Design of a Weapon Assignment Subsystem within a Ground-Based Air Defence Environment

Daniel Petrus Lötter

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

Promoter: Prof JH van Vuuren

December 2017

Declaration

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

Date: December 2017

Abstract

A number of assets on the ground typically require protection from aerial threats in a military ground-based air defence environment. The problem of defending these assets is twofold: Incoming aircraft first have to be identified and classified as friendly or hostile, and the level of threat posed to defended assets by each hostile aircraft has to be assessed, after which available ground-based weapon systems secondly have to be assigned to engage aerial threats with a view to scare them away or to neutralise them. The latter problem is known in the military operations research literature as the *weapon assignment problem*. A fire control officer is responsible for solving both these sub-problems in real time, usually under very stressful conditions. The officer therefore typically employs a computerised threat evaluation and weapon assignment decision support system to aid him in this task.

An architecture is put forward in this dissertation for the weapon assignment part of such a decision support system. The proposed architecture contains two subsystems, namely an engagement quantisation subsystem and a weapon assignment subsystem. The purpose of the engagement quantisation subsystem is to quantify single shot hit probabilities achievable by weapon systems in conjunction with other information within the format required by the weapon assignment subsystem. The working of the various components of the engagement quantisation subsystem is illustrated by means of a series of small numerical examples.

The weapon assignment subsystem forms the heart of the proposed architecture and a weapon assignment model classification is proposed for use in this subsystem. This classification consists of four classes of weapon assignment models ranging in different levels of complexity. The classes are single-objective static weapon assignment models, multi-objective static weapon assignment models, single-objective dynamic weapon assignment models and multi-objective dynamic weapon assignment models.

A model prototype is proposed for default inclusion in each of the aforementioned weapon assignment model classes. The working of each of these models is illustrated by solving it in the context of a hypothetical, but realistic, ground-based air defence environment. A conventional genetic algorithm is used to solve the single-objective static weapon assignment model prototype, while an extension of this algorithm, a nondominated sorting genetic algorithm (specifically designed for solving multi-objective optimisation problems) is used to solve the multi-objective static weapon assignment model prototype. The method of simulated annealing is used to solve the single-objective dynamic weapon assignment model prototype, while a variant of the aforementioned nondominated sorting genetic algorithm is used to solve the multi-objective dynamic weapon assignment model prototype.

The results returned by the algorithms are discussed and validated by means of three methods, including a subjective face validation, a random benchmark validation and a validation consultation with two independent military experts. It is found that the results are plausible in terms of realism and practical executability. The models also outperform solutions put forward by the military experts when asked to solve the models by hand in the context of the same ground-based air defence scenario.

Uittreksel

’n Aantal bates op die grond word tipies in ’n militêre grond-gebaseerde lugafweer-omgewing teen lugbedreigings beskerm. Hierdie bateverdedigingsprobleem is tweeledig: Inkomende vliegtuie moet eerstens geïdentifiseer en as vriendelik of vyandig geklassifiseer word, en die mate van bedreiging wat elke vyandige vliegtuig vir die grondbates inhou, moet afgeskat word, waarna grond-gebaseerde wapenstelsels tweedens aan vliegtuie wat as vyandig geklassifiseer is, toegewys moet word met die oog om hulle te verjaag of te neutraliseer. Die laasgenoemde probleem staan in die militêre operasionele navorsingsliteratuur as die *wapentoewysingsprobleem* bekend. ’n Afvuurbeheer-offisier is verantwoordelik vir die intydse oplossing van die bogenoemde twee deelprobleme, gewoonlik onder baie stresvolle omstandighede. Die offisier maak dus tipies van ’n gerekenariseerde besluitsteunstelsel vir bedreigingsafskatting en wapentoewysing vir hierdie doel gebruik.

’n Argitektuur word vir die wapentoewysingsdeel van só ’n besluitsteunstelsel in hierdie proefskrif daargestel. Die voorgestelde argitektuur bevat twee hoofkomponente, naamlik ’n deelstelsel vir toewysingskwantifisering en ’n deelstelsel vir wapentoewysing. Die doel van die eersgenoemde deelstelsel is om enkelskoot-trefwaarskynlikhede wat deur wapenstelsels behaal kan word, saam met ander inligting oor vyandelike vegvliegtuie in ’n formaat te kwantifiseer soos deur die laasgenoemde deelstelsel vereis. Die werking van die onderskeie komponente van die deelstelsel vir toewysingskwantifisering word aan die hand van ’n reeks klein, hipotetiese numeriese voorbeelde geïllustreer.

Die wapentoewysingsdeelstelsel vorm die hart van die voorgestelde argitektuur en ’n wapentoewysingsmodelklassifikasie word vir gebruik in hierdie deelstelsel daargestel. Hierdie klassifikasie bestaan uit vier klasse wapentoewysingsmodelle van verskillende vlakke van kompleksiteit. Die klasse is enkeldoelige statiese wapentoewysingsmodelle, meerdoelige statiese wapentoewysingsmodelle, enkeldoelige dinamiese wapentoewysingsmodelle en meerdoelige dinamiese wapentoewysingsmodelle.

’n Modelprototipe word vir versuiminsluiting in elkeen van die bogenoemde wapentoewysingsmodelklasse voorgestel. Die werking van elkeen van hierdie modelle word geïllustreer deur hul in die konteks van ’n hipotetiese, maar realistiese, grond-gebaseerde lugafweerscenario op te los. ’n Konvensionele genetiese algoritme word gebruik om die enkeldoelige statiese wapentoewysingsmodelprototipe op te los, terwyl ’n veralgemening van hierdie algoritme, naamlik ’n nie-gedomineerde sorterings-genetiese algoritme (spesifiek ontwerp vir die oplos van meerdoelige optimeringsprobleme) gebruik word om die meerdoelige statiese wapentoewysingsmodelprototipe op te los. Die metode van gesimuleerde tempering word gebruik om die enkeldoelige dinamiese wapentoewysingsmodelprototipe op te los, terwyl ’n variasie op die bogenoemde nie-gedomineerde sorterings-genetiese algoritme gebruik word om die meerdoelige dinamiese wapentoewysingsmodelprototipe op te los.

Die afvoer van hierdie algoritmes word bespreek en deur middel van drie tegnieke gevalideer, insluitend subjektiewe sigvalidering, lukrake-maatstaf validering en validering deur twee onafhanklike militêre kenners. Daar word bevind dat die algoritmiese resultate geloofwaardig is in terme van realisme en praktiese uitvoerbaarheid. Die modelle vaar ook beter as oplossings wat deur die kenners vir dieselfde grond-gebaseerde lugafweerscenario voorgestel word.

Acknowledgements

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

- I wish to thank my colleague and promoter, Prof Jan van Vuuren, for his support and guidance throughout this project. I truly admire his passion for the subject field — his enthusiasm is a true inspiration. I would like to thank him for his patience and hard work towards the end of work on this dissertation. His professionalism in the working environment and dedication to ensure that work of a high standard is delivered are admired. Thank you also for providing me with the time to finish this project during the past few months.

- I wish to thank my colleagues at the Department of Industrial Engineering at Stellenbosch University for their support and assistance during the completion of this project.

- I wish to thank my friends and colleagues in the SUnORE research group who provided me with an academic home for the past three years. I appreciate the great experiences we shared. Their assistance and support is much appreciated.

- I wish to thank my family for their loving support and in, particular, my parents for their considerable moral and financial support during the past years, providing me with the opportunity to finish this project. Furthermore, I wish to thank my sister and my brother-in-law for their love and support. I wish to extend my deepest gratitude to Wayne Lötter for his support and guidance throughout the past years; it is much appreciated.

- I wish to thank my friends for their love, support and interest, especially during the past few months.

- Finally, I wish to thank the Armaments Corporation of South Africa for funding the research reported in this dissertation.

Table of Contents

Apology

It is customary in the military training literature and in the military operations research literature to use acronyms abundantly. These literatures are, in fact, littered with acronyms to the extent that military practitioners may often be unable to recall the original, full expansions of certain acronyms due to their exclusive, widespread use instead of the phrases or concepts they abbreviate. This convention may make it exceedingly difficult for the uninitiated reader to find his or her way in technical military reports and academic papers within the military operations research literature.

In order to render the exposition of this dissertation as credible, plausible and convincing as possible to the intended target audience, *viz* high-ranking military practitioners and commanders, the convention is also adopted in this dissertation of employing acronyms that are well known in the military domain instead of the full phrases they represent. An attempt has nevertheless also been made to render the dissertation accessible to non-military readers by clearly defining the meaning (in italics) of every acronym employed at its first occurrence and by providing a detailed list of acronyms and their meanings, arranged in alphabetic order, in the preamble of the dissertation.

The understanding of the reader is therefore requested in this respect, and the author unreservedly issues an apology in advance for any frustration that may be experienced by the cumbersome action of having to look up the forgotten meanings of acronyms during the course of reading this document.

Dedication

This dissertation is dedicated in loving memory to:

Daan & Dorothy Lötter
and
Piet & Annatjie Griesel.

List of Reserved Symbols

Symbols in this dissertation conform to the following font conventions:		
$\boldsymbol{x}$	Symbol denoting a vector	(Bold font)
$\mathbb{A}$	Symbol denoting a set	(Blackboard capitals)

The following symbols are reserved for exclusive use in the following contexts.

Symbol	Meaning
$\mathbb{A}$	A set of n_a attack techniques that can be flown by the aircraft in $\mathbb{V}$
$\mathbb{E}$	A set of n_e weapons that may be carried by the aircraft in $\mathbb{V}$
$\mathbb{L}_u$	A set of n_ℓ intensity levels associated with environmental condition $u \in \mathbb{U}$
$\mathbb{U}$	A set of n_u environmental conditions affecting the effectiveness of WSs
$\mathbb{V}$	A set of n_v aircraft types available to the opposing force
$\mathbb{W}$	A set of n_w GBAD WS types available to the own force

Symbol	Meaning
$A_i(\tau)$	The number of ammunition units available to WS i during time stage τ
c_{ij}	The cost of assigning WS i to threat j
d_i	The setup time of WS i
e_{ij}	The start of a temporal interval during which WS i can engage threat j
e_{ijk}	The FTTF for WS i when engaging threat j during the pair's k^{th} FW
F_{ij}	The number of feasible FTTF values for WS-threat pair (i, j)
f_{ij}	The number of distinct FWs for WS-threat pair (i, j)
κ	The maximum number of WSs that may be assigned to any aerial threat
$L_i^{\min}$	The minimum pre-specified length of a FW for WS i
L_{ij}	The length of a FW for WS-threat pair (i, j)
ℓ_{ij}	The end of a temporal interval during which WS i can engage threat j
ℓ_{ijk}	The LTTF for WS i when engaging threat j during the pair's k^{th} FW
M	The number of objectives in a multi-objective optimisation problem
M_{ijk}	The set of consecutive time stages utilised in calculating the fixed-mean priority-weighted survival probability of the k^{th} FW for WS-threat pair (i, j)
$m(\tau)$	The number of ground-based WSs (at time stage τ)
μ_{ijk}	The fixed-mean priority-weighted survival probability calculated for WS-threat pair (i, j)'s k^{th} FW
$n(\tau)$	The number of aerial threats (at time stage τ)
p_{ij}	The SSHP value of WS i when assigned to threat j
p_{ijk}	The SSHP value of WS i when assigned to threat j during the WS-threat pair's k^{th} FW

q_{ij}	The survival probability of threat j when engaged by WS i
q_{ijk}	The survival probability of threat j when assigned by WS i during the pair's k^{th} FW
s_{ijk}	The engagement time duration of threat j by WS i during the pair's k^{th} FW
t	The number of discrete time stages in the temporal period T
τ	An indexing symbol for time stages
V_j	The threat or elimination priority value of threat j
x_{ij}	A binary decision variable taking the value 1 if WS i is assigned to threat j, or the value of 0 otherwise
x_{ijk}	A binary decision variable taking the value 1 if WS i is assigned to threat j during the pair's k^{th} FW, or the value zero otherwise
y_{ihj}	A binary decision variable taking the value 1 if threat h directly precedes threat j in a sequence of engagements by WS i, or the value 0 otherwise

Symbol	Meaning
$E_\tau(i, j, T_\ell)$	The efficiency of WS i predicted at time stage τ for future time stage $\tau + T_\ell$ in respect of threat j
$\boldsymbol{M}_\tau(w, v, e, a)$	The SSHP matrix for time stage τ corresponding to a WS of type $w \in \mathbb{W}$ for the following formative threat combination: an aircraft of type $v \in \mathbb{V}$ carrying weapons of type $e \in \mathbb{E}$ and executing an attack technique $a \in \mathbb{A}$
$\tilde{\boldsymbol{M}}_\tau^{u_\ell}(w, v, e, a)$	The efficiency matrix containing the discounted SSHP value entry at time stage τ for environmental condition $u \in \mathbb{U}$ at intensity level $u_\ell \in \mathbb{L}_u$ for a WS of type $w \in \mathbb{W}$ and for the following threat combination: an aircraft of type $v \in \mathbb{V}$, carrying weapons of type $e \in \mathbb{E}$ and executing an attack technique $a \in \mathbb{A}$
$\overline{\boldsymbol{M}}_\tau(w, v, e, a)$	The WSEM containing SSHP values discounted for environmental conditions and terrain restrictions at time stage τ for a WS of type $w \in \mathbb{W}$ and for the following threat combination: an aircraft of type $v \in \mathbb{V}$, carrying weapons of type $e \in \mathbb{E}$ and executing an attack technique $a \in \mathbb{A}$
$\boldsymbol{M}_{t+T_\ell}(w, v, e, a)$	the WSEM at time $t + T_\ell$ for a WS of type $w \in \mathbb{W}$ and for the following threat combination: an aircraft of type $v \in \mathbb{V}$ carrying a weapon of type $e \in \mathbb{E}$ and executing an attack technique of type $a \in \mathbb{A}$
$\boldsymbol{P}_{t+T_\ell}(w, v, e, a)$	The PPM at time stage τ predicted for time stage $\tau + T_\ell$

List of Acronyms

AD: Air Defence

ADC: Air Defence Control

AM: Attribute Management

C2: Command and Control

CIWS: Close-In Weapon System

DA: Defended Asset

DBMOSA: Dominance-Based Multi-Objective Simulated Annealing

DS: Decision Support

DSS: Decision Support System

EEM: Engagement Efficiency Matrix

EQ: Engagement Quantisation

EW: Electronic Warfare

FCO: Fire Control Officer

FNSA: Fast Nondominated Sorting Algorithm

FPP: Flight Path Prediction

FTTF: First-Time-To-Fire

FW: Fire Window

GBAD: Ground-Based Air Defence

HCI: Hostility Classification/Identification

HMI: Human Machine Interface

IFF: Identification — Friend or Foe

IPB: Intelligence Preparation of the Battlefield

LRSAM: Long-Range Surface-to-Air Missiles

LTTF: Last-Time-To-Fire

MCDA: Multi-Criteria Decision Analysis

MMI: Man-Machine Interface

MRSAM: Medium-Range Surface-to-Air Missiles

NCW: Network-Centric Warfare

NSGA II: Nondominated Sorting Genetic Algorithm II

OODA: Observe-Orientate-Decide-Act

OPTEMPO: OPerational TEMPO

P: Priority

PEF: Physical Element Filter

PFP: Predicted Flight Path

PPI: Plan Position Indicator

PPM: Predicted Position Matrix

R: Repairability

SA: Situation Awareness

SBAD: Surface-Based Air Defense

SHORADS: SHOrt-Range Air Defence System

SSHP: Single Shot Hit Probability

TADMUS: TActical Decision Making Under Stress

TCI: Type Classification/Identification

TE: Threat Evaluation

TEM: Threat Evaluation Model

TEFM: Threat Evaluation Fusion Model

TM: Track Management

TEWA: Threat Evaluation and Weapon Assignment

V: Vulnerability

VI: Vital Importance

VRPTW: Vehicle Routing Problem with Time Windows

VSHORADS: Very SHOrt Range Air Defence System

WA: Weapon Assignment

WAM: Weapon Assignment Model

WAP: Weapon Assignment Problem

WASP: Weapon Assignment Scheduling Problem

WASS: Weapon Assignment Solution Selection

WS: Weapon System

WSEM: Weapon System Efficiency Matrix

WTA: Weapon-Target Assignment

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Background

The USS Vincennes (CG 49) is a United States Navy guided missile cruiser which was commissioned for service in 1985 [36]. It originally saw service in the Pacific Ocean before it was dispatched to the Persian Gulf (in particular to the *Straight of Hormuz*[1]) in 1988 to support *Operation Earnest Will*[2] during the Iran-Iraq War which took place during the period 1980–1988 [75]. A picture of the USS Vincennes is shown in Figure 1.1(a). At the time, the USS Vincennes was equipped with the United States' then state-of-the-art naval combat information centre. The heart of this centre was the well-known *Aegis combat system*[3]. It included advanced *Command and Control* (C2) and weapon control systems which were able to track and guide *Weapon Systems* (WSs) in order to engage and destroy enemy targets [76]. Aegis was developed as a fully integrated *Decision Support System* (DSS) boasting state-of-the-art radar and missile systems and display screens for depicting the surrounding air picture, making it the first integrated system providing real-time *Decision Support* (DS) to operators on several fronts, including air, surface and subsurface DS [81]. The Aegis display screens in the combat information centre aboard the USS Vincennes are shown in Figure 1.1(b).

By 1984, the war between Iraq and Iran had evolved to include air attacks against oil tankers and merchant shipping of neighbouring countries. On 17 May 1987, the Iraqi Air force attacked the United States guided missile frigate, the USS Stark (FGG-31), which resulted in severe damage

[1] *The Straight of Hormuz* provides the only sea passage for sea vessels between the Persian Gulf and the Gulf of Oman [181].

[2] *Operation Earnest Will* was launched on 24 July 1987 and lasted until 26 September 1988. It involved the United States military protecting Kuwaiti-owned tankers from Iranian attacks during the period 1987–1988. This operation was launched three years into the so-called *Tanker War phase* of the Iran-Iraq war [75].

[3] The word *Aegis* stems from Greek mythology and means *protective shield* [76].

(a)

(b)

FIGURE 1.1: *(a) The USS Vincennes which shot down Iran Air flight 655 [36] and (b) the Aegis Combat System information centre on board the USS Vincennes [76].*

to the port side of the vessel and the death of thirty seven American sailors [183]. Furthermore, United States naval forces also exchanged gunfire with Iranian gunboats late in 1987 and in April 1988 the United States guided missile frigate, the USS Samuel B. Roberts, struck an Iranian sea mine which severely damaged the vessel, resulting in it being out of commission for several months [183]. All these events exacerbated an already tense relationship between the United States, Iraq and Iran in the Persian Gulf at the time.

Then, on 3 July 1988, a civilian jet airliner, Iran Air flight 655, carrying 290 passengers and crew members was flying over the Persian Gulf. It was crossing the Straight of Hormuz *en route* to Dubai in the United Arab Emirates on its usual flight path. On the morning of 3 July, however, the USS Vincennes received intelligence from one of its helicopters that it had received small arms fire from Iranian gunboats operating within Iranian territorial limits [183]. The USS Vincennes then entered Iranian territorial waters to open fire on the Iranian gunboats. While the USS Vincennes was retaliating against the gunboats, confusion reigned aboard the vessel as the tracking of aircraft in the area had become muddled. Amidst the chaos, a USS Vincennes operator spotted Iran Air flight 655 on the Aegis radar. The passenger aircraft was climbing at the time and its *Identification Friend or Foe*[4] (IFF) transponder was set to *Mode III civilian* code rather than to the purely *Mode II military* code. The USS Vincennes tried several times to make contact with the airliner, but the airliner failed to respond to any of the communication attempts as recorded by the Aegis onboard combat system. The crew onboard the USS Vincennes then mistakenly concluded that the airliner was a hostile F-14 Tomcat fighter aircraft. It fired two radar-guided missiles at the airliner "in self-defence" and shot down the airliner over Iranian airspace [183]. Only after the missiles had been launched did the crew realise that they misclassified the airliner. The consequences of the crew's decision were catastrophic, resulting in the deaths of all 290 passengers and crew members on board the airliner.

The disaster prompted a number of formal investigations by the *United States Navy* and the *International Civil Aviation Organisation* [126]. During these investigations it was concluded that the misidentification of the aircraft may have been due to a phenomenon known as *compression of time*[5] in conjunction with the contemporaneous engagement with Iranian gunboats as well

[4]IFF communication systems are used to confirm the nature and intent of an aircraft, and to differentiate between friendly and non-friendly aircraft [35].

[5]*Compression of time* is a phenomenon in which a person's perceived duration of a time interval understates its true duration [155].

as emotional stress and a psychological phenomenon known as *scenario fulfilment*[6]. Prompted by the findings of these investigations, the *United States Office of Naval Research* sponsored research resulting in the establishment of a programme called *Tactical Decision Making Under Stress* (TADMUS) [155].

The main focus of the TADMUS programme was to investigate the effect that stress has on the decision making abilities of military operators, to use this information to improve the combat performance of team operators under stress by means of enhanced training and to provide them with suitable DSS during combat situations [30, 98, 81]. The focus in the design of the DSS was on enhancing the performance of tactical operators who act as decision makers in combat situations. The approach followed in the development of the DSS was to analyse the cognitive tasks performed by operators in a ship's combat information centre and then to develop a set of display concepts which may be used in support of tasks performed by the operators based on the underlying decision making process typically followed by these operators [93].

The results of the cognitive task analysis highlighted two higher-order tasks performed by operators — the assessment of the current situation and the selection of alternative courses of action. In respect of the first higher-order task (*i.e.* information transactions associated with assessing the current tactical situation), it was found that in 87% of the decisions made, operators tried to match the observed events in the scenario to similar scenarios they had previously encountered, while in 12% of the decisions made, operators developed novel hypothetical explanations in order to make sense of the observed scenarios [81, 93]. When considering the second higher-order task (*i.e.* selecting alternative courses of action), it was found that 94% of the operators based their alternative courses of action solely on established rules of engagement, while the remaining 6% of the operators developed strategies extrapolated from previous encounters to base their alternative courses of action upon. Furthermore, it was found that experienced operators were not particularly well served by current DSSs in demanding missions and that operators typically experienced periodic loss of *Situation Awareness* (SA) often due to limitations in human memory and shared attention capacity.

Based on the aforementioned results of the cognitive task analysis, a prototype DSS was developed with the following three objectives in mind: (1) to minimise the mismatches between cognitive processes and the data available in the combat information centre in order to facilitate decision making, (2) to mitigate the shortcomings of current display sets in the combat information centre when imposing high information processing demands and exceeding the limitations of human memory, and (3) to transfer the data in the current combat information centre from numeric to graphical representations wherever appropriate [81, 93]. The aim in designing the improved DSS was to reduce errors, reduce workload on the operators and to improve adherence to the rules of engagement. The integrated display of the set of display concepts deriving from the TADMUS programme may be found in Figure 1.2.

The DSSs and human operators responsible for the detection of threats, evaluating these threats and determining alternative courses of action in order to eliminate or deter such threats in combat situations, typically have to make these important decisions within very small time frames. Although most DSSs are fully automated, many commanding officers fail to entrust the safety of their commands to a software program and end up rather trusting their own intuition. The effect of this mistrust may be detrimental to the safety of own forces and the protection of assets. One example of such an incident is the attack on the United States guided missile frigate, the USS Stark, mentioned previously. An Iraqi fighter jet fired two French-built

[6] *Scenario fulfilment* occurs when operators work under stressful conditions, confusing a training scenario with reality and responding accordingly [65]. Operators typically ignore sensory information which contradicts the scenario. Roberts [156] describes scenario fulfilment as "you see what you expect."

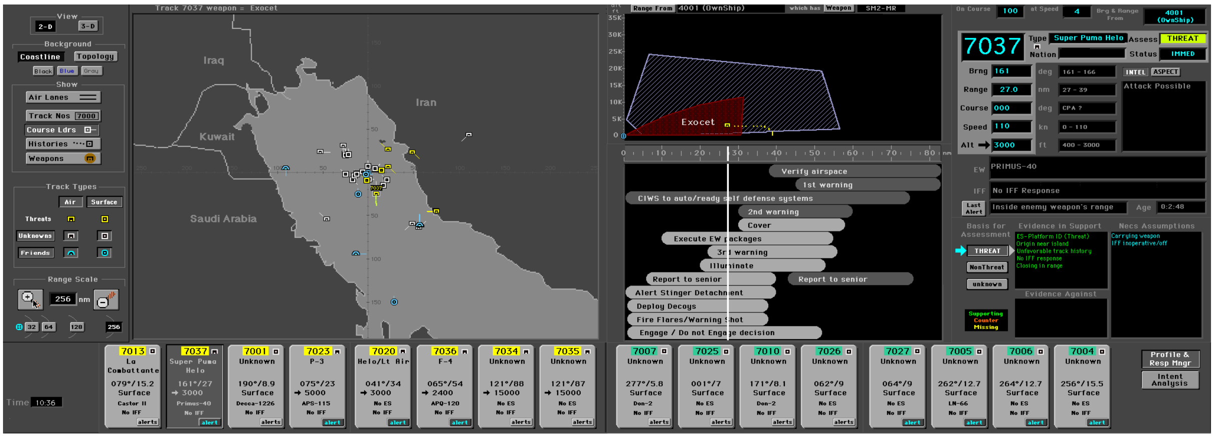


FIGURE 1.2: *The integrated display of the set of display concepts developed in the TADMUS DSS [93].*

Exocet anti-ship missiles at the USS Stark [183]. The Aegis system on board the USS Stark automatically detected, classified and tracked the two missiles as hostile, but the commanding officer did not trust the results proposed by the DSSs. This resulted in the missiles hitting the frigate unchallenged.

The disaster which emanated from the erroneous decisions made on both the USS Vincennes and USS Stark, as well as the research findings of the TADMUS programme described earlier, accentuate the importance of providing efficient DS to operators when they have to make tactical decisions in combat situations. Furthermore, it is crucial that the DS provided by a DSS be of such a nature that operators are able to trust the courses of action proposed — the idea is that the results of the DSS conform to the judgement of the operators and that it should only confirm the courses of action that the operators would, in fact, have made unassisted if allowed ample response time. It is also important that the output of the DSS conforms to the standard rules of engagement contained in military doctrine.

1.2 Informal problem description

In a military *Ground-based Air Defense* (GBAD) environment, a number of *Defended Assets* (DAs) on the surface[7], such as sea-faring vessels, air strips, main bridges or fuel depots, typically require protection from enemy aerial vehicles entering the three-dimensional airspace surrounding these DAs, known as the *defended airspace*. A network of sensors is responsible for detecting these aircraft when they enter the defended airspace and protection against these aircraft is afforded by a number of pre-deployed ground-based WSs which are available for engagement in the immediate tactical area.

The problem of defending DAs effectively is twofold. The first part of the problem is to identify any aircraft which enter the defended airspace and to then classify them according to force (*i.e.* enemy or friendly) as well as according to type (*e.g.* as fixed wing aircraft or rotary wing aircraft). After the aircraft have thus been classified, the next step is to investigate the perceived level of threat that each of the enemy aircraft poses to the DAs and to assign a numerical value to each of the enemy aircraft according to some appropriate measure which provides a good indication of the level of threat the aircraft poses to DAs. The processes involved in detecting

[7] In the context of GBAD, the term *surface* refers to the surface of a portion of land, lake or ocean.

aircraft, classifying them and estimating the perceived level of threat that they pose to DAs are collectively known in the military literature as *Threat Evaluation* (TE).

The second part of the problem of defending DAs in a GBAD environment is to assign one or more available surface-based WSs to engage those aircraft classified as threats in a bid to neutralise or deter them. The process involved in assigning available WSs to threats is known in the military literature as *Weapon Assignment* (WA) and the underlying combinatorial optimisation problem of assigning available WS to threats so as to achieve the best effect is known in the operations research literature as the *Weapon Assignment Problem* (WAP).

A human operator, called a *Fire Control Officer* (FCO), is responsible for performing TE and solving the WAP in real-time under severely stressful conditions. The speeds at which enemy aerial vehicles typically approach DAs leave very short time frames for analysis or delays in reaction time. Not only does the FCO have to decide which WSs to assign to the threats, but he also has to choose the number of WSs to assign to each individual threat as well as the timings of these engagements. The assignment of multiple WSs to a threat at any point in time may result in an increased probability of eliminating that specific threat, but may simultaneously compromise the number of WSs available for assignment during future time intervals called *time stages*.

After deciding which WSs to assign to a specific threat, the FCO also has to decide whether he should assign any WS to engage an aerial threat during the current time stage, or whether he should rather wait until a later time stage before initiating such an engagement when the probability of eliminating the threat may be higher. Moreover, a WS achieving a longer range typically involves a higher monetary cost of assignment than does a WS achieving a shorter range. A longer-range WS may be assigned to aerial threats earlier, but at an increased assignment cost.

When a large number of aircraft approach the defended airspace almost simultaneously from various directions (an attack scenario typically employed by opposing forces in a bid to overwhelm FCOs), TE and solution of the WAP become very complex and almost impossible for the FCO to carry out effectively in real time. A computerised *Threat Evaluation and Weapon Assignment* (TEWA) system may therefore be employed to perform TE and solve the WAP in real time, and to provide the results obtained as DS to the FCO. By using such a DSS in real time, overall system performance may be improved, and operator stress and workload may be alleviated significantly. Using his own judgement, which is typically based on considerable experience and training, in conjunction with the alternatives recommended by the TEWA DSS, the FCO is expected to be able to make more effective assignment decisions with respect to the engagement of aerial threats by WSs, and to make these assignment decisions with more confidence.

Many TEWA DSSs exist in the military defence industry. Examples of such systems include the Aegis system mentioned earlier in §1.1, the *Battlefield Command Support System* [163] and *Tactical Command and Control System* [164], developed by SAAB, as well as the *GENESIS Ship Integrated Combat Management System*, developed by the Turkish navy [191]. These "off-the-shelf" systems are typically commercially available as "black box" systems in the defence industry. The design rationales of such systems are usually not disclosed in the open literature, which limits the possibility of refining the software systems and fully understanding the working of these systems.

In 2007, Roux and Van Vuuren [161], reviewed the state of the art of such TEWA DSSs about which information was available in the open literature at that stage. In 2008, they went further by suggesting a design for a complete, generic TE subsystem for use in a GBAD context for the South African military [162]. Since then there has largely been a void in the design of a fully-fledged TEWA DSS within the South African context, especially on the WA side of such a larger TEWA DSS. The aim in this dissertation is therefore to build on the work of Roux and Van Vuuren, by putting forward a design of a first-order generic WA subsystem as counterpart to the TE subsystem already designed for inclusion in a larger TEWA DSS system. The purpose of this subsystem is to provide real-time DS to FCOs with respect to high-quality WS-threat assignment alternatives. The design follows the structured approach proposed by Roux and Van Vuuren [161] in the context of TE and contains a refinement of the substructures contained in such a subsystem.

1.3 Dissertation objectives

The following nine objectives are pursued in this dissertation:

I To *conduct* a comprehensive survey of the literature with respect to:

   (a) the processes involved in TE and WA, as well as the interaction between these processes in the context of a GBAD context,

   (b) GBAD procedures and modelling approaches in general, with an emphasis on WA DS,

   (c) the physical, functional and cognitive elements encompassed within the WA subsystem of a TEWA DSS, accentuating the requirements for the successful working of such a subsystem,

   (d) the current status of WA-related research in a GBAD environment within the South African military domain,

   (e) available single and multi-objective *Weapon Assignment Models* (WAMs) in the operations research literature which may be adapted for inclusion in a WA subsystem so as to facilitate recommendation of high-quality WS-threat assignment alternatives,

   (f) the notion of Pareto optimality within the realm of multi-objective optimisation,

   (g) available solution methodologies for solving the WAMs of Objective I(e) and the requirements of each methodology for the successful implementation thereof within a GBAD WA subsystem, and

   (h) DSSs in general, with an emphasis on the design of DSSs in a military setting.

II To *establish* a suitable, novel framework design for a WA subsystem intended for use as real-time DS within a larger TEWA system.

III To *propose* a classification scheme for WAMs, encompassing models of different levels of complexity, for inclusion in the framework design of Objective II which are able to propose high-quality WS-threat assignment pairs based on WAM configurations specified by the FCO, and also to *identify* prototypes of WAMS from the literature for use in each of the proposed classes.

IV To *formulate* a novel tri-objective dynamic WAM which aims to minimise the accumulated survival probabilities of aerial threats (by incorporating their respective threat values), to minimise the accumulated cost of the assignment proposed and to maximise the number of times that the least re-engagable WS in the set of WSs can be used in future assignments, after the current assignment WA window has elapsed. The dynamic element of the WAM should facilitate the scheduling of subsets of future time stages during which WS-threat pair assignments are to occur. It is envisaged that this WAM may be taken as prototype in the most complex class of WAMs proposed in the classification scheme of Objective III.

V To *propose* appropriate solution methodologies for the WAMs in each of the classes of the classification scheme proposed in pursuit of Objective III, elucidating the input parameters required for the timeous and successful solution implementation of each WAM.

VI To *suggest* appropriate approaches, procedures and methodologies that may be considered during a practical implementation of the proposed WA subsystem of Objective II.

VII To *illustrate* the workability of each WAM prototype of Objective III by solving it in the context of a simulated, realistic GBAD scenario according to the solution methodologies of Objective V.

VIII To *evaluate* the success, significance and usefulness of the design framework of Objective II in the context of the simulated GBAD scenario of Objective VII.

IX To *suggest* sensible ideas for possible further work related to the research conducted in this dissertation.

1.4 Dissertation scope

The following delimitations of research focus are adopted in this dissertation. For the purposes of this dissertation, it was decided to pursue the design of a WA subsystem as part of an *Air Defence Control* (ADC) system within a GBAD environment specifically. This seemed to be the natural choice based on access the author had to a number of military experts residing within the South African National Defence Force who are GBAD experts in respect of the substructures of a GBAD environment, including its physical, functional and cognitive elements. The mathematical WA models considered in this dissertation also take as input the output of the TE subsystem previously developed by Roux and Van Vuuren [162], therefore also making it natural to delimit the scope of the work to the development of a WA subsystem within a GBAD environment.

In addition, the scope is delimited to consider only *fixed wing* aircraft as threats and also to consider the elements in the deployment of the *Air Defence* (AD) as *fixed* (*i.e.* the positions of the sensor systems, the DAs, the WSs and other AD physical elements are fixed during a mission).

Furthermore, the objective in the development of the WA models in the dissertation is that they should be of a generic, flexible and adaptive nature. This implies that the WA subsystem architecture proposed should be able to accommodate temporal updates of data related to its functional elements or additional functional elements (*i.e.* not restricting the type, number or related attributes of physical, functional or cognitive elements considered during WA).

1.5 Research methodology

The execution of the research reported in this dissertation consists of five stages. The first stage is mainly concerned with reviewing the literature on various topics related to the contents of the dissertation. First, a typical GBAD environment and the elements that such an environment encompasses are reviewed in order to establish a framework for understanding the context in which a GBAD TEWA system is expected to operate. Next, the typical working of a TEWA system is reviewed in some detail within a GBAD environment, with an emphasis on the processes involved in the TE and WA subsystems individually as well as how these processes interact with one another. WA DS within the South African military domain is also reviewed in order to establish the current state of the art of WA DS tools available to FCOs in a GBAD environment. WAMs from the military operations research literature are reviewed next. The literature review on WA DS contains descriptions of WAMs since the inception of the first WAM during the 1950s up to the current state of the art of WAMs of the early 21st century.

Furthermore, a number of mathematical prerequisites are established, thereby paving the way for the TEWA framework design. The literature review concludes with descriptions of solution methodologies from the operations research literature which may be used to the solve the WAMs found in the military decision support literature.

The next stage in the execution of the research of this dissertation involves the establishment of a novel framework design for WA DS to FCOs as part of a larger TEWA system for use in a GBAD environment. The architecture proposed includes four components which function in a sequential manner to provide real-time DS to FCOs. The four components are described comprehensively and examples are given of how these components may function both individually and together. The interaction between these components, as well as the flow of information between the components, are highlighted.

A novel WAM component is proposed as part of the aforementioned WA subsystem architecture. This component contains four classes of WAMs which may be used to propose WS-threat assignment pairs. These classes range over different levels of complexity in terms of the quality and level of realism of the solutions that they are able to generate, as well as the difficulty associated with implementing them. WAM prototypes for use in three of these classes are taken from the literature and are discussed in detail. A novel tri-objective, dynamic WAM is finally formulated and proposed for use in the final, most complex class of WAMs, thereby filling the current void in this class of WAMs in the military operations research literature.

The next research stage involves illustrating the working of each of the WAM prototypes in the context of a simulated, but realistic, GBAD scenario. This is achieved by adopting appropriate solution methodologies from the operations research literature. Emphasis is placed on a study of the input parameters required for the successful implementation of each WAM. The results returned by the WAM prototypes (*i.e.* the assignments and/or engagement schedules proposed by each WAM prototype) are tabulated numerically and also displayed graphically in the form of a top-view of the scenario in which the WS-threat assignment pairs are illustrated.

The results obtained by each WAM prototype are then validated by means of three validation techniques. The first technique involves a face validation in which the numerical and graphical results are analysed critically. The second validation involves a comparison of the results returned by each of the WAM prototypes and those of a random benchmark of the problem instance. More specifically, thirty feasible WS-threat assignment solutions are generated for each WAM prototype and compared with the results returned by each WAM prototype. Finally, two military experts are consulted in respect of the validation of the results returned by each WAM prototype. The problem instance for each WAM is presented to the experts and they are asked to solve the problem manually based on their intuition, knowledge and experience. Their results are then compared with the results obtained by the WAM prototypes. The results obtained by each of the WAM prototypes are finally also presented to the military experts to analyse critically. They are then asked to voice their opinions on the quality, realism and usefulness of the results returned by each of the WAM prototypes.

The final stage in the execution of the research of this dissertation involves the identification of voids in the proposed WA subsystem architecture and to propose suitable ideas for further work that will lead to a refinement of the architecture proposed with a view to filling these voids.

1.6 Dissertation organisation

The nine chapters in the remainder of this dissertation are organised into four parts. Part I contains two chapters pertaining to the open literature relevant to the context of the problem considered in this dissertation. In Chapter 2, a theoretical, integrated framework for GBAD is reviewed. This is achieved by providing detailed descriptions of the processes typically involved in TE and WA, as well as the interaction between them, in fulfilment of Dissertation Objective I(a) of §1.3. An introduction to GBAD-related research in the literature is also provided in fulfilment of Dissertation Objective I(b). The chapter further includes a discussion on available AD artillery in the South African military domain as well as existing ADC measures employed. WA DS is also discussed in a theoretical setting, in fulfilment of Dissertation Objective I(d). Next, detailed discussions are provided on important practical concepts such as SA, C2 measures and network-centric warfare, which should be accommodated in the design of any modern TEWA DSSs. The physical, functional and cognitive elements of a TEWA DSS are furthermore elucidated in fulfilment of Dissertation Objective I(c). The physical elements (the hardware systems) typically present within the tactical GBAD environment in which a TEWA DSS is employed are discussed in some detail. These elements include DAs, sensor systems, aerial threats to the system and GBAD WSs. Next, the functional elements (the software systems) which should be considered in the design of a TEWA DSS are described. The current functional elements in the South African ADC system are discussed and a functional high-level design of an integrated TEWA DSS is presented. Testing and training measures, as well as maintenance procedures which should be included in a TEWA DSS design are also discussed in Chapter 2. In addition, the cognitive elements of a TEWA DSS are also discussed. This includes the cognitive behaviour of FCOs and WS operators under various conditions of stress, how these operators are expected to interact with the TEWA DSS and how they may react to the DS provided by computerised DSSs. The chapter closes with a discussion on the design and implementation of military DSSs that are described in the open literature.

An exposition on the evolution of WAMs since their inception during the 1950s up to the current state of the art WAMs of the early 21$^{\text{st}}$ century is offered in Chapter 3, in fulfilment of Dissertation Objective I(e). The chapter opens with a brief introduction to the notion of **NP**-completeness since even the simplest WAMs in the literature are **NP**-complete. The remainder of the chapter contains detailed descriptions of various WAMs available in the military operations research literature.

Part II of the dissertation also contains two chapters and is concerned with establishing a collection of mathematical prerequisites for the development of a TEWA framework design put forward later in the dissertation. Chapter 4 is dedicated to a discussion on multi-objective optimisation, in fulfilment of Dissertation Objective I(f), since multi-objective WAMs exist in the literature. Moreover, a (strong) case may be made that the inherent nature of the WAP requires complex trade-off decisions which necessitate a multi-objective WA modelling approach. Notions from the literature on multi-objective optimisation covered in Chapter 4 include nonconvexity, dominance, Pareto optimality and necessary conditions for Pareto optimality.

In Chapter 5, various solution methodologies are reviewed from the operations research literature, in fulfilment of Dissertation Objective I(g), which may be employed when solving the

WAMs reviewed in pursuit of Dissertation Objective I(e). The chapter opens with a discussion on exact model solution methodologies. This is followed by a discussion on various heuristic model solution approaches and, finally, metaheuristic solution approaches.

