABABLEY M, 2011, *Storage and supply chain management*, (Unpublished) Technical Report, World Health Organisation, Geneva.
- [168] YOO T, CHO H & YÜCESAN E, 2010, *Hybrid algorithm for discrete event simulation based supply chain optimization*, Expert Systems with Applications, **37(3)**, pp. 2354–2361.
- [169] YU MM, TING SC & CHEN MC, 2010, *Evaluating the cross-efficiency of information sharing in supply chains*, Expert Systems with Applications, **37(4)**, pp. 2891–2897.
- [170] ZHAO X, XIE J & ZHANG WJ, 2002, *The impact of information sharing and ordering co-ordination on supply chain performance*, Supply Chain Management, **7(1)**, pp. 24–40.
- [171] ZIUKOV S, 2015, *A literature review on models of inventory management under uncertainty*, Business Systems and Economics, **5(1)**, pp. 26–35.