Part III constitutes the main contribution of the dissertation and is dedicated to a novel, detailed WA system framework design. This part contains three chapters. Chapter 6 is the heart of the dissertation and opens with a discussion on the various approaches towards the design and implementation of DSSs that are described in the open literature, in fulfilment of Dissertation Objective I(h). Emphasis is placed on important aspects that have to be considered in the design of DSSs intended for use in the military domain, especially in the context of GBAD. Next, a generic WA subsystem architecture is put forward in fulfilment of Dissertation Objective II. This chapter also includes descriptions of all the substructures contained in the proposed framework, the flow of data between these substructures and how the proposed framework may fit into a larger TEWA DSS. A WAM component, which forms the heart of the proposed WA subsystem framework, is proposed and discussed in detail. Four classes of models are also proposed and a desirable WAM prototype from the literature is described for implementation in each class of the framework, in fulfilment of Dissertation Objective III. Furthermore, a novel tri-objective, dynamic WAM is formulated which aims to achieve suitable trade-offs between the minimisation of the accumulated survival probabilities of the threats, the minimisation of the accumulated cost of the proposed assignment and the maximisation of the number of times that the least re-engagable WS in the set of WS can be used for future assignments *after* the current assignments have occurred, in fulfilment of Dissertation Objective IV. This is followed by a discussion on various solution methodologies which may be employed in the proposed WA framework in order to solve the WAMs of Dissertation Objectives III and IV within suitable time frames, in fulfilment of Dissertation Objective V.

In Chapter 7, a number of important practical implementation suggestions are put forward in respect of the implementation of the WA subsystem architecture proposed in pursuit of Dissertation Objective II, in fulfilment of Dissertation Objective VI. These suggestions include how to avoid overwhelming the FCO with information when providing DS during combat situations and to allow the FCO to configure the models for use in the WAM component during the pre-deployment stages of a mission in order to avoid confusion when he has to make important WA engagement decisions under considerable stress during the mission.

Part III closes in Chapter 8 where a simulated, but realistic, GBAD scenario is employed to illustrate the working of the four model prototypes of Dissertation Objective III, solving them according to various solution methodologies researched in pursuit of Dissertation Objective V, in fulfilment of Dissertation Objective VII. The scenario is described in detail, and the numerical results returned by each of the WAM prototypes are presented and interpreted. The chapter closes with an evaluation and validation of the results obtained by each WAM, in fulfilment of Dissertation Objective VIII.

The dissertation finally closes in Part IV, which contains the final two chapters. A brief summary of the dissertation contents as well as an appraisal of the dissertation contributions may be found in Chapter 9. This is followed in Chapter 10 by various suggestions for possible further work, in fulfilment of Dissertation Objective IX.

Part I

Literature review

CHAPTER 2

Ground-based air defence

Contents

In this dissertation, WA is considered as part of the larger TEWA system in the context of a GBAD environment. Although the focus in this dissertation is on WA, the processes involved in WA are closely knit with the processes involved in TE — the mathematical models typically employed in the WA process rely on the output received from the TE subsystem as input in order to propose assignments of WSs to aerial threats. TE processes and the influence they have on WA processes are therefore discussed in some detail in this chapter.

The aim in this chapter is to provide the reader with some insight into the complexity, relationships and contextual orientation of the TE and WA processes in GBAD and how modern fundamental notions within the theory of warfare, including network centric warfare, C2 and situation awareness, underlie these processes.

The chapter opens in §2.1 with a discussion aimed at contextualising the process of TE, the process of WA and the integration of these two processes. Next, in §2.2, the notion of C2 is discussed in some detail. This includes a specification of the facilities, equipment, communications, procedures and personnel that are considered essential for a commander to plan, direct and control operations during a mission. The section also includes a detailed description of the well-known *Observe-Orientate-Decide-Act* (OODA) loop (a very basic C2 model) due to Boyd [20]. A discussion on extensions of this model by other authors is finally also provided.

In §2.3, the notion of *SA* is discussed. This concept has had a large impact on operators in complex systems of warfare since its inception in the mid 1970s. In essence, SA may be thought of as a thorough understanding of the state of the surrounding tactical environment. This includes an understanding of the parameters relevant to a TEWA system.

Next, in §2.4, the focus shifts to a discussion on one of the key concepts for future warfare, known as *network centric warfare* (NCW). This notion aims to link GBAD physical elements (such as sensors and WSs) and cognitive elements (such as decision makers) into an effective and responsive control unit.

In §2.5, the focus shifts yet again to a discussion on the elements typically contained within a GBAD system. These include functional elements, physical elements and cognitive elements. The chapter finally closes in §2.6 with a brief summary of the chapter contents.

2.1 TEWA within a GBAD in context

The TE process should be seen as part of a larger process known as *Intelligence Preparation of the Battlefield* (IPB) which is a systematic, continuous approach followed to assess and analyse the environment and to quantify the various levels of threat in the specific geographical area surrounding the battlefield — intelligence with respect to the effects that the environment has on the own force's ability to achieve its mission are collected, organised and analysed [192]. IPB is a continuous process, implying that it is performed both prior to and during a mission in order to ensure that the results of the IPB process remain complete and valid. It is defined as a four-step process which includes defining the battlefield environment, describing the effects of the battlefield, evaluating the levels of threat posed by the opposing force and determining an appropriate course of action [192]. In the context of IPB, TE may be defined as the process by which the own force applies its knowledge of the opposing force, obtained from its doctrine, tactics and capabilities, so as to ascertain the nature and level of the threat to which the own force is exposed.

Up to 2008, TE in the context of a GBAD TEWA system was a poorly defined process in the academic literature due to the difficulty associated with capturing operator thought in real time — it forms part of the cognitive domain and since each operator is unique, operator thought attributes may be difficult to define. According to Paradis *et al.* [144], TE in the defence domain may be defined as "the part of threat analysis concerned with the ongoing process of determining if an entity intends to inflict evil, injury, or damage to the own force and its interests, along with the ranking of such entities according to the level of threat they pose." TE as part of a GBAD TEWA DSS therefore depends largely on the situational picture, which may include the states of enemy aircraft in the area of defence, the locations of DAs, attributes of enemy aircraft platform types and attack techniques, available WSs and surveillance systems, opposing force doctrine, intelligence reports gathered, terrain features of the tactical environment, and knowledge pertaining to the opposing force's recent behaviour within the tactical environment [159].

Aerial threats are typically analysed according to their *capability* and *intent* [144]. The former refers to the ability of a threat to inflict injury or damage to the DAs, whereas the latter refers to a threat's willingness or determination to inflict injury or damage to the DAs. The capability of a threat is assessed by taking into account factors such as the size and aircraft composition of a group of threats, its proximity to the DAs in question and attributes of its WSs carried, its surveillance systems, and various attack techniques that may be utilised. The intent of a threat, on the other hand, is generally more difficult to assess since intent is assessed based on subjective measures. The intent of a threat is usually estimated by considering factors such as the speed, acceleration and heading of the threat with respect to the DA(s), the detection of emissions from its fire control radar and an estimation of its possible courses of action based on its observed pattern of movement, and the events and activities in which it has previously participated.

In 2008, Roux and Van Vuuren [162] put forward a generic design of a TE subsystem specifically intended for use in the context of a GBAD environment to assist FCOs in quantifying the level of threat that enemy aircraft may pose to DAs. They proposed use of three classes of TE models, each containing a suite of mathematical models (including both qualitative and quantitative models), ranging in different levels of complexity and sophistication, and operating in parallel to produce results which are fused together in order to provide the desired DS to FCOs with respect to the level of threat posed by enemy aircraft to DAs.

WA in a GBAD environment, on the other hand, entails the reactive assignment of available WSs to threats in an attempt to engage or counter aerial threats identified in the defended airspace [144]. It involves making decisions in real time that are aligned with own force mission objectives and which are consistent with the rules of engagement, WS characteristics and constraints posed by the tactical environment [159]. According to Paradis *et al.* [144], the problem of assigning WS to threats may be approached in two different ways. The first is a *single-platform* approach in which a single platform protects itself from threats — the most suitable WS is chosen for assignment to counter a threat. The second approach is a *force-coordination* approach in which a C2 platform provides protection against third-party DAs — the most suitable armed platform is identified to engage or counter the threats.

Furthermore, Paradis *et al.* [144] claimed that the assignment of WSs to threats may be executed by following a *threat-by-threat* approach or a *multi-threat* approach. In the threat-by-threat approach, WSs are assigned to threats in a sequential manner in such a manner that the best WS is assigned to each threat (from the highest priority to the lowest priority) in turn — that is, the best WS is assigned to the threat with the highest priority, followed by the second best WS being assigned to the threat with the second highest priority, and so on. This approach therefore involves an optimisation element, although it typically manifests itself in the form of a greedy algorithm. When a multi-threat approach is adopted, however, the assignments of available WSs to the current set of threats are made concurrently in such a manner that some overarching set of objective functions is optimised. Paradis *et al.* [144] pointed out that it is the optimisation of a set of objective functions which distinguishes the two approaches from one another. In this dissertation, it is assumed that WA is approached from a force coordination point of view — a number of DAs require joint protection against a number of aerial threats evaluated by the TE subsystem and a number of WSs are available to engage or to counter these threats. Furthermore, a multi-threat approach is adopted in this dissertation when WA decisions are proposed.

The topic of WA is well documented in the operations research literature and a number of researchers have contributed towards the topic since its inception in 1958 when Manne [125] formulated the first documented weapon assignment model — a detailed description of contributions by various authors towards the formulation of models for the WAP is provided in the following chapter of this dissertation. Central to the successful implementation of a WA subsystem, however, are a number of functional elements which are discussed in detail towards the end of this chapter. At the core of these elements are three important notions, namely the notions of C2, SA and NCW which form a basis for efficient TEWA DSS provided to FCOs. These notions are discussed in some detail in the following sections.

2.2 Command and control

Providing effective TEWA DS to FCOs in a GBAD environment requires a thorough understanding of the underlying C2 processes followed. The term *command* (in a general sense) refers to "directing with authority" or to "giving order to," while the term *control* (in a general sense) refers to "exercising authoritative or dominating influence over." The notion of C2 is very well documented in the military operations research literature and hence, a number of formal definitions for C2 exist. A widely adopted formal definition of C2 is the definition by the *United States Defense Technical Information Center* [33] which defined C2 as "exercise authority and direct by a properly designated commander over assigned and attached forces in the accomplishment of a mission."

The earliest form of C2 measures (prior to the 19$^{\text{th}}$ century) typically involved a single commander who executed all planning, direction and monitoring functions with the assistance of a small number of aides and messengers [208]. Later (during the 19$^{\text{th}}$ century), the Prussian Army developed a so-called *staff system* which involved a commander being assisted by a small number of well-trained officers — the idea behind this approach to C2 was that these officers could assist the commander by carrying out routine planning and monitoring functions in order to leave the commander free to "concentrate on the bigger picture" [208].

The notion of C2 later evolved (during the 20$^{\text{th}}$ century) into a so-called C2 *system*. In this context, the term *command* refers to an "authorative act of making decisions and ordering action," while the term *control* refers to an "act of monitoring and influencing this action" [159]. The *United States Joint Chiefs of Staff* [99] defined a C2 system as follows: "The facilities, equipment, communications, procedures and personnel essential to a commander for planning, directing and controlling operations." According to Wilson [208], this definition accentuates the importance of the cooperation required between human, doctrinal procedures and information technology elements in order to ensure effective C2.

According to Krulak [110], C2 may be seen as the most important element in warfare since it is here where the numerous activities that a military force must execute, gain purpose and direction — effective C2 aids commanders in making the most of what they have (*e.g.* people, information, material and time). Some variations of C2 are primarily of a procedural or technical nature, such as the control of air traffic and air space, the coordination of supporting arms or the fire control of a WS. Others are concerned with the overall conduct of military actions, such as formulating concepts, deploying forces, allocating resources and supervising officers. It is clear that the realm within which a TEWA system operates, falls within the range of both these two variations of C2 [159]. It is therefore essential that a thorough understanding of the applicable operational processes involved in C2 is required for the design of an efficient WA subsystem.

A very basic, well-known model of the C2 decision cycle is known as the *OODA loop*, introduced by Boyd [19, 20]. The OODA loop is a widely accepted model of C2 and is recognised as a decision cycle model that is applicable to all C2 systems — both friendly and adversary [110]. It is acknowledged that other C2 models also exist, such as Lawson's C2 process model [142], the monitor-assess-plan-execute model [82], the input-process-output model [170] and the find-fix-track-target-execute model [185]. The underlying functional elements of all these models are, in essence, the same.

The OODA loop may be represented by a decision cycle containing four phases, *i.e.* to observe, to orientate, to decide and to act, as illustrated graphically in Figure 2.1 [20]. These four phases also constitute the naming of the OODA loop and are now discussed in further detail:

Observation. The first phase in the OODA loop is to *observe*, which entails gathering information with respect to the current decision at hand in terms of own force status, the surrounding area and the opposing force. Information may be collected from internal or external sources [110, 159]. Internal sources typically yield information received from a feedback loop within the decision-making entity, whereas external sources typically yield information gathered from sensors or information from sources outside the decision-making entity.

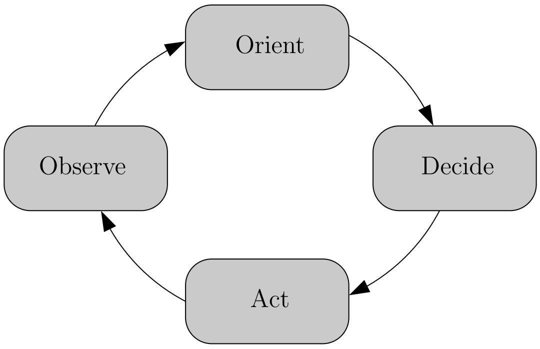


FIGURE 2.1: *The well-known C2 OODA decision loop of Boyd [20].*

Orientation. After observing the situation, the decision-making entity should *orientate* itself with respect to the current situation in order to ascertain certain estimates, the validity of various assumptions, and the results of analyses and judgements in order to form a cohesive picture of the situation at hand. Boyd [20] pointed out that a large portion of the cognitive effort of the decision-making entity resides within the orientation phase. He further stated that this phase consists of two subphases, *i.e. destruction and creation.* The destruction subphase describes a decision-maker's natural instinct to decompose a problem into smaller subproblems until they are close to situations or problems for which the decision-making entity has a solution. Familiarity with problems of this type is typically gained through education, training, experience and instruction, and solutions to these problems are usually a collection of doctrinal contingency plans. Once the decision maker has deconstructed the problem it simply matches a solution to a problem which has already been encountered (through thought, experience or instruction upon) to the problem currently faced — an overall solution to a problem is therefore *created* by combining the contingency plans of all the respective contingency plans of the subproblems.

Decision. The third phase in the OODA loop is to *decide* upon a feasible course of action which is based on the solution obtained from the orientation phase. If a single plan is derived in the orientation phase, the decision is simply whether or not to implement the decision. If, however, multiple plans have been derived, then the decision becomes more complex — one of the plans must be chosen for implementation. Such decisions may take longer than when the decision is to simply implement the decision or not. If a number of good plans are derived, the focus shifts to having to weigh the risk or cost involved in executing the plans with their potential benefit, which may take time [185]. Similarly, having to choose a decision from amongst a number of poor plans derived, the focus shifts to having to choose the plan with the highest probability of success or choosing the plan with the lowest risk of failure, which may also take time, thereby again rendering the decision phase a complex one.

Action. The last phase in the OODA loop is to implement the decision that was made during the decision phase. This decision may include the action of attacking a threat or issuing an order by a commander or repositioning of sensor systems in order to obtain an improved observation during the next loop of the cycle.

From the descriptions above, it is clear that WA in a GBADS context resides within the orientation phase of the OODA loop of Figure 2.1. For complex problems, FCOs typically break down the problem into a number of smaller subproblems and match solutions to similar problems which they have encountered in the past so as to construct an overall solution to the current problem they are facing.

A very common misconception of the OODA decision loop of Figure 2.1 is that it is sometimes viewed as a single cycle of sequential events. The *Joint Chiefs of Staff* [99] rather described the OODA decision loop as continuous, even implying that multiple concurrent OODA decision loops are executed at any given time. Tighe *et al.* [185] went further to propose the OODA decision process as an OODA cable, rather than an OODA loop. This cable approach contains four seperate intertwined strands of cable (differing in size), spliced together at the OODA loop phases of Figure 2.1. The rationale behind the OODA cable is that it is able to portray the inherent need to filter and consolidate information moving through the OODA decision cycle — thicker cables represent an increased ability to carry information.

Another contribution towards the OODA decision loop of Boyd [20] is the TEWA C2 system proposed by Cramer [41], illustrated in Figure 2.2. Cramer [41] proposed a C2 system which includes "a sequence of activities that continually occur within an organisation's C2 process". The process in Figure 2.2 is initiated by obtaining intelligence and fusing this intelligence together with information from sensor systems in order to develop a sense of awareness of the situation or context in which the decision has to be made. Next, threat lists are compiled and WS-threat assignment pairs are proposed for engagement purposes. The loop closes with damage assessment after which a new loop of the cycle commences with new intelligence and information from sensors, creating a revised awareness picture of the situation. Cramer [41] explained that the nodes in Figure 2.2 may be decomposed further into a number of smaller OODA loops which may result in a further refinement of C2. The idea is that these OODA loops run in parallel and that further, overarching OODA loops are formed on higher levels of decision making.

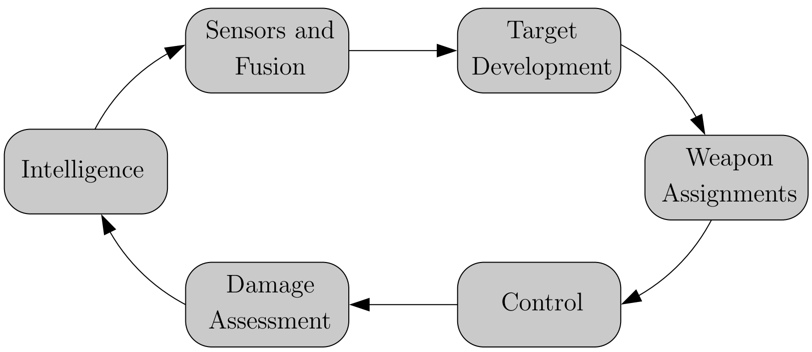


FIGURE 2.2: *An augmented OODA loop decision cycle for C2 [41].*

A question that arises from the theory behind C2 is how C2 may be carried out effectively in missions. According to the *Marine Corps Doctrinal Publication* [147], the antagonist who can maintain a higher *OPerational TEMPO* (OPTEMPO) of actions in any situation of conflict (*i.e.* to cycle through the OODA loop of decisions accurately and consistently at a faster tempo) will gain an ever increasing advantage over the opposing force with each successive cycle completed. The converse of this is also true — the slower antagonist will fall behind with each decision cycle and become increasingly more unable to cope with the situation (*i.e.* C2 itself will deteriorate).

It is therefore evident that speed is a very important factor in executing C2 and the aim should be to generate OPTEMPO in C2 (that is, executing the OODA decision loop faster and faster during each cycle). Boyd [20, 19] reinforced the importance of speed by stating that "in order to win, we should operate at a faster tempo or rhythm than our adversaries — or better yet, get inside (the) adversary's OODA time cycle or loop." He also added that a faster OPTEMPO may cause confusion and disorder within the opposing force's ranks and that the own force aim should be to "collapse the adversary's system into confusion and disorder by causing him/her to over and under react to activity that appears simultaneously menacing as well as ambiguous, chaotic or misleading." The implication of increasing OPTEMPO is that a shorter time frame is available to the own force for planning, making decisions, coordinating and communication. It is here where the provision of effective WA DS may bring relief to the C2 operators. Ultimately, the aim of a WA subsystem should be to aid in reducing operator uncertainty, achieving an increased state of awareness of the battle situation which may result in increased OPTEMPO.

2.3 Situation awareness

The notion of SA is widely described as the perception of environmental elements surrounding a particular situation and understanding these elements. In simple terms, SA may be described as knowing what is going on in one's vicinity and understanding how this information will impact one's goals, both immediately and in the near future. Although widespread adoption of the term SA is relatively new, specialised use of this notion dates back to World War I and has its roots in the domains of air traffic control, aircraft cockpit control, manufacturing process control, military C2, and information warfare [171]. Today, the application of SA may be found in many complex, dynamic areas of decision making, including aviation, air traffic control, ship navigation, power plant operations, military C2 and emergency service applications (such as fire fighting and policing).

In the context of GBAD, SA forms a foundation for the decision-making process (of the decision-making phase in the OODA loop described earlier) as well as for performance of complex, dynamic TEWA DSSs. A number of authors have contributed towards technical definitions of SA. Dennehy and Deighton [54], for example, defined SA as "the ability to maintain the 'big picture' and think ahead," while Dominquez *et al.* [58] defined SA as "the continuous extraction of environmental information along with integration of this information with previous knowledge to form a coherent mental picture, and the end use of that mental picture in directing further perception and anticipating future need." A number of other definitions of SA also exist and although many authors have acknowledged the positive impact that good SA may have on decisions in complex, dynamic systems, no single, universally accepted theory for SA has emerged [166].

A popular and tractable definition of SA (which has been acknowledged by a number of authors, such as Stanton *et al.* [176], Wallenius [204] and Wickens [206]) is the definition by Endsley [66], who defined SA as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." Endsley [66] went further to provide a three-tiered hierarchy portraying various phases (and primary components of each phase) within this definition of SA. The three tiers are:

Level 1: Perception of the elements in the environment. The first and lowest level of achieving SA is to perceive the status, attributes and dynamics of relevant elements in the environment within a volume of space and time. For example, an FCO requires accurate data on the location, type, number, capabilities and dynamics of all opposing and own force elements in the area of responsibility as well as their relationships with other points of reference.

Level 2: Comprehension of the current situation. The second level of achieving SA is based on a synthesis of the disjointed elements contained in level 1 SA. Level 2 SA promotes an accurate understanding of the significance of the level 1 elements in view of pertinent operator goals. Level 2 SA enables the decision maker to form a holistic view of the surrounding environment, discerning the significance of objects and events. For example, an FCO must comprehend that the appearance of, say, three aircraft within close proximity in a specific geographical area within the air picture indicates certain facts about their objectives and motives.

Level 3: Projection of future status. The third and highest level of SA is the ability to project the future actions of level 1 SA elements in the environment — at the very least in the near future. This is typically achieved by obtaining knowledge of the status and dynamics of the level 1 elements and a comprehension of the SA achieved in levels 1 and 2. For example, if an FCO knows that an aircraft is currently offensive while in a certain geographical location, it allows the FCO to project that the aircraft is likely to attack in a given manner and, hence, knowledge and time are afforded to the FCO to decide on the most favourable course of action in order to reach some pre-specified mission objective.

In this dissertation, the assumption is made that SA is considered as a purely cognitive process enhanced by DS and the fusion of data. The TEWA cognitive process is described later in this chapter. Wallenius [204] provided a framework for complete SA specifically in the context of C2. An adapted version of this framework is presented graphically in Figure 2.3.

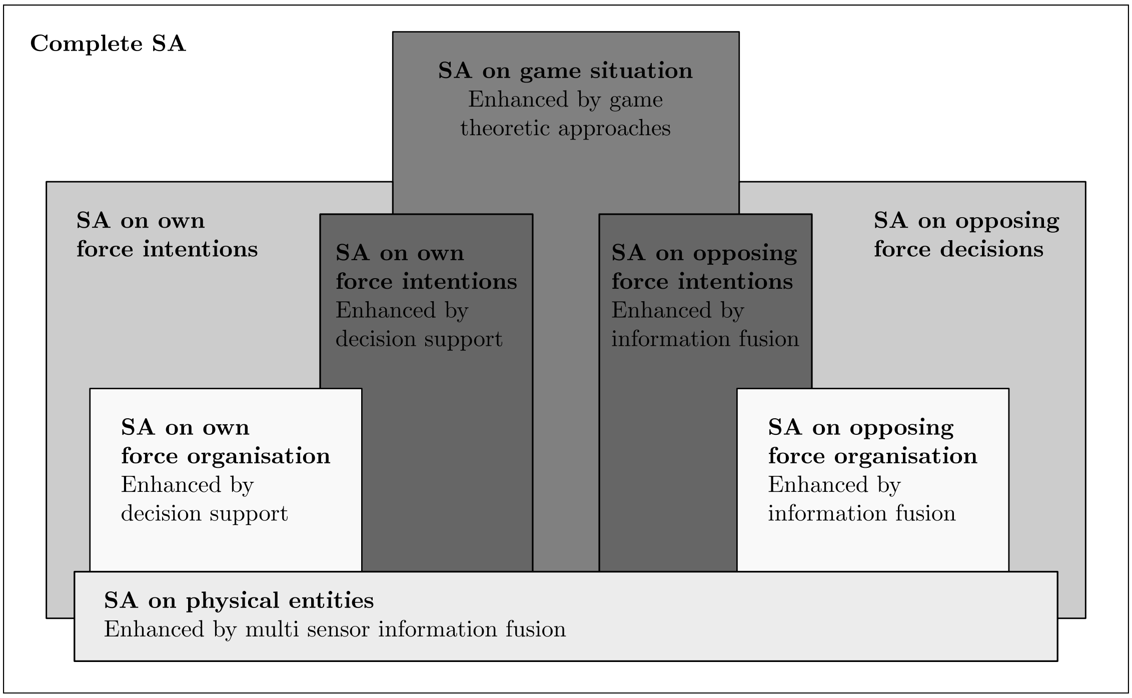


FIGURE 2.3: *A framework containing three levels of complete SA for C2 (adapted from [204]).*

Wallenius [204] proposed that complete SA comprises three levels of SA conforming to the three-tiered SA hierarchy of Endsley [66]. At the lowest level is *SA on physical elements* which entails assessing properties or states of physical elements, such as vehicles, soldiers, WSs and sensors, and to convert this information into useable information through data fusion processes. This lowest level of SA is aimed at aiding the operator during the *observation* phase of the OODA decision loop in Figure 2.1.

At the intermediate level of the framework, Wallenius [204] differentiated between SA of the own force (as displayed on the left-hand side in Figure 2.3) and SA of the opposing force (as displayed on the right-hand side of the figure). SA of the own force is typically provided by means of DS and encompasses decisions on an organisational level pertaining to physical resources on the lowest level and decisions on the intentional level which involve potential plans with goals including the assessed consequences and potential benefits of these plans.

On the other hand, SA of opposing force environments (displayed on the right-hand side of Figure 2.3), entails information in respect of what is known about the organisation and intention of the hostile elements — these decisions are similar to SA of the own force (the left-hand side of the figure), although such information for the opposing force may typically be unknown and inferences with respect to such information may have to be made. A large degree of uncertainty may therefore be present in these inferences due to the actual decisions of the hostile elements being unknown. At the intermediate level of SA in Figure 2.3, the FCO is provided with information that is capable of improving the *orientation* phase of the OODA decision loop of Figure 2.1.

Finally, the most advanced level of SA proposed by Wallenius [204] is *SA on game situation* where the assessment of hostile decisions depends on friendly decisions, and *vice versa*. Here, the notion of *game theory* may typically be employed to contribute towards decision making, by enabling the decision maker to exploit *what-if* scenarios.

From the discussion above, it is evident that the elements or processes involved in TE and WA contribute towards SA to some extent. Although the availability of data may restrict the working of the framework provided by Wallenius [204], an improvement in these processes through SA will evidently improve the overall SA of the FCO and, in turn, improve the OPTEMPO of the system.

2.4 Network centric warfare

NCW[1] is a theory of warfare (developed by the *United States Department of Defence*) that emerged during the 1990s. It involves the study of human and organisational behaviour based on adopting a new method of thinking and applying it to operations within a military environment [5]. The focus in NCW operations is on generating combat power from the effective linking and networking of own force elements. NCW is typically characterised according to the ability of the geographical deployment (or dispersion) of own force elements of being able to achieve a high level of collective battle space awareness. This may then be exploited by means of self-synchronisation and operations within NCW to achieve favourable outcomes during a mission. Recent advances in sensor technologies, the alarming rate at which data are distributed across the globe, and the ever-increasing computational power of computer processors has led to what is currently known as the *information age* [80]. This new age facilitates an increased ability to generate, organise and distribute information by employing sophisticated modern sensor technologies and information structures. The underlying theory emerging from the information age is that military forces may achieve improved SA by exploiting the benefits that this volume of information has to offer. This has formed the stepping stone from which the concept of NCW has emerged.

[1]NCW is sometimes also referred to in the literature as *network centric operations* or *net centric warfare* [207].

NCW may be thought of as an inter-woven system (network) of sensors, WSs, decision makers and information — an effective whole system so as to create a decisive war fighting advantage [18, 31]. This concept is based on the hypothesis that combat effectiveness may be improved greatly when warfare is conducted in a networked environment rather than separately on a collection of individual platforms [18]. Many definitions of NCW exist in the military operations research literature. There is, however, no single, universally accepted definition for the concept of NCW. Dahl [44] wrote as follows about the concept: "NCW may be broadly defined as deriving power from the rapid and robust networking of well-informed, geographically dispersed military resources. Thereby creating a supreme tempo and precise, agile form of manoeuvre warfare. NCW focuses on operational and tactical warfare, but can even affect the strategic domain. It is the dominating theory of war for the information age." Raduege [148] added that "NCW is not just about technology; it is an emerging theory of war and the next art and science of warfare to be exploited."

An important prerequisite for effectively incorporating NCW into military doctrine is a thorough understanding of the interaction of the physical, information, social and cognitive domains of warfare. These domains are now described in some detail.

The physical domain. In this domain, warfare operations, such as striking, protecting and manoeuvring, are conducted within the tactical environment consisting of land, sea, air and space. The physical elements (*e.g.* WSs and sensor systems) and the communication infrastructure which connects them all reside within this domain. Therefore, the challenges that arise from employing and exploiting the elements in this domain are a result of the laws of physics. In comparison with the other domains, the performances of the elements contained in the physical domain are the easiest to measure and comprehend. Consequently, combat power has traditionally primarily been measured according to the nature and number of the elements in this domain [31].

The information domain. In this domain, information is created, collected, stored and processed within the network. It is also here where communication and information sharing are facilitated among war fighters. The C2 of modern military forces are therefore communicated and the commander's intent is also conveyed among war fighters in this domain [4, 18, 31, 32]. As a result of a recent increase in cyber-warfare, the information domain, in particular, is becoming more vulnerable to hostile attacks [207]. It is therefore extremely important to protect and defend the information domain so as to enable the generation of combat power in the face of offensive actions by an adversary in order to ensure sustained information superiority [31].

The cognitive domain. This domain constitutes the collective minds of the war fighters — it is here where many battles, campaigns and wars are won or lost [18, 31]. Elements within this domain include the intangibles of leadership, morale, unit cohesion, level of training and experience, SA and public opinion. According to Cebrowski [31], it is also in this domain where the commander's intent, as well as doctrine, tactics, techniques and procedures reside — decisive battle space concepts and tactics emerge in this domain. One of the challenges associated with this domain is the difficulty of measuring the attributes contained therein. It is for this reason that explicit treatment of this domain by means of analytic models of warfare is rare [32]. Herman [84] has, however, proposed a methodology that addresses key attributes and relationships between them in the context of *entropy-based warfare.*

The social domain. In this domain, the necessary elements of any human enterprise are described, including the ways in which humans interact, exchange information, form shared awareness and understanding, and make collaborative decisions. In addition, other elements, such as culture, the set of values, attitudes and beliefs held and conveyed by leaders to society, whether military or civil, also reside within this domain. This domain overlaps with the cognitive and information domains, although distinct differences between these domains do exist. Naturally, cognitive activities are individualistic, implying that they occur in the minds of individuals. On the other hand, the social domain is, in essence, shared sense-making and involves the process of going from shared awareness to shared understanding to making collaborative decisions. The social domain is concerned with socio-economic activity since an individual's cognitive-related activities are directly influenced by the social nature of the exchange and *vice versa* [31].

Based on the individual discussions of each of the domains above, it is evident that these domains interact and intersect with each other to form a universal concept of NCW. The four domains, as well as their intersections and the interactions between them, are presented graphically in Figure 2.4. According to Cebrowski [31], the intersections between these domains should be considered important since they represent dynamic areas within which concept-focused experimentation should be conducted.

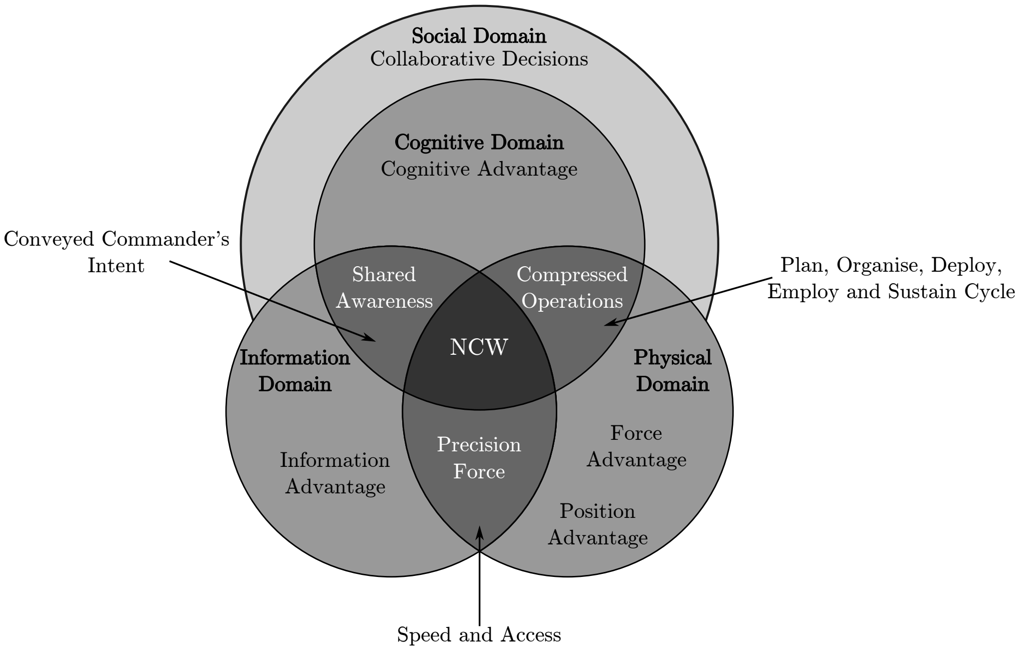


FIGURE 2.4: *The physical, information, cognitive and social domains of NCW, as well as these intersections between the domains [31, 159].*

Consider the intersection between the information and physical domains in Figure 2.4. This intersection is referred to as the *precision force* and is vital to conducting successful joint operations. The intersection between the information and cognitive domains creates *shared awareness* and *tactical innovation*. This is an important intersection since, as mentioned earlier, many battles are won or lost in the cognitive domain. Finally, the intersection between the cognitive and physical domains creates *compressed operations* which include the phenomena of time compression and "lock-out[2]." It is in this intersection where tactics achieve operational (and even

[2]According to Alberts [4], lock-out refers to the situation in which an adversary's strategic objectives have been *locked out*, because it has no remaining viable courses of action — the opposing force is no longer able to react coherently.

strategic) effects as well as high rates of change — the ability to quickly change from one rapid operation to another. At the centre of all four of the domains lies NCW as illustrated in Figure 2.4.

The four-domain NCW construct builds on an earlier model proposed by Fuller [71] in 1925. It was later refined in 1999 in a paper by Booz [18] to the structure presented in Figure 2.4. The importance of these four domains and their contributions towards NCW have been highlighted by many authors in the military literature [5, 18, 32, 31]. When considering the TEWA process described in §2.1, it is clear that this process extends over all four the domains of NCW. For example, information from the sensor systems (part of the physical domain) is used to create and share DS information (part of the information domain) which conforms to operator thought and collaborative thought (part of the cognitive and social domains) [159]. Similarly, data emanating from TE (part of the information domain) and WSs (part of the physical domain) are used to create and share DS information (part of the information domain) which also conforms to operator thought (part of the cognitive and social domains). Cebrowski [31] proposed a logical framework for the notion of NCW and Roux [159] slightly adapted this framework. The adapted framework is presented graphically in Figure 2.5.

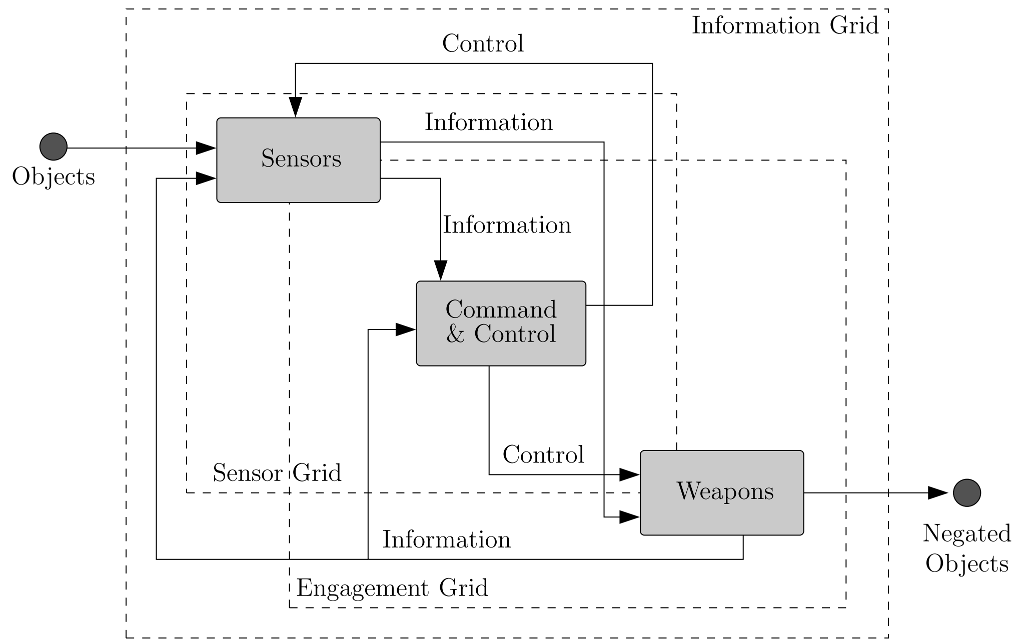


FIGURE 2.5: *A first-order, logical framework for NCW in the context of the TEWA process [31, 159].*

The adapted framework features the network centric flow of information between sensor systems, C2 (the FCO) and WSs. Three substructures are present in the framework:

(1) **The information grid.** This infrastructure facilitates receiving, processing, transportation, storage and protection of information of a joint force. The embedding capabilities for information assurance prevent intrusive attacks and provide commanders with the assurance that their information is valid and uncompromised [18, 158].

(2) **The sensor grid.** This infrastructure provides a high degree of awareness of the various elements of the own force and opposing force, as well as the tactical environment across the battlefield [18, 158]. Elements residing within this infrastructure include dedicated sensors,

WS-based sensors, man portable sensors and logistics sensors which are embedded in the air, land, surface, subsurface and space environments.

(3) **The engagement grid.** This infrastructure provides a means for war fighters to utilise speed of command and achieve overwhelming effects at precise places and times [18, 158]. Elements residing within this infrastructure include air, land, surface, subsurface and space WSs.

Lately, two distinct views have been formed in respect of the effects of NCW within the context of modern warfare [6, 80]. The first view is considered conservative where NCW is simply seen as the evolution that accompanies the digitisation of the process of conducting war, while the second view is an extreme perspective in the sense that the information age is revolutionising the nature of conflict in a fundamental manner.

The first view above implies that only small, incremental improvements towards modernising current technologies are required to suit the information age. Alberts and Hayes [6], however, argued that this perspective may result in NCW not reaching the full potential that it can add to success on the battlefield. They suggested that a more introspective approach be followed where a more intensive and intentional change is enforced. This disruptive transformation will require that current concepts within C2 (lying very close to military operations) will have to be reconsidered and redeveloped.

Smith [172] suggested three concurrent technologies which may be very beneficial to NCW, if exploited. These technologies include advanced sensor technology, an increase in computing power and more sophisticated WSs. Smith [172] further advocated that these technologies should interact in conjunction with each other in such a way that they create novel emergent properties in order to reach an efficient state of NCW. If this can be achieved, it may change the characteristics of war altogether. The process of revolutionising military forces for the information age is currently the prevailing idea in NCW [80, 172, 207].

From the arguments reviewed above, it would seem that NCW is a central focus in modern-day improvements of combat capabilities. Moreover, the advancement in technology since the turn of the 21$^{\text{st}}$ century provides a means to achieve the desired levels of NCW capabilities, greater sharing of improved information (in real time) and overall improved SA [80]. It is therefore extremely important that the developers of any military DSS, such as a TEWA system, incorporate the concepts of NCW in their design.

2.5 Elements of a GBAD system

The elements of a GBAD TEWA system may be subdivided into three distinct groups *i.e. the physical elements*, *the functional elements* and *the cognitive elements*. The physical elements contain all the hardware elements of a TEWA system. According to Roux and Van Vuuren [161], the physical elements may be identified by considering a typical AD scenario in which an area of responsibility within a *tactical environment* is observed. Such an observation of the tactical environment is typically facilitated by means of various types of *sensors*. An attack on the own force typically constitutes an enemy pilot flying a specific *aircraft* (called a threat from the own force perspective) with a view to execute a specific *attack technique* (also known as a flight profile) in order to deliver a specific type of *weapon* in an attempt to damage or destroy a specific *asset* on the ground. Furthermore, these orchestrated actions are typically executed according to a pre-defined *tactic* by enemy commanders. In the case where a number of enemy aircraft are present in the area of responsibility, specific *flight formations* are typically used.

The functional elements, on the other hand, contain all the software elements of a TEWA system which link the physical elements in some way and provide a means for providing improved DS to FCOs. Roux [159] proposed consideration of a number of functional elements in a GBAD TEWA system. These functional elements are also considered in this dissertation and include operator DS, testing and training, effecting and sensing, *Track Management* (TM), maintenance and testing, remote system processing, map processing, TEWA data management, *Attribute Management* (AM), TE, *Engagement Quantisation* (EQ), and WA. The aforementioned physical, functional and cognitive elements are discussed in some detail in this section.

2.5.1 Physical elements

The physical elements of the system comprise the actual hardware elements of the TEWA system and include sensor systems, defended assets, aerial threats to the system, and ground-based WSs. Although the tactical environment, attack techniques and flight formations of enemy aircraft do not constitute actual physical elements, they have an effect on some of the physical elements and hence are also discussed here.

The tactical environment

The natural spatial volume in which a GBADS operates is known as the *tactical environment*. This is the space in which own forces combat opposing forces during battle. The tactical environment potentially consists of five subenvironments, namely space, air, land, surface and subsurface, as observed by Roux [159]. These five subenvironments are illustrated graphically in Figure 2.6.

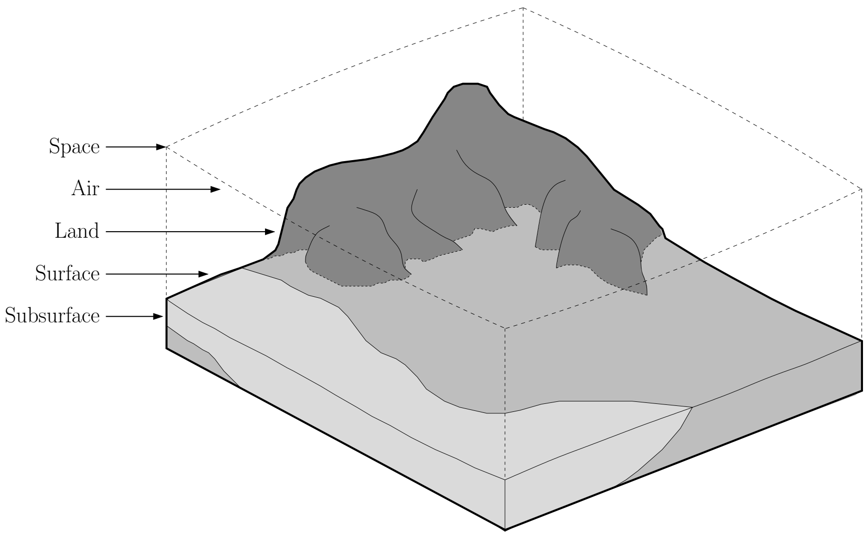


FIGURE 2.6: *The five subenvironments of the tactical environment in which a GBADS operates [159].*

The *space* environment refers to the three-dimensional space beyond the air environment which contains satellites and affords spacecraft a substantial volume for manoeuverability. The space environment is, however, primarily dominated by satellites which may be classified into three broad categories, namely satellites explicitly and specifically employed for military use, satellites officially recognised as dual-use satellites[3] and satellites which are classified as civilian

[3]In the context of military operations, the term dual-use satellites refers to satellites employed for military

satellites [152]. Satellites specifically dedicated for military use include early warning systems, military communication satellites and technology demonstrators, while dual-use satellites include navigation and reconnaissance spacecraft[4]. Civilian satellites include navigational satellites (which facilitate global position systems) and weather satellites, but according to the Yearbook on Space Policy of 2008/2009 [152], this category of satellites is debatable since it is suspected that a substantial portion of these satellites serve to achieve both civilian and military goals.

The use of space in a military context is limited to sensing [159], since international laws exist which place limitations on the military use of space. One such law is the *Outer Space Treaty* of 1967. This law contains an undertaking not to place in orbit around the earth, install on the moon or any other celestial body, or otherwise station in outer space, nuclear or any other weapons of mass destruction. Furthermore, it limits the use of the moon and other celestial bodies exclusively to peaceful purposes and expressly prohibits their use for establishing military bases, installations or fortifications, testing weapons of any kind, or conducting military manoeuvres[5] [193].

The *air* environment also provides aerial vehicles with a three-dimensional space in which they may operate. The nature and dimension of the air environment offers aerial vehicles a considerable amount of space for manoeuvreability. This environment is largely dominated by aircraft of the opposing force in a GBADS context and may include *electronic warfare* (EW) platforms, unmanned aircraft and manned aircraft such as fixed-wing or rotary wing aircraft. WSs operating in the air environment consist mainly of missile systems [159]. The aforementioned aircraft and WSs operating in the air environment are considered GBAD threats and are therefore considered in more detail later in this chapter.

The *land* environment affords vehicles a two-dimensional space for operational purposes. This environment contains a number of so-called *terrain features* which may restrict the manoeuvereability of vehicles and WSs. Terrain features may also restrict the manoeuvereability of vehicles and WSs operating in the air environment. The terrain features of the land environment is usually categorised into five different geographical environments, namely mountains, jungles, deserts, arctic areas and urbanised terrain [73].

The *surface* environment refers to the surfaces of dams, lakes, oceans and rivers, and also provides vehicles with a two-dimensional space for operation. These vehicles consist mainly of watercraft, including aircraft carriers, amphibious warfare craft, coastal defence craft, combat logistics craft, mine warfare craft, mobile logistics craft, patrol combatant craft, service and support craft, and surface combatant craft [131]. Watercraft may be classified into the different aforementioned categories by means of a so-called hull classification symbol[6] which represents the type of hull these watercraft bear.

The *subsurface* environment refers to the space beneath the surface environment and offers a three-dimensional submerged space to submarine craft. Submarines are watercraft capable of operating independently when submersed and can be propelled forward by means of diesel-electric propulsion, air-independent propulsion[7] or nuclear power [182]. In a military environment, submarines may be categorised into two major classes, namely attack submarines and ballistic

purposes as well as civilian purposes, whether proven or potential [7].

[4]Reconnaissance satellites, also known as *spy satellites*, refer to observation or communication satellites employed to obtain intelligence on the military activities of foreign countries [137]. The full capabilities of these satellites are usually unknown, since countries which launch these satellites tend to keep any information with respect to their reconnaissance satellites classified [136].

[5]The reader is referred to [94] for an extensive list of other laws on the limitation of the usage of space.

[6]The reader is referred to [131] for a full discussion on the various types of available hull classification symbols.

[7]Air-independent propulsion refers to the ability of a submarine to be propelled forward without the need for atmospheric oxygen [182].

missile submarines [37]. Another class of submarines exists, known as *bathyscaphe* submarines, which is a classification of submarines having the ability to reach great depths.

A number of possible physical conditions, such as *terrain* and *weather elements*, reside within the tactical environment which may have a significant effect on the physical elements within a GBAD environment. Not only do these physical conditions affect operations and systems, but also the personnel operating these systems [74]. Examples of weather elements include cloud cover, electrical thunder storms, humidity, precipitation, temperature, visibility, wind direction and wind speed [56]. According to the United States Army's Field Manual FM34-81-1 [74], weather tactical decision aids may be employed to assist commanders to make decisions with respect to operations in different weather conditions. Weather tactical decision aids usually consist of simple two-sided matrices or lookup tables, which may be used to assist in pairing a specific weather condition with its related impact on the battlefield [74].

Although adverse weather conditions may have a critical impact on battlefield operations, it may also be beneficial in, for example, aviation operations. The United States Army's Field Manual FM1-100 [73] gives examples of fallen snow, extreme cold weather or muddy terrain which may restrict the manoeuvreability of ground personnel, while minimally affecting aviation operations. Low cloud cover and ceilings may restrict high-performance aircraft operations, while simultaneously enhancing aviation operations by providing low-level cover and concealment.

Furthermore, the environmental conditions affecting the tactical environment may be divided into *natural environmental conditions* and *induced environmental conditions*. Natural environmental conditions typically refer to the weather conditions described above, while induced environmental conditions refer to dust, smoke and debris on the battlefield, typically caused by encounters with the opposing force. Induced environmental conditions are, therefore, much harder to predict than natural environmental conditions [159]. Hence, induced environmental conditions are omitted from all analyses in the remainder of this dissertation.

It is crucial to gather a substantial amount of weather intelligence by means of weather forecasts, weather observations, climatological data, and atmospheric and astronomical information affecting radar and wireless communications [56] on a continual basis for use in all GBAD deployment, planning and tactical operations. Ignoring the effects of weather conditions on the tactical environment may lead to an inefficiency of battlefield operations [74], resulting in the vulnerability of the own force.

Considering the above-mentioned discussions, it is evident that the tactical environment and the physical conditions that prevail within this environment should be examined carefully and thoroughly in the formulation of GBAD DS, since terrain types and weather conditions may have a significant impact on the physical elements such as DAs, sensors, WSs and attack techniques employed by enemy aircraft. Furthermore, the modelling of the tactical environment is extremely important when investigating potential threats entering the defended airspace [159].

Defended assets

In any military tactical situation there are typically a number of assets which require protection from opposing forces. This gives rise to the notions of *critical assets* and *DAs*. According to [189], critical assets may be defined as "*all* those assets identified for possible AD by the supported force commander and his supporting staff during the planning/appreciation process that is deemed/foreseen as critical in the achievement of the force's desired end state." Typical examples of critical assets include main bridges on the main supply route of the own force, a fuel depot or an electricity supply source. According to Oerlikon Military Products [139], critical assets may be divided into the following types: air bases (including runways, aircraft protected by shelters, unprotected aircraft, fuel depots, hangars, buildings and ammunition depots), oil refineries (including oil tanks and installations), railway installations (including railway installations, trains, signal boxes and buildings), harbours (including ships and installations), crossroads, main bridges, combat bridges, power plants, factories, field fortifications, command posts, gun and troop emplacements, anti-aircraft defence, early warning and allocation radars, soft skinned vehicle accumulations and armoured vehicle formations. Note that critical assets are either fixed installations or moving objects.

Prior to a mission, a list of critical assets are typically compiled by the own force commander. These assets are then prioritised by assigning a priority value to each. The idea behind prioritising assets is that more important assets should be afforded better protection than lesser important assets. Chastain [34] introduced the concept of asset prioritisation in a way that identifies the general attributes to be addressed when prioritising assets.

Once the own force commander has compiled a list of critical assets, some subset of these assets are selected to constitute the set of DAs requiring protection against opposing forces. The rationale behind a reduction in the number of assets which should be considered for protection is that it may result in a reduction in reaction time (which may lead to an improved OPTEMPO) and more importantly, because there are typically a limited number of AD resources available for the protection of assets [159]. According to the South African National Defence Force *Field Pamphlet Number 2* [189], the task of prioritising critical assets and identifying DAs is not a trivial task. The task of prioritising threats is typically done manually and the own force commander should take into account elements such as the mission objectives, the characteristics of the opposing force, terrain features, IPB results, doctrinal rules, and guidelines for operations to establish critical asset priority values. Priority values may, of course, also be obtained by including certain underlying attributes of the assets themselves. From the prioritised list of critical assets, the own force commander may then choose a subset of assets achieving the highest priority values as the set of DAs.

According to Roux [159], own force commanders are typically involved in extensive planning procedures before assigning asset priorities to critical assets within the South African AD context. These procedures involve reconnaissance, site visits to possible asset positions and an intensive study of aerial photographs and maps of the area of responsibility. Only thereafter is a priority value derived by taking the following three aspects into account, namely *Vital Importance* (VI), *Vulnerability* (V), and *Repairability* (R):

Vital Importance (VI). The degree to which an asset is critical in terms of achieving mission accomplishment. One way of measuring the VI of a DA is to estimate the impact that damage to or destruction of the asset may have on the success of the mission — usually the damage to or destruction of vitally important assets is of such significance that mission objectives may not be achieved. Within the context of the South African AD, the following classification scheme is used for assessing the VI of DAs [189]: (1) Not important = 0, (2) Important = 2 and (3) Vitally important = 4.

Vulnerability (V). The degree to which an asset is susceptible in terms of surveillance and/or attack or the degree to which damage may be inflicted to the DA should it be attacked. In contrast with VI, damage or destruction of vulnerable assets will typically result in having a detrimental effect on achieving mission objectives, but will not result in the mission objectives being unaccomplishable. In determining the V of an asset it should be kept in mind that certain assets may be deployed in hardened casings or may be hidden (resulting in the asset being less vulnerable), while others are required to be in the open (resulting in the asset being exposed and therefore more vulnerable). Nevertheless, within the context of the South African AD, the following classification scheme is used for assessing V of DAs [189]: Not vulnerable = 0, Vulnerable = 1 and Very vulnerable = 2.

Repairability (R). The degree to which an asset can recover or be restored to a working condition (*i.e.* to continue functioning properly during the mission) after damage has been inflicted, in terms of time, available equipment and manpower required for such recovery. Within the context of the South Africa AD the following classification scheme is used for assessing the R of DAs [189]: Easily/Quickly repairable = 0, Difficult/Long time to repair = 1 and No repair possible/Needs replacement = 2.

The *Priority* (P) of a critical asset may then be determined by

$$P = VI + R + V. \tag{2.1}$$

Although this formula seems to be sufficient for determining priority values for critical assets within a South African AD context, the formula is not scientifically justifiable — the three aspects are not measured in any specific units and it is not mathematically sound or justifiable to add the values together or to subtract the values from each other [159]. Nevertheless, the method is acknowledged by South African military experts and also considered sufficient for prioritising critical assets [63]. It also provides some understanding of the problem of prioritising assets and illustrates the complexity associated with developing a mathematically sound method for quantifying the prioritisation of assets.

In 2005, du Toit [63] developed an alternative, novel method for prioritising assets in an attempt to improve on the formula (2.1). His method was based on the formulation of a mathematical performance index which is able to portray the success of a porftolio manager according to personal characteristics, developed earlier by Wagenaar and Van Vuuren in 1999 [203]. Du Toit [63] implemented his method in a pilot study (in conjunction with a number of military experts) comprising three stages. The first stage involved identifying the preferences of a core group of military experts with respect to similarities and differences between asset attributes *in terms of achieving mission success*. This was achieved by conducting interviews with the military experts and using the well-known *Repertory Grid method*[8]. The second stage involved the completion of a questionnaire by a larger group of military experts. In this questionnaire, the military experts were required to rate a number of assets with respect to the asset attributes identified during the first stage. Finally, during the third stage, an index for asset prioritisation was derived. This was achieved by including those asset attributes that were deemed statistically significant from the output obtained during the second stage. When a *new* critical asset is therefore added to the list of existing critical assets, a priority value may be determined for the asset by simply using the established index.

Although the pilot study of Du Toit [63] seemed to yield plausible results (as stated by the military experts), Roux [159] claimed that a proper execution of this prioritisation method is required, implying that the method should be carefully calibrated and that the interviews should be conducted with a sufficient number of military experts.

For the purpose of this dissertation, only fixed DAs are considered and it is assumed that the own force commander has already compiled the critical asset list and that he has also selected a

[8]The Repertory Grid method is a method for eliciting personal constructs and was originally developed by Kelly in 1955 for use in psychological analysis [178]. It has since been widely used in the study of consumer preference. The method is based on the identification of elements and the comparison of attributes associated with these elements by means of the formulation of bipolar constructs [203].

number of these assets as DAs by prioritising them according to some preferred method — for the purposes of this dissertation, it is therefore assumed that the DAs are given.

Sensor systems

The physical elements used to observe the tactical environment are sensor systems. The goal of such observation is twofold, namely to detect any new aerial objects that are entering the defended airspace and to monitor the objects that have already been detected in the defended airspace. The sensor systems may be thought of as the “eyes” of the system and are able to detect and monitor the altitude, direction, and speed of moving aircraft. Roux [159] goes as far as to claim that the sensor systems form the core of the TE decision making process. The accuracy and update rate (*i.e.* the quality and quantity) of sensor data may indeed have a vast influence on the quality of the results provided by TE models, especially on the more complex stochastic TE models [158].

In the context of NCW (discussed in §2.4), sensors are typically deployed in a network called a *sensor grid*. The goal is to ensure that at least one of the sensors in the network covers every significant portion of the defended airspace. The sensors employed in such a grid may include thermal (*e.g.* infra-red sensor), electromagnetic (*e.g.* radar), mechanical (*e.g.* position sensor), optical radiation (*e.g.* light sensor) and acoustic (*e.g.* sonar) sensor systems [87]. The sensors employed in a GBADS typically consist of two-dimensional surveillance (search) radar systems and the characteristics associated with such a system are given in Table 2.1.

	Update rate	Range	Range accuracy	Azimuth accuracy
Long range	10 seconds	200+ km	100–500 m	0.5–2 degrees
Medium range	2–4 seconds	50–200 km	20–100 m	0.1–0.5 degrees
Short range	≤ 1	0–50 km	5–20 m	0.1 degrees

Table 2.1: *Characteristics associated with a two-dimensional surveillance (search) radar system [210].*

It is important to acknowledge that in modern-day GBAD, the capabilities and characteristics of sensor systems employed should be considered in detail. The information provided in Table 2.1 represent typical ranges for radar sensors from a global perspective. When considering these ranges from a South African perspective, Oosthuizen [140] states that the update rate, range accuracy and azimuth accuracy values are correct, but that the range values have to be adjusted. He suggests that the range values should be lowered to 100+ km for long range radar sensors, lowered to 20–100 km for medium range radar sensors and lowered to 5–20 km for short range sensor systems. These suggested changes are in line with the capabilities of the sensors employed within the South African AD and the three sensor classes of low-, medium- and short range directly relate to the sensor classes of early warning, local warning and WS/target engagement sensors, respectively [159].

The aforementioned three classes of sensors may be used as general sensor layers to provide a comprehensive SA picture on the various levels of C2 which may, in turn, result in an increased level of SA for various decision makers. According to the South African National Defence Force *Field Pamphlet Number 1* [188], “the SA picture consists of an in-time air picture and ADC information, which is displayed to aid FCOs in operational and tactical level decision-making” — the SA picture contains information with respect to the entire aerial picture (*i.e.* threats to the system and own force aircraft) as well as the position and other attributes of other physical elements (*i.e.* DAs, sensors and WSs). The systems responsible for obtaining the SA picture may therefore be thought of as the “eyes” of the FCO and their results are typically displayed graphically to the FCO on an HMI [159]. For the purpose of this dissertation, it is assumed that an all-encompassing SA picture is available to the FCO. This dissertation is therefore not concerned with *how* this SA picture was obtained.

Four different sensor classes exist within the South African AD [190]. These classes are optic, electro-optic, infra-red and radar sensors. Although it is important to include all these classes of sensors within a NCW environment, only radar sensors are considered in this dissertation. A number of different types of radar sensors exist. When having to choose between these sensors, a trade-off between effective range, range accuracy and update rate typically has to be considered. Surveillance sensors, which have a three-dimensional capability, brings the third dimension of *elevation* data as input to the TE subsystem. Such sensors typically achieve elevation accuracy between 0.1 and 5 degrees within elevation angles of 0–30 degrees. Elevation angles of 30–75 degrees are less common and only in exceptional cases are elevation angles larger that 75 degrees achieved. In addition, short to medium range sensors are able to detect threats up to an altitude of 2 000–3 500 metres, while longer range sensors are able to detect threats up to an altitude of 30 000 metres [210].

Two types of sensor systems typically used within a GBAD scenario are search and track radar systems. Search radar systems act as early warning systems aimed at providing early warning of enemy aircraft approaching as well as providing accurate initial positions of these threats for use by track radar systems. Such sensor systems typically have slow refresh rates, since they have to scan a wide area. Track radar systems, on the other hand, are dedicated to track only a single object or a small number of objects at a time at relatively small effective ranges (typically 15–20 km). Furthermore, these systems are able to achieve high levels of accuracy (range accuracies of up to 5 metres and azimuth accuracies of 1 up to milli-radian) as well as fast update rates (up to once every 20 milli-seconds).

Apart from searching and tracking targets, the *target radar signature* may be employed to aid in the hostility and platform type classification of aircraft. Techniques typically employed to achieve this include non-cooperative target recognition techniques [42, 154] and jet engine modulation[9] based target identification techniques. It should be noted that these techniques typically require short range surveillance or tracking radars and long target dwell times.

It is very important that high-quality data be used as input to a TEWA system. This quality depends on the type of radar sensor used, but may deteriorate due to internal or external interferences caused by unwanted signals. Internal interferences are typically caused by signal noise, while external interferences are typically caused by clutter (*e.g.* actual radio frequency echoes returned from other threats, multipath echoes from the related threat due to ground reflection, atmospheric ducting or ionospheric reflection/refraction) [159].

In the context of the South African GBADS, the kinematic data that sensor systems are able to obtain about aircraft are known as aircraft *attributes.* In the case where multiple radar sensors are used, the values are typically fused together to obtain a single value for a specific attribute. Roux [159] describes the notion of attributes as “the various factors or variables necessary for effective evaluation of, and assignment of WSs to the threat.” He distinguishes between two classes of attributes, namely *measured* and *derived* attributes.

Measured attributes are obtained from sensor systems; examples of these attributes include the *altitude, position, speed* and *course* of aircraft. The values associated with these attributes are typically measured with respect to some global reference point and are associated with two parameters, namely the index or identification number of the aircraft in question, and a time

[9]The jet engine modulation phenomenon stems from radar returns caused by the rotating structure of jet engines [13].

stamp associated with the value measured [159]. For each aircraft, a so-called *system track* is created which contains a series of measured attribute values associated with successive time stamps for each aircraft which enters the defended airspace. This is achieved in the TM which is described in more detail later in this chapter when the functional elements of a GBAD system are considered.

Derived attributes are (as the name suggests) derived or computed from combinations of measured attributes for each DA; examples of these attributes include *range* to the DA or the *bearing* with respect to the DA. These values are computed using the latest (real-time) value of a combination of measured attributes at a specific time [159]. Derived attributes are associated with four parameters, namely the index or identification number of the aircraft in question, an index or identification number allocated to the DA, the time stamp related to the attribute and a number of historical values of the derived attribute (if available).

Aerial threats to the system

Aerial threats to the system are those aircraft (of the opposing force) which enter the defended airspace with the intent to inflict damage to own force DAs. These threats are highly manoeuverable and their aerodynamic movement may typically be described by means of *six-degrees-of-freedom models* which take into account the following six degrees of freedom of a rigid body: (1) moving up or down (heaving), (2) moving left or right (swaying), (3) moving forward and backward (surging), (4) tilting up or down (pitching), (5) turning left or right (yawing) and (6) tilting side to side (rolling) [159]. Although the manoeuverability of an opposing force aircraft depends strongly on the actual aircraft employed and the capabilities of the pilot, these threatening manoeuvres are typically restricted by obstructions in the land environment and weather conditions.

There are a vast number of aircraft available today which may be utilised to attack own force DAs. These aircraft may be divided into the following categories: missiles, EW platforms, unmanned aircraft and manned aircraft (which includes fixed wing and rotary wing aircraft). The choice of which type of aerial threat to employ by the opposing force is not a trivial task since the various categories of threats each embodies a variety of aircraft and weapon characteristics including attack speed, attack density, effective range, typical missions (objectives) and targets, weapons carried (ordnance and armament), associated attack tactics, radar signature and physical size, environmental constraints and economics [159]. In addition, a number of other factors may influence the choice of craft, weapon or attack technique employed by the opposing force. These factors include minimisation of the cost of a mission, maximisation of human survivability, own force doctrine, similar previous encounters and personal preferences of the commanding officer. The manoeuvres executed by the aircraft of the opposing force and its doctrine are almost impossible to ascertain. Furthermore, the results returned by the TE subsystem depend heavily on the quality and availability of data pertaining to the aforementioned factors [175]. The complexity involved in the analysis of system threat categories therefore necessitates considering only a small subset of threats. The scope of the aerial threats considered in this dissertation is therefore limited to include fixed wing aircraft only. Although delimiting the scope of the dissertation to include only fixed wing aircraft, it is important to note that a WA subsystem should be able to include all threat categories listed in this section. Therefore, the WA models proposed later in this dissertation are formulated in such a way that they are generic and versatile in order to ensure that no or minimum changes are required when other threat categories are included.

Fixed wing aircraft may be divided into three distinct groups: fighter (or bomber) aircraft, reconnaissance aircraft and transport aircraft (or tankers):

Fighter/Bomber. This category of fixed wing aircraft is considered the main threat against which DAs have to be protected. These types of fixed wing aircraft typically attack in groups (also referred to as the attack density and are typically limited to six aircraft within two minutes [212]), employ specific flight formations and execute specific attack techniques to deliver their WSs. They are categorised by their speed, which may fall within the range 200–300 ms^{-1} (depending on the armament carried[10]), and their low-altitude approach in order to avoid detection by radar sensors. In order to ensure effective neutralisation of this kind of fixed wing aircraft, the TEWA system should facilitate neutralisation of such an aircraft before it delivers its WSs. Examples of fighter (or bomber) fixed wing aircraft include: F-14 Tomcat Standard Fleet Fighter Aircraft (USA), MiG-29 Fulcrum High Performance Combat Aircraft (Russia), Mirage 2000 Multi-Role Combat Fighter (France), B-1B Lancer Strategic Bomber (USA) and B-2 Spirit Stealth Bomber (USA) [159].

Reconnaissance. This category of fixed wing aircraft is typically used for reconnoitering important areas and the surrounding defences. They are equipped with various camera systems in order to obtain vertical, forward, sideways and panoramic photographs[11]. These fixed wing aircraft are categorised by their ability to fly at high altitudes and at speeds within the range 150–300 ms^{-1}. Typical armament carried by these aircraft include guns and cannons (although this is only for self protection). Examples of reconnaissance fixed wing aircraft include: Nimrod MRA4 Maritime Reconnaissance Aircraft (UK), U-2 High Altitude Reconnaissance Aircraft — Dragon Lady (USA) and E-2C Hawkeye Airborne Early Warning Aircraft (USA).

Transporters/Tankers. This category of fixed wing aircraft is typically used to transport personnel and materials (or goods) into the combat zone. They are categorised by their large size, low level of manoeuvrability and low speed which falls within the range 100–200 ms^{-1}. These characteristics make this type of fixed wing aircraft extremely vulnerable, especially during landing and take-off. They also carry no WSs — protection is typically provided by accompanying fighter aircraft or attack helicopters [159]. Examples of transporter (or tanker) fixed wing aircraft include: AN-124 Condor Long Range Heavy Transport Aircraft (Russia), C-130J Hercules Tactical Transport Aircraft (USA) and V-22 Osprey Medium-Lift, Multi-mission, Tilt-Rotor Aircraft (USA).

The armament carried by opposing force fixed wing aircraft usually depends on the type and size of the DA of the own force and may be divided into a number of categories [159]:

Guns/Cannons. These WSs are known for their ability to produce small dispersions of ammunition and achieving high hit probabilities, and are typically used to attack soft DAs. Their ammunition is categorised by high-explosive/incendiary or armour-piercing/incendiary, high initial velocity and heavy impact energy, and a short delay fuse enabling the ammunition to pass through the outer case of the DA [159]. Examples of guns and cannons include: the 20 mm M-61 VULGAN Gun Vulgan (USA), the 23 mm GSh-23 Guns (USA) and the 30 mm DEFA Cannon (France).

[10]The armament typically depends on the type of aircraft used.

[11]Modern equipment which may typically be found on reconnaissance fixed wing aircraft include infra-red, forward-down and sideways-looking systems as well as sideways-looking radars used to detect moving targets at night [159].

Rockets. Rockets are typically fired in salvos from pods at a high rate of fire and are typically used to attack soft or hard DAs. Their ammunition is categorised by high explosive/incendiary or armour piercing with hollow-charge warheads and smoke-markers for designating DAs, and high-explosive warheads with a dense fragmentation effect. Examples of rockets include: the 68 mm SNEB Rocket (France) and 57 mm Rockets (Russia).

Guided missiles. Guided missiles are known as stand-off WSs which are able to achieve very high hit probabilities. They are typically used to attack tactically important and well-defended assets when certainty of success is important and cannot be guaranteed by ordinary WSs. This is due to their warheads being relatively heavy with a penetration (hollow charge), defration, blast and incendiary effect. Examples of guided missiles include: the AGM-88 HARM (USA), the AGM-65 Maverick (USA) and the AS-7 Kerry (Russia) [9].

Guided ballistic bombs. These WSs are known for their high hit probabilities and are categorised by the following characteristics and effects: heavy blast (instantaneous fuse), penetration, cratering (delay fuse), and deflagration (cluster submunition and instantaneous fuse) [159]. This type of WS is typically used to attack soft or hard DAs and examples include: the AGM-62 Walleye II (USA), the Homing Bomb System (HOBOS) guided bomb (USA) and the Paveway I/II bombs (USA).

Free falling bombs. This type of WS is typically released in ripple-series in order to achieve a relatively wide dispersion. They may either be fitted with instantaneous fuses (in order to obtain a heavy surface blast and fragmentation effect) or with delay fuses (in order to achieve a penetration and cratering effect). If a time-delay fuse is fitted to these bombs, typical delays may range from five seconds to twenty four hours. Examples of free falling bombs include: the Mk 80-series of General Purpose Bombs and the UK 1000-lb bomb (UK).

Container/Cluster bombs. These bombs comprise a number of bomblets which form a cluster bomb falling nearly vertically onto the target and in large numbers. Such a bomb is typically delivered at low altitude and at high speed in conjunction with simultaneous attacks from several other aircraft. Examples of container/cluster bombs include: the JP-233 Submunitions dispenser (USA), the MK-20 Rockeye bomb (USA) and the BL-755 Cluster bomb (UK) [139, 78].

Fire bombs. This type of bomb induces a chemical incendiary which is very effective against inflammable materials and installations. They are especially effective against personnel and combustible material by producing a heat and conflagration effect, although the area they affect is relatively small. Examples of fire bombs include: the BLU-1 bomb (USA), the Mark 77 (USA) and the BLU-76/B Pave Pat 1 (USA) [78].

A number of the WSs described above are equipped with weapon guidance systems. The aim of such systems is to guide the WSs towards their targets for penetration or to bring the WSs close enough to the targets for the warheads to detonate. Examples of guidance systems which may be found on various WSs include: active radar homing, anti-radiation, command guidance, guide by wire, infra-red homing, optical guidance, semi-active laser homing, semi-active radar homing and smart guidance [159]. Although the use of weapon guidance systems may improve the accuracy of WSs, the only way of effectively neutralising these WSs is to employ appropriate counter measures which implies that such WSs should be considered as threats to the system in their own right — this, however, falls outside the scope of this dissertation (recall that only fixed wing aircraft are considered as threats). It is nevertheless important to have sufficient knowledge about the properties and vulnerabilities of these systems since they may impose restrictions on certain aircraft attack techniques used by the opposing force when specific WSs are delivered to DAs.

In order for the opposing force to deliver their WSs effectively, their aircraft have to fly a specific profile called an *attack technique*. The form (or shape), constraints and number of stages of such a technique is typically determined by various factors including: doctrinal rules, rules of engagement, terrain features, weather conditions, DA characteristics and geometry [158]. Five popular aircraft attack techniques are reviewed here. They are the pitch-and-dive technique, the high level dive technique, the combat turn dive technique, the toss bombing technique and the low level attack technique.

The *pitch-and-dive* attack technique, also known as the *combat hump dive* attack technique [162] is typically used to attack static DAs. The technique consists of an aerial threat approaching a DA at low altitude so as to attempt avoiding radar detection [162]. This approach phase is followed by a manoeuvre phase during which the threat pulls up (pitches) at a distance of approximately 3–9 km from the DA; small manoeuvres are possible during this stage. The threat then turns in to the DA so as to point its nose directly at the DA. The threat now enters the attack phase during which the aircraft is stabilised before aiming at the DA. The weapons are released at a distance of approximately 800–2 500 m from the DA. Once the weapons have been released, the threat pulls up from the dive to manoeuvre away from the DA as quickly as possible. The pitch-and-dive flight path attack technique is illustrated graphically in Figure 2.7.

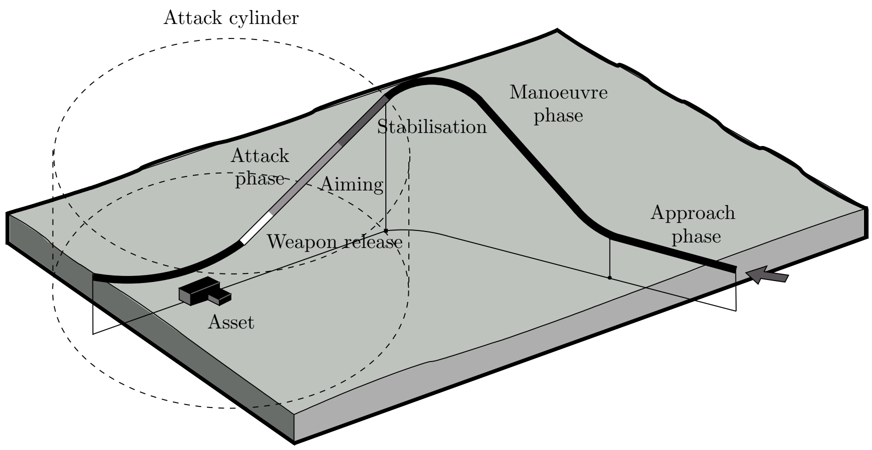


FIGURE 2.7: *Graphical illustration of the pitch-and-dive attack technique [162].*

The *high level dive* attack technique is also employed to attack static DAs. The technique entails an aerial threat approaching a DA at high altitude (more than 200 m above ground level). To render this approach successful, cloud cover or high terrain is required in order to avoid long range radar detection from own forces. The approach phase is followed by a manoeuvre phase which is carried out at approximately the same altitude as the approach phase. The aircraft then turns in towards the DA so as to point directly at the DA. The remainder of the attack stages follow exactly as for the pitch-and-dive attack technique. The main advantage of adopting this technique is that the aircraft is usually able to remain out of range of small-calibre guns as well as hand-held missile systems. The success of this technique may be enhanced if executed from behind terrain obstacles, clouds or out of the sun — a greater element of surprise is thus created. The high level dive attack technique is illustrated graphically in Figure 2.8.

The *toss bomb* attack technique is typically used to attack heavily defended static DAs. The technique consists of an aerial threat which approaches a DA at a low altitude, approximately

60–150 m above the ground [162]. During the manoeuvre phase, the threat pulls up at an exact, pre-determined position (typically 4.5–6 km from the DA) and starts aiming its weapon release system at the DA. It releases its weapons at a slant distance of approximately 3.5–4.7 km from the DA. After releasing its weapons, the threat pulls around in order to manoeuvre away from the DA as quickly as possible. The toss bomb attack technique is illustrated graphically in Figure 2.9.

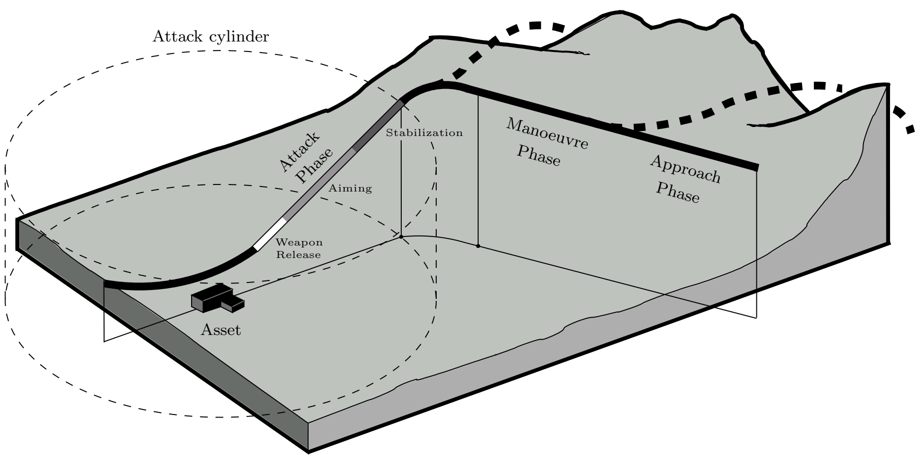


Figure 2.8: *Graphical illustration of the high level dive attack technique [159].*

The *combat turn dive* attack technique is usually employed to attack moving DAs and is typically executed as a primary attack after an armed passing reconnaissance, as a secondary attack with alternative WSs or against DAs of opportunity encountered by chance in passing flight. This technique requires a flat terrain surface in order to ensure good visibility. The technique consists of an aerial threat approaching a DA at high altitude, approximately 150–300 m above the ground. After having detected the moving DA, the threat lowers its altitude to approximately 60 m above the ground. After approximately 5–10 seconds, the manoeuvre phase commences during which the threat pulls up and turns in towards the DA to deliver its WSs. The remainder of the attack stages follow exactly as for the combat hump dive attack technique. The combat turn dive attack technique is illustrated graphically in Figure 2.10.

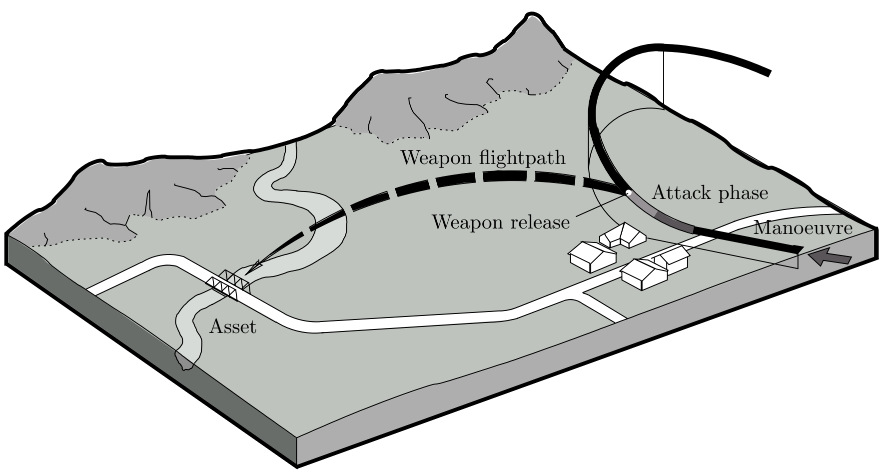


Figure 2.9: *Graphical illustration of the toss bomb flight path attack technique [162].*

The *low level attack* attack technique is also employed to attack moving DAs. In this attack technique, the aerial threat typically assumes a horizontal flight path, if possible, and aims to execute the technique at an altitude as low as possible — the lower the altitude of the aircraft, the smaller the chance of the aircraft being hit by the own force's WSs. If the combat zone falls within a hilly terrain area, a flat dive angle has to be adopted in order to ensure detection of the DA. This technique is similar to the combat hump dive technique, except that the dive angle is flatter [159]. Four variations in this technique exist, namely: a low level attack involving the threat executing a shallow dive with a long WS release range, a low level attack involving a threat executing a shallow dive with a short WS release range, a low level attack involving the threat executing a horizontal attack with a high WS release altitude and a low level attack involving the threat executing a horizontal attack with a low WS release altitude. The low level attack technique involving the threat executing a shallow dive with a long WS release is illustrated graphically in Figure 2.11.

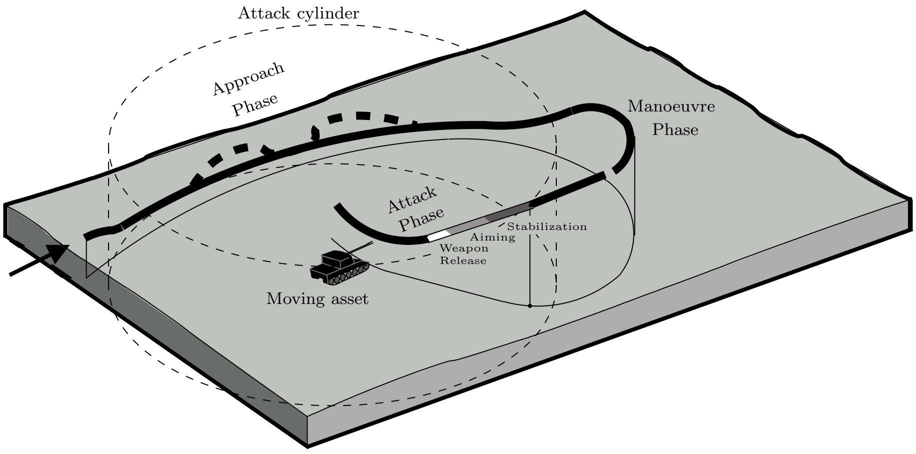


FIGURE 2.10: *Graphical illustration of the combat turn dive flight path attack technique [159].*

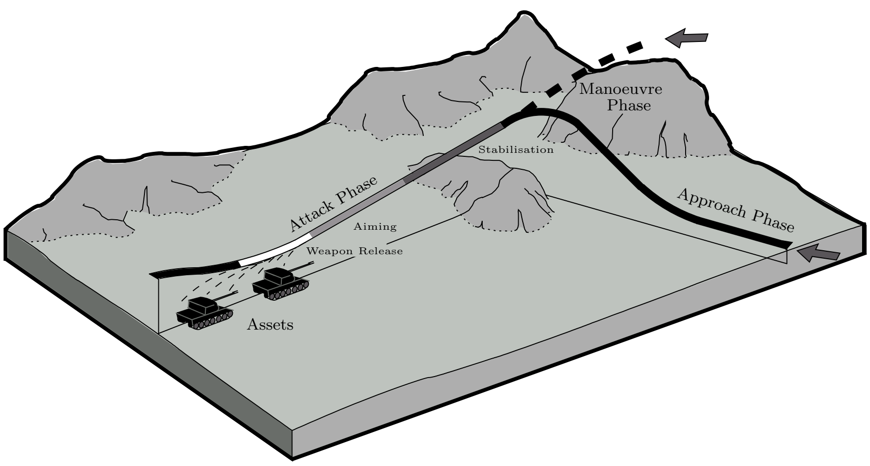


FIGURE 2.11: *Graphical illustration of the low level attack technique involving the threat executing a shallow dive with a long WS release [159].*

Ground-based weapon systems

Ground-based WSs are used to afford protection of DAs against aerial threats. These WSs are typically deployed in a strategic manner around the DAs so as to afford the maximum level of protection against attacks from the opposing force. Efficient deployment of WSs naturally gives rise to the notion of *layered AD* which is a well-documented topic in the military literature [22, 52, 100]. It is a theoretical construct which involves dividing the defended airspace surrounding the DAs into a number of layers depending on the capabilities of the various types of WSs used in the deployment [190]. Within the context of South African AD, there are four layers of AD which are typically employed. These layers comprise an inner, middle, outer and in-depth layer, as illustrated graphically in Figure 2.12. The figure illustrates a two-dimensional side view of the four layers based on the ranges (both vertically and horizontally) of the WSs typically employed in each layer.

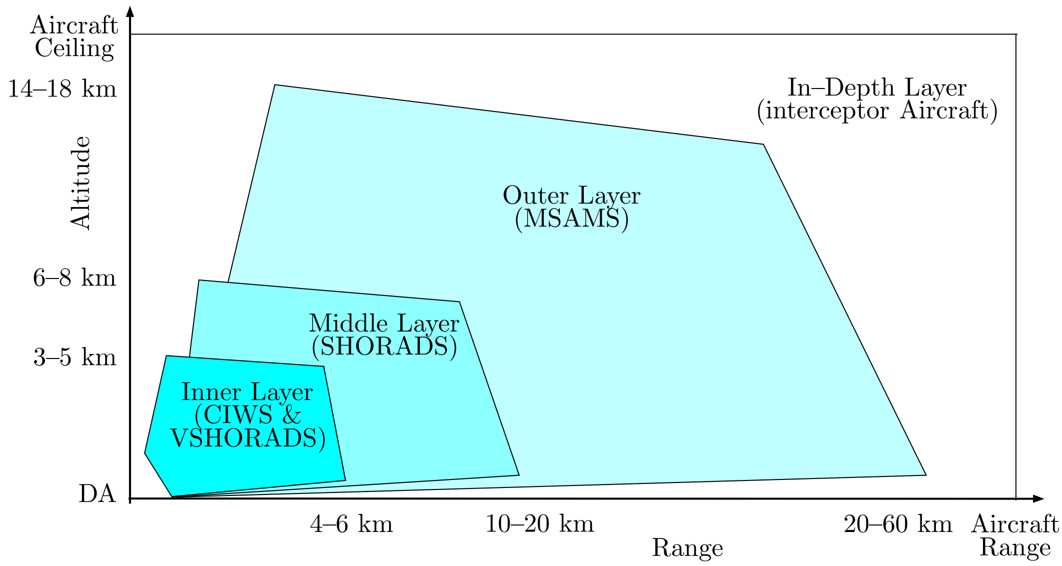


FIGURE 2.12: *Four layers of AD based on the ranges of WSs contained in each layer [159].*

The first layer of AD, the inner layer, is typically afforded by *Close-In Weapon Systems* (CIWSs) and *Very SHOrt Range Air Defence Systems* (VSHORADSs). CIWSs usually constitute guns (*i.e.* 35mm dual purpose guns) and are known for their short reaction times, high fire rates, short effective ranges (less than 4 000 m), lengthy deployment procedures (due to allignment procedures that have to be followed), all-weather operation and intensive maintenance requirements [190]. VSHORADs are characterised by their light weight (which enables them to be man portable), rapid deployment procedures and short effective ranges (less than 6 000 m). Starstreak missile systems are an example of a VSHORADs.

The second layer of AD, the middle layer, is usually afforded by *SHOrt Range Air Defence Systems* (SHORADSs) of which the South African Umkhonto missile system is an example. SHORADSs are characterised by their ability to reach effective ranges of up to 20 000 m, possible vertical launch and day/night as well as all-weather operation [159].

The third layer of AD, the outer layer, typically consists of *Medium-Range Surface to Air Missiles* (MRSAM) and are known for their ability to reach effective ranges up to 60 000 m and high altitudes up to 18 000 m [159]. An example of an MRSAM is the Royal Navy Seadart system.

Finally, the fourth layer of AD is typically provided by *Long-Range Surface to Air Missiles* (LRSAMs) and interceptor aircraft. LRSAMs are characterised by their effective ranges (larger than 80 000 m) and large coverage (depending on the specific surface to air missile used and the targets engaged) [159].

Currently, the available AD artillery within the South African GBADS is limited to include only WSs from the inner and middle layers of AD [159]. The scope of this dissertation is therefore limited to include only WSs from these two layers. These WSs are CIWSs and VSHORADSs.

It should be noted that a number of WS parameters affect the WA process [145]. These parameters include the ammunition available for each WS in the deployment, the velocity and cost of the ammunition, the time that is required to reload WSs and the so-called *Single Shot Hit Probability*[12] (SSHP) of the WSs. The WA process is further complicated by the WS operators who are governed by doctrinal rules, readiness and control measures [159].

2.5.2 Functional elements

In 2010, Roux [159] proposed a number of functional elements (or processes) that may be used for AD decision support — in accordance with the existing South African GBAD functional elements of that time. These functional elements, as well as the data flow between the processes, are illustrated graphically in Figure 2.13 and are described in some detail in this section.

The first functional element in Figure 2.13 is *operator decision support*, which enables the FCO to communicate with the physical elements in the system. It may also be referred to as the link between man and machine, and is typically implemented by means of a so-called *Human Machine Interface* (HMI)[13] which provides the FCO with an interactive graphic user interface [143, 149]. The HMI displays the air picture on a screen, enables the FCO to view results generated by the other functional elements (*e.g.* the TE and WA results), and performs decisions with respect to the evaluation and prioritisation of threats as well as the assignment of WSs for engaging aerial threats. A well-designed and lean user interface is essential since the FCO may easily be overwhelmed by the amount of data displayed on the HMI. The HMI should be able to provide the functionality that the FCO requires with the minimum amount of effort on the operator's part whilst improving his productivity [143]. The effectiveness of the HMI is therefore important. The ISO 9241 standard defines the following three components of quality in the design of HMIs: *effectiveness* (Does the product do what the users require? Does it do the right thing?), *efficiency* (Can the users learn the HMI quickly? Can they carry out their tasks with minimum expended effort, including a minimum of errors? Does it improve the productivity/effort ratio? Does it do things right?) and *satisfaction* (Do users express satisfaction with the product? Does the new product reduce stress?) [149].

In the context of a GBADS, an HMI is required to operate in various modes depending on the authority that the operator has and the role he fulfils. The inputs and outputs of the HMI are therefore customised and filtered accordingly when operators log in to perform different tasks. Functionalities of the HMI include ADC health display, air picture management, sensor control, diagnostics, error notification, fire control, training, and maintenance [159]. In addition, the HMI should also be able to provide the operator with a *Plan Position Indicator* (PPI)[14].

[12] The probability of hitting a target at a specific location relative to the WS with a single unit of ammunition.

[13] The HMI is sometimes also referred to as a *man machine interface* (MMI) [149].

[14] A PPI is a means of representing radar output which involves a display unit resembling a map with the radar site at the centre of the display [8]. The physical elements of an ADC system is typically represented by means of MIL-STD-2525B symbols [134] on the PPI.

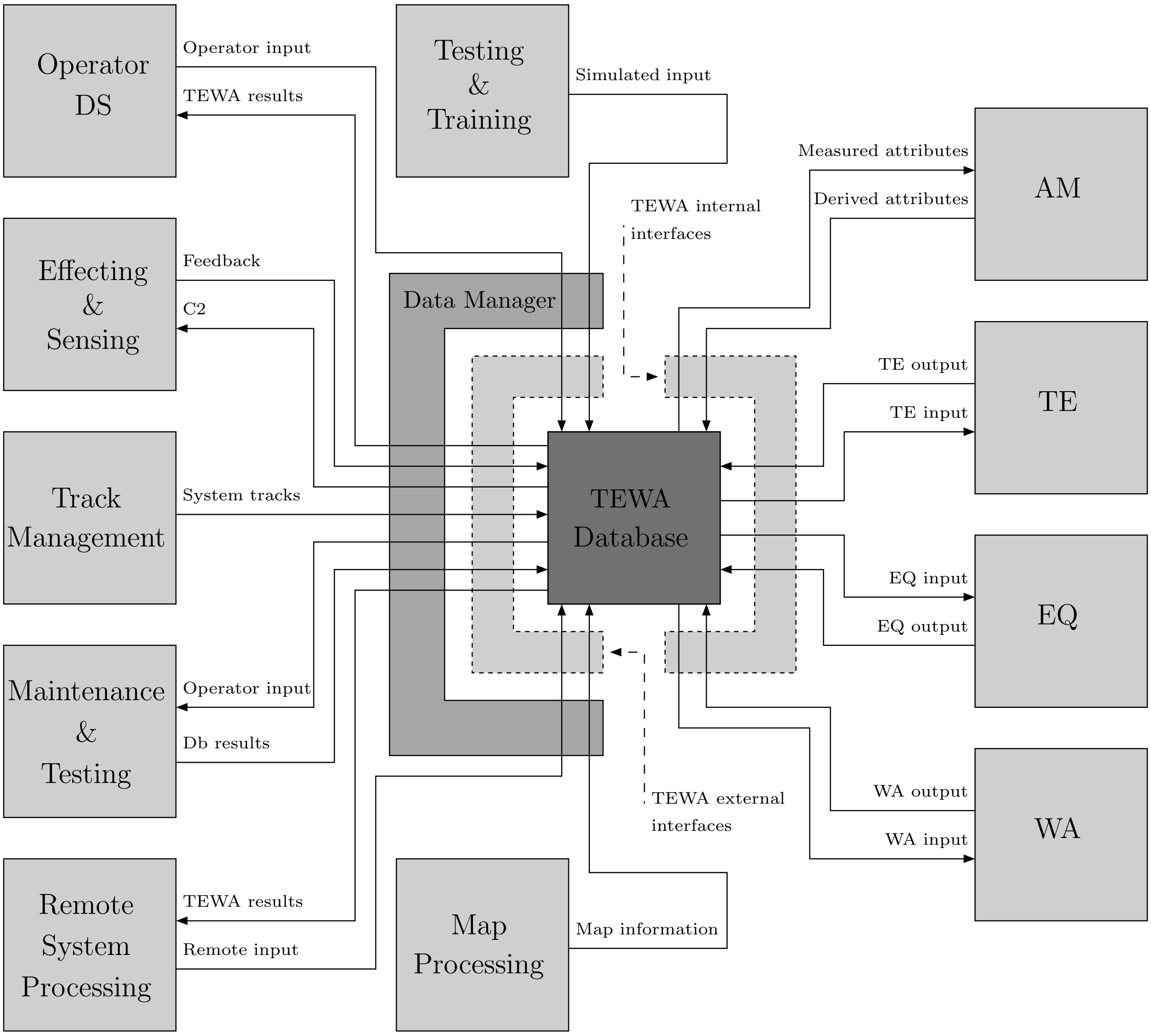


FIGURE 2.13: *The flow of information between the internal-, external functional elements and the database of a TEWA system. Adapted from Roux [159].*

Military personnel have to be trained so that they are able to operate the functional elements of a GBAD system effectively. This may be achieved by *testing and training* procedures that are carried out during the pre-deployment stage of a mission, or at an earlier stage. The type of training required depends on the various capacities in which military personnel are deployed. Training procedures typically involve simulation functions that are responsible for imitating diverse tactical GBAD scenarios in which operators have to operate [162]. A scenario generator is typically employed to assist in the training of operators in the respective roles they fulfil within an ADC system. The scenario generator is capable of creating, storing and playing back pre-configured scenarios, including DAs and aircraft-WS delivery profiles [159]. It should be noted that the scenario generator is typically only activated if the HMI is in training mode. During these training procedures, actual software may therefore be disabled or disconnected. Testing usually involves evaluating the success of the integration of the system as an entity. This may be achieved by testing the performance of the system under controlled conditions while analysing the results [90]. During these testing procedures, scenarios may be employed that are similar to those generated for training purposes.

The TE and WA functional elements are not necessarily directly connected to the *effecting and sensing* functional element. Information (such as feedback messages) forms part of the sensor or WS system tracks which are handled by the TM process and relayed via the data management process. In some situations (*e.g.* certain TEWA modes or enhanced reaction modes), however, time-critical feedback or response time may induce the use of direct links between these elements. The effecting and sensing process provides this functionality.

Typical information received from the sensor systems are the measured (kinematic) attributes of incoming aircraft *e.g.* the speed or altitude at which they are travelling. Recall (from the discussion on sensor systems in §2.5.1) that the TM system uses this information to produce a single system track for each aircraft which enters the defended airspace. This system track should correctly indicate the current location and movement of each observed aircraft in the defended airspace, is generated by fusing together various aircraft characteristics, and may be of use when identifying and classifying the observed aircraft[15] [194]. Functionalities provided by the TM system include performing operations such as correlation, fusion, *Hostility Classification/Identification* (HCI), triangulation and *Type Classification/Identification* (TCI) [159]. The HCI operation involves the classification of observed aircraft as hostile or friendly, which is supported by means of intelligence reports, IFF interrogations and other specialised systems. The TCI operation involves the classification of all observed aircraft (*e.g.* fixed wing or rotary) and is also supported by intelligence reports. Only those aircraft that are classified as hostile are considered for further evaluation with respect to the level of threat they pose to DAs. It is evident that the TM system relies heavily on information from the sensors. It can therefore only produce system tracks at the same rate at which aircraft kinematic information is received from sensor systems.

Apart from creating system tracks and classifying aircraft, the TM system also includes a so-called *Flight Path Prediction* (FPP) module. The function of this module is to predict the flight paths of observed threats over a pre-specified number of future time stages [194]. This is achieved by employing prediction models which utilise system track information from the TM. The output of the FPP module is an array of predicted flight paths for each threat combined with a probability distribution for each of these paths, which is usually fused into a single, expected flight path for each threat. This single expected flight path is stored as part of the system track of each threat in the TEWA database, is updated at the same rate at which the system tracks are updated and serves as input to the TE and WA subsystems [146, 197]. The number of future time stages over which flight paths are predicted should be chosen carefully, since the statistical confidence associated with a predicted flight path diminishes as the number of future time stages increases.

During the pre-deployment stages of a mission, the various models residing in the TE and WA processes require certain initialisation parameters based on intelligence with respect to the aircraft types, modes of armament, and attack profiles typically employed by the opposing force, as well as various WS parameter threshold values. A set of initialisation parameters may be required for each TE model and each WA model, and these parameters may be updated during the mission by means of carefully established *maintenance and testing procedures.*

A number of sensor, WS and DA attributes are typically determined during the deployment of a mission and are also stored in the TEWA database. The attributes may be received from *remote system* processes. The attributes are stored as system tracks and should be updated in real time (*e.g.* if the position of a mobile WS changes), similar to the system tracks of aerial threats [159]. Remote system processes may increase the flexibility of the system, but may result

[15] Tadil [180] defines a track as "a collated set of data associated with a track number for the purposes of representing the position and/or characteristics of a specific object, point or bearing."

in an increase in the processing power required by the TEWA system.

A *map processing server*, as part of an ADC system, contains information with respect to all functionality and data pertaining to geographical maps. In the context of a GBADS, the MAP server functionality may typically include a component responsible for line-of-sight computations, a display component for map planning and orientation processes, and the ability to toggle map features (*e.g.* national and regional boundaries, coastlines, contours, roads, rivers and cities) [159]. Furthermore, the map server may include a local database for storing information regarding maps and a display component for planning and prioritisation.

A central functional element of an ADC system is the *data manager*. It serves the purpose of communicator between interfaces of other systems contained in the ADC system, resulting in the continuous relay of data between the interfaces of these systems. These system interfaces may be subdivided into internal interfaces and external interfaces. The TEWA interface is an example of an internal interface, whereas WS and sensor interfaces are examples of external interfaces. The data manager is also able to provide functionalities such as configuration functionality, diagnostic and system health monitoring, the processing of sensor and system track data, the storage of data in a central repository, and recording and playback functionality [159].

The measured attributes of aircraft obtained from the sensor systems are used by an AM system to compute further derived attributes (as mentioned earlier) associated with each system track (*i.e.* each threat) [162]. An example of such an attribute is the acceleration of an aircraft. Once the derived attributes have been calculated, they are stored in real time as part of the system track of each threat in the TEWA database. The derived attributes are employed in the TE functional element.

The TE subsystem proposed by Roux and Van Vuuren [162] comprises two substructures, known as the *Threat Evaluation Model* (TEM) component and the *Threat Evaluation Fusion Model* (TEFM) component. The former component employs the measured attributes obtained from the sensors, the derived attributes retrieved from the AM system, as well as pre-deployment data and initialisation parameters to estimate the level of threat that each of the hostile aircraft in the defended airspace poses to the DAs. This may be achieved by means of the suite of mathematical threat evaluation models functioning concurrently.

The TEM component proposed by Roux and Van Vuuren [162], in fact, comprises three classes of TE models which are distinguished from one another by the level of complexity and sophistication of the models contained in each class. The three classes of models are, in increasing order of complexity, *flagging models*, *deterministic models*, and *stochastic models*:

1. Flagging models are a suite of binary output TE models. These models are qualitative in nature and simply alert the operator by flagging an aircraft if sudden changes in the kinematic behaviour of the aircraft are observed or if the aircraft seems to be engaging in some kind of hostile behaviour. Hence, flagging models are not able to distinguish between different levels of threat posed by aircraft.

2. Deterministic models, on the other hand, are quantitative models, each based on some measure of threat, such as the expected travel time to a DA or some course/bearing-related measure. Each model in this class takes the observed kinematic data of aircraft, as well as the DA deployment data, as input and generates as output a normalised, deterministic threat value for each aircraft.

3. The TE models exhibiting the highest level of sophistication are the class of quantitative, stochastic models. Such a model takes the observed kinematic data of aircraft, DA deployment data, enemy arsenal intelligence, and doctrine as input and also generates as output

a single threat value for each aircraft-DA pair. This output value is, however, probability-based and is typically an estimate of the probability that an aircraft will attack and/or kill a specific DA.

The results generated by the TEM component are fused together by the TEFM component to obtain a single threat value for each threat so as to arrive at a single prioritised list of threats (*i.e.* a list containing a threat value for each threat) with respect to the entire collection of DAs. This may be achieved by means of some *Multi-Criteria Decision Analysis* (MCDA) value function method. The single threat list is presented to the FCO as DS in real time.

The remaining functional elements are the EQ subsystem and the WA subsystem. Since EQ entails the quantisation of the effectiveness of an engagement of an aerial threat by a WS, the EQ subsystem is considered a WA-related component. These two components form the main contribution of this dissertation and are hence described in detail later in this dissertation.

2.5.3 Cognitive elements

The third important element to consider in the design of a TEWA DSS is the cognitive element. Recall from §2.4 that the cognitive domain constitutes the collective minds of the war fighters and hence the cognitive element refers to how these war fighters (or operators) perceive attacks in a mission, how they interpret DS results provided to them by a TEWA DSS and the decisions they make based on these results. Recall from §2.2 that the individual which is central to C2 is the commander. Roux [159] argues that the entire OODA loop (see Figure 2.1) of the operator may be seen as a cognitive process. In the context of an ADC domain, information such as raw data on physical elements and TEWA DS results are presented to the operator in real time. The human mind then has to perceive this information, *process* it (which typically differs for each individual) and decide on a specific course of action to take, given the current tactical situation. In order to understand this process thoroughly, Roux [159] combined an information hierarchy developed in the *Australian Marine Corps Doctrinal Publication* [147] and a cognitive hierarchy developed by Roodt *et al.* [157] to arrive at a so-called *information-to-action* hierarchy via the cognitive domain. This hierarchy is presented graphically in Figure 2.14.

The hierarchy in Figure 2.14 constitutes three main levels, namely information, intelligence and command. At the lowest level (*i.e.* information), data are processed and converted into information by including it in a situational context. This information (or processed data) includes all the relevant data on TEWA results presented to the operator. It therefore includes all raw data on sensing, tracking, TM, TE and WA results. This lower level of the hierarchy therefore intends to provide all the information to the operator which may aid him to enhance his SA.

Information may typically be provided in two ways, namely in a supply-push manner or a demand-pull manner [96]. In a supply-push system, information is typically *pushed* from the source to the user either as the information becomes available or according to a pre-determined schedule. In a demand-pull system, on the other hand, information is typically demanded by the user and therefore does not rely on the ability to anticipate information needs. Furthermore, information may also be transmitted by either broadcast or point-to-point transmission [147]. Broadcasting entails transmitting information to a broad audience which proves to be very effective, although information cannot be tailored for a specific user. Point-to-point broadcasting, on the other hand, entails information being sent to a specific user or users — appropriate information is therefore transmitted sequentially from one user to the next.

The middle level of the hierarchy in Figure 2.14 constitutes intelligence. At this level in the hierarchy, information from the lower information level has reached the operator. The intelligence

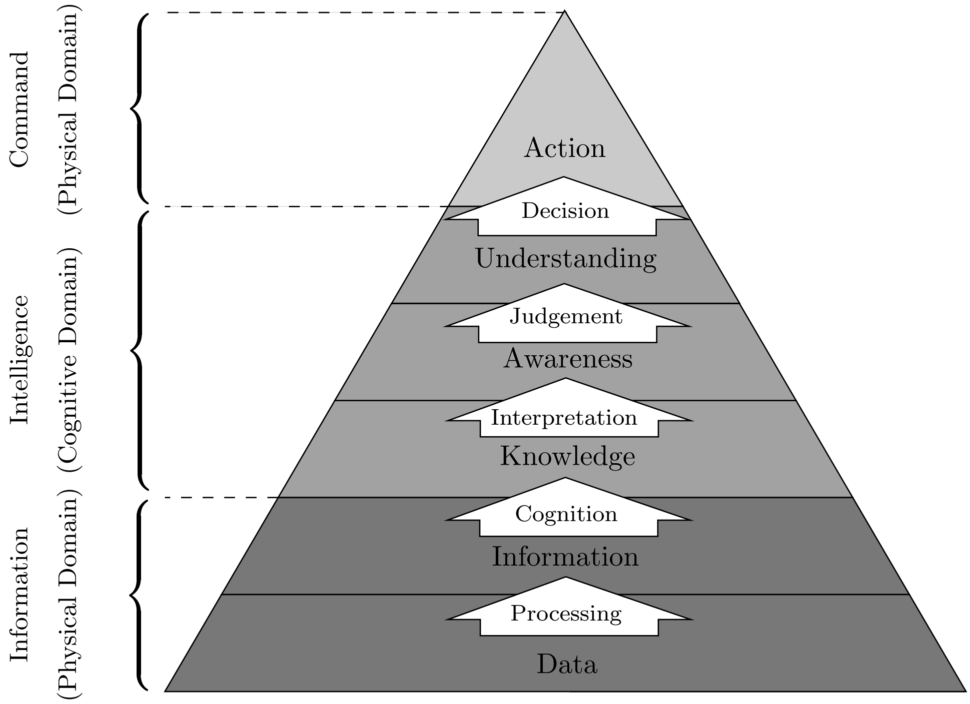


FIGURE 2.14: *Information-to-action by means of the cognitive domain [159].*

level involves processing (by means of cognition) the information from the information level into *knowledge*.

Von Clausewitz [201] defines the term *coup d'oeil*[16] as the ability of a commander to intuitively grasp what is happening on the battle field. A higher level of C2 typically requires that the operator (or commander) relies more on information from others and less on own observations. Roux [159] states that when observing a situation directly, an intuitive appreciation of the level of uncertainty may be achieved — this sense is typically lost when "second hand" information is received. In addition, information may become distorted as it progresses through every node in the hierarchy before it reaches the operator. A time-delay to the information due to the point-to-point transmission described above may also be experienced [147]. This problem may, however, be overcome by providing data of certain critical events to the operator directly.

When considering the hierarchy in Figure 2.14, it is observed that the cognitive process of the operator starts with knowledge and involves the operator receiving information from the information tier and observing this information. This aids the operator in reaching a certain level of SA which directs the cognitive process to the next level in the hierarchy, namely the awareness level.

The awareness level is therefore reached by the operator interpreting the knowledge obtained from the information level. Here, the cognitive process of information employs observation information, orientation information, experience and training in order to reach a level of SA. Note that the level of SA reached on this level corresponds to the intermediate levels of SA of Figure 2.3 — the assessment of the intent and organisation of the opposing force.

[16]The term *coup d'oeil* literally translates to "stroke of the eye."

The next level in the operator cognitive process hierarchy is understanding. In order to reach this level of the cognitive process, operator judgement is required with respect to *what-if* scenarios which takes into consideration the organisation and intention of the own force and opposing force. Such judgement typically depends on operator experience, SA, doctrinal rules, and orientation information.

At this level in the hierarchy, the operator has observed the tactical environment and has oriented himself with respect to the tactical situation. The next steps in the OODA loop is therefore for the operator to decide and act accordingly. Decisions typically involve some level of uncertainty and although such uncertainty may be reduced with the help of information, the corresponding reduction usually comes at the expense of time [147]. Klein [106, 105] describes two basic approaches which may be followed when making decisions. These approaches are an analytical approach and an intuitive approach.

The analytical approach involves generating a number of different alternatives, comparing these alternatives according to a given set of criteria and choosing the best alternative. This approach tends to be methodological (analytical) and very time consuming since reasoning power matters more than experience.

An intuitive approach, on the other hand, involves no computational effort whatsoever and rather relies on the intuition of an experienced operator to identify important aspects of a specific problem so as to arrive at a suitable solution (decision). In this approach, the focus is to find the first solution that will solve the problem satisfactorily rather than finding an optimal solution [169]. In order to promote this approach, Patton [23] suggested that: "A good plan violently executed now is better than a perfect plan next week."

It is therefore evident that the correct decision making approach to be adopted in an ADC scenario should be based on the current tactical situation at hand, the time available for the decision, and the knowledge (experience) and level of SA of the operator. Roux [159] suggests that for an efficient decision making process, a combination of these approaches should be used. Furthermore, he claims that the goal of TEWA DSS should be to automate certain elements of the system (if possible) which may streamline the decision making process, ultimately leading to a reduction in the time required to make the decision. Such automated procedures are especially applicable to analytical decision processes which are fairly easy to automate and are aimed at producing alternatives (or an optimal alternative), depending on operator preference. This provides the operator with ample time to consider the result(s) and to act accordingly based on his judgement (experience and knowledge). The final level in the operator cognitive process hierarchy of Figure 2.14 is then to take action by issuing some form of command.

In order to understand the role and responsibilities of the FCO, it is also necessary to understand the role of the commander. According to Roux [159] the training of a commander and that of an FCO are similar in terms of courses completed and practical experience gained. The difference between the commander and the FCO, however, is that the commander is trained to command an entire mission and to also interact with other ADA functions. The roles of the commander and the FCO are nevertheless very intertwined and they work together to command an ADC deployment during a mission.

The commander typically utilises the intelligence (information) which he has received and develops a so-called *ADA battle plan* for a particular mission [115]. In order to develop an efficient battle plan, the commander orientates himself with respect to the current situation and he conceptualises the options available to the own force as well as the opposing force within the context of the terrain, the civil population, meteorological conditions and the current air situation [112, 113, 114]. The commander then finally chooses the *most probable* and *most dangerous* enemy courses of action from an own force perspective. He then sends a request for the deployment of the physical elements in such a way so as to counter these enemy courses of action as efficiently as possible.

Roux [159] states that it is important for the commander and the FCO to work closely together. The FCO has to be part of the planning process described above since he is the one who will ultimately fight the battle. The FCO should therefore support the commander in the planning phase during the pre-deployment stages of a mission — the FCO should ensure that the communication channels between himself and the physical elements are set up correctly. Furthermore, the FCO should also use the appropriate intelligence available, as well as the conclusions and assumptions made by the commander, to set up the TEWA system. Once the physical elements have been deployed, all functional elements are operational, all required communications have been established and all cognitive elements are in position, the commander declares the deployment as 'Ready for Action.'

At this stage of the mission, the responsibility of the successful completion of the mission shifts completely to the FCO. The commander may typically still be involved, but only on a high level (for example, observing and monitoring the ADC activities). The commander usually starts with planning for future missions at this stage. Apart from supporting the commander during the planning phases of a mission, the responsibilities of the FCO may be divided into the following three distinct time frames [159]:

Early Warning. During this period, early warning reports are received, indicating that possible aerial threats are entering the defended airspace and heading towards the DAs. The FCO then reacts to these early warning reports by orientating himself with respect to the possible threat(s) and the various courses of action that he can take according to the ADA battle plan previously established by the commander together with the FCO. The FCO may also warn and alert the various physical elements in order for them to ready themselves for possible courses of action.

Local Warning. During this period, local warning reports are received from sensor systems. The FCO then typically sends local warning reports containing additional information about the approaching threat(s) to ADC elements (*e.g.* WSs) so as to alert them to track the threats when they come within range of their tracking sensors.

Fire Control. The fire control period commences once the FCO sends the first engagement order. Such an engagement order may be sent to one WS or a number of WSs at a time and is typically monitored by the FCO on an *ad hoc* basis. Automated feedback of the engagement orders are relayed continuously from the WSs to the TEWA system. In the event of communication between certain functional elements failing, the FCO is required to obtain voice feedback from the WSs. In return, he will then provide manual feedback via the HMI to the TEWA system.

2.6 Chapter summary

This chapter was dedicated to a discussion on general GBAD. The chapter opened in §2.1 with a brief review of TEWA DS within a GBAD context. The section included a description of IPB, which is central to the TE process, a coherent definition of TE as well as approaches that aerial threats may follow to attack the own force. The different platforms from which the process of WA may be approached was also discussed in some detail.

The remainder of the chapter focussed on three important fundamental theories of warfare, as well as the physical, functional and cognitive elements which form a central part of an effective TEWA DSS. In §2.2, a description was given of C2 (which included a description of the well-known C2 model of Boyd [20]). Variations on this model by other authors were also touched upon. The importance of speed in terms of increasing the OPTEMPO of C2 was also briefly mentioned.

In §2.3, the focus shifted to a discussion on SA. The importance thereof in military missions was highlighted and a three-tiered hierarchy for achieving SA due to Endsley [66] was also discussed. Furthermore, a framework (also containing three levels of SA) for achieving complete SA in the context of C2 was also described. This framework was proposed by Wallenius [204].

In §2.4, the notion of NCW was considered. The importance thereof between sensors, WSs, decision makers and information was briefly discussed, and this was followed by a detailed discussion on the social, cognitive, information and physical domains which surround NCW. The interactions between the four domains were also highlighted. The section closed with a discussion on a logical framework for NCW between sensors, C2 and WSs proposed by Cebrowski [31].

In §2.5, the physical-, functional- and cognitive elements of a TEWA DSS were highlighted. The section opened with a discussion on the physical (hardware) elements, including DAs and how they may be prioritised, sensor systems and their various capabilities with respect to detecting and tracking aerial threats, various aircraft and their manoeuvring capabilities (including possible WS armament carried by them and attack profiles that they may execute) when attacking DAs and ground-based WSs that the own force may employ to counter aerial threats.

The focus next shifted to a discussion on the functional (software) elements of an effective TEWA DSS. This discussion highlighted functional elements pertaining to operator DS, testing and training, effecting and sensing, TM, data management, maintenance and testing, remote system processing, map processing, AM, and TE. The chapter finally closed with a discussion on the cognitive elements of a TEWA system. The discussion focussed on the roles and responsibilities of the commander and FCO.

CHAPTER 3

The evolution of weapon assignment modelling

Contents

Ever since aerial vehicles have proven a threat to defended assets on the battlefield during the Second World War, high-quality assignments of available WS have been sought to deter or eliminate these aircraft in ground-based air defence scenarios. The notion of WA has since been researched extensively by a number of research institutions over the world and various authors have contributed significantly to the topic. This chapter is devoted to a detailed discussion on the evolution of WA modelling over time since its inception in 1952. The chapter opens in §3.1 with a brief introduction to the notion of **NP**-completeness, because the decision problems associated with even the simplest WAMs are **NP**-complete. Next, a detailed description follows in §3.2 of the origin of the first formulation of a WAM by Manne [125] in 1957. The three sections that follow contain similarly detailed descriptions of various contributions towards WA modelling since the 1950s up to the current state of the art WAMs available in the military operations research literature. In §3.3, a number of WAM contributions from the early 1960s to the mid 1980s are reviewed in some detail. A major breakthrough in the development of WA modelling was the formulation of the first dynamic WAM by Hosein *et al.* [89] in 1989. This model, as well as various extensions to and generalisations of the model up to the late 1990s, are discussed in §3.4. In §3.5, the focus shifts to a discussion on the current state of the art of WAMs which have been contributed to the military operations research literature during the 21$^{\text{st}}$ century. The chapter finally closes in §3.6 with a brief summary of the chapter contents.

3.1 The notion of NP-completeness

Complexity theory underlies the study of the *intrinsic complexity* of computational tasks. One of the main goals of complexity theory is to establish means for classifying computational tasks according to the resources required to solve them [128]. Examples of such resources include time, storage space, random bits or number of processors, but the focus is typically on time and space.

An *algorithm* is an ordered sequence of procedural operations that may be used to solve a problem within a finite number of steps [83]. Algorithms are widely used to solve different kinds of computational problems and in most cases it is important to analyse the efficiency of an algorithm in terms of its speed and the amount of computer memory required to execute the algorithm. The reason for this is to ensure that for a computational problem of a given size, the algorithm will be able to solve the problem within a realistic time frame on a computer with a realistic memory capacity.

Algorithmic complexity is typically measured by means of two variables: the *time complexity* $T_c(n)$ and the *space complexity* $S_c(n)$ of the algorithm, where n refers to the size of the input to the algorithm [83]. The time complexity of an algorithm represents the amount of time required by a computer to execute an algorithm, while the space complexity represents the amount of space (*i.e.* memory) required by a computer to execute an algorithm. The *worst-case* complexity of an algorithm is the largest values that $T_c(n)$ and $S_c(n)$ may assume for an algorithm with input size n. It is often difficult to determine exact counts of the resources required to execute an algorithm and in such cases so-called *asymptotic upper bounds* on the functions $T_c(n)$ and $S_c(n)$ are usually estimated rather than seeking exact upper bounds — these asymptotic bounds describe how $T_c(n)$ and $S_c(n)$ increase as $n \to \infty$. For two functions $f(n)$ and $g(n)$, the notion $f(n) = \mathcal{O}(g(n))$ means that there exist constants $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$ [128]. In such a case, the function $g(n)$ is an asymptotic upper bound for the function $f(n)$ as $n \to \infty$ and it is said that the function $f(n)$ is *of the order of* the function $g(n)$.

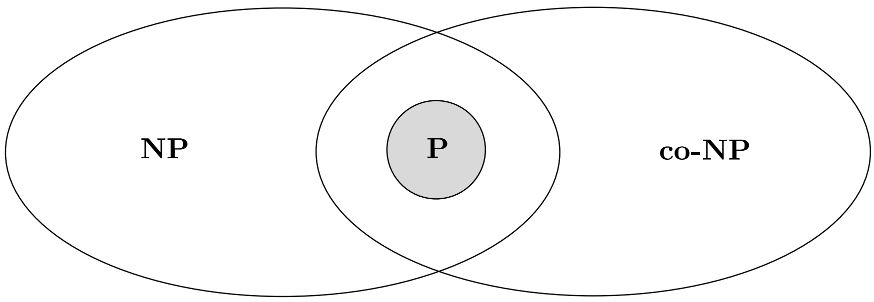


FIGURE 3.1: *The complexity classes* **NP, P** *and* **co-NP** *of decision problems [47].*

A *polynomial-time* algorithm is an algorithm having time complexity $\mathcal{O}(n^k)$, where n is the size of the input instance to the algorithm and $k \in \mathbb{R}^+$. *Decision theory* is that part of complexity theory that focusses on binary output to problems to which the answer may be interpreted as a boolean value `true` or `false` [47, 83]. The class of decision problems which may be solved by polynomial-time algorithms is denoted by **P** (an abbreviation for *Polynomial*). Furthermore, the class of **NP** (an abbreviation for *Non-deterministic Polynomial*) problems contains all those decision problems for which the answer `true` may be verified by a polynomial-time algorithm, given additional information related to the problem instance (known as a *certificate*). The class of **co-NP** problems, on the other hand, contains all the decision problems for which the answer `false` may be verified by a polynomial-time algorithm, given additional information related to the problem instance (again called a certificate) [128]. Although a certificate may exist for decision problems in the classes **NP** or **co-NP**, it may sometimes be difficult to find such certificates. The various complexity classes of decision problems discussed above are illustrated graphically in Figure 3.1.

The notion of *reducibility* may be used to determine whether one decision problem is at least as hard to solve as another. Let D_1 and D_2 denote two decision problems. The decision problem D_1 is then *polynomial-time reducible* to the decision problem D_2 if an algorithm A_1 exists which can solve all instances of D_1 and also contains as subroutine an algorithm A_2 which can solve all instances of D_2 in such a way that A_1 is a polynomial time algorithm if A_2 is a polynomial time algorithm [83, 128].

A decision problem D may be classified as **NP**-hard if D_1 is polynomial-time reducible to D for all $D_1 \in$ **NP**. Moreover, a decision problem D may be classified as **NP**-complete if $D \in$ **NP** and if D is **NP**-hard [47, 83, 128]. An updated classification scheme of decision problems described so far in this section is illustrated graphically in Figure 3.2. The class of **NP**-complete problems may therefore be considered the most restrictive class of decision problems since this class is a subset of the class of **NP** problems. **NP**-complete problems are also considered the most difficult class of decision problems in Figure 3.2 since they are computationally at least as hard to solve as any other decision problem in **NP**.

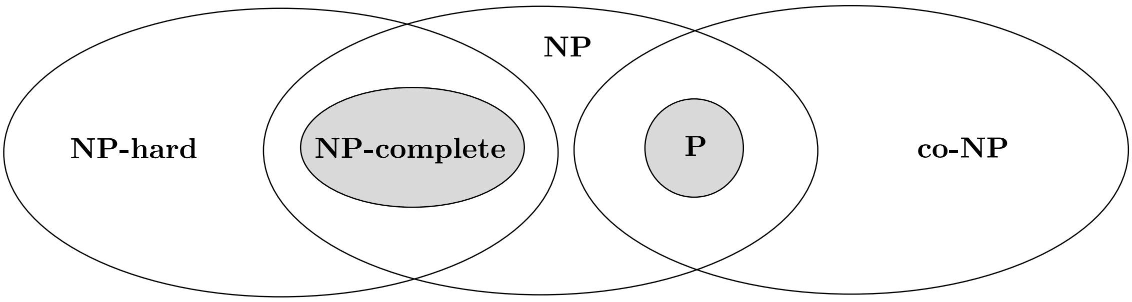


FIGURE 3.2: *The decision problem complexity classes* **P, NP, co-NP, NP**-*hard and* **NP**-*complete [47].*

Computational problems are problems having a real number (or a collection of real numbers) as solution rather than a binary value. Decision problems may therefore be considered as special cases of computational problems. Computational problems may often be solved efficiently in terms of algorithmic procedures by repeatedly solving their associated decision problems. For this reason, the notion of reducibility may also be employed to determine whether one *computational* problem is at least as hard to solve as another.

Let C_1 and C_2 denote two computational problems. The computational problem C_1 is then polynomial time reducible to the computational problem C_2 if an Algorithm A_3 exists which can solve C_1 and uses, as a subroutine, an algorithm A_4 to solve computational problem C_2, and which runs in polynomial time if the algorithm for computational problem C_2 does [128]. Therefore, if the computational problem C_1 is polynomial time reducible to the computational problem C_2, then C_2 is computationally at least as difficult as C_1 (that is, the computational problem C_1 is no harder than the computational problem C_2). The complexity classes of decision problems in Figure 3.2, the notions of polynomial-time reducibility and the notion of **NP**-hardness may also be generalised to accommodate computational problems in the obvious manner [83].

3.2 The inception of weapon assignment modelling

The art of WA modelling has its roots in the *classical assignment problem* of Votaw and Orden [202] dating back to 1952. This problem entails finding an assignment of n agents to n tasks which minimises the overall cost of the assignment of agents to tasks, where the assignment of each agent to a task is associated with a cost depending on the specific agent and the specific task. Denote the cost of assigning agent i to task j by c_{ij} and consider a binary decision variable x_{ij} which takes the value 1 if agent i is assigned to task j, or 0 otherwise. Then the objective in the classical assignment problem is to

$$\text{minimise} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{3.1}$$

$$\begin{aligned} \text{subject to the constraints} \quad & \sum_{i=1}^{n} x_{ij} = 1, && j = 1,\ldots,n, && (3.2)\\ & \sum_{j=1}^{n} x_{ij} = 1, && i = 1,\ldots,n, && (3.3)\\ & x_{ij} \in \{0,1\}, && i = 1,\ldots,n, && (3.4)\\ & && j = 1,\ldots,n. && \end{aligned}$$

Constraint set (3.2) ensures that each of the n tasks is assigned to exactly one agent, while constraint set (3.3) ensures that each of the n agents is assigned to exactly one task. Note that in the classical assignment problem, the strong assumption is made that the number of agents and the number of tasks are equal[1]. Furthermore, it is assumed that the cost of assigning an agent to a task is independent of which tasks are assigned to the other agents.

The classical assignment formulation (3.1)–(3.4) served as a stepping stone for many WAMs that exist in the military operations research literature today. The first incarnation of a WAM was introduced informally by Flood [69] in 1957, but was only formulated mathematically as the so-called *weapon-target assignment* (WTA) model by Manne [125] in 1958. The model is based on the classical assignment problem (3.1)–(3.4) in which the agents are replaced by m WSs and the tasks are replaced by n targets (or threats). The objective in this model is to ascertain a set of WS-threat assignment pairs which minimises the overall expected survival probabilities of the threats. Let V_j denote the importance of eliminating threat j, let p_{ij} denote the probability that WS i will hit threat j with a single shot and finally, let X_{ij} denote the probability with which WS i is assigned to threat j. Then the objective in the WAM formulation of Manne [125] is to

$$\text{minimise} \quad \sum_{j=1}^{n} V_j \prod_{i=1}^{m}(1 - p_{ij}X_{ij}) \tag{3.5}$$

$$\begin{aligned} \text{subject to the constraints} \quad & \sum_{j=1}^{n} X_{ij} = 1, && i = 1,\ldots,m, && (3.6)\\ & X_{ij} \geq 0, && i = 1,\ldots,m, && (3.7)\\ & && j = 1,\ldots,n. && \end{aligned}$$

Constraint set (3.6) ensures that the probabilities of assigning each WS accumulates to the value one over all the threats, while constraint set (3.7) ensures that the individual probability values

[1] This assumption may be relaxed in the case of m agents and n tasks, where $n \neq m$. If $n > m$, then an agent may be assigned to more than one task. Similarly, if $m > n$, then dummy tasks may be created which results in certain agents being unassigned.

of assigning WS i to threat j is greater than or equal to zero. Note that it is assumed in the objective function (3.5) that the events of hitting a threat by the various WSs are independent of one another — similar to the independence assumption of the costs in the classical assignment problem (3.1)–(3.4). Note also that whereas the classical assignment problem (3.1)–(3.4) is a combinatorial optimisation problem[2] (its decision variables x_{ij} are required to assume binary values), this is not the case in the WTA model (3.5)–(3.7) of Manne [125] (where the decision variables X_{ij} are allowed to assume continuous, nonnegative values). It should be noted here that this early WAM of Manne [125] is known to be **NP**-complete [135].

In his paper [125], Manne acknowledged that the minimand in (3.5)–(3.7) is nonlinear. In order to simplify the problem he suggested that the assumption be made that only uniform WSs are used in the WAM formulation. From a practical point of view, this assumption implies that WSs of the same type are used in the deployment and that they are grouped closely together in such a way that each WS in the deployment achieves the same hit probability p_j with respect to a given threat j. The formulation in which uniform WSs are assumed is known in the military operations research literature as the WTA problem with uniform WSs. This model formulation is similar to the WAM (3.5)–(3.7), except for the objective function which rather aims to

$$\text{minimise} \qquad \sum_{j=1}^{n} V_j \prod_{i=1}^{m} (1 - p_j y_j). \tag{3.8}$$

Note that unlike the case in (3.5)–(3.7), integer decision variables $y_1, \ldots, y_m$ are adopted in the WTA problem with uniform WSs, where y_j represents the number of WSs assigned to threat j. Manne noted in [125] that although the objective function (3.8) is still nonlinear, it is similar to the objective function of the well-known transportation problem[3] which may be solved by linear programming solution techniques.

The research conducted by Manne on the WAP was considered a significant breakthrough in the military defence realm and the WTA problem with uniform WSs has since been used extensively in the strategic planning of defence strategies on the battlefield [125]. Many authors have subsequently extended and improved Manne's original model. These extensions and contributions are described in the following sections.

3.3 WAM contributions from the early 1960s to the mid 1980s

In 1959, Den Broeder *et al.* [53] formulated a WAM in which the aim is to find the number of WSs to assign to a particular threat at a time, which maximises the expected value of destroying at least a given number of threats. Two assumptions are made in the model. First, it is assumed that the probability p_j of hitting threat j with a single shot from a WS is independent of the WS selected and secondly it is assumed, if y_j WSs are assigned to a threat j, that its survival probability is $q_j^{m_j}$, where $q_j = 1 - p_j$ (in other words the events of hitting the threat by various

[2] It is, however, acknowledged that the combinatorial optimisation problem (3.1)–(3.4) is very easy to solve in the sense that its linear programming relaxation yields an optimal solution in which all variable values are already binary [202]. There also exist efficient solution techniques tailor-designed for the model, such as the celebrated Hungarian method [209].

[3] The transportation problem is a type of linear programming problem in which the aim is to find the optimal path of distribution of goods from several points of origin (called *supply points*) to several different destinations (called *demand points*) so as to minimise the cost involved in shipping the goods [209].

WSs are independent). Let a_d denote the probability of destroying exactly d threats where

$$a_d = \prod_{j=1}^{d} q_j^{m_j}$$

and let

$$P_d = \sum_{j=d}^{n} a_d$$

denote the probability of destroying d or more threats. Furthermore, let V_d denote the priority value associated with eliminating exactly d threats. Then the objective in the WAM of Den Broeder *et al.* [53] is to

$$\text{maximise} \quad \sum_{d=1}^{n} V_d a_d \tag{3.9}$$

$$\text{subject to the constraints} \quad \sum_{j=1}^{n} m_j = m, \tag{3.10}$$

$$m_j \in \mathbb{N}_0, \qquad j = 1, \ldots, n, \tag{3.11}$$

where constraint (3.10) ensures that exactly m WSs are assigned to the combined set of threats, while constraint set (3.11) ensures that the number of WSs assigned to threat j assumes a nonnegative integer value.

In 1961, Bradford [21] formulated the WAP as a dynamic program. In his model, the assumption is made that a single type of WS is used, but that the targets are not necessarily of the same type. Suppose there are m WSs available for assignment to n threats. Let $E_j(m_j)$ denote the so-called effective function (or payoff function) representing the effectiveness value for assigning m_j WSs to threat j. Then the objective in the WAM of Bradford [21] is to

$$\text{maximise} \quad \sum_{j=1}^{n} E_j(m_j) \tag{3.12}$$

$$\text{subject to the constraints} \quad \sum_{j=1}^{n} m_j = m, \tag{3.13}$$

$$m_j \in \mathbb{N}_0, \qquad j = 1, \ldots, n. \tag{3.14}$$

Constraint set (3.13) ensures that all the WSs are assigned to the threats in the system, while constraint set (3.14) ensures that the number of WSs assigned to threat j assumes a nonnegative integer value.

In 1966, Day [46] also contributed to the WAP by introducing a three-stage method for assigning WSs to threats. In this model, the aerial threats in the system are partitioned into a number of so-called *threat complexes*. Each of these threat complexes contains a subset of threats that are located within close geographical proximity of one another in such a way that if a WS were to be assigned to one threat within the complex, it is expected to cause some damage to other threats in the same complex but would, however, cause negligible damage to threats in any other complex. The method therefore decomposes the general WAP into two phases: (1) a set of smaller WS-threat assignment problems (*i.e.* a WS-threat assignment problem for each threat complex) are solved during the first phase and (2) a larger WS-threat assignment problem, which combines the results of the smaller problems, is solved during the second phase. The smaller WS-threat complex assignment problems are solved sequentially and the results thus obtained are used to solve the WAP in the larger nonlinear form — the larger WAP is a nonlinear combination of the smaller WA-threat complex problems. Suppose there are n_i WSs of type i available for assignment to q threat complexes in the system. Let D_j denote the expected proportion of total weighted damage to threat complex j and let m_{ij} denote the number of WSs of type i that are assigned to threat complex j. Furthermore, let p_{ij} denote the hit probability of threat complex j if a WS of type i is assigned to it and let W_j denote the weighted damage achievable in threat complex j. Furthermore, let q_{ij} denote the survival probability of threat complex j if a WS of type i is assigned to it, where $q_{ij} = 1 - p_{ij}$. Then the objective in the nonlinear formulation of Day [46] is to

$$\text{maximise} \quad \sum_{j=1}^{q} W_j(q_{1j}^{m_{1j}}, q_{2j}^{m_{2j}}, \ldots, q_{mj}^{m_{ij}}), \tag{3.15}$$

$$\text{subject to the constraints} \quad \sum_{j=1}^{q} m_{ij} = n_i, \qquad i = 1, \ldots, m, \tag{3.16}$$

$$1 - \prod_{i=1}^{n_i} p_{ij}^{m_{ij}} \geq R_j, \qquad j = 1, \ldots, n, \tag{3.17}$$

$$\sum_{i=1}^{n_i} m_{ij} \geq N_j, \qquad j = 1, \ldots, n, \tag{3.18}$$

where R_j is the minimum proportionate damage to complex j desired by the operator and N_j is the minimum number of WSs that have to be assigned to threat complex j.

The 1970s and early 1980s proved to be a seemingly uneventful period in the evolution of WAMs in the sense that no significant contribution was made in respect of developing new WAMs.

3.4 WAM contributions from the late 1980s to the late 1990s

The next major contribution towards WA modelling was the introduction of the first so-called *dynamic WAP formulation* by Hosein and Athans [89] in 1989. The class of dynamic WAPs, also known as *multi-stage problems*, involve the introduction of a temporal period T in the problem formulation. This temporal period is typically partitioned into $\tau \in \{0, \ldots, t\}$ shorter discrete time stages of equal duration. The aim is then to find an assignment of WS-threat pairs during each of the $t + 1$ discrete time stages which maximises some aspiration criterion at the end of the temporal period T under consideration. This temporal aspect of the formulation is the main difference between the static WAPs (discussed in §3.2 and §3.3) and dynamic WAPs. Furthermore, WS-threat assignment pairs are made sequentially in dynamic WAPs and not simultaneously as in static WAPs. WAMs in the dynamic class are therefore able to reserve a WS-threat assignment pair for a later time stage during which the WS may achieve a higher efficiency value with respect to the threat than during the current time stage. Dynamic WAPs are formulated in two fundamentally different ways in the literature: (1) by assuming that the number and locations of the threats are known in advance for each time stage, and (2) by assuming that not all the threats are known at the start of the temporal period (*i.e.* the number and the locations of the threats are known only stochastically).

The dynamic WAM formulation of Hosein and Athans [89] is known in the military operations research literature as the *shoot-look-shoot* WAM. As the name indicates, the model is based on a shoot-look-shoot strategy in which the outcomes of WS-threat assignment pairs are observed (perfectly) during every time stage, before WS-threat assignment pairs are proposed for the next time stage. The WAP is therefore restated during every time stage in the time continuum under consideration. The first stage in the dynamic WAM entails finding a subset of the available WSs and assigning them to the observed threats in the system. These WSs are all assigned to engage the threats simultaneously during the first time stage. The next step involves assessing the outcome of the engagements during the first time stage, ascertaining the subset of remaining threats (if any threat(s) were eliminated during the first time stage) and selecting a subset of available WSs at the beginning of the second time stage to assign to the remaining subset of threats during the second time stage. This process is repeated for each successive time stage τ in the time continuum under consideration. The aim of the model is to minimise the accumulated survival probabilities of the threats, weighted by their respective estimated threat priority values, at the end of the temporal period T under consideration. The only information therefore required by the model during every time stage are the available number of WSs, the remaining subset of threats (together with their updated positions) and the remaining time stages in the time continuum — this implies that the shoot-look-shoot WAM may be implemented as a *dynamic programming model*[4]. This methodology yields an optimal solution in terms of WS-threat assignment pairs for the first time stage although assignments over the entire temporal period T have to be taken into account. The methodology is reinstated during every time stage and solved for the remainder of the time stages — hence, each of the remaining time stages (iteratively taken as the initial interval) will yield an optimal solution over the remainder of the entire time continuum.

The model functions under a number of assumptions. The first assumption is that the number of threats, as well as their locations, are known in advance. Next, it is assumed that the outcomes of the WS-threat pair assignments during each time stage (*i.e.* hits or misses) are observed (perfectly) before WS-threat assignment pairs for the next time stage are considered. Consequently, it is assumed that the duration of each time stage is long enough in order for each WS to engage the threat to which it has been assigned and to observe (perfectly) the outcomes of these WS-threat assignment pair engagements. Furthermore, it is assumed that a WS may only be used once during the entire temporal period T. The WAM therefore returns an optimal solution of WS-threat pairs for the entire temporal period T under consideration in which each WS is utilised during at most one stage τ.

Let $p_{ij}(\tau)$ denote the probability that WS i will hit threat j with a single shot if WS i is assigned to threat j during time stage τ and let the corresponding survival probability be $q_{ij}(\tau) = 1 - p_{ij}(\tau)$. Consider a binary decision variable $x_{ij}(\tau)$ which takes the value 1 if WS i is assigned to threat j during time stage τ, or zero otherwise. Define the *threat state*, denoted by an n-dimensional binary vector $\boldsymbol{u} \in \{0,1\}^n$ with unit entries corresponding to the set of surviving threats, and the *weapon state*, denoted by an m-dimensional vector $\boldsymbol{w} \in \{0,1\}^m$ with unit entries corresponding to the set of WSs available for assignment after the previous time stage, $\tau - 1$. Here again, n denotes the number of threats and m denotes the number of available WSs to assign to the threats. Therefore, u_j is 1 if threat j survives the previous stage, or 0

[4]Dynamic programming is an algorithmic paradigm in which a large unwieldy problem is partitioned into a number of so-called *temporal stages* where a decision is required during each stage — during each temporal stage, a small part of the problem is therefore solved optimally and this solution is stored. The solution to the problem is obtained by starting at the last stage and working backwards iteratively until an optimal solution is found. During each iteration of the program, the problem is enlarged slightly by adding a stage (one-at-a-time) and solving the problem optimally by using the optimal solutions from the previous stage. The program is iterated in this fashion until all the stages are included in the problem — this should represent the original large problem.

otherwise, and similarly w_i is 1 if WS i has not been assigned up to the end of the previous stage, or 0 otherwise. Since the probability distribution of u_j depends on the probability that WS i will hit threat j during the first time stage, the probability that threat j survives or does not survive time stage $\tau - 1$ is given by

$$P(u_j = k_j) = k_j \prod_{i=1}^{m}[q_{ij}(\tau-1)]^{x_{ij}(\tau)} + (1-k_j)\left(1 - \prod_{i=1}^{m}[q_{ij}(\tau)(\tau-1)]^{x_{ij}(\tau)}\right), \tag{3.19}$$

where $k_j \in \{0,1\}$. Let $\boldsymbol{k} = [k_1, \ldots, k_m]$. Then the probability of observing the state vector $\boldsymbol{k}$ is given by

$$P(\boldsymbol{u} = \boldsymbol{k}) = \prod_{j=1}^{n} P(u_j = k_j), \tag{3.20}$$

where the assumption is made that the state vector entries are independent. The objective during time stage $\tau \in \{1, \ldots, t\}$ of the dynamic shoot-look-shoot WAM of Hosein and Athans [89] is to

$$\text{minimise} \quad \sum_{\boldsymbol{k}\in\{0,1\}^n} P(\boldsymbol{u} = \boldsymbol{k}) F_\tau^*(\boldsymbol{k}, \boldsymbol{w}) \tag{3.21}$$

$$\text{subject to the constraints} \quad \sum_{j=1}^{n} x_{ij}(\tau) = 1 - w_i, \qquad i = 1, \ldots, m, \tag{3.22}$$

$$x_{ij}(\tau) \in \{0,1\}, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n, \tag{3.23}$$

where $F_\tau^*(\boldsymbol{k}, \boldsymbol{w})$ denotes the optimal assignment cost for a $(\tau-1)$-interval problem with an initial threat state $\boldsymbol{k}$ and initial weapon state $\boldsymbol{w}$. When $\tau = t$, the final-stage expected cost

$$F_T^*(\boldsymbol{k}, \boldsymbol{w}) = \sum_{j=1}^{n} V_j u_j \tag{3.24}$$

is the accumulated threat priority values of the threats that survived to the final stage.

Furthermore, constraint set (3.22) ensures that a WS is assigned only once during the temporal period T, while constraint set (3.23) ensures the binary nature of the decision variables. It should be noted here that the dynamic shoot-look-shoot WAM (3.21)–(3.23) is computationally expensive to solve since it is essentially required that a static WAM be solved during every time stage in the time continuum under consideration.

Although the above formulation is generaly considered to be a dynamic WAM, Huaiping *et al.* [91] rightly criticised the model by pointing out that it merely involves solving a static WAM repeatedly. It is therefore not able to take the entire time continuum into consideration by looking ahead and taking a number of future time stages into account during which WS-threat pair assignments are proposed.

3.5 WAM contributions from the early 21st century

In 2000, Murphey [135] also made a substantial contribution to the class of dynamic WAMs by formulating the so-called *stochastic demand dynamic* WAM. It differs from the dynamic WAM

(3.21)–(3.23) of Hosein and Athans [89] in the sense that the number of threats in the system and their positions are unknown in advance — only a subset of the threats and their positions are known during each time stage in a discretisation of the temporal period under consideration. The model is formulated in such a way that an assignment of available WSs may be made to one of the known threats, or the assignment may be postponed to occur at a later stage in favour of a threat that is yet unknown, but which may be detected in the future. Cost coefficients which are monotonically increasing functions of time are incorporated in the objective function, penalising the prolonged assignment of WSs. In essence, the model aims to find a balance between proposing WS assignments to currently known threats immediately or rather reserving WS ammunition for as yet, undetermined future threats.

Suppose $n(\tau)$ threats and their positions are known during time stage τ. Let $c(\tau)$ denote a monotonically increasing cost function which aims to penalise the postponement of the assignment of a WS. Following the same notation as in the previous WAMs discussed, the objective in the stochastic demand dynamic WAM is to

$$\text{minimise} \quad \sum_{\tau=1}^{t} c(\tau) \sum_{j=1}^{n(\tau)} V_j \prod_{i=1}^{m} q_{ij}(\tau)^{x_{ij}(\tau)} \tag{3.25}$$

subject to the constraints

$$\sum_{\tau=1}^{t} \sum_{j=1}^{n(\tau)} x_{ij}(\tau) = 1, \qquad i = 1, \ldots, m, \tag{3.26}$$

$$x_{ij}(\tau) \in \{0, 1\}, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n, \; \tau = 1, \ldots, t, \tag{3.27}$$

where constraint set (3.26) ensures that each WS is assigned exactly once during the entire time continuum under consideration and constraint set (3.27) ensures the binary nature of the decision variables.

In [135], Murphey also suggested that the stochastic demand dynamic WAM (3.25)–(3.27) may be simplified by assuming the use of uniform WSs. This change results in the simplified decision variable $y_j(\tau)$ denoting the number of WSs which are assigned to threat j during time stage τ. This variable may therefore assume any integer value in the range $[0, m]$. Furthermore, let $q_j(\tau)$ denote the survival probability of threat j during time stage τ. Adopting the same notation as in the stochastic demand dynamic WAM (3.25)–(3.27), the objective when using uniform WSs is to

$$\text{minimise} \quad \sum_{\tau=1}^{t} c(\tau) \sum_{j=1}^{n(\tau)} V_j q_j(\tau)^{y_j(\tau)} \tag{3.28}$$

subject to the constraints

$$\sum_{\tau=1}^{t} \sum_{j=1}^{n(\tau)} y_j(\tau) \leq m, \qquad i = 1, \ldots, m, \tag{3.29}$$

$$y_j(\tau) \in \mathbb{N}_0, \qquad j = 1, \ldots, n, \; \tau = 1, \ldots, t, \tag{3.30}$$

where constraint set (3.29) ensures that no more than m WSs are assigned to the threats during the time continuum under consideration and constraint set (3.30) ensures that the decision variables assume nonnegative, integer values.

In 2003, Ahuja *et al.* [3] contributed to the class of static WAMs by adapting the WAM (3.5)–(3.7) of Manne [125] to make provision for the use of different types of WSs. Let W_i denote the number of WSs of type i available for assignment. Following the same notation as in the previous models, the objective in the WAM of Ahuja *et al.* [3] is to

$$\text{minimise} \quad \sum_{j=1}^{n} V_j \prod_{i=1}^{m} q_{ij}^{x_{ij}} \tag{3.31}$$

subject to the constraints

$$\sum_{j=1}^{n} x_{ij} \leq W_i, \qquad i = 1, \ldots, m, \tag{3.32}$$

$$x_{ij} \in \{0, 1\}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n, \tag{3.33}$$

where constraint set (3.32) ensures that no more than the available number of WSs of type i is assigned to threats and constraint set (3.33) ensures that the decision variables assume binary values.

In 2006, Du Toit [63] contributed to the class of dynamic WAMs by formulating the so-called *WA Scheduling Problem* (WASP). This model formulation is based on the stochastic demand model of Murphey [135], but it is assumed that the SSHP values achieved by the WSs with respect to the threats are independent of one another and hence the cost coefficients may be omitted, resulting in a simpler objective function than the one in Murphey's formulation. The WASP formulation of Du Toit [63] is therefore the same as the stochastic demand model (3.25)–(3.27) except for the objective function. Adopting the same notation as before, the survival probability of threat j over the time interval $[1, \bar{\tau}]$ is

$$\prod_{\tau=1}^{\bar{\tau}} \prod_{i=1}^{m} q_{ij}(\tau)^{x_{ij}(\tau)}, \tag{3.34}$$

so that the objective function aims to

$$\text{minimise} \quad \sum_{\tau=1}^{t} \sum_{j=1}^{n(\tau)} V_j \prod_{\tau=1}^{\bar{\tau}} \prod_{i=1}^{m} q_{ij}(\tau)^{x_{ij}(\tau)}. \tag{3.35}$$

The constraint sets of the model (3.35) are the same as constraint sets (3.26)–(3.27).

In 2008, Potgieter [145] further extended the static model of Manne to formulate the so-called *κ-WA problem* in which the original model is constrained to allow for a maximum of κ WSs to be assigned to any threat. Following the same notation as in the original WAM formulation (3.5)–(3.7) of Manne [125], the objective function in the κ-WA problem is to

$$\text{minimise} \quad \sum_{j=1}^{n} V_j \prod_{i=1}^{m} q_{ij}^{x_{ij}} \tag{3.36}$$

subject to the constraints

$$\sum_{j=1}^{n} x_{ij} \leq 1, \qquad i = 1, \ldots, m \tag{3.37}$$

$$\sum_{i=1}^{m} x_{ij} \leq \kappa, \qquad j = 1, \ldots, n, \tag{3.38}$$

$$x_{ij} \in \{0, 1\}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n, \tag{3.39}$$

where constraint set (3.37) ensures that each WS may be assigned at most once, constraint set (3.38) ensures that no more than κ WS are assigned to any threat in the defended airspace and constraint set (3.7) ensures the binary nature of the decision variables.

Potgieter [145], also adjusted the WASP formulation of Du Toit in order to make provision for the inclusion of non-renewable WSs and to allow for a maximum of κ WSs to be assigned to any given threat (similar to the extension in the κ-WA WAM (3.36)–(3.39)) over the entire temporal period under consideration. He also contributed a so-called *multi-period WAP* formulation which is an extension of the k-WAP, and is based on the same fundamental approach as that of the shoot-look-shoot model of Hosein and Athans [89].

Another single-objective, dynamic formulation of the WAP is due to Du Toit [64], who, in 2009, formulated the so-called *expected threat priority accumulation dynamic weapon target assignment* model. This model follows the same fundamental approach as the shoot-look-shoot model of Hosein and Athans [89], but with the difference that earlier engagements by WSs are encouraged by assigning higher priorities to these assignments during the evaluation of the expected survival probabilities of the threats within each time stage, rather than only during the final time stage of the temporal period under consideration, as is the case in the original shoot-look-shoot model. Let τ_α and τ_ω denote the first and last of the discretised time stages, respectively, in the time continuum under consideration. Furthermore, let $n(\tau_\alpha)$ denote the number of threats considered during the first time stage, let $m(\tau_\alpha)$ denote the number of WSs available for assignment during the first time stage and let $\bar{\phi}_i$ denote the number of initial time stages during which WS i is not available for assignment as a result of an assignment prior to time stage τ_α. If WS i is assigned to engage a threat, let ω_i denote the number of time stages (that follow after the assignment) during which WS i cannot engage another threat. The objective in the expected threat priority accumulation WAM is then to

$$\text{minimise} \quad \sum_{\bar{\tau}=\tau_\alpha}^{\tau_\omega} \sum_{j=1}^{n(\tau_\alpha)} V_j(\bar{\tau}) \prod_{\tau=\tau_\alpha}^{\bar{\tau}} \prod_{i=1}^{m(\tau_\alpha)} q_{ij}(\tau)^{x_{ij}(\tau)} \tag{3.40}$$

subject to the constraints

$$\sum_{j=1}^{n(\tau_\alpha)} \sum_{\tau=\tau_\alpha}^{\tau_\alpha-1+\bar{\phi}_i} x_{ij}(\tau) = 0, \qquad i = 1,\dots,m(\tau_\alpha),\ \tau = \tau_\alpha,\dots,\tau_\omega, \tag{3.41}$$

$$\sum_{j=1}^{n(\tau_\alpha)} \sum_{\tau=\tau_\alpha+\bar{\phi}_i}^{\tau_\omega} x_{ij}(\tau) \leq 1, \qquad i = 1,\dots,m(\tau_\alpha),\ \tau = \tau_\alpha,\dots,\tau_\omega, \tag{3.42}$$

$$x_{ij}(\tau) \in \{0,1\}, \qquad i = 1,\dots,m(\tau_\alpha),\ j = 1,\dots,n(\tau_\alpha),\ \tau = \tau_\alpha,\dots,\tau_\omega. \tag{3.43}$$

Constraint set (3.41) ensures that WS i is not assigned for engagement during the first $\bar{\phi}_i$ time stages, while constraint set (3.42) ensures that WS i is not considered for assignment more than once during the remaining time intervals in the time continuum under consideration. Constraint set (3.43) finally ensures the binary nature of the decision variables.

In 2013, Lötter *et al.* [120] added a new dimension to the WAP in which they modelled the WAP as a bi-objective decision problem, hence contributing towards the class of multi-objective WAMs. They consulted six military experts from the South African National Defence Force in

No	Objective	Possible constraints
1	Minimise aggregated survivability of observed aerial threats as a result of assignments, weighted by the priorities of eliminating the threats	Maximum number of WSs that may be assigned
2	Minimise aggregated cost of assignments	Budget threshold on accumulated assignment cost
3	Minimise the maximum engagement time in the assignment	Maximum length threshold for fire windows or minimum SSHP threshold within fire windows
4	Maximise minimum number of times a WA can re-engage after the assignment	Number of available ammunition rounds at WSs

TABLE 3.1: *A list of possible objectives and constraints from which to populate WAMs [120]*

a bid to identify possible objectives deemed important when proposing WS-threat assignment pairs. During the consultation process, the military experts were requested to complete an electronic questionnaire consisting of a set of questions relating to various possible factors which may influence the choice of WSs to assign to threats. After carefully analysing the results from these questionnaires, a number of possible objectives were derived for use in the formulation of WAMs. A subset of these objectives, as well as possible constraints related to each of these objectives, is listed in Table 3.1. Lötter *et al.* [120] selected two objectives from this list in order to formulate a static, bi-objective WAM in which the aim is to minimise the accumulated survival probabilities of the threats as well as to minimise the cost of the ammunition of the WSs used in the assignments. Let c_{ij} denote the cost of assigning WS i to threat j. Then, following the same notation as in the previous WAMs, the objective in the bi-objective WAM is to

$$\text{minimise} \quad \sum_{j=1}^{n} V_j \prod_{i=1}^{m} q_{ij}^{x_{ij}} \tag{3.44}$$

and

$$\text{minimise} \quad \sum_{i=1}^{m} c_{ij} \sum_{j=1}^{n} x_{ij} \tag{3.45}$$

subject to the constraints

$$\sum_{j=1}^{n} x_{ij} \leq 1, \qquad i = 1, \ldots, m \tag{3.46}$$

$$\sum_{i=1}^{m} x_{ij} \leq \kappa, \qquad j = 1, \ldots, n, \tag{3.47}$$

$$x_{ij} \in \{0, 1\}, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n. \tag{3.48}$$

Constraint sets (3.46)–(3.48) may be interpreted in the same way as constraint sets (3.37)–(3.39).

In 2014, Van der Merwe [195] put forward a single-objective, dynamic WAM which is able to schedule a subset of consecutive future time stages (*i.e.* time windows) during which WS-threat assignments should occur. This model characteristic distinguishes it from the dynamic WAMs preceding it in the sense that the model is able to look ahead in time and consider subsets of future time stages during which assignments may occur in answer to the criticism by Huaiping *et al.* [91] directed at the shoot-look-shoot model of Hosein and Athens [89]. The model formulation is based on the fundamentals of the classical *Vehicle Routing Problem with Time Windows* (VRPTW) in which the WSs are modelled as virtual vehicles which deliver commodities (ammunition) to customers (threats) within specific pre-specified time windows (fire windows). The aim in Van der Merwe's model is to minimise the total accumulated "dissatisfaction" associated with delivering the ammunition to the threats (measured in terms of the survival probabilities of the threats). The model therefore not only brings a dynamic element to the WAP by considering *when* in the future WS-threat assignment pairs should occur, but also a scheduling element indicating the subset of consecutive future time stages during which a specific WS-threat pair assignment should ideally occur.

As mentioned, the model has its roots in the celebrated classical VRPTW, which is a well-known combinatorial optimisation problem and has been researched extensively in the operations research literature [186]. It involves finding an optimal composition of a fleet of heterogeneous vehicles, as well as a specific set of routes along which each of these vehicles should travel, in order to serve a set of customers with known demands, simultaneously [186]. A typical objective adopted in vehicle routing problems is to minimise the accumulated service cost of delivering commodities to customers. This service cost is typically calculated as the total travel cost incurred by all the vehicles [38]. In the VRTPW, each customer is associated with a so-called service time frame or time window which is a subset of feasible consecutive time stages in the time continuum under consideration during which the customer has to be served [174]. In addition, each customer is also associated with a service time duration which is the total number of time stages that the customer requires to be served. Furthermore, a central depot is also included in the problem which acts as a *base* from which each vehicle route is required to depart and return to. The depot is associated with a so-called *scheduling horizon* (similar to the time window associated with each customer) during which all customers are allowed to be served. Furthermore, a vehicle is allowed to serve more than one customer along a specific route, but a time constraint ensures that the time stage during which service at a particular customer ends plus the number of time stages which is required by a vehicle to travel to the next customer on the route should neither precede the earliest time stage nor exceed the latest time stage during which service may start at the next customer. This constraint ensures that the service windows of customers served consecutively along the same route by some vehicle do not overlap in time. Finally, the commodity carrying capacity of a vehicle assigned to a route should equal or exceed the demand of the customers on that route.

In the context of a GBAD system, Van der Merwe [195] modelled the WSs as vehicles having to deliver ammunition (commodities) to the threats (customers) in order to conform to the classical VRPTW modelling approach. Her approach towards the scheduling of successive engagements of threats by WSs over time may be interpreted as the scheduling of virtual vehicles (the WSs) delivering real commodities (the ammunition) to real customers (the aerial threats) as in the VRPTW. In the VRPTW, the vehicles move between the customers to deliver the commodities requested by the customers, but in the WAP within a GBAD context, the WSs are stationary. Therefore, the WSs may be thought of as virtual vehicles which have to service threats by delivering WS ammunition to the threats.

Furthermore, the assumption is made that a WS delivers a single unit of ammunition to a threat if it is utilised to service a threat. This single unit of ammunition represents the number of rounds[5] of ammunition fired at a threat. In addition, the WAP involves a time delay which represents

[5]The number of rounds depends on the specific type of WS used. In the case of an Umkhonto missile, for example, this number is typically one unit, whereas in the case of automated WSs, such as 35 mm cannons, this number entails a pre-determined quantity due to these systems being able to fire in a setting known as *burst mode*.

the number of (consecutive) time stages from when an engagement order is sent until the time stage during which the shot is fired at the threat. This time delay is known as the *weapon setup time* (analogous to the time is takes for a vehicle to travel between customers in the classical VRPTW) and includes the number of time stages it takes for the FCO to send the engagement order, for the WS operator to respond to the engagement order and for him to load the WS with ammunition, to turn the muzzle of the WS and to track the threat before the engagement occurs. In the single-objective, dynamic WA scheduling model of Van der Merwe [195], it is assumed that the travel time (*i.e.* the weapon setup time) is fixed at a pre-determined number of time stages.

The main difference between the single objective, dynamic WA scheduling model formulation and the classical VRPTW formulation is the way in which the objective functions are constructed. In the original VRPTW formulation, the objective function aims to minimise the accumulated cost of the vehicles used, which is typically associated with the combined distance that the vehicles have to travel. The model therefore requires a distance matrix containing the distances between all the customers in order to assign these vehicles to routes which minimise the overall distance covered by the vehicles. In the WAM formulation, however, the WSs are stationary and the distance over which a WS needs to travel is not considered in the objective function. Instead, an *Engagement Efficiency Matrix* (EEM) is employed which contains the probability values that the threats will be hit when WSs are assigned to them during the various time stages. Finally, the capacity of each vehicle in the WAP is taken as the number of units of ammunition available for use by each WS.

Each threat in the WAP is associated with a number of so-called engagement *fire windows* (FWs) analogous to the single time window of a customer in the VRPTW. Each FW is delimited by an earliest time stage, known as the *first-time-to-fire* (FTTF), during which a WS may start to engage a threat as well as a latest time stage, known as the *last-time-to-fire* (LTTF), after which a WS is no longer allowed to engage the threat. The number of time stages from the FTTF to the LTTF is known as the *length* of the FW. A WS is allowed to engage a threat during any time stage within the FW (including the FTTF and LTTF). A WS is, however, required to achieve a positive SSHP value during *each* time stage within the FW of the threat under consideration. It is therefore presumed that the set of FWs associated with a WS-threat pair is induced by a combination of terrain surface masking, meteorological conditions, the position of the WS and the predicted flight path of the threat.

Furthermore, if a WS is assigned to engage more than one threat during the scheduling horizon, the FWs of these threats are not allowed to overlap. The WS setup time should also be taken into account when considering a WS for assignment in a particular threat's FW by ensuring that the number of time stages preceding the FW of the WS-threat pair is equal to or exceeds the WS setup time of that WS. Consider the following example in which the notion of a FW is illustrated for a single WS-threat pair.

Example 3.1 *Consider the assignment of a single WS to a particular aerial threat. Suppose the scheduling horizon comprises eleven time stages with SSHP values as shown in Figure 3.3, that the minimum length of any FW in the scheduling horizon is three time stages and that the WS setup time is three time stages. Note that the smallest possible value for the FTTF in the example is time stage 5, due to the weapon setup time (i.e. three time stages). One example of a FW has a FTTF at time stage 5 and a LTTF at time stage 10, which results in a FW length of six time stages, as illustrated graphically by means of a dashed arrow at the top of Figure 3.3. This example also allows for nine other possible FWs with (FTTF, LTTF)-pair values given by (5,7), (6,8), (7,9), (8,10), (5,8), (6,9), (7,10), (5,9) and (6,10), respectively, as indicated in the* *figure. Note that the lengths of the FWs are always at least the specified minimum length of a FW, namely three time stages.* ■

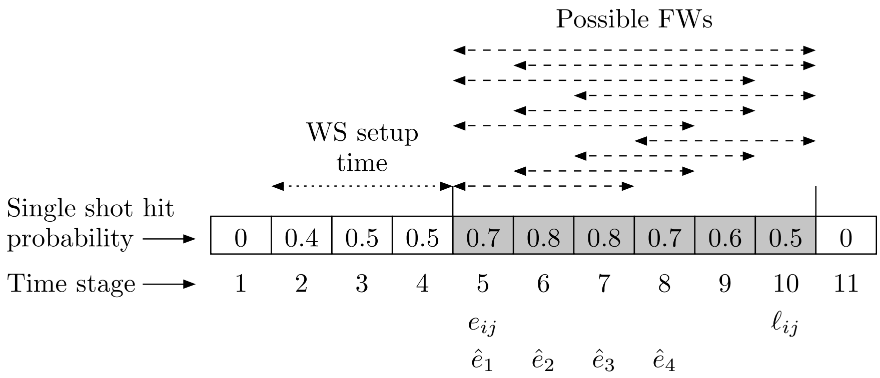


FIGURE 3.3: *A number of different FWs for a single WS-threat pair.*

It is clear from Example 3.1 and Figure 3.3 that there are various combinations of FWs for each WS-threat pair. Define, for each WS-threat pair, a time interval $[e_{ij}, \ell_{ij}]$[6], starting at the beginning of stage e_{ij} and ending at the end of stage ℓ_{ij}. During this interval, a number of possible FWs have various FTTFs and LTTFs (as illustrated in Example 3.1) for the pair during which engagement of threat j by WS i may commence and end. Let L_i^{min} denote the minimum pre-specified length of a FW for WS i and let $L_{ij} = \ell_{ij} - e_{ij} + 1$. Then, for each WS-threat pair (i, j), there are $F_{ij} = L_{ij} - L_i^{min} + 1$ different feasible FTTF stages $\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_{F_{ij}}$ having values $e_{ij}, e_{ij} + 1, \ldots, \ell_{ij} - L_i^{min} + 1$, respectively, and for each different FTTF z there are $\ell_{ij} + 1 - \hat{e}_z - (L_i^{min} - 1)$ different LTTF stages. It may therefore be deduced that there are

$$f_{ij} = \sum_{z=1}^{F_{ij}} (\ell_{ij} + 1 - \hat{e}_z - (L_i^{min} - 1)) \qquad (3.49)$$

distinct possible FWs for each WS-threat pair (i, j) [196].

As in the VRPTW, a virtual depot is included in the WAM which may be seen as an artificial construct representing an idle state during which no WS engages any threats. Let the virtual depot be indexed by both threat 0 and threat $n + 1$, where n is the number of threats in the system. The depot has a demand of zero and the system is required to start from this state and to return to it again after all WS-threat pair engagements have been carried out. Let d_i denote the WS setup time of WS i. Define a FTTF e_{ijk} and a LTTF ℓ_{ijk} for WS i when engaging threat j during the pair's k^{th} FW. Let s_{ijk} denote the engagement time duration[7] of threat j by WS i during the pair's k^{th} FW, where $s_{ijk} = \ell_{ijk} - e_{ijk} + 1$, and let f_{ij} be the number of distinct FWs

[6]The value for e_{ij} is typically taken as the first possible stage during which a WS-threat pair may be assigned (*e.g.* time stage 5 in Example 3.1) and the value for ℓ_{ij} is typically taken as the last possible stage during which a WS-threat pair may be assigned (*e.g.* time stage 10 in Example 3.1).

[7]The engagement time duration refers to the number of time stages required by a WS to guide its ammunition in the direction of the threat from the time stage during which the ammunition is launched (the time stage during which the trigger is pulled) until the time stage during which the ammunition hits the threat. This is only applicable to WSs which require operator guidance, such as cannons which are able to fire multiple bursts with the pull of the trigger, but require manual WS operator assistance to guide the ammunition in the direction of the threat. In some instances, WSs have on-board guidance systems, such as certain missiles, and hence require no guidance from the WS operator, in which case the value s_{ijk} is assumed to be zero. The value s_{ijk} is nevertheless included in the model formulation for the sake of flexibility.

for WS-threat pair ij. Furthermore, let p_{ijk} denote the SSHP value associated with threat j if engaged by WS i during the pair's k^{th} FW, and let $q_{ijk} = 1 - p_{ijk}$.

A binary decision variable x_{ijk} is adopted in the formulation which takes the value 1 if WS i engages threat j during the k^{th} FW associated with the WS-threat pair, or a value of 0 otherwise. A binary auxilliary variable y_{ihj} is also incorporated, which may be interpreted as a vehicle flow variable, and takes a value of 1 if threat h directly precedes threat j in a sequence of engagements by WS i, or a value of 0 otherwise. Adopting the same notation as used in the previous WAMs discussed in this section, the objective in the single-objective, dynamic WAM is to

$$\text{minimise} \quad \sum_{j=1}^{n} V_j \prod_{i=1}^{m} \prod_{k=1}^{f_{ij}} (q_{ijk})^{x_{ijk}} \tag{3.50}$$

subject to the constraints

$$\sum_{i=1}^{m}\sum_{h=0}^{n+1} y_{ihj} \leq \kappa, \qquad j = 1, \ldots, n, \tag{3.51}$$

$$\sum_{h=1}^{n+1} y_{i0h} = 1, \qquad i = 1, \ldots, m, \tag{3.52}$$

$$\sum_{h=1}^{n+1} y_{i,h,n+1} = 1, \qquad i = 1, \ldots, m, \tag{3.53}$$

$$\sum_{h=0}^{n+1} y_{ihj} - \sum_{h=0}^{n+1} y_{ijh} = 0, \qquad i = 1, \ldots, m, \; j = 0, \ldots, n+1, \tag{3.54}$$

$$\sum_{k=1}^{f_{ij}} x_{ijk} = \sum_{\substack{h=1 \\ h \neq j}}^{n+1} y_{ihj}, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n, \tag{3.55}$$

$$\sum_{k=1}^{f_{ih}} (e_{ihk} + s_{ihk}) x_{ihk} - \sum_{k=1}^{f_{ij}} e_{ijk} x_{ijk} + d_i < (1 - y_{ihj}) L, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n, \; h = 1, \ldots, n, \tag{3.56}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{f_{ij}} x_{ijk} \leq A_i, \qquad i = 1, \ldots, m, \tag{3.57}$$

$$y_{ihj} \in \{0, 1\}, \qquad i = 1, \ldots, m, \; j = 0, \ldots, n+1, \; h = 0, \ldots, n+1, \tag{3.58}$$

$$x_{ijk} \in \{0, 1\}, \qquad i = 1, \ldots, m, \; j = 0, \ldots, n+1, \; k = 1, \ldots, f_{ij}. \tag{3.59}$$

Constraint set (3.51) ensures that at most κ WSs are assigned to engage any threat over the scheduling horizon. Constraint set (3.52) ensures that WS i "leaves the depot" (idle state) exactly once, if it is assigned to engage threats at all, while constraint set (3.53) ensures that WS i "returns to the depot" exactly once after being used to engage threats. Constraint set (3.54) ensures, if a threat is serviced by WS i, that the WS "leaves the threat" again in order to "move on" to engage the next threat assigned to it. Constraint set (3.55) ensures, if threat h precedes threat j for engagement by WS i, that threat h is engaged during exactly one stage. If threat h is engaged by WS i directly before threat j, constraint set (3.56) ensures that the time stage during which the engagement of threat h starts plus the time it takes to engage threat h plus the time it takes for WS i to "travel" from threat h to threat j does not exceed the time stage during which engagement of threat j starts, where L is a large number. Furthermore, constraint set (3.56) also ensures that the stage during which WS i engages threat j is within a FW associated with the (WS, threat)-pair. Constraint set (3.57) ensures that the capacity of WS i is not exceeded and finally, constraint sets (3.58)–(3.59) ensure the binary nature of the decision and auxilliary variables.

3.6 Chapter summary

The aim in this chapter was to present a review of the existing WAMs in the military operations research literature since the inception of WA modelling during the early 1950s until the current state of the art of WAMs during the early 21$^{\text{st}}$ century. These WAMs were presented in chronological order.

The chapter opened in §3.1, with a brief introduction to the notion of **NP**-completeness since even the simplest WAMs presented in this chapter are **NP**-complete. Section §3.2 opened with a description of the classical assignment problem of Votaw and Orden [202] which was formulated in 1952. The formulation of the assignment problem formed the cornerstone for Manne [125] on which he formulated the first WAM in 1958. A detailed description of this formulation followed. The WAM formulation of Manne [125] paved the way for many authors to contribute towards the formulation of WAMs in many different incarnations. The remainder of the chapter was partitioned into three sections and each of these sections contained similar descriptions of WAMs put forward from the late 1950s up to the current state of WAMs in the military operations research literature.

In §3.3, a number of WAM contributions from the early 1960s to the mid 1980s were described in some detail. Hosein *et al.* [89] made a substantial contribution to the existing WAM literature in 1989 by formulating the first dynamic WAM. This formulation, as well as dynamic WAMs contributed by other authors from 1989 up to the late 1990s, were reviewed in §3.4. In §3.5, the focus shifted to brief descriptions of WAM contributions during the early 21$^{\text{st}}$ century.

Part II

Mathematical prerequisites

CHAPTER 4

Multi-objective optimisation

Contents

This chapter is devoted to a literature review on fundamental concepts related to multi-objective optimisation. First, the notion of multi-objective optimisation is described in general in §4.1 and a general mathematical formulation of a multi-objective optimisation problem is given. Two search spaces associated with multi-objective optimisation problems are also discussed in this section. The focus in the remainder of the chapter is on various important principles involved in multi-objective optimisation and broadly follows the general exposition in the opening chapters of Deb's influential text *Multi-objective optimisation using evolutionary algorithms* [50], supplemented by discussions on the contributions of other authors.

In §4.2, the importance of the requirement of convexity of multi-objective optimisation problems is discussed since such convexity plays an important role in the computational effort expended in solving multi-objective optimisation problems effectively. In §4.3, the notion of solution dominance is introduced and the properties of the dominance relation are discussed briefly. The notion of Pareto optimality is also reviewed in this section.

In §4.4, the focus shifts to a discussion on the notion hypervolume in objective space as a measure of nondominated front quality. This is followed, in §4.5, by a description of sufficient conditions for a solution to a multi-objective optimisation problem to be Pareto optimal and a description of necessary conditions for a solution to be Pareto optimal.

Descriptions of three methods, ranging in different levels of complexity, for computing nondominated sets of solutions to multi-objective optimisation problems may be found in §4.6. This is followed in §4.7 by a discussion on the well-known fast nondominated sorting algorithm which may be used to partition a finite set of candidate solutions to a multi-objective optimisation problem into different classes according to their degree of dominance. A number of multi-objective optimisation techniques from the operations research literature require such a partition of the set of solutions.

Section 4.8 contains a discussion on problems that may arise in the classical approach of weighting each of the objectives in a multi-objective optimisation problem so as to scalarise the optimisation problem. The chapter finally closes in §4.9 with a brief summary of the chapter contents.

4.1 Multi-objective problem formulation

An *optimisation* problem is one in which feasible solutions corresponding to extremal values of one or more objectives are sought [50]. When there is only one objective function in such a problem, the problem is known as a *single-objective optimisation* problem. When multiple objectives are, however, present, the problem is known as a *multi-objective optimisation* problem [39, 48]. Multi-objective optimisation problems typically involve a number of conflicting objectives that have to be optimised simultaneously. The notion of multi-objective optimisation is evident in many practical decision making problems — *e.g.* an individual who is in the market for a new vehicle and has the following two conflicting objectives: to minimise the purchase price of the new vehicle and to maximise the reliability of the new vehicle. In contrast with a single-objective optimisation problem, where the aim is to find a single, globally optimal solution to the problem, a single solution which simultaneously optimises all of the objectives in a multi-objective problem typically does not exist if the objectives are conflicting [116]. Instead, the aim is to find a set of high-quality solutions which achieve acceptable compromises between the objective function values.

Denote a solution to a multi-objective optimisation problem by $\boldsymbol{x} = [x_1, \ldots, x_n]$, where n represents the number of decision variables. Moreover, let $\mathcal{D}$ denote the set of all such solutions to the multi-objective optimisation problem, called the *decision space* of the problem. Suppose M objective functions are considered in the multi-objective optimisation problem and that there are two types of constraints: J *inequality* constraints and K *equality* constraints. Then the general form of such a multi-objective optimisation problem is to

$$\text{minimise/maximise} \qquad z_m = f_m(\boldsymbol{x}), \qquad m = 1, \ldots, M, \tag{4.1}$$

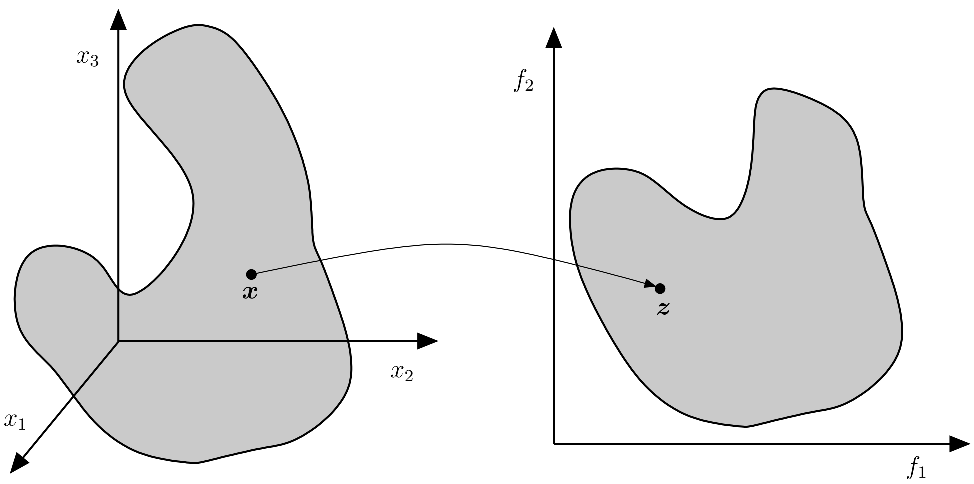
subject to the constraints

$$\begin{aligned} g_j(\boldsymbol{x}) &\geq 0, & j &= 1, \ldots, J, & (4.2)\\ h_k(\boldsymbol{x}) &= 0, & k &= 1, \ldots, K, & (4.3)\\ x_i^{(L)} \leq x_i &\leq x_i^{(U)}, & i &= 1, \ldots, n, & (4.4) \end{aligned}$$

where $x_i^{(L)}$ and $x_i^{(U)}$ are constant lower and upper bounds on the decision variable x_i, respectively, for all $i = 1, \ldots, n$ [50]. Constraint set (4.2) contains the inequality constraints, while constraint set (4.3) contains the equality constraints. If a solution $\boldsymbol{x}$ satisfies *all* the constraints in (4.2)–(4.4), the solution is said to be *feasible*. If, however, at least one of the constraints in (4.2)–(4.4) is violated by a vector $\boldsymbol{x}$, the vector is said to be *infeasible*[1]. The set of all feasible solutions to the multi-objective optimisation problem is known as the *feasible region* $\mathcal{S}$ or the *decision space* of the problem. Therefore, $\mathcal{S} \subseteq \mathcal{D}$.

The objective functions encapsulated in vector form by $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]$ each either has to be maximised or minimised [50]. Many multi-objective optimisation solution methodologies require that the problem involves either only maximisation objectives or only minimisation objectives. When an optimisation problem contains some objective functions that have to be minimised and others that have to be maximised, the *principle of duality*[2] may be used to transform all maximisation objective functions into minimisation objective functions or *vice versa* in order to arrive at a formulation containing only minimisation objective functions or only maximisation objective functions.

In addition to the usual decision variable space $\mathcal{D}$, which is also present in a single-objective optimisation problem, a multi-objective optimisation problem with M objectives is equipped with an M-dimensional space called the *objective space* $\mathcal{Z}$ [50]. For each solution $\boldsymbol{x} \in \mathcal{D}$ there exists a point $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{z} = [z_1, \ldots, z_m]$ in objective space. An n-dimensional solution vector $\boldsymbol{x}$ in decision space is therefore mapped to an M-dimensional objective vector $\boldsymbol{z}$ in objective space by the optimisation problem (4.1)–(4.4). This mapping is illustrated graphically for $n = 3$ and $M = 2$ in Figure 4.1.



(a) Decision space (b) Objective space

FIGURE 4.1: *The three-dimensional decision space of a bi-objective problem containing three decision variables illustrated in (a), and its corresponding two-dimensional objective space is illustrated in (b). The mapping of a solution $\boldsymbol{x}$ in decision space to a point $\boldsymbol{z}$ in objective space is also shown [50].*

4.2 Nonconvexity in multi-objective optimisation problems

A set of points $\mathcal{B} \subseteq \mathbb{R}^n$ is said to be *convex* if the line segment joining any pair of points in $\mathcal{B}$ is wholly contained in $\mathcal{B}$ [209]. A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is furthermore *convex* if, for any pair of

[1]It should be noted that the entire decision space $\mathcal{D}$ need not be feasible.

[2]The principle of duality in an optimisation context implies that a minimisation problem may be converted to a maximisation problem by multiplying the objective function by -1 [150, 153]. A maximisation problem may similarly be converted to a minimisation problem by multiplying the objective function by -1.

vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \in \mathbb{R}^n$, the inequality

$$f\big(\lambda\boldsymbol{x}^{(1)} + (1-\lambda)\boldsymbol{x}^{(2)}\big) \leq \lambda\boldsymbol{x}^{(1)} + (1-\lambda)\boldsymbol{x}^{(2)} \tag{4.5}$$

holds for all $0 \leq \lambda \leq 1$ [209]. Conversely, if the inequality sign $\leq$ is replaced with the inequality sign $>$ in (4.5), the function f is said to be *nonconvex* [209]. Note, therefore, that a function may be neither convex nor nonconvex. The notion of a convex function is illustrated graphically for $n = 1$ in Figure 4.2.

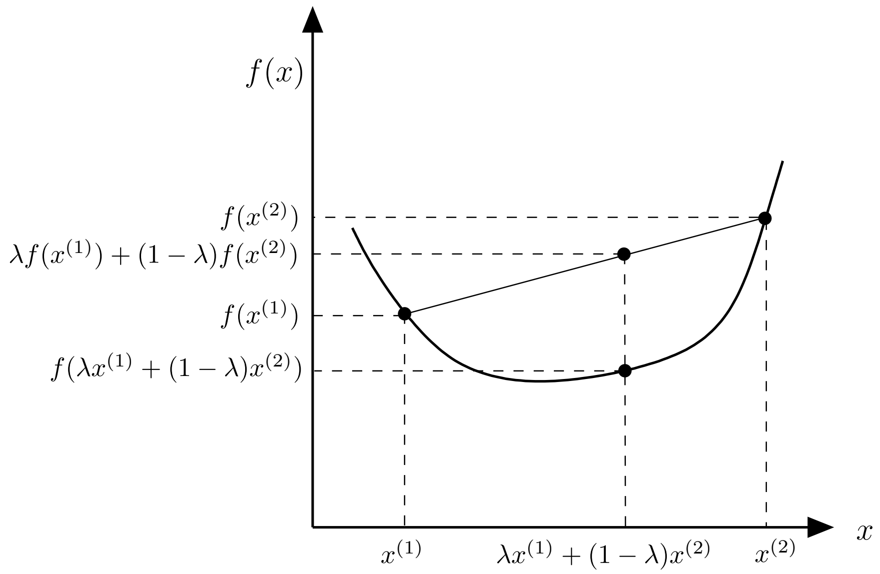


FIGURE 4.2: *A convex function (adapted from [50]).*

A convex function $f(\boldsymbol{x})$ has the following properties [50]: (1) a linear approximation of $f(\boldsymbol{x})$ at any point along the straight line in $\mathbb{R}^n$ joining $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ *underestimates* the value of $f(\boldsymbol{x})$, (2) the Hessian[3] of $f(\boldsymbol{x})$ is *positive definite*[4] for all $\boldsymbol{x} \in \mathbb{R}^n$ and (3) any local minimum of $f(\boldsymbol{x})$ is also a global minimum of $f(\boldsymbol{x})$.

In order to test whether a function is convex within a given region, the Hessian, denoted by $\nabla^2 f(\boldsymbol{x})$, may be calculated and tested for positive definiteness for all $\boldsymbol{x}$ in the region [50]. If all the principal minors[5] of the Hessian are nonnegative for all $\boldsymbol{x}$, then the function $f(\boldsymbol{x})$ is convex [209]. This may, of course, also be achieved by calculating the eigenvalues of the Hessian and verifying that they are all positive [50]. Similarly, if all the principal minors of the negation of the Hessian matrix $-\nabla^2 f(\boldsymbol{x})$ are nonnegative for all $\boldsymbol{x}$ in a region, then the function $f(\boldsymbol{x})$ is nonconvex in that region [209]. If a function $f(\boldsymbol{x})$ is nonconvex on $\mathbb{R}^n$, it may be shown that all vectors $\boldsymbol{x} \in \mathbb{R}^n$ satisfying $f(\boldsymbol{x}) \geq 0$ collectively form a convex set in $\mathbb{R}^n$ [50]. If the feasible region of a multi-objective optimisation problem is formed by a set of nonconvex constraints, it follows that the feasible region will enclose a convex space.

A multi-objective optimisation problem is therefore *convex* if all the objective functions in (4.1) are convex and if the set of feasible solutions (*i.e.* the feasible regions) to the problem is convex (*i.e.* all the functions $g_j(\boldsymbol{x})$ in (4.2) are nonconvex and all the functions $h_k(\boldsymbol{x})$ in (4.3) are

[3]The Hessian of a function is a square matrix of all the second order partial derivatives of the function, which describes its curvature [209].

[4]A positive definite matrix is a symmetric matrix which has the property that all its eigenvalues are positive.

[5]An i^{th} principal minor of an $n \times n$ matrix is the determinant of any $i \times i$ matrix obtained by deleting any $n - i$ rows and corresponding columns of the matrix [209].

linear). The notion of convexity of multi-objective optimisation problems is important since many multi-objective optimisation methodologies are able to solve convex multi-objective optimisation problems easily, but are ineffective in solving nonconvex multi-objective optimisation problems [50].

4.3 The notion of solution dominance

The concept of multi-objective optimisation naturally gives rise to the notion of *dominance* when comparing candidate solutions in objective space with one another in order to uncover a set of solutions in decision space which are superior to the remaining solutions in terms of all the objective functions of a given multi-objective optimisation problem. Such a set of solutions is known as a *nondominated* set of solutions. The notion of dominance may be defined more formally as follows. A solution $\boldsymbol{x}^{(1)}$ to (4.1)–(4.4) is said to *dominate* another solution $\boldsymbol{x}^{(2)}$ to the problem, denoted by $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$, if the following two conditions hold: (1) solution $\boldsymbol{x}^{(1)}$ is no worse than solution $\boldsymbol{x}^{(2)}$ in all the objective functions and (2) solution $\boldsymbol{x}^{(1)}$ is strictly better than solution $\boldsymbol{x}^{(2)}$ in at least one objective function [39, 50, 48]. The relation $\preceq$ is known as the *dominance relation* between solutions. If any one of the two conditions for dominance is violated, the solution $\boldsymbol{x}^{(1)}$ does not dominate the solution $\boldsymbol{x}^{(2)}$.

To illustrate the notion of dominance in a practical sense, consider a bi-objective optimisation problem in which an objective function f_1 has to be maximised and another objective function f_2 has to be minimised. Consider five candidate solutions to the problem with objective function values as illustrated in Figure 4.3.

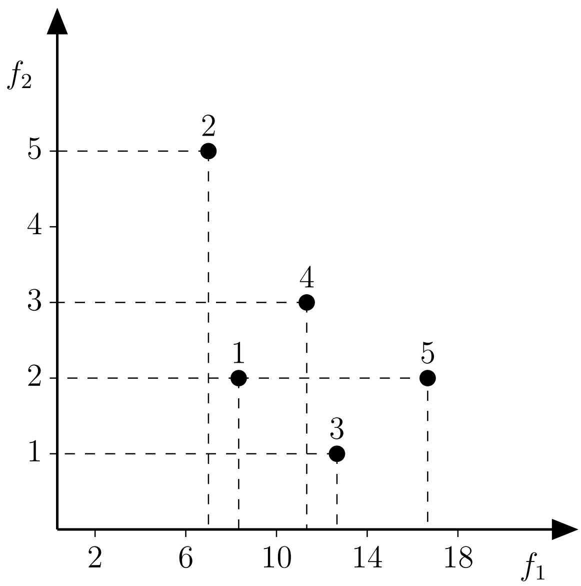


FIGURE 4.3: *A set of five candidate solutions in the objective space of a bi-objective optimisation problem with objective functions f_1 and f_2 illustrating the notion of dominance [50]. Objective function f_1 is to be maximised, while f_2 is to be minimised.*

When considering the solutions in the figure, it is difficult to find a solution which is best with respect to both objective functions. The definition of dominance may therefore be used to decide which solution is better between any pair of solutions in objective space. Consider, for example, solutions 1 and 2. Solution 1 achieves a value of 9 in objective function f_1 and a value of 2 in objective function f_2, while solution 2 achieves a value of 8 in objective function f_1 and a value of 5 in objective function f_2. Solution 1 is therefore better than solution 2 in objective function f_1 and also better than solution 1 in objective function f_2. Both the conditions for dominance are thus satisfied and it may be concluded that solution 1 dominates solution 2 (*i.e.* solution 1 $\preceq$ solution 2) — in other words, solution 1 is superior to solution 2. In a more general sense, if $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$, then solution $\boldsymbol{x}^{(1)}$ is considered superior to $\boldsymbol{x}^{(2)}$.

Now consider solutions 3 and 5. Solution 3 achieves a value of 12 in objective function f_1 and a value of 1 in objective function f_2, while solution 5 achieves a value of 16 in objective function f_1 and a value of 2 in objective function f_2. Solution 3 is therefore better than solution 5 in objective function f_2, but solution 5 is better than solution 3 in objective function f_1. Neither of the conditions for dominance are therefore satisfied, and it may be concluded that solutions 3 and 5 do not dominate each other — in other words, neither of the solutions 3 and 5 is superior to the other.

The concept of dominance provides an intuitive way of comparing solutions in the objective space of a multi-objective optimisation problem. It is therefore employed within a number of multi-objective optimisation solution methodologies to find nondominated solutions in the objective space [39, 50, 48].

4.3.1 Properties of the dominance relation

The notion of dominance allows for three possible outcomes when a solution $\boldsymbol{x}^{(1)}$ is compared with a solution $\boldsymbol{x}^{(2)}$ [50]. These outcomes are (1) $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$, (2) $\boldsymbol{x}^{(2)} \preceq \boldsymbol{x}^{(1)}$ or (3) neither of the solutions $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ dominates the other. These outcomes lead to three distinct dominance relation properties. These properties are:

Non-reflexivity. According to the definition of dominance, a solution $\boldsymbol{x}^{(1)}$ cannot dominate itself. Therefore $\boldsymbol{x}^{(1)} \not\preceq \boldsymbol{x}^{(1)}$ and hence the dominance relation is *non-reflexive.*

Asymmetry. If $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$, then $\boldsymbol{x}^{(2)} \not\preceq \boldsymbol{x}^{(1)}$ (*i.e.* if $\boldsymbol{x}^{(1)}$ dominates $\boldsymbol{x}^{(2)}$, then $\boldsymbol{x}^{(2)}$ cannot dominate $\boldsymbol{x}^{(1)}$). Therefore, the dominance relation is *asymmetric.*

Transivity. If $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(2)} \preceq \boldsymbol{x}^{(3)}$, then $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(3)}$. Hence, the dominance relation is *transitive.*

It is stressed that if a solution $\boldsymbol{x}^{(1)}$ does not dominate a solution $\boldsymbol{x}^{(2)}$ (*i.e.* $\boldsymbol{x}^{(1)} \not\preceq \boldsymbol{x}^{(2)}$), it does not necessarily imply that $\boldsymbol{x}^{(2)}$ dominates $\boldsymbol{x}^{(1)}$ (*i.e.* that $\boldsymbol{x}^{(2)} \preceq \boldsymbol{x}^{(1)}$).

4.3.2 The notion of Pareto optimality

Consider again the candidate solutions in objective space of Figure 4.3. When continuing with the comparison of the solutions in a pairwise manner, it is easy to see that solution 5 dominates solutions 1, 2 and 4. Furthermore, solution 3 also dominates solutions 1, 2 and 4. Recall, however, that neither of the solutions 3 or 5 dominates the other.

If all the solutions in a finite set of candidate solutions to a multi-objective optimisation problem are compared in a pairwise manner, a set of solutions is obtained in which no solution dominates another. Furthermore, this set of solutions dominates all other candidate solutions. This set of solutions is called the *nondominated set of solutions* to the multi-objective optimisation problem. The nondominated set of solutions is therefore *superior* to the remaining candidate solutions and no solution in the nondominated set is superior (or *inferior*) to any other solution in the nondominated set [39, 50]. Such a nondominated set of solutions may formally be defined as follows. Among a finite set $\mathcal{P}$ of solutions to a multi-objective optimisation problem, the nondominated solutions $\mathcal{P}' \subseteq \mathcal{P}$ are those that are not dominated by any member of $\mathcal{P}$. In Figure 4.3, solutions 3 and 5 are the nondominated set of solutions among the five solutions.

When the above set $\mathcal{P}$ is the entire set of feasible solutions to a multi-objective optimisation problem, the resulting set $\mathcal{P}'$ of nondominated solutions are, together, called the *Pareto optimal solutions* and form a so-called *Pareto front* in objective space. This front typically comprises a set of solutions on a hyper-edge of the feasible region, oriented depending on the type of objective functions (*i.e.* to maximise or minimise). The nondominated set of solutions of the entire feasible region of the multi-objective optimisation problem is called the *globally Pareto optimal set* of solutions to the problem.

Figure 4.4 contains illustrations of sets of globally Pareto fronts for four different scenarios. In Figure 4.4(a), a Pareto front of a bi-objective problem is portrayed in which both objective functions f_1 and f_2 are to be minimised. The shaded area represents the feasible region in objective space and the solid dark curve represents the Pareto front. If, however, the objective function f_1 is to be minimised and the objective function f_2 is to be maximised for the same feasible region, the Pareto front illustrated in Figure 4.4(b) is obtained. Note that the Pareto front in this scenario is a union of two disconnected Pareto fronts. If the objective function f_1 is to be maximised and the objective function f_2 is to be minimised for the same feasible region, the Pareto front in Figure 4.4(c) is obtained. Finally, if both objective functions f_1 and f_2 are to be maximised, the Pareto front shown in Figure 4.4(d) is obtained.

As is the case with single-objective optimisation, multi-objective problems may, however, also exhibit locally Pareto optimal solutions. Locally Pareto optimal solutions may be defined as follows. If, for any member $\boldsymbol{x} \in \mathcal{P}$, there exists no solution $\boldsymbol{y}$ to a multi-objective optimisation problem in the neighbourhood $||\boldsymbol{x} - \boldsymbol{y}|| < \epsilon$, where $|| \cdot ||$ is a norm and ϵ is a small, positive real number, then the solutions in $\mathcal{P}$ form a locally Pareto optimal set of solutions to the multi-objective optimisation problem [49, 130]. By this definition it may be concluded that a globally Pareto optimal set of solutions is also a locally Pareto optimal set, but the converse is not necessarily true. An example of a globally Pareto front (represented by a dark solid curve) and two locally Pareto fronts (represented by dark dotted curves) are illustrated in Figure 4.5(a). When any solution in a set of locally Pareto optimal set of solutions is perturbed locally in the solution space, no solution can be found which dominates any member of this set. This phenomenon is illustrated by solution B in Figure 4.5(b).

4.3.3 Strong dominance and weak Pareto optimality

The notion of dominance described earlier in this section is often referred to as a *weak* dominance relation. The definition may, however, be strengthened as follows to obtain a relation with stronger dominating properties. A solution $\boldsymbol{x}^{(1)}$ strongly dominates a solution $\boldsymbol{x}^{(2)}$ if $\boldsymbol{x}^{(1)}$ is strictly better than $\boldsymbol{x}^{(2)}$ in all M objectives of (4.1)–(4.4) [39, 48].

Consider again the example in Figure 4.3. It was shown earlier that solution 5 weakly dominates solution 1. According to the above definition of strong dominance it is, however, clear that solution 5 does not strongly dominate solution 1. From the figure, it is also clear that solution 3 strongly dominates solution 1 since it achieves superior values in both objective functions. It may therefore be concluded that if a solution $\boldsymbol{x}^{(1)}$ strongly dominates a solution $\boldsymbol{x}^{(2)}$, then $\boldsymbol{x}^{(1)}$ also dominates $\boldsymbol{x}^{(2)}$ weakly, but not *vica versa.*

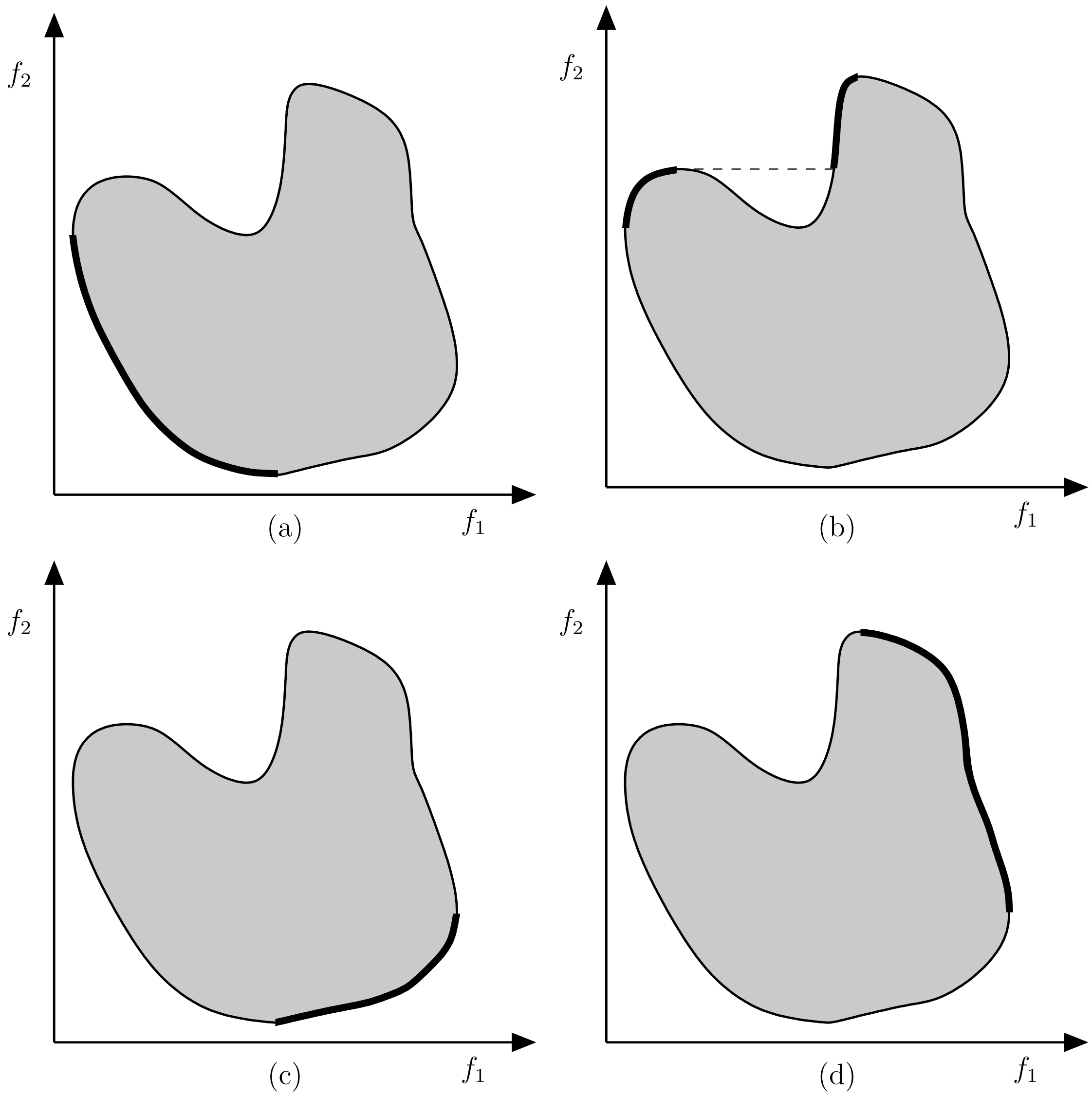
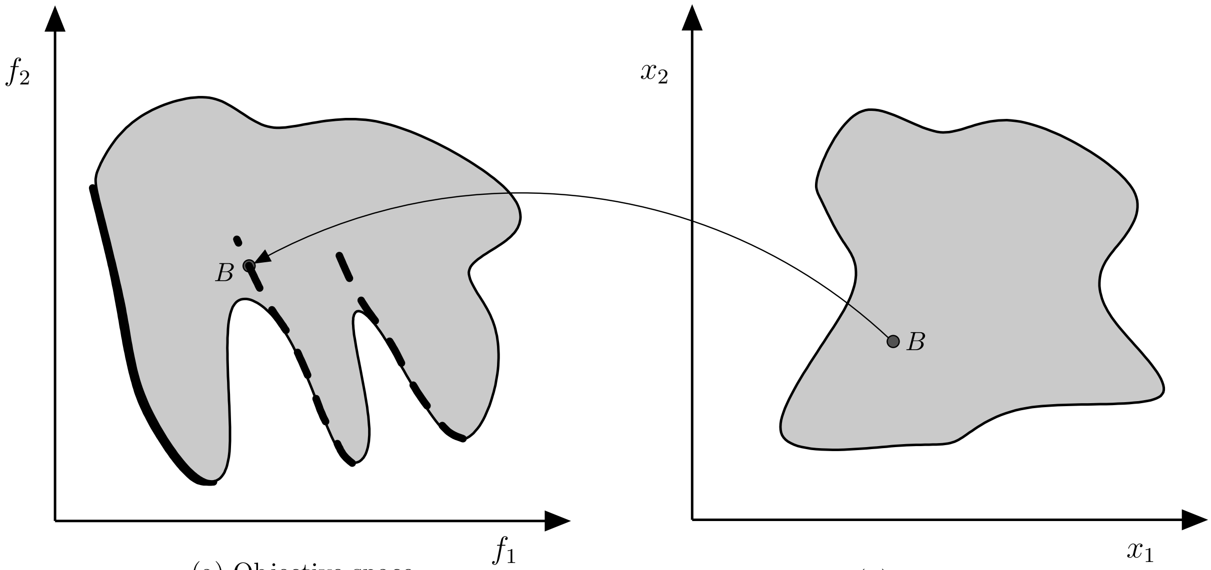


FIGURE 4.4: *Continuous sets of globally Pareto optimal curves for four scenarios containing different Pareto optimal sets. In (a), objective functions f_1 and f_2 are minimised, in (b), objective function f_1 is minimised while objective function f_2 is maximised, in (c), objective function f_1 is maximised and objective function f_2 is minimised and in (d), objective functions f_1 and f_2 have to be maximised.*



(a) Objective space

(b) Decision space

FIGURE 4.5: *A globally Pareto front denoted by the dark solid curve and two locally Pareto fronts denoted by the dark, dotted curves, for a bi-objective optimisation problem in which both objective functions f_1 and f_2 are to be minimised [50].*

The strong dominance relation may also be used to define the notion of a *weakly nondominated set*. A weakly nondominated set of solutions $\mathcal{P}'$ in a set of solutions $\mathcal{P}$ are those that are not strongly dominated by any other member in the set of solutions $\mathcal{P}$ [39, 50]. The set of weakly nondominated solutions therefore contains all the members of the nondominated set of solutions, as defined earlier in this chapter. The cardinality of the set of weakly nondominated solutions of a multi-objective optimisation problem is therefore equal to or greater than the cardinality of the set of solutions obtained by the dominance relation defined here.

A Pareto optimal set of a multi-objective optimisation problem always comprises a nondominated set of solutions to the problem. It is, however, possible that nondominated sets may contain some Pareto optimal solutions and some non-Pareto optimal solutions. Furthermore, the nondominated set of solutions obtained by approximate multi-objective solution techniques do not necessarily represent the true Pareto optimal set of solutions. The nondominated set of solutions obtained by these algorithms are therefore typically referred to as *approximately Pareto optimal solutions* in decision space forming an *approximate Pareto front* in objective space.

4.4 Hypervolume as a measure of nondominated front quality

The aim in multi-objective optimisation techniques is usually to approximate Pareto optimal solutions to a problem instance as closely as possible. Thiele *et al.* [184] state that multi-objective search techniques should, in fact, aim to achieve the following three objectives: (1) the nondominated front attained by the multi-objective solution technique should be as close as possible to the true Pareto front in objective space — the nondominated front should, in fact, ideally be a subset of the true Pareto front, (2) the set of nondominated solutions should be uniformly distributed and diverse along the Pareto front in all the objective functions — the aim is to provide the decision maker with a comprehensive view of trade-off decisions in objective space, and (3) the nondominated front should be such that it includes solutions which cover the full spectrum of values along the true Pareto front in all the objective functions. It is, however, typically difficult to obtain a close approximation of the entire Pareto front in a single run of a multi-objective optimisation technique, especially for complex multi-obective optimisation problem instances.

A very well-known *indicator measure* often used to evaluate the quality of Pareto front estimates is the *hypervolume indicator* (also known as the *S-metric* or *Lebesgue measure*) [12]. This indicator was originally introduced by Thiele and Zitzler [184], and belongs to the class of *unary* quality indicators[6] which involves mapping a nondominated set of solutions to the set of positive real numbers. The aim in the hypervolume measure is to determine the portion of objective space which is dominated by the set of solutions in the Pareto front estimate with respect to some pre-defined reference solution vector in objective space [24]. This solution vector is known as a *reference point* and is typically chosen in such a way that it bounds the nondominated objective space. The aim is to maximise the hypervolume measure when estimating the true Pareto front. Note that such a maximum is achieved if and only if the nondominated front is equal to the true Pareto front.

Define $\mathcal{S}_{\boldsymbol{x}}= \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ to be the set of nondominated solutions to a multi-objective optimisation problem with corresponding objective function vectors $\mathcal{S}_{\boldsymbol{z}}= \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^N\}$ in objective space. The hypervolume of the set $\mathcal{S}_{\boldsymbol{x}}$ may then be defined as the region dominated by $\mathcal{S}_{\boldsymbol{z}}$ in objective space from a specific reference point $\bar{\boldsymbol{z}}$ which satisfies $\boldsymbol{z}^\ell \succeq \bar{\boldsymbol{z}}$ for all $\ell \in \{1, \ldots, N\}$. The hypervolume metric is illustrated graphically in Figure 4.6 for a bi-objective optimisation problem in which both objectives are to be minimised.

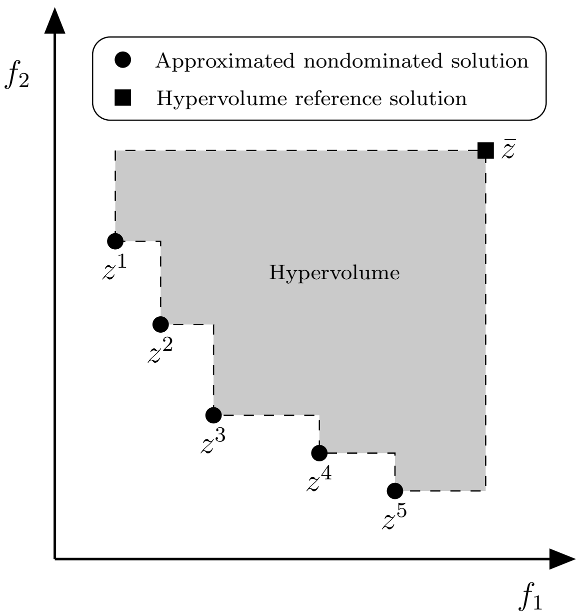


FIGURE 4.6: *The hypervolume (indicated by the shaded region) of a set $\mathcal{S}_{\boldsymbol{z}}= \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^5\}$ of nondominated solutions for a bi-objective optimisation problem in which both objective functions f_1 and f_2 are to be minimised, with reference point $\bar{\boldsymbol{z}}$.*

The main advantage of adopting the hypervolume measure is that it is able to encapsulate in a single measure the closeness of nondominated solutions to the true Pareto front (if it is known),

[6]Unary quality indicators take a single nondominated set of solutions, and returns a scalar evaluation of its quality [117].

evenness in the spread of nondominated solutions along the Pareto front as well as the diversity of these solutions in objective space, simultaneously [25]. Although the hypervolume measure seems to achieve these goals effectively, the method has three major drawbacks. The first drawback is that the computational complexity associated with the hypervolume is relatively high — it has been shown that the calculation of the hypervolume is classified as **NP**-hard [11]. The second drawback associated with the hypervolume measure is that it requires the nondominated set of solutions to be normalised so as to ensure that the objectives contribute towards the hypervolume calculations in equal proportions. This, in turn, requires that approximate bounds on the objective functions across the Pareto front be known *a priori*. While *et al.* [12], however, suggested that if the maximum or minimum in all the objectives of the true Pareto optimal solutions are not known, then it is acceptable to take, for each objective, the best or worse value amongst all the nondominated solutions being compared. The last major drawback associated with the hypervolume is related to the choice of an appropriate reference point $\bar{\boldsymbol{z}}$ (described earlier). Caution should be taken so as not to choose $\bar{\boldsymbol{z}}$ in such a way that it allows certain objectives to contribute more towards the hypervolume value than others. Various "rules of thumb" exist for selecting $\bar{\boldsymbol{z}}$ [11, 12]. A popular method entails taking $\bar{\boldsymbol{z}}$ as a fixed percentage above or below bounds on the objective function values.

4.5 Necessary and sufficient conditions for Pareto optimality

Now that the notions of dominance and Pareto optimality in multi-objective optimisation problems have been discussed, theoretical conditions for Pareto optimality in a multi-objective optimisation problem such as (4.1)–(4.4) may be described. Two such conditions are reviewed in this section. The first condition is a *necessary condition* for a solution to be Pareto optimal and the second condition is a *sufficient condition* for a solution to be Pareto optimal. The underlying assumption of both these conditions is that all the objective functions in (4.1) and constraint functions in (4.2)–(4.3) are continuously differentiable.

The first-order necessary conditions for Pareto optimality described here are known as the *Fritz-John necessary conditions* [43, 50]. Necessary conditions for $\boldsymbol{x}^*$ to be a Pareto optimal solution to the multi-objective optimisation problem (4.1)–(4.4) is that there should exist non-negative vectors $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M]$ and $\boldsymbol{u} = [u_1, \ldots, u_J]$ (both not identically zero) so that

1. $\sum_{m=1}^{M} \lambda_m \nabla f_m(\boldsymbol{x}^*) - \sum_{j=1}^{J} u_j \nabla g_j(\boldsymbol{x}^*) = \mathbf{0}$ and
2. $u_j g_j(\boldsymbol{x}^*) = 0$ for all $j = 1, \ldots, J$.

For unconstrained multi-objective optimisation problems (*i.e.* if $J = K = 0$), the above conditions reduce to the single condition

$$\sum_{m=1}^{M} \lambda_m \nabla f_m(\boldsymbol{x}^*) = \mathbf{0}. \tag{4.6}$$

If the Fritz-John conditions are not satisfied for a solution corresponding to the vector $\boldsymbol{\lambda}$, it may be concluded that the solution is not a Pareto optimal solution to (4.1)–(4.4). If, however, a solution does satisfy the Fritz-John conditions, it does not necessarily hold that the solution is a Pareto optimal solution to (4.1)–(4.4). A further sufficient condition for Pareto optimality is required to verify this.

The second-order sufficient conditions for Pareto optimality described here are the well-known *Karush-Kuhn-Tucker sufficient conditions* for Pareto optimality [50, 130]. Suppose the objective functions $f_1, \ldots, f_M$ of the multi-objective optimisation problem (4.1)–(4.4) are convex, that the inequality constraint functions $g_1, \ldots, g_J$ are all non-convex, that the equality constraint functions $h_1, \ldots, h_K$ are all linear and that all of these functions are continuously differentiable at a feasible solution $\boldsymbol{x}^*$ to (4.1)–(4.4). Then $\boldsymbol{x}^*$ is a Pareto optimal solution if there exist a *positive* real vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M]$ and a *non-negative* real vector $\boldsymbol{u} = [u_1, \ldots, u_J]$ such that

1. $\sum_{m=1}^{M} \lambda_m \nabla f_m(\boldsymbol{x}^*) - \sum_{j=1}^{J} u_j \nabla g_j(\boldsymbol{x}^*) = \mathbf{0}$ and
2. $u_j g_j(\boldsymbol{x}^*) = 0$ for all $j = 1, \ldots, J$.

It should, however, be noted that the above second-order conditions are not sufficient if the multi-objective optimisation problem is not convex (*i.e.* if the multi-objective optimisation problem (4.1)–(4.4) is not convex, then the above second-order conditions are not sufficient to conclude that $\boldsymbol{x}^*$ is Pareto optimal).

4.6 Methods for calculating nondominated sets

An important quest in multi-objective optimisation is to find computationally efficient methods for computing high-quality sets of nondominated solutions lying as close as possible to the true Pareto optimal set of solutions. Many such methods, ranging in different levels of complexity, exist in the literature. In this section, three well-known methods ranging from a slow and naive approach to a more complex, efficient method are described for identifying the set of nondominated solutions from a given finite set of candidate solutions to (4.1)–(4.4).

4.6.1 A naive and slow approach

The first method for identifying a nondominated set of solutions among a given, finite set of solutions to (4.1)–(4.4) is the so-called *naive and slow* method. The method works in such a way that each solution i in the set of given candidate solutions within the feasible region of (4.1)–(4.4) is compared, based on the dominance relation described in §4.3, with every other solution j in the set. If no solution j is found which dominates solution i, then solution i is added to the set of nondominated solutions. If, however, any solution j is found which dominates solution i, then solution i cannot form part of the set of nondominated solutions. Solution i is then flagged to indicate that it cannot belong to the nondominated set. A pseudocode description of the naive and slow method is given as Algorithm 4.1.

The working of the naive and slow approach is elucidated in the following example.

Example 4.1 (Naive and slow method) *Consider the set of solutions $\mathcal{P} = \{1, 2, 3, 4, 5\}$ in Figure 4.3. Set $i = 0$ and $\mathcal{P}' = \emptyset$. Solution 1 is compared with all the other solutions in the set $\mathcal{P}$, starting with solution 2. Solution 2 does not dominate solution 1. Next, solution 1 is compared with solution 3 and it is found that solution 3 dominates solution 1. Solution 1 therefore cannot belong to the nondominated set $\mathcal{P}'$. Hence, i is incremented to 2.*

Solution 2 is compared with all the solutions in the set $\mathcal{P}$, starting with solution 1. It is found that solution 1 dominates solution 2, implying that solution 2 cannot belong to the nondominated set $\mathcal{P}'$, and so i is incremented to 3.

Solution 3 is next compared with all solutions in the set $\mathcal{P}$, starting with solution 1. It is found that no solution in the set $\mathcal{P}$ dominates solution 3 and solution 3 is therefore included in the nondominated set to obtain $\mathcal{P}' = \{3\}$.

Algorithm 4.1: Nondominated set: Naive and slow method [50]

Input : A set $\mathcal{P}$ of candidate solutions to (4.1)–(4.4).
Output: The nondominated solutions $\mathcal{P}' \subseteq \mathcal{P}$.

$i \leftarrow 0$;
$\mathcal{P}' \leftarrow \emptyset$;
while $i \leq |\mathcal{P}|$ **do**
    $j \leftarrow 0$;
    while $j \leq |\mathcal{P}|$ **do**
        if $j \neq i$ **then**
            if $x_j \preceq x_i$ **then**
                Continue;
            else
                $j \leftarrow j + 1$;
    if $j = |\mathcal{P}|$ **then**
        $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{i\}$;
    $i \leftarrow i + 1$;
return $[\mathcal{P}']$;

Next, i is incremented to 4 and solution 4 is compared with all the solutions in the set $\mathcal{P}$. It is found that solution 5 dominates solution 4, implying that solution 4 cannot belong to the nondominated set $\mathcal{P}'$ and so i is incremented to 5.

Solution 5 is finally compared with all the other solutions in the set $\mathcal{P}$. It is found that no solution in the set $\mathcal{P}$ dominates solution 5 and so solution 5 is included in the nondominated set to obtain $\mathcal{P}' = \{3, 5\}$.

Since all the solutions in the set $\mathcal{P}$ have been considered, the algorithm terminates. The nondominated set $\mathcal{P}' = \{3, 5\}$ is returned as output. ■

Testing for domination requires $\mathcal{O}(|\mathcal{P}|)$ comparisons, while each comparison for domination requires M objective function comparisons. The complexity of testing for domination is therefore $\mathcal{O}(M|\mathcal{P}|)$. The algorithm performs at most $\mathcal{O}(M|\mathcal{P}|^2)$ computations.

4.6.2 A continuously updating approach

Another approach towards identifying a nondominated set of solutions among a given, finite set of solutions to (4.1)–(4.4) is the so-called *continuously updating* method which is based on the naive and slow method, but is able to find nondominated solutions much quicker. According to the continuously updating approach, every solution in the given, finite set of solutions is compared, based on the dominance relation in §4.3, with a smaller and smaller subset of solutions as the algorithm progresses. The method is initiated by choosing a solution randomly from the given set $\mathcal{P}$ of candidate solutions to (4.1)–(4.4), inserting it into an empty set $\mathcal{P}'$ and removing it from $\mathcal{P}$. Each solution i in $\mathcal{P}$ is then compared with each member in the set $\mathcal{P}'$, one-by-one. If a solution $i \in \mathcal{P}$ dominates any member of $\mathcal{P}'$, then the solution dominated by i is removed from the set $\mathcal{P}'$. If, however, a solution $i \in \mathcal{P}$ is dominated by any solution in the set $\mathcal{P}'$, then i is ignored. Finally, if solution i is not dominated by any solution in $\mathcal{P}'$, then solution i is included in the set $\mathcal{P}'$. The algorithm terminates once all the solutions in $\mathcal{P}$ have been compared with each solution in the set $\mathcal{P}'$. The members of the set $\mathcal{P}'$ then constitute the set of nondominated solutions of $\mathcal{P}$. A pseudocode description of the continuous method is given in Algorithm 4.2.

Algorithm 4.2: Nondominated set: Continuously updating method [50]

```
Input  : A set P of candidate solutions to (4.1)–(4.4).
Output: The nondominated solutions P' ⊆ P.
i ← 2;
P' ← {1};
while i ≤ |P| do
    j ← 1;
    while j ≤ |P'| do
        if x_i ⪯ x_j then
            P' ← P'\{j};
        else if x_j ⪯ x_i then
            Continue;
        j ← j + 1;
    if j = |P| then
        P' ← P' ∪ {i};
    i ← i + 1;
return [P'];
```

The working of the continuously updating approach is illustrated in the following example.

Example 4.2 (Continuously updating method) *Consider again the set of solutions* $\mathcal{P} = \{1, 2, 3, 4, 5\}$ *in Figure 4.3. Suppose solution 1 is initially included in the nondominated set to obtain* $\mathcal{P}' = \{1\}$*, when* $i = 2$*. Solution 2 is now compared to the member of* $\mathcal{P}'$ (i.e. *solution 1) for dominance. It is found that solution 1 dominates solution 2. Since solution 1 dominates solution 2, i is incremented to 3 and solution 3 is compared with solution 1 (solution 2 cannot belong to the nondominated set* $\mathcal{P}'$*). Note that still* $\mathcal{P}' = \{1\}$.

When comparing solution 3 with solution 1, it is found that solution 3 dominates solution 1. Solution 1 is therefore removed from the nondominated set to yield $\mathcal{P}' = \emptyset$*. This implies that* $|\mathcal{P}'| = 0$.

Next, solution 3 is included in the nondominated set to yield $\mathcal{P}' = \{3\}$*. Since* $i \leq 5$*, i is incremented to 4 and solution 4 is compared with the solution in* $\mathcal{P}'$*. It is found that solution 3 dominates solution 4, and so i is incremented to 5. Note that still* $\mathcal{P}' = \{3\}$.

Solution 5 is finally compared with solution 3 and it is found that neither of them dominates each other. Solution 5 is therefore inserted in the nondominated set to obtain $\mathcal{P}' = \{3, 5\}$*. Since* $i = 5$*, the algorithm terminates to yield the nondominated set* $\mathcal{P}' = \{3, 5\}$. ■

When considering the complexity of the method described above, it is clear that when the algorithm is initiated, a solution is compared with one solution in the set $\mathcal{P}'$, then the next solution is compared with at most two solutions in the set $\mathcal{P}'$, and so on. Therefore, a total of at most $1 + 2 + \cdots + |\mathcal{P}| - 1$ or $|\mathcal{P}|(|\mathcal{P}| - 1)/2$ dominance comparisons are required. This results in a complexity of $\mathcal{O}(M|\mathcal{P}|^2)$. Although this is the same complexity as that of the naive and slow approach, the actual number of comparisons in the continuous method is usually much smaller than in the naive and slow approach — this is estimated to be approximately half the number of computations performed according to the naive and slow approach [50].

4.6.3 Kung *et al.*'s efficient method

The final method for identifying a nondominated set of solutions from among a given, finite set $\mathcal{P}$ of solutions to (4.1)–(4.4) considered here is *Kung et al.'s efficient method* [111], which is computationally faster than the naive and slow method as well as the continuously updated method. The method sorts the entire set of solutions $\mathcal{P}$ in non-improving order with respect to their first objective function values. The sorted set is then recursively halved to obtain a top subset of solutions, denoted by $\mathcal{T}$, and a bottom subset of solutions, denoted by $\mathcal{B}$ — the subset $\mathcal{T}$ therefore contains the better solutions with respect to the first objective function. Each solution i in the subset $\mathcal{B}$ is then compared, based on the dominance relation of §4.3, with each solution j in the subset $\mathcal{T}$. The solutions in the subset $\mathcal{B}$ which are not dominated by any solution in $\mathcal{T}$, are then included in $\mathcal{T}$ to form a merged set $\mathcal{S}$. The process is repeated recursively in respect of the set $\mathcal{S}$ until no solution in the subset $\mathcal{B}$ is dominated by any solution in the subset $\mathcal{T}$, in which case the subset $\mathcal{T}$ contains the set of nondominated solutions. Algorithm 4.3 contains a pseudocode description of the working of the efficient method of Kung *et al.* [111].

Algorithm 4.3: Front ($\mathcal{P}$)

Input : A set $\mathcal{P}$ of candidate solutions to (4.1)–(4.4).
Output: The nondominated solutions $\mathcal{P}' \subseteq \mathcal{P}$.

Sort($\mathcal{P}$);
if $|\mathcal{P}| = 1$ **then**
  return $[\mathcal{P}]$;
else
  $\mathcal{T} \leftarrow$ **Front**$([\mathcal{P}_1, \ldots, \mathcal{P}_{\lfloor|\mathcal{P}|/2\rfloor}])$;
  $\mathcal{B} \leftarrow$ **Front**$([\mathcal{P}_{\lfloor|\mathcal{P}|/2\rfloor+1}], \ldots, \mathcal{P}_{|\mathcal{P}|}])$;
  $i \leftarrow 1$;
  $\mathcal{S} \leftarrow \emptyset$;
  while $i \leq |\mathcal{B}|$ **do**
    $j \leftarrow 1$;
    while $j \leq |\mathcal{T}|$ **do**
      if $x_j \not\preceq x_i$ **then**
        $j \leftarrow j + 1$;
      else
        Continue;
    if $j = |\mathcal{T}|$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$;
    $i \leftarrow i + 1$;

The working of the efficient method of Kung *et al.* is illustrated in the following example.

Example 4.3 (Kung *et al.*'s efficient method) *Consider again the set of solutions $\mathcal{P} = \{1, 2, 3, 4, 5\}$ in Figure 4.3. Since the first objective is to maximise the function f_1, the solutions in $\mathcal{P}$ are sorted in decreasing order of magnitude for the first objective. This yields the ordered set $\mathcal{P} = \{5, 3, 4, 1, 2\}$. Since the size of $\mathcal{P}$ is five (and not one as required for termination of the algorithm), $\mathcal{P}$ is partitioned into two sets $\mathcal{T} =$ Front$(\{5, 3\})$ and $\mathcal{B} =$ Front$(\{4, 1, 2\})$ as illustrated graphically in the top-most branch of Figure 4.7.*

Recursively, the set $\{5, 3\}$ *in* $\mathsf{Front}(\{5, 3\})$ *is partitioned, in turn, to yield the sets* $\mathcal{T} = \mathsf{Front}(\{5\})$, *achieving the output* $\{5\}$, *and* $\mathcal{B} = \mathsf{Front}(\{3\})$, *achieving output* $\{3\}$, *as the next inner set of fronts. Since both of these inner sets have a size of one, they are returned as output to the previous* $\mathsf{Front}(\{5, 3\})$. *The inner sets* $\mathcal{T} = \{5\}$ *and* $\mathcal{B} = \{3\}$ *may now be tested for dominance. It is found that solution 3 is not dominated by solution 5 and so a merged set* $\mathcal{S} = \{5, 3\}$ *is populated. The outcome of* $\mathsf{Front}(\{5, 3\})$ *is therefore the nondominated set* $\mathcal{S} = \{5, 3\}$, *which is returned to* $\mathsf{Front}(\{5, 3, 4, 1, 2\})$ *as output.*

In a similar fashion, the set $\{4, 1, 2\}$ *in* $\mathsf{Front}(\{4, 1, 2\})$ *is partitioned into the inner sets* $\mathcal{T} = \mathsf{Front}(\{4\})$, *achieving the output* $\{4\}$, *and* $\mathcal{B} = \mathsf{Front}(\{1, 2\})$. *Since* $\mathcal{T} = \mathsf{Front}(\{4\})$ *has a size of one, the output* $\{4\}$ *is returned to the previous front,* $\mathsf{Front}(\{4, 1, 2\})$.

Since $\mathcal{B} = \mathsf{Front}(\{1, 2\})$ *has a size of two, it is partitioned to yield the sets* $\mathcal{T} = \mathsf{Front}(\{1\})$, *achieving the output* $\{1\}$, *and* $\mathcal{B} = \mathsf{Front}(\{2\})$, *achieving the output* $\{2\}$. *Next, solution 2 is compared with solution 1 in order test whether solution 2 is dominated by solution 1. It is found that solution 2 is indeed dominated by solution 1 and so solution 1 is included in the nondominated set* $\mathcal{S} = \{1\}$ *which serves as the output of* $\mathsf{Front}(\{1, 2\})$. *Next, solution 4 is compared with solution 1 to test whether solution 1 is dominated by solution 4. Since solution 4 does not, however, dominate solution 1, the merged set* $\mathcal{S} = \{4, 1\}$ *is formed. The output of* $\mathsf{Front}(\{4, 1, 2\})$ *is therefore the nondominated set* $\mathcal{S} = \{4, 1\}$, *which is returned to the call* $\mathsf{Front}(\{5, 3, 4, 1, 2\})$.

Finally, the sets $\mathcal{T} = \{5, 3\}$ *and* $\mathcal{B} = \{4, 1\}$ *are compared for dominance by testing whether the nondominated solution in the set* $\mathcal{B}$ *(i.e. solution 1) is dominated (in terms of the second objective) by the nondominated solution in the set* $\mathcal{T}$ *(i.e. solution 3). It is found that solution 3 dominates solution 1 with respect to the second objective, and the output of* $\mathsf{Front}(\{5, 3, 4, 1, 2\})$ *is therefore the nondominated set* $\{5, 3\}$. ■

Kung *et al.* [111] showed that for $M = 2$ or 3, the complexity of their method is $\mathcal{O}(|\mathcal{P}| \log |\mathcal{P}|)$ and for $M \geq 4$, the complexity is $\mathcal{O}(|\mathcal{P}|(\log |\mathcal{P}|)^{M-2})$.

4.7 Nondominated sorting of a set of solutions

Multi-objective optimisation techniques generally aim to find high-quality sets of nondominated solutions within the feasible region of a problem instance. These techniques therefore typically partition candidate solutions into two sets, namely the nondominated set and the remaining set of dominated solutions. Although interest naturally lies in the set of nondominated solutions, there are, however, algorithms which require that the set of candidate solutions be partitioned into different classes based on their relative degrees of dominance. This may be achieved by means of a nondominated sorting algorithm, such as the *fast nondominated sorting algorithm* developed by Deb [51], which is discussed in some detail in this section.

The algorithm identifies the nondominated set of solutions among a finite set $\mathcal{P}$ of candidate solutions to (4.1)–(4.4) by using any one of the methods described in §4.6. This set is called the front 1 nondominated solutions, denoted by F_1. The solutions in F_1 are then temporarily removed from the set $\mathcal{P}$ and the nondominated solutions of the new (smaller) set is calculated. These nondominated solutions within the smaller set of candidate solutions are called the front 2 nondominated solutions, collectively denoted by F_2. The nondominated solutions of fronts 1 and 2 are then temporarily removed from $\mathcal{P}$ and the front 3 nondominated solutions, denoted by F_3, are calculated similarly. This process is repeated in iterated fashion until the entire set of candidate solutions is classified into fronts. For each solution i, a dominance count dc_i (*i.e.* the

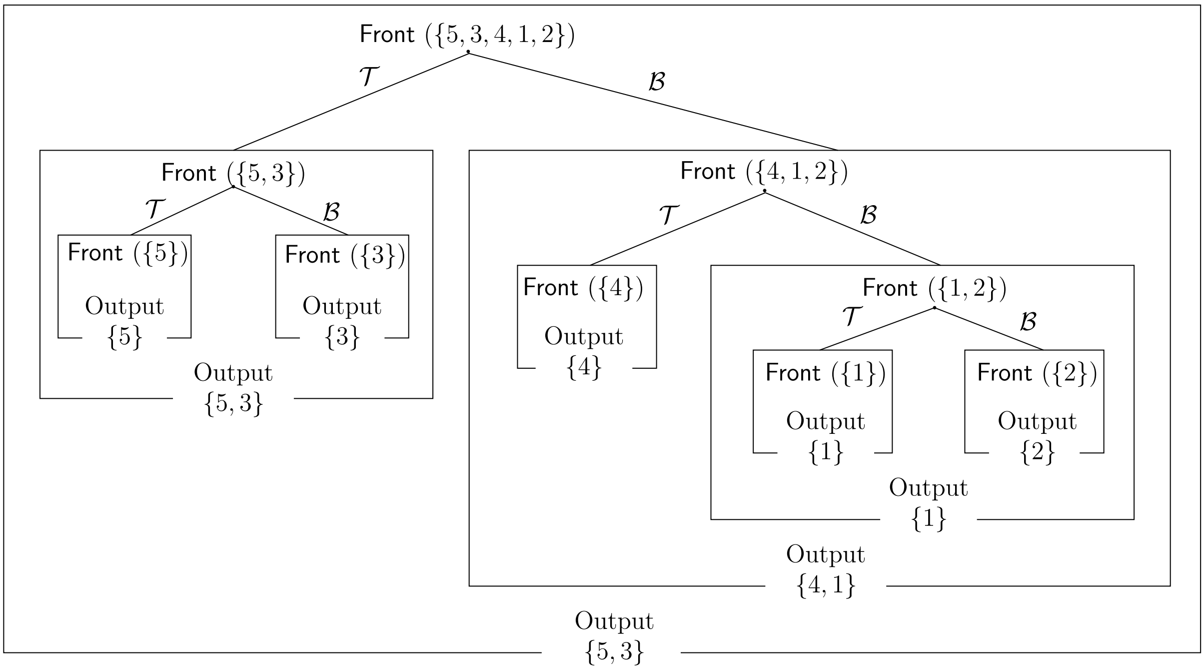


FIGURE 4.7: *Kung et al.'s efficient method illustrated in the context of finding the set of nondominated solutions for the set $\mathcal{P} = \{1, 2, 3, 4, 5\}$ of solutions in Figure 4.3.*

number of solutions that dominate i), and a set of solutions S_i dominated by i, are computed. The dominance count of a solution depends on the nondominated front into which it is classified. More specifically, if a solution i is a member of the set F_j computed as described above, then $dc_{j-1} = j-1$. The entire process requires $\mathcal{O}(M|\mathcal{P}|^2)$ comparisons, where M denotes the number of objectives in objective space and $|\mathcal{P}|$ denotes the number of decision variables in decision space as before. A pseudocode description of the working of the fast nondominated sorting algorithm is provided in Algorithm 4.4 and the working of the algorithm is illustrated in the following example.

Example 4.4 (Fast nondominated sorting algorithm) *Consider again the set of solutions $\mathcal{P} = \{1, 2, 3, 4, 5\}$ in Figure 4.3. From the results obtained in Examples 4.1, 4.2 and 4.3 it is found that solutions 3 and 5 form the nondominated set of solutions of the set $\mathcal{P}$. These solutions therefore belong to the front F_1.*

Solutions 3 and 5 are then temporarily removed from the set $\mathcal{P}$ and, again, the nondominated solutions are sought using any one of the three methods discussed in §4.6. When comparing the solutions in the reduced set {1, 2, 4} with one another it is found that solutions 2 and 4 dominate solution 1, but solutions 2 and 4 do not dominate each other. The set of solutions {2, 4} therefore form the set of nondominated solutions of the set {1, 2, 4} of solutions. Hence, these solutions are contained in the second front F_2 of nondominated solutions.

Next, solutions 2 and 4 are temporarily removed from the reduced set {1, 2, 4} leaving only solution 1 in the set. Since solution 1 is the only solution left, it forms the front F_3 of nondominated solutions on its own. Each candidate solution in $\mathcal{P}$ now belongs to a front and the algorithm terminates. The three fronts of nondominated solutions F_1, F_2 and F_3 are illustrated graphically in Figure 4.8. ∎

Algorithm 4.4: Fast Nondominated Sorting Algorithm [51].

Input : A finite set $\mathcal{P}$ of candidate solutions to a multi-objective optimisation problem of the form (4.1)–(4.4).

Output: The set of nondominated fronts F_1, F_2, F_3, ...

```
F_1 ← ∅
for i ∈ P do
    S_i ← ∅
    d_i^c = 0
    for j ∈ P do
        if i ≺ j then
            S_i ← S_i ∪ {j}
        else if j ≺ i then
            d_i^c ← d_i^c + 1
    if d_i^c = 0 then
        i_rank ← 1
        F_1 ← F_1 ∪ {i}
m ← 1
while F_m ≠ ∅ do
    A ← ∅
    for i ∈ F_m do
        for j ∈ S_i do
            d_j^c ← d_j^c − 1
            if d_j^c = 0 then
                j_rank ← m + 1
                A ← A ∪ {j}
    m ← m + 1
    F_m ← A
```

4.8 Weighted sums of objectives

One of the classical multi-objective optimisation techniques is the so-called *weighted-sum method*. It is considered the simplest multi-objective optimisation approach and is widely used to solve multi-objective optimisation problems [50]. The method requires a user-specified weight for each of the objective functions in (4.1) and involves scalarisation of the set of objective functions into a single objective which is to be optimised. Let $w_m \in [0, 1]$ denote the weight of objective $m \in \{1, \ldots, M\}$, where $\sum_{m=1}^{M} w_m = 1$. Then the problem (4.1)–(4.4) may be scalarised

$$\text{minimising} \quad F(\boldsymbol{x}) = \sum_{m=1}^{M} w_m f_m(\boldsymbol{x}) \tag{4.7}$$

$$\text{subject to the constraints} \quad g_j(\boldsymbol{x}) \geq 0, \quad j = 1, \ldots, J, \tag{4.8}$$

$$h_k(\boldsymbol{x}) = 0, \quad k = 1, \ldots, K, \tag{4.9}$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, \ldots, n. \tag{4.10}$$

The problem (4.7)–(4.10) is now a single-objective optimisation problem. In order to ensure that the objectives are of the same order of magnitude, they are typically normalised before including them in the objective function (4.7).

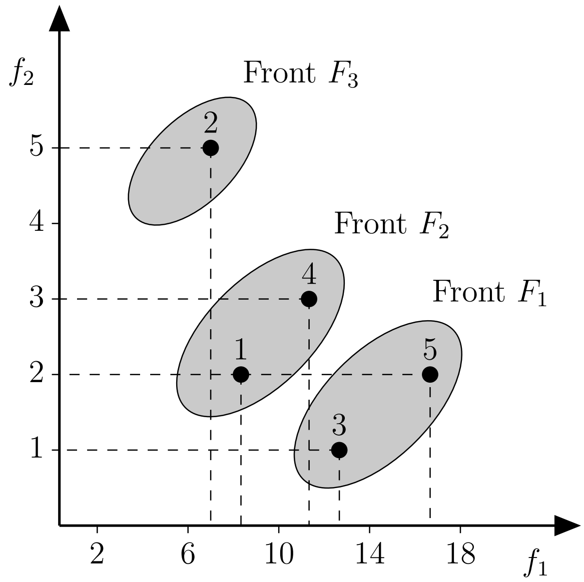


FIGURE 4.8: *Three nondominated fronts F_1, F_2 and F_3 of solutions of Figure 4.3 obtained by means of the fast nondominated sorting algorithm (Algorithm 4.4). Objective function f_1 is maximised, while objective function f_2 if minimised.*

Miettinen [130] showed that if the weights $w_1, \ldots, w_M$ in (4.7) are all positive, then any optimal solution to (4.7)–(4.10) is a Pareto optimal solution to (4.1)–(4.4). Miettinen [130] went on to show that the converse of this result is also true *if* (4.1)–(4.4) is a convex problem. More specifically, he showed that if $\boldsymbol{x}^*$ is a Pareto optimal solution to a convex multi-objective optimisation problem of the form (4.1)–(4.4), then there exists a positive weight vector $\boldsymbol{w}^* = [w_1^*, \ldots, w_M^*]$ such that $\boldsymbol{x}^*$ is an optimal solution to (4.7)–(4.10). This does not, however, suggest that any Pareto optimal solution to (4.1)–(4.4) can be obtained by solving an approximately weighted version of (4.7)–(4.10). More specifically, if (4.7)–(4.10) is not convex, then Pareto optimal solutions to (4.1)–(4.4) exist which do not correspond to optimal solutions to (4.7)–(4.10) for any positive choice of weight vector $\boldsymbol{w}$.

In order to illustrate how a nondominated set of solutions may be obtained by solving differently weighted versions of (4.7)–(4.10), consider Figure 4.9 which contains an illustration of the feasible region (in objective space) of a bi-objective optimisation problem in which both objective functions f_1 and f_2 have to be minimised. The feasible region is denoted by the shaded region and the Pareto front is indicated by means of the dark black curve. Suppose objective function f_1 is assigned a weight of w_1 and that objective function f_2 is assigned a weight of w_2 in (4.7), with $w_1 + w_2 = 1$. Then the objective function (4.7) is a convex combination of the functions f_1 and f_2 (where $M = 2$). The contour surfaces of the function F may therefore be represented by straight lines of slope $-w_1/w_2$ in objective space. Examples of such lines are the lines denoted a, b, c and d in Figure 4.9. The location of a contour line depends on the value of the objective function F. Since F is to be minimised, the aim is to find the contour line corresponding to the smallest value of F. This occurs when the contour line is tangential to the feasible region (the point A in Figure 4.9). This point corresponds to a Pareto optimal solution of (4.1)–(4.4) associated with the weight vector $w = [w_1, w_2]$. Using a different set of weights will result in the contour line having a different slope and consequently another Pareto optimal solution will be uncovered. If (4.1)–(4.4) is convex, then multiple Pareto optimal solutions to this problem may therefore be found by solving (4.7)–(4.10) for various sets of positive weight vectors, one at a time.

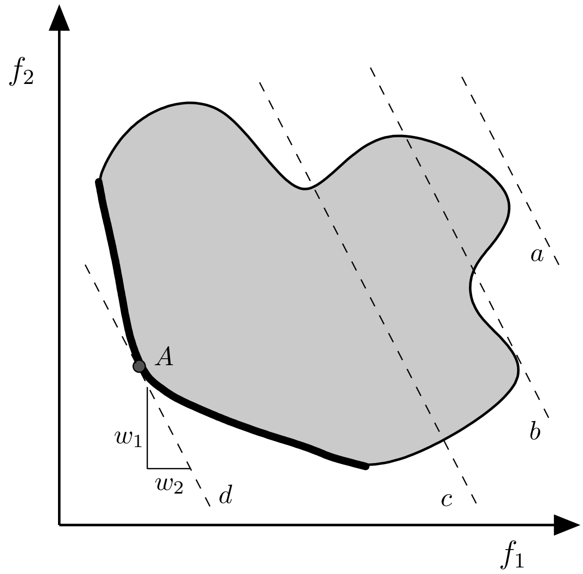


FIGURE 4.9: *Uncovering Pareto optimal solutions by means of the weighted-sum approach for a bi-objective optimisation problem (in which both objective functions f_1 and f_2 are to be minimised) [50].*

The same procedure described above may also be followed to find *certain* Pareto optimal solutions to (4.1)–(4.4) even if the problem is not convex. The problem, however, is that some Pareto optimal solutions are not thus discoverable if (4.1)–(4.4) is not convex. Consider, for example, Figure 4.10 which relates to a bi-objective optimisation problem in which the objective functions f_1 and f_2 again have to be minimised and which has a feasible region that is not convex. If a weight vector w is chosen which results in consideration of the contour lines a or b, then the Pareto optimal solutions A, B and C are discoverable. When considering the line segment BC, it is clear that no contour line will produce a tangential point in the feasible region BC. The reason for this is that before a contour line becomes tangent to any point on the line segment BC, it will also become tangent to another (improved) point in the feasible region. Therefore no Pareto optimal solution along the curve BC is computable via the weighted-sum method.

The weighted-sum method described above should therefore be implemented with great caution [177], and should be avoided altogether in the case of nonconvex problems [50]. For these reasons the weighted-sum method is not adopted in this dissertation.

4.9 Chapter summary

This chapter was dedicated to a discussion on the basic principles involved in multi-objective optimisation. The chapter opened in §4.1 with an introduction to multi-objective optimisation and a mathematical formulation of the general form of a multi-objective optimisation problem. The section also familiarised the reader with the notions of decision and objective spaces. In §4.2, the importance of the notion of convexity in multi-optimisation problems was discussed and it was pointed out that nonconvex multi-objective problems are harder to solve than convex problems.

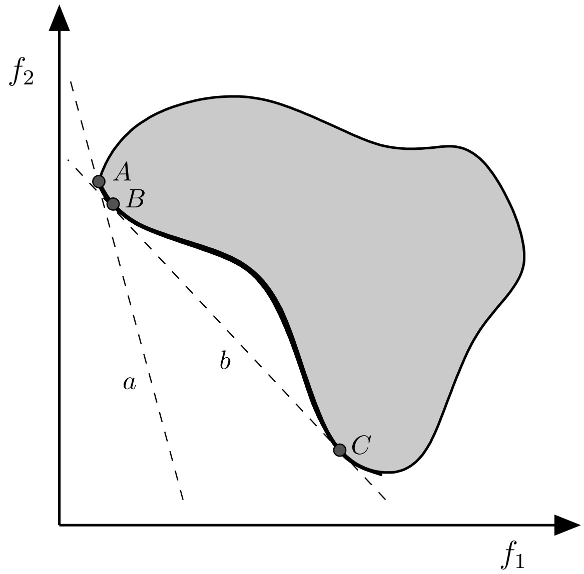


FIGURE 4.10: *Failure of the weighted-sum approach to uncover Pareto optimal solutions for a bi-objective optimisation problem involving a nonconvex feasible region (in which both the objective functions f_1 and f_2 are to be minimised) [50].*

In §4.3, the focus shifted to the notion of solution dominance. The section contained a description of the dominance relation and brief discussions on the properties of this relation as well as the notion of Pareto optimality.

The notion of hypervolume was next elucidated in §4.4 as a performance measure capable of measuring the quality of a nondominated set of solutions as an estimate of the true Pareto front of a multi-objective problem.

In §4.5, two sets of theoretical conditions for Pareto optimality were reviewed. The first was a set of necessary conditions for a solution to (4.1)–(4.4) to be Pareto optimal and the second was a set of sufficient conditions for a solution to (4.1)–(4.4) to be Pareto optimal.

Three methods, ranging in different levels of complexity, were described in §4.6 for identifying a nondominated set of solutions from among a given, finite set of candidate solutions to (4.1)–(4.4). These methods included a naive and slow approach, a continuously updating approach and a more robust method developed by Kung *et al.* [111].

Many multi-objective optimisation techniques in the operations research literature require that candidate solutions to (4.1)–(4.4) be sorted into different classes based on their respective degrees of dominance. In §4.7, a description was given of the well-known fast nondominated sorting algorithm developed by Deb [51] for this purpose.

The chapter finally closed, in §4.8, with a discussion on problems which arise when adopting the classical multi-objective optimisation method of weighting objective functions into a single objective when (4.1)–(4.4) is transformed into a single-objective optimisation problem of the form (4.7)–(4.10).

CHAPTER 5

Solution methodologies for solving WAMs

Contents

A number of solution approaches are available in the operations research literature for solving various types of optimisation problems. These approaches may be partitioned into different classes (depending on the particular problem at hand) and include exact solution approaches, heuristic solution approaches and metaheuristic solution approaches. The approaches in each of these classes may furthermore be tailored to solve single-objective optimisation problems or multi-objective optimisation problems. In this chapter, a number of solution approaches are discussed with a specific focus on those solution approaches that are used later in this dissertation to solve WAMs (*i.e.* problems emanating from the realm of assignment problems). The solution approaches discussed here may therefore be incorporated into a WA subsystem.

The chapter opens in §5.1 with a discussion on two exact solution approaches that are applicable to certain types of WAMs: the method of total enumeration and the well-known branch-and-bound method. Although such exact solution approaches are able to provide optimal solutions to optimisation problems, these approaches are computationally expensive when the dimensions of these optimisation problems become large. These solution approaches are therefore typically abandoned in favour of heuristic or metaheuristic solution approaches which are often able to provide near-optimal solutions in much shorter computational times than those required by exact solution approaches.

Next, heuristic optimisation approaches are touched upon briefly in §5.2 and a number of heuristics designed specifically for use in the context of WA are reviewed. This is followed in §5.3 by a discussion on metaheuristic optimisation approaches. This class of optimisation techniques may be classified into trajectory-based approaches and population-based approaches. Examples of methods in each class are mentioned briefly, after which the working of two metaheuristics that are actually applied later in this dissertation to solve WAMs (approximately) are described in some detail. The first of these metaheuristics is the method of simulated annealing (a trajectory-based approach) and the second is a genetic algorithm (a population-based approach). The methods are first described in §5.3 within the context of single-optimisation and the extensions to the algorithms required in order to render them applicable to the case of multi-objective optimisation are then discussed in §5.4.

5.1 Exact solution approaches

An exact solution approach aims to find a globally optimal solution to an optimisation problem by executing an exhaustive search through a very problem-specific search space — this is achieved either implicitly or explicitly. The advantage of adopting an exact approach to solving an optimisation problem is that it returns an exact solution to the problem (if the dimensions of the problem allow for this), but this boon typically comes at a high computational expense. Two exact optimisation approaches are discussed briefly in this section, namely the (explicit) *method of total enumeration* and the (implicit) *branch-and-bound* search algorithm.

One of the simplest exact solution approaches for combinatorial optimisation problems is the method of *total enumeration*. As the name suggests, the method involves carrying out a complete enumeration of all the solution possibilities in an iterative fashion, keeping track of those alternatives that score best in terms of the objective function [151]. Due to the high computational complexity involved in this method for large problem instances, it is typically only employed in the context of small problems.

For larger problems, an exact method that is computationally less expensive is rather recommended. One such an exact method is the celebrated *branch-and-bound* method, developed by Doig and Land [57] in 1960. This well-known algorithm is a design paradigm for solving combinatorial optimisation problems. The method systematically, but implicitly, enumerates a set of candidate solutions to an optimisation problem by discarding subsets of unsuccessful candidates [151, 168]. This feature of the method often renders it substantially less computationally expensive than the total enumeration method.

Suppose a single-objective optimisation problem has the objective function $f(\boldsymbol{x})$ which is to be maximised, where $\boldsymbol{x}$ is a vector of discrete decision variables. Let $\mathbb{S}$ denote the set of candidate solutions (also called the *search space*) of an optimisation problem over which the decision variable vector $\boldsymbol{x}$ ranges [168]. As the name suggests, the branch-and-bound algorithm employs two user-specified procedures, namely a *branching* procedure and a *bounding* procedure. According to the branching procedure, two or more (smaller) sets $\mathbb{S}'_1, \mathbb{S}'_2, \ldots$ of a given subset $\mathbb{S}' \subseteq \mathbb{S}$ of candidate solutions is returned — the union of these sets is $\mathbb{S}'$. A tree-like data structure, known as the *search tree*, is built up by the branch-and-bound method. The nodes of the tree represent the subsets $\mathbb{S}'_1, \mathbb{S}'_2, \ldots$ of $\mathbb{S}'$. The maximum value of $f(\boldsymbol{x})$ over the subset $\mathbb{S}'$ is computed as the maximum value from the set of solutions $\{f(\boldsymbol{x}_1), f(\boldsymbol{x}_2) \ldots\}$, where $f(\boldsymbol{x}_i)$ attains a maximum function value at $\boldsymbol{x}_i$ within $\mathbb{S}'_i$. Note that a branching procedure should be chosen which is able to produce non-overlapping subsets. The bounding procedure, on the other hand, is employed by the algorithm to determine lower and upper bounds on the maximum value of $f(\boldsymbol{x})$ within a given subset $\mathbb{S}' \subseteq \mathbb{S}$ [168]. A number of methods may be used to achieve this, although no universal bounding procedure exists in the literature (that is, a specialised bounding procedure is employed for the specific problem at hand).

During each iteration of the branch-and-bound algorithm, a branch is formed on a subset $\mathbb{S}' \subseteq \mathbb{S}$ of candidate solutions. Suppose, in a maximisation problem, an upper bound on the objective function for some node $\mathbb{S}'_i$ of the search tree is *smaller* than the lower bound of some other node $\mathbb{S}'_j$ $(i \neq j)$. Then the node $\mathbb{S}'_i$ may be discarded entirely from the search. A global variable is utilised to keep track of the largest lower bound uncovered, denoted by ℓ_b, among all subregions of the search space explored throughout the execution of the branch-and-bound method — a node which achieves an upper bound value smaller than the value of ℓ_b may safely be discarded from the tree. Eliminating a node such as $\mathbb{S}'_i$ from the search tree is known as *pruning* [151, 168]. A stopping criterion is employed in the branch-and-bound algorithm, indicating when the algorithm should terminate. Typical stopping criteria include terminating the algorithm once the current candidate set $\mathbb{S}$ reduces to a single element or when the objective function upper bound of the set $\mathbb{S}$ corresponds to any available objective function lower bound — in such a case, the objective function $f(\boldsymbol{x})$ attains a maximum within $\mathbb{S}$ at any element of $\mathbb{S}$.

The dimensions of the decision spaces of even the simplest WAMs reviewed in Chapter 3 typically grow dramatically as the numbers of WSs and aerial threats increase. For example, if there are m distinct WSs that can be assigned to engage n distinct aerial threats under the very simple assumptions that (a) WSs are assigned to engage threats independently, (b) each WS can be assigned to at most one aerial threat, and (c) all assignments occur simultaneously (*i.e.* no temporal scheduling of WS engagements is required), then there are $(n+1)^m$ distinct possible WS assignments. This number grows very rapidly as m and/or n increases, as illustrated in Table 5.1 for $m, n \in \{5, 10, 15, 20\}$.

	$m = 5$	$m = 10$	$m = 15$	$m = 20$
$n = 5$	7 776	60 466 176	470 184 984 576	3 656 158 440 062 976
$n = 10$	161 051	25 937 424 601	4 177 248 169 415 651	672 749 994 932 560 009 201
$n = 15$	1 048 576	1 099 511 627 776	1 152 921 504 606 846 976	1 208 925 819 614 629 174 706 176
$n = 20$	4 084 101	16 679 880 978 201	68 122 318 582 951 682 301	278 218 429 446 951 548 637 196 401

Table 5.1: *The number of ways in which m distinct WSs can be assigned simultaneously to n distinct aerial threats under the assumptions that WSs are assigned independently and that each WS can be assigned to at most one aerial threat.*

The kind of growth demonstrated in Table 5.1 clearly rules out an explicit total enumeration approach towards solving the WAMs of Chapter 6 for all but the tiniest instances. Furthermore, the growth of the search space of more complicated WAMs requiring temporal scheduling of WAs is significantly more severe than that demonstrated in Table 5.1. Even implicit enumeration schemes, such as the aforementioned branch-and-bound method, cannot accommodate this kind of computational burden for realistic values of m and n. Although it is possible to adopt more sophisticated decomposition or dimensionality reduction approaches, such as Benders decomposition [15] or column generation [45], such methods often struggle to deal with the peculiar structures of certain WAMs and the dimensions of realistic instances of these models. For this reason, researchers have in the past traditionally turned to heuristic solution methodologies for when solving WAM instances.

5.2 Heuristic optimisation approaches

Heuristic[1] solution approaches aim to find near-optimal solutions to optimisation problems which are too complex to solve exactly [86] — these approaches usually involve the pursuit of high-

[1]The word *heuristic* stems from the Greek word *heuriskein* which means *to discover* or *to find*. Perhaps a more modest choice would have been to choose a word deriving from the Greek for *to seek*, for what heuristics *find* sometimes leave much to be desired.

quality feasible solutions instead of necessarily optimal solutions. Three well-known classes of heuristic optimisation approaches that are widely used in the operations research literature are, *iterative algorithms*, *constructive algorithms* (*e.g.* usually algorithms following a greedy incremental solution construction approach) and *local search algorithms* (*e.g.* algorithms following a hill climbing approach).

Iterative heuristics are simple, rigid procedures that are carried out iteratively as the name implies [86]. During each iteration, a new, hopefully better solution is considered. This procedure often incorporates intuitive or expert knowledge rules of thumb and is repeated until some stopping condition is met. The best solution discovered throughout the process is recorded and returned as output. Such optimisation methods typically yield solutions of inferior quality.

Constructive heuristics, on the other hand, usually work by iteratively selecting solution components that are best possible without consideration of any future consequences of the greedy choice during each iteration. A constructive heuristic starts with an empty solution and builds the solution iteratively according to a pre-defined set of rules until a complete solution is constructed. During each iteration, an element to the solution which will yield the best improvement in the solution is usually chosen for insertion in the solution. Constructive heuristics also typically yield solutions of inferior quality.

Local search algorithms (such as hill climbing procedures) involve initially generating an arbitrary, complete candidate solution to an optimisation problem instance and changing a single element of the solution in an iterated fashion in a bid to uncover a sequence of improved successive solutions. If a change produces an improved solution, this solution is used in the next iteration of the algorithm to perform another incremental change. If, however, the change produces a worse solution, an alternative change is made to the solution until an improved solution is found. The algorithm is typically iterated in this fashion until no further improvements can be found. Local searches therefore typically yield locally optimal solutions.

In 2008, Potgieter [145] developed a number of greedy algorithms for solving the WAP. One of these algorithms, called *Greedy Weapon Assignment,* considers the assignment of every WS in turn to the threat which achieves the largest threat priority value and assigns the WS achieving the best effectiveness with respect to that threat. The WS is then removed from the list of available WSs considered for WA engagements and the algorithm iterates in this fashion until all the threats have been considered for WA proposals. The other greedy algorithms proposed by Potgieter [145] are *Threshold Greedy Weapon Assignment, Weighted Greedy Weapon Assignment, Multi-Period Greedy Weapon Assignment* and *Layered Greedy Weapon Assignment.* These algorithms are all specialised extensions of the Greedy Weapon Assigment algorithm.

It is reiterated that heuristic optimisation approaches are *ad hoc* in nature, implying that they are typically designed to fit a specific type of optimisation problem rather than a number of types of problems [86]. The disadvantage of this characteristic is that a new optimisation procedure typically has to be developed for each new optimisation problem considered — a tailored heuristic approach is thus required for each optimisation problem.

Since the aim in this dissertation is to put forward a *generic* WA subsystem architecture design framework in the sense that it should be flexible and easily adaptable, heuristic solution approaches may not be the best choice to solve the WAM prototypes selected for default inclusion in the system design proposed later in this dissertation. Since such solution approaches are typically problem specific and unadaptable, new heuristic solution approach may have to be developed for each new WAM included in the WA subsystem. Furthermore, heuristic approaches may also exhibit overly greedy behaviour which may result in the search becoming trapped at locally optimal solutions. Metaheuristic solution approaches are therefore rather considered to solve the WAM prototypes selected for default inclusion in the WA subsystem later in this dissertation, due to their enhanced flexibility.

5.3 Single-objective metaheuristic optimisation approaches

In recent years, the development of *metaheuristic*[2] optimisation approaches has addressed (to some extent) the known drawbacks associated with heuristic approaches. Hillier and Lieberman [86] define a metaheuristic as "a general kind of solution method that orchestrates the interaction between local improvement procedures and higher level strategies to create a process that is capable of escaping from local optima and performing a robust search of a feasible region." Metaheuristics are therefore methods that are able to provide a general structure and guidelines for developing a tailored solution approach suited to more than one type of problem. They are frequently used to solve combinatorial optimisation problems approximately in view of the phenomenon of *combinatorial explosion*[3].

Metaheuristics may be divided into two distinct classes, *i.e. trajectory-based methods* and *population-based methods*. In trajectory-based methods, a single candidate solution is iteratively maintained and improved in search of a globally optimal solution. Popular examples of trajectory-based metaheuristics include the method of *tabu search* (due to Glover [77]), the method of *simulated annealing* (due to Kirkpatrick *et al.* [104]), the method of *harmony search* (due to Geem *et al.* [72]) and the method of *variable neighbourhood search* (due to Hansen and Mladenovic [132]). Population-based methods, on the other hand, entail modifying and improving multiple candidate solutions at once (often employing population characteristics) in an iterative fashion in order to guide the search towards uncovering a high-quality locally optimal solution. Popular examples of population-based metaheuristics include the *genetic algorithm* (due to Holland [88]), *particle swarm optimisation* (due to Kennedy and Eberhart [102]) and *ant colony optimisation* (due to Dorigo [59]).

	Single-objective	Multi-objective
Trajectory-based	Simulated annealing	DBMOSA
Population-based	Genetic algorithm	NSGA II

FIGURE 5.1: *Classification scheme for the metaheuristic optimisation approaches together with an example methodology in each class that is employed to solve the WAMs later in this dissertation.*

Two metaheuristic optimisation methods are employed later in this dissertation to solve the WAMs proposed for the basis of WA DS. These methods are the method of simulated annealing originally proposed in 1983 by Kirkpatrick *et al.* [104] and the genetic algorithm originally proposed by Holland [88] in 1975. The remainder of this chapter is devoted to detailed discussions on the general working of these two solution approaches. Specific choices made in terms of

[2]The prefix *meta* in the word *metaheuristic* means *beyond, after* or *higher level* implying that metaheuristics generally perform better than heuristics in view of their superior flexibility.

[3]The notion of *combinatorial explosion* is a phenomenon that occurs during the solution of combinatorial optimisation problems. It is the fundamental problem of experiencing exponential growth in the number of combinations that have to be considered in a specific combinatorial optimisation problem. This type of growth typically occurs at such a fast pace that even the fastest computers require a very large amount of computation time to consider all the combinations if the problem dimensions grows past some finite value.

algorithmic building blocks and parameter values for implementation of the algorithms within the context of WA are described in a later chapter. Both the aforementioned solution approaches were, however, in their original incarnations only able to solve single-objective optimisation problems (approximately). Since some of the WAMs proposed later in this dissertation involve the optimisation of multiple objectives simultaneously, multi-objective optimisation extensions to the algorithms are also described toward the end of the chapter. The first of these extensions applies to the simulated annealing algorithm and is called the DBMOSA, originally proposed by Smith *et al.* [173] in 2008. The second extension is to the genetic algorithm and is called the NSGA II, originally proposed by Agarwal *et al.* [2] in 2002. A slight alteration to the latter extension, due to Bullinaria and Garcia-Najera [26], is also touched upon. The classification of the metaheuristic solution methodologies discussed in the remainder of this chapter is illustrated graphically in Figure 5.1.

5.3.1 The method of simulated annealing: A trajectory-based approach

Simulated annealing is a metaheuristic optimisation technique developed by Kirkpatrick *et al.* [104] in 1983 which mimics the physical annealing process involved in the strengthening of solids in metallurgy. During this process, heat treatment is applied to a metal in order to change its molecular properties. The metal is typically heated above its recrystallization temperature, after which it is allowed to cool down very slowly. The heating of the material causes its atoms to become excited and to start vibrating randomly through higher energy states [17]. As the metal is then allowed to cool down slowly, its atoms vibrate less until they settle down into a low energy state. This slow cooling process allows for a better chance of the atoms settling into lower energy states than their initial energy states.

Simulated annealing algorithms are able to control the cycling phenomenon typically induced by accepting non-improving moves in local searches according to probabilities tested by means of computer-generated random numbers. The method is a trajectory-based method since it iteratively considers and performs operations on a single solution at a time. A pseudocode description of the method of simulated annealing is given in Algorithm 5.1[4] (for a minimisation problem).

The simulated annealing algorithm requires as input an initial solution vector $\boldsymbol{x}^{(0)}$, the maximum number of iterations $t_{\max}$ over which the search may be carried out and an initial temperature q_0. The initial solution vector is taken as the incumbent solution at algorithm initialisation. During iteration $t \in \{0, 1, 2, 3, \ldots, t_{\max} - 1\}$ of the algorithm, a neighbourhood solution $\boldsymbol{x}^{(t+1)}$ is generated from $\boldsymbol{x}^{(t)}$ by performing small alterations to $\boldsymbol{x}^{(t)}$ as a result of the application of a so-called *neighbourhood move operator*. The move operator typically performs a random perturbation to $\boldsymbol{x}^{(t)}$ from a set of possible moves. This set of possible moves is typically problem type-specific with the specific combinatorial context in mind.

A move that results in an improvement upon the objective function value of $\boldsymbol{x}^{(t)}$ is always accepted, while a move that does not achieve such an improvement is accepted according to the classical *Metropolis rule* [29], *i.e.* with probability

$$\exp(-\Delta_{obj}/q_i),$$

where Δ_{obj} denotes the change in the objective function when moving from the current solution

[4]It should be noted that the pseudocode description of the method of simulated annealing in Algorithm 5.1 follows a general outline of the algorithm. The description of the algorithm in the text, however, includes extensions of the method which may promote diversity and improved search methods which may uncover larger portions of the search space when employed, for the sake of completeness.

Algorithm 5.1: Simulated annealing (for a minimisation problem)

Input : An initial candidate solution $\boldsymbol{x}^{(0)}$, a maximum allowable number of iterations $t_{\max}$ and an initial temperature q_0.
Output: An approximately optimal solution vector $\hat{\boldsymbol{x}}$ to a single-objective minimisation problem.

$t \leftarrow 0$
$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(0)}$
while $t < t_{\max}$ **and** *all feasible moves have not been rejected* **do**
  Randomly choose a feasible move $\Delta\boldsymbol{x}^{(t+1)}$
  Compute the (possibly negative) net objective function improvement Δobj for moving from $\boldsymbol{x}^{(t)}$ to $\boldsymbol{x}^{(t)} + \Delta\boldsymbol{x}^{(t+1)}$
  if $\Delta\text{obj} < 0$ **or** *with probability* $e^{-\Delta\text{obj}/q}$ *if* $\Delta\text{obj} \geq 0$, **then**
    $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{x}^{(t)} + \Delta\boldsymbol{x}^{(t+1)}$
  if *the objective function value of* $\boldsymbol{x}^{(t+1)}$ *is superior to that of the incumbent solution* $\hat{\boldsymbol{x}}$ **then**
    $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(t+1)}$
  If a sufficient number of iterations have passed since the last temperature change, reduce temperature q
  $t \leftarrow t + 1$
return $\hat{\boldsymbol{x}}$

$\boldsymbol{x}^{(t)}$ to the randomly chosen neighbouring solution and q_i denotes the temperature during stage or epoch i of the search, which controls the randomness of the search. If the neighbouring solution is accepted, it becomes the new current solution $\boldsymbol{x}^{(t+1)}$ during the next iteration of the algorithm, while if the neighbouring solution is rejected, the move operator is applied to $\boldsymbol{x}^{(t)}$ once more, repeating this process until acceptance occurs. If $\boldsymbol{x}^{(t+1)}$ performs better than the current incumbent $\hat{\boldsymbol{x}}$, then $\boldsymbol{x}^{(t+1)}$ becomes the new incumbent.

For large values of q_i, the majority of new solutions are accepted — this provides a mechanism for the search to escape when trapped at a local optimum. When q_i is small, on the other hand, only new worsening solutions that result in small degradations of the objective function value are typically accepted. The algorithm is therefore usually initialised with a large value of q_0 in order to allow for as much exploration of the decision space during the early stages of the search as possible.

According to Busetti [28], the initial temperature used in the search should be such that approximately eighty percent of all the non-improving moves are accepted at the beginning of the search. One way of approximating such an initial temperature value is to conduct a trial search during which all non-improving moves are accepted. A good estimation of the initial temperature may then be taken as

$$q_0 = \frac{\Delta^+}{\ln 0.8}, \tag{5.1}$$

where Δ^+ denotes the average change of the objective function values resulting from accepting non-improving moves during the trial search.

The temperature is typically kept constant for a number of consecutive iterations. A set of successive iterations over which the temperature remains constant is known as an *epoch*. The length of an epoch is not a fixed parameter value — an epoch rather has a maximum number of iterations associated with it. The lengths of various epochs may therefore differ from one another. The length of an epoch (*i.e.* the number of iterations the search spends in temperature stage i) is typically determined by a Markov chain of length L_i. According to Busetti [28], this value should be customised to the optimisation problem at hand rather than being a function of i. It seems intuitive to require a pre-specified minimum number of move acceptances during any temperature stage before lowering the temperature and initiating the next epoch. Let $A_{\min}$ denote this pre-specified number of move acceptances. As the temperature q_i approaches zero as $i \to \infty$, non-improving moves are typically accepted with decreasing probability — this results in the number of trials expected before accepting $A_{\min}$ moves becoming large (without bound) as the search progresses, irrespective of the value of $A_{\min}$. In order to counter this situation, the epoch may be terminated once L moves have been attempted (by increasing the temperature) or once $A_{\min}$ moves have been accepted (by lowering the temperature), where $L > A_{\min}$. According to Dreo *et al.* [62], a good rule of thumb is to take $L = 100$ and $A_{\min} = 12N$, where N is some measure of the number of degrees of freedom related to the optimisation problem in question.

Cooling and/or reheating can therefore occur multiple times over the course of the algorithm according to the scheme oulined above. Whereas cooling is aimed at making it harder to accept worsening solutions (in a bid to promote exploitation), reheating is aimed at making it easier to accept worsening solutions (in a bid to promote exploration by facilitating escape from local optima). The algorithm iterates in this fashion until a stopping criterion is reached. When the algorithm terminates, the incumbent solution vector $\hat{\boldsymbol{x}}$ is returned by the algorithm as output.

The aim in simulated annealing is to find a locally optimal solution of high quality with minimal computational effort. In order to achieve this, a tailor-made set of input parameters (*i.e.* the starting temperature, the stopping criterion, the cooling schedule, the heating schedule and an epoch termination criterion) is required for every optimisation problem instance. A popular stopping criterion involves the specification of a maximum allowable number of epochs. A stopping temperature may also be employed in which case the algorithm terminates when the temperature is close to zero for an extended period of time — this allows for the algorithm to converge towards a locally optimal solution.

Typical cooling schedules which may be employed in the algorithm include geometric, linear and adaptive schedules [1, 92, 17, 199]. Of these cooling schedules, the geometric schedule is the most common. Some of the other schedules have, however, proven to be more efficient than the geometric schedule in certain contexts [199]. According to the geometric cooling schedule, the temperature at each epoch is reduced by a fixed factor α (ranging between 0 and 1). The temperature during epoch $i + 1$ is therefore

$$q_{i+1} = \alpha q_i, \qquad i = 0, 1, 2, \ldots \tag{5.2}$$

An analysis by Vigeh [199] on good values of α concluded that the best results are generally found when α is chosen between 0.8 and 0.99. Note that the computational effort required to execute the algorithm will be greater for a large value of α, since the total number of temperature values to explore will typically be greater.

According to the alternative linear cooling schedule, the temperature at the end of an epoch is lowered by subtracting a constant cooling factor from the current temperature. The temperature during epoch $i + 1$ is therefore

$$q_{i+1} = q_i - c, \qquad i = 0, 1, 2, \ldots, \tag{5.3}$$

where c is a constant cooling factor. In contrast to the geometric cooling schedule, the linear cooling schedule reduces the temperature by the same amount throughout the execution of the algorithm.

An example of an adaptive cooling schedule is that of Huang *et al.* [92], according to which

$$q_{i+1} = q_i \exp\Big(-\frac{\Lambda q_i}{\sigma(q_i)}\Big), \qquad i = 0, 1, 2, \ldots, \tag{5.4}$$

where $\Lambda \in (0, 1]$ is a parameter whose value has to be determined empirically and $\sigma(q_t)$ denotes the standard deviation achieved in the changing objective function values since the start of epoch i. A typical value of Λ is 0.7 [92].

Reheating is implemented in a similar fashion as cooling, except that the temperature is increased by some factor when reheating. According to a geometric reheating schedule, the temperature at each epoch is increased by a fixed factor β (chosen larger than one). Following the same notation as in the cooling schedules (5.2)–(5.4), the temperature during epoch $i+1$ is therefore

$$q_{i+1} = \beta q_i, \qquad i = 0, 1, 2, \ldots. \tag{5.5}$$

A linear reheating schedule may also be employed in which case the temperature at each epoch is increased by adding a positive constant reheating factor to the current temperature. The temperature during epoch $i+1$ in such a case is therefore

$$q_{i+1} = q_i + h, \qquad i = 0, 1, 2, \ldots, \tag{5.6}$$

where h is the reheating factor. As with the linear cooling schedule (5.3), the temperature is increased by the same amount during the execution of the algorithm.

5.3.2 The genetic algorithm: A population-based approach

A genetic algorithm is an evolutionary algorithm based on Darwin's theory of evolution by means of natural selection. According to this algorithm, originally proposed by Holland [88], a population of candidate solutions, also called *chromosomes*, are allowed to evolve over a number of iterations (representing time) within a carefully controlled environment in an attempt at uncovering near-optimal solutions to an optimisation problem instance. During each iteration, parent solutions in the population are selected, based on a pre-specified set of criteria collected into a so-called *fitness measure*, in order to produce offspring solutions which populate the next generation of candidate solutions. The fitness of a candidate solution is an indication of the quality of the solution relative to other solutions with respect to the objective function. A typical fitness measure employed in genetic algorithms is the objective function value of a solution itself in the case of a maximisation problem. Offspring solutions are produced by applying a *selection* operator to the current generation of solutions in order to select parent solutions and by then applying *recombination* operators to the selected parent solutions. The newly generated offspring solutions populate the next generation of candidate solutions (typically replacing their parents). This process is iterated over a number of solution generations until significantly fitter solutions can no longer be found or until a pre-specified number of generations is reached. A pseudocode description of a generic genetic algorithm is given in Algorithm 5.2 and the working of the algorithm is described in some detail in the remainder of this section.

The algorithm requires as input the size N of the population, the maximum number of iterations $t_{\max}$ over which the algorithm has to be executed, the recombination probabilities p_c and p_m and the size of a tour S_t to be employed in the selection procedure. The algorithm is initiated by generating an initial candidate solution population $\hat{\boldsymbol{P}}^{(0)}$ of size N which is typically selected in a random fashion. Next, the fitness of each candidate solution in the population is calculated.

Algorithm 5.2: Genetic algorithm

Input : A combinatorial optimisation problem and a domain set for each decision variable, a population size N, the maximum number of iterations $t_{\max}$, a mutation probability p_m, a crossover probability p_c and a tour size S_t.

Output: A converged population of solutions $\hat{\boldsymbol{P}}$ containing an approximation of a globally optimal solution $\hat{\boldsymbol{x}}$ to the problem.

Generate an initial population of solutions $\hat{\boldsymbol{P}}^{(0)}$ of size N randomly
Calculate the fitness of each solution in $\boldsymbol{P}^{(0)}$
$t \leftarrow 0$
$\hat{\boldsymbol{P}} \leftarrow \boldsymbol{P}^{(0)}$
while $t < t_{\max}$ **do**
    Apply a tournament selection operator to $\boldsymbol{P}^{(t)}$ to choose parent solutions for crossover based on the fitness values of candidate solutions
    Apply with probability p_c a crossover operator to the parent solutions selected
    Apply with probability p_m a mutation operator to the offspring solutions
    Calculate the fitness of each solution in $\boldsymbol{P}^{(t)}$
    $t \leftarrow t + 1$
return $\hat{\boldsymbol{P}}$, containing $\hat{\boldsymbol{x}}$

The selection operator and the recombination operators (often called *crossover* and *mutation* operators) are next applied to the population [151].

First, two parent solutions are selected for crossover. A popular selection method is the so-called *tournament selection* procedure [61]. In tournament selection, a small subset of solutions, known as a *tour*, is chosen randomly, and the number of solutions in the subset is called the *tour size*. One solution is then chosen from the tour to include as a parent solution within a so-called *mating pool* of solutions. The number of parent solutions included in the mating pool is called the *pool size*. Solutions are typically selected for inclusion in the mating pool by adopting their fitness function values as selection criterion (*i.e.* a candidate solution achieving a very good objective function value has a very good chance of being chosen as a parent solution). A simple way of achieving this is by the popular *roulette wheel selection* procedure [10]. The method employs the fitness assigned to chromosomes in order to calculate a probability of being selected for each chromosome. Let f_i denote the fitness value assigned to chromosome i. The probability of selection for chromosome i is then taken as

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}.$$

The approach involves normalisation of the fitness values of the chromosomes. Next, a circular wheel is partitioned into arcs (similar to a roulette wheel in a casino), each arc representing a chromosome, with these arcs spanning angles that are proportional to the fitness values of the corresponding chromosomes. A fixed point is then chosen on the wheel and it is rotated. The chromosome corresponding to the arc of the wheel which coincides with the fixed point is then chosen as the first parent solution for crossover. The procedure is repeated to find a second parent solution for crossover, after having removed the first parent from the population.

Once two parent solutions have been selected for recombination, a crossover operator is applied to the pair in order to produce an offspring solution. The crossover procedure is stochastic and is associated with a probability p_c of occurring. The value $p_c \in (0, 1)$ is typically chosen large, since the crossover procedure is considered a central part of introducing variation (diversity) into the population of candidate solutions [86, 109, 151]. A number of crossover techniques exist in the operations research literature, including single-point crossover, two-point crossover, cut-and-splice crossover and uniform crossover [86, 108, 167]. The method employed typically depends on the encoding of a solution to the problem under consideration. To illustrate the notion of crossover, single-point crossover is considered and illustrated in Figure 5.2. This method entails uniformly selecting a *single* point along the parent solution encodings and slicing both the encodings at that point. Two offspring solutions are then obtained by interchanging the elements in the parent solution encodings beyond this point.

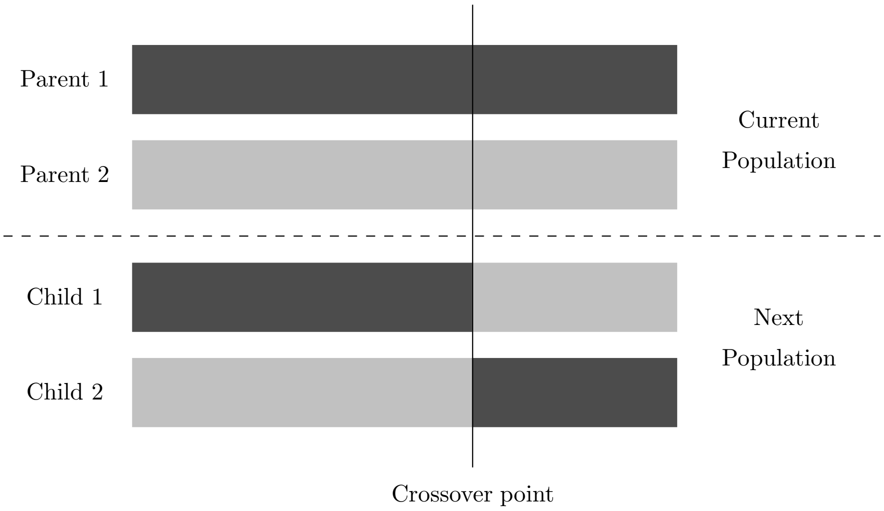


FIGURE 5.2: *Single-point crossover applied to the encodings of two parent solutions in a genetic algorithm.*

A two-point crossover method similarly entails uniformly selecting *two* points along the parent solution encodings and slicing both the solution encodings at that point. Two offspring solutions are then obtained by interchanging the elements in the parent solution encodings in between these two points.

In uniform crossover, the parent solution encodings are not partitioned into segments — each element in an encoding is considered individually for inclusion in the offspring solutions. Solution elements are selected from the parent solutions based on probabilities — bias may therefore be introduced with respect to one of the parent solutions.

After offspring solutions have been generated, a mutation operator is applied to certain offspring solutions. Mutation is aimed at further diversifying solutions in an attempt to explore new regions of the solution space by avoiding premature convergence of the algorithm to poor local optima and involves altering the value of one or more of the entries in a solution encoding from its original state [86]. Mutation is typically executed in a random fashion, and is also a stochastic process (*i.e.* is associated with a probability p_m of occurring). In contrast to the crossover operator, mutating a solution is typically associated with a very small probability of occurring since a large value may result in the algorithm reducing to a more random search [109]. There are a number of mutation operators available in the operations research literature to choose from, including bit flip mutation, uniform mutation and Gaussian mutation [88, 109, 167]. In order to illustrate the notion of mutation, consider the notion of bit flip mutation as illustrated in Figure 5.3. This method is typically employed in optimisation problems with binary-coded solutions and entails inverting the bit (gene) values (*i.e.* flipping the value of a gene from 0 to 1 or *vice versa*).

$$\text{Child before mutation } \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & \mathbf{0} & 1 & 0 \end{bmatrix}$$
$$\downarrow$$
$$\text{Child after mutation } \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & \mathbf{1} & 1 & 0 \end{bmatrix}$$

Figure 5.3: *Bit flip mutation applied to a binary-coded offspring solution.*

In a uniform mutation, a gene in an offspring solution is chosen randomly and the value of this gene is replaced with a uniformly random value (which falls within user-specified upper and lower bounds for the gene). This method of mutation is only used for problems involving real or integer-coded genes. Gaussian mutation, on the other hand, entails choosing a gene randomly from an offspring solution and adding a Gaussian distributed random value to the selected gene. If the value of the gene falls outside user-specified upper and lower bounds for the gene after the addition, then the value of the gene is taken as the user-specified upper or lower bound of the gene, whichever is applicable.

The algorithm terminates once the specified stopping criterion is satisfied. Typical stopping criteria employed in genetic algorithms involve termination once a pre-determined number of iterations is reached (as in Algorithm 5.2), a satisfactory level of fitness is achieved or the algorithm fails to improve on the quality of solutions sufficiently [86, 109, 151]. Once the algorithm terminates, the solution achieving the best fitness value $\hat{\boldsymbol{x}}$ in the converged population of solutions $\hat{\boldsymbol{P}}$, called the *incumbent*, is returned by the algorithm as output.

5.4 Multi-objective metaheuristic optimisation approaches

The two single-objective metaheuristics described in §5.3 have been extended to solve multi-objective optimisation problems. Recall, from Chapter 4, that the aim in multi-objective optimisation problems is to approximate a good spread of solutions along the Pareto frontier as close as possible to the true Pareto frontier. A number of multi-objective metaheuristic solution methodologies in the operations research literature may be employed for this purpose, such as the NSGA II of Agarwal *et al.* [2], the multi-objective particle swarm optimisation technique of Moore and Chapman [133] and the DBMOSA of Smith *et al.* [173] to name but a few. In this section, and extension of the general genetic algorithm, namely the NSGA II and an extension of the general simulated annealing algorithm, namely the DBMOSA are discussed in some detail.

5.4.1 The dominance-based multi-objective method of simulated annealing

The DBMOSA was developed by Smith *et al.* [173] in 2008 and is an extension of the original simulated annealing algorithm of Kirkpatrick [104] (described in §5.3.1), making provision for the solution of multi-objective optimisation problems of the form (4.1)–(4.4). It therefore features many of the operations of the original simulated annealing algorithm described earlier — the main difference is that the notion of *archiving* is introduced in the DBMOSA.

Recall that the simulated annealing algorithm generates only a single solution during each iteration in the execution thereof (*i.e.* it is a trajectory-based metaheuristic). Therefore, the DBMOSA maintains an external set of solutions known as an *archive*, denoted here by $\mathcal{A}$, which contains all the nondominated solutions uncovered by the algorithm during its search progression. The DBMOSA works in such a way that all the solutions generated during the course of the search are candidates for becoming members of the archive. New candidate solutions are compared for dominance with respect to existing members of the archive when deciding whether or not to include them in the archive. The notion of archiving is illustrated graphically in Figure 5.4 for a bi-objective optimisation problem in which both objective functions f_1 and f_2 are to be maximised.

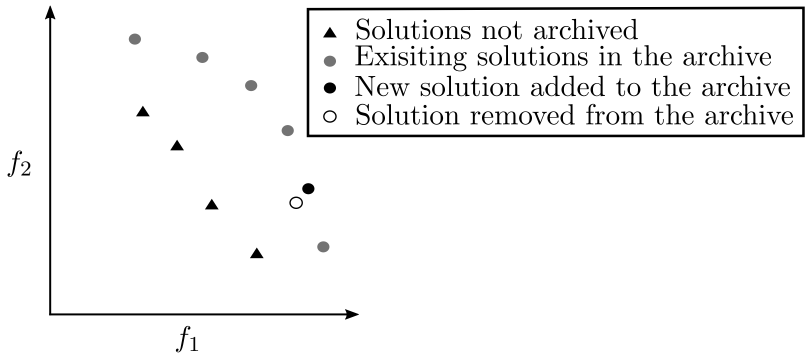


FIGURE 5.4: *The notion of archiving employed in the DBMOSA in a bi-objective maximisation problem with objective functions f_1 and f_2.*

Consider the candidate solutions in Figure 5.4. The solutions uncovered during the search thus far are the solutions represented by the light grey circles (*i.e.* the existing nondominated solutions in the archive), the open circle (*i.e.* a nondominated solution which also forms part of the archive) and the black triangles (*i.e.* the solutions found thus far, which are dominated and do not form part of the archive). Suppose a new candidate solution (represented by the black circle) is generated by the algorithm. This solution is compared with every solution in the archive and it is clear from the figure that it only dominates the solution represented by the open circle — it does not dominate the other solutions in the archive, nor do the other solutions in the archive dominate it. The solution represented by the black circle is therefore included in the existing archive and the solution represented by the white open circle is removed from the archive.

In the simulated annealing algorithm for single-objective optimisation problems, the sign of the difference in energy denoted by Δobj in §5.3.1, provided information on the "quality" of the neighbouring solution $\boldsymbol{x}'$ with respect to the current solution $\boldsymbol{x}$ — an indication as to whether the neighbouring solution $\boldsymbol{x}'$ performs better or worse than the current solution $\boldsymbol{x}$ in terms of the single objective function. Suppose, in the multi-objective case, that the true Pareto front $\mathcal{P}_F$ were available *a priori*. Then it would instead be possible to define the energy of a solution $\boldsymbol{x}$ as a measure of the portion of the true Pareto front that dominates $\boldsymbol{x}$. Let $\mathcal{P}_F(\boldsymbol{x})$ denote the portion of the true Pareto front that dominates $\boldsymbol{x}$, then

$$\mathcal{P}_F(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{P} \mid \boldsymbol{y} \prec \boldsymbol{x}\}.$$

The energy of $\boldsymbol{x}$ may be defined as

$$E(\boldsymbol{x}) = \mu(\mathcal{P}_F(\boldsymbol{x})),$$

where μ denotes the aforementioned measure defined on $\mathcal{P}_F$. Note that $\mathcal{P}_F$ may be either discrete or continuous. If $\mathcal{P}_F$ is discrete, $\mu(\mathcal{P}_F(\boldsymbol{x}))$ may be taken as the cardinality of $\mathcal{P}_F(\boldsymbol{x})$ — the number of solutions forming part of $\mathcal{P}_F$ which dominate $\boldsymbol{x}$. If, on the other hand, $\mathcal{P}_F$ is continuous, μ may be taken as a Lebesgue measure — the length, the area or volume in the cases of two, three or four objectives, respectively, of the portion of the Pareto front which dominates $\boldsymbol{x}$.

Recall, from Chapter 4, that the true Pareto front of solutions is typically unknown during the optimisation process, which renders use of estimation of the energy function described above rather difficult. Smith *et al.* [173] instead proposed that the energy function be defined in terms of the current estimate of the Pareto front — the set of mutually nondominated solutions uncovered during the course of the search (*i.e.* the solutions contained in the archive $\mathcal{A}$). Therefore, the difference in energy may be estimated by taking the difference between the energy of the solution in the archive and the neighbouring solution and normalising this difference by the size of the archive. Smith *et al.* [173] claimed that the use of this energy measure promotes both coverage of and convergence towards the true Pareto front.

A pseudocode description of the working of the DBMOSA is provided in Algorithm 5.3 and the working of the algorithm is briefly discussed in the remainder of this section. The algorithm is initialised by generating an initial feasible solution $\boldsymbol{x}$ randomly and placing it in the archive $\mathcal{A}$. Define $\tilde{\mathcal{A}} = \mathcal{A} \cup \{\boldsymbol{x}\} \cup \{\boldsymbol{x}'\}$, where $\boldsymbol{x}'$ denotes the neighbouring solution generated in the neighbourhood of the current $\boldsymbol{x}$. Furthermore, define $\tilde{\mathcal{A}}_{\boldsymbol{x}} = \{\boldsymbol{y} \in \tilde{\mathcal{P}} \mid \boldsymbol{y} \prec \boldsymbol{x}\}$, where $\mu(\tilde{\mathcal{A}}_{\boldsymbol{x}}) = |\tilde{\mathcal{A}}_{\boldsymbol{x}}| + 1$, *i.e.* $\tilde{\mathcal{A}}_{\boldsymbol{x}}$ denotes the subset of solutions in $\tilde{\mathcal{A}}$ that dominates $\boldsymbol{x}$. The difference in energy between $\boldsymbol{x}$ and $\boldsymbol{x}'$ may then be estimated as

$$\Delta_E(\boldsymbol{x}', \boldsymbol{x}) = \frac{|\tilde{\mathcal{A}}_{\boldsymbol{x}'}| - |\tilde{\mathcal{A}}_{\boldsymbol{x}}|}{|\tilde{\mathcal{A}}|}. \tag{5.7}$$

In (5.7), division by $|\tilde{\mathcal{A}}|$ ensures that the estimated energy value $\Delta_E(x', x)$ remains below unity. One advantage of this is that it may provide some degree of robustness against fluctuations in the number of solutions contained in $\mathcal{A}$ during execution of the search. This implies that if $\tilde{\mathcal{A}}$ is a nondominated set, then the difference in energy between any two of the elements contained in $\tilde{\mathcal{A}}$ is zero. If $\boldsymbol{x}' \preceq \boldsymbol{x}$, then $\Delta_E(\boldsymbol{x}', \boldsymbol{x}) < 0$, hence the inclusion of the current as well as the neighbouring solution in the energy function in (5.7). Furthermore, another advantage of the energy function in (5.7) is that it encourages the search to explore regions of the estimated Pareto front in $\mathcal{A}$ which are sparsely populated, regardless of the portion of the true Pareto front that dominates the current solution $\boldsymbol{x}$ and the neighbouring solution $\boldsymbol{x}'$. This characteristic is illustrated graphically in Figure 5.5.

It appears as though $\mu(\mathcal{P}_{\boldsymbol{x}'}) > \mu(\mathcal{P}_{\boldsymbol{x}})$ in Figure 5.5, when considering the portion of the true Pareto front that dominates the neighbouring solution $\boldsymbol{x}'$ and that which dominates the current solution $\boldsymbol{x}$ in objective space. This is, however, not the case because $|\tilde{\mathcal{A}}_{\boldsymbol{x}'}| = 1$ and $|\tilde{\mathcal{A}}_{\boldsymbol{x}}| = 3$, from which follows that $|\tilde{\mathcal{A}}_{\boldsymbol{x}'}| = 1 < 3 = |\tilde{\mathcal{A}}_{\boldsymbol{x}}|$.

Once a neighbouring solution $\boldsymbol{x}'$ of the current solution $\boldsymbol{x}$ has been generated, its fitness value is compared to the fitness value of $\boldsymbol{x}$. If the fitness value of $\boldsymbol{x}'$ is larger than the fitness value of $\boldsymbol{x}$, the neighbouring solution $\boldsymbol{x}'$ is accepted with probability 1. If, on the other hand, the fitness value of $\boldsymbol{x}'$ is not better than that of $\boldsymbol{x}$, then $\boldsymbol{x}'$ is accepted according to the *Metropolis acceptance rule* [173] which states that $\boldsymbol{x}'$ should be accepted as the new current solution for the next iteration with probability $\exp(-\frac{\Delta_E(\boldsymbol{x}', \boldsymbol{x})}{q})$, where q denotes the current temperature of the search. The acceptance probability of $\boldsymbol{x}'$ may thus be summarised as

$$P(\boldsymbol{x}') = \min\left\{1, \exp\left(-\frac{\Delta_E(\boldsymbol{x}', \boldsymbol{x})}{q}\right)\right\}.$$

It should be noted that a neighbouring solution which is dominated by fewer elements of the currently estimated Pareto front in $\mathcal{A}$ naturally achieves a lower energy value and is, hence, automatically accepted as the new current solution — it is an improving move in the search. If, on the other hand, there is a large energy difference between the current solution and the

---

Algorithm 5.3: Dominance-based multi-objective simulated annealing [173]

Input : A multi-objective maximisation problem instance of the form (4.1)–(4.4), the maximum allowable number of iterations per epoch I_{max}, the minimum number of accepted moves A_{min} per epoch, the cooling function used to determine the cooling schedule, and a restriction on the maximum number C_{max} of epochs which may pass without the acceptance of any new solutions.

Output: A nondominated set of solutions $\mathcal{P}_S$ approximating the Pareto front of the given instance of (4.1)–(4.4).

Generate an initial feasible solution $\boldsymbol{x}$;
Initialise the archive $\mathcal{A} = \{\boldsymbol{x}\}$;
Initialise the cooling schedule epoch $c \leftarrow 1$;
Initialise the number of iterations $t \leftarrow 1$;
Initialise the number of epochs without an accepted solution $\xi \leftarrow 0$;
while $\xi \leq C_{max}$ **do**
    $A \leftarrow 0$;
    while $t \leq t + L_c$ *and* $A < A_{min}$ **do**
        Generate a neighbouring solution $\boldsymbol{x}'$ from the current solution $\boldsymbol{x}$;
        Assess the energy difference $\triangle_E(\boldsymbol{x}', \boldsymbol{x})$ according to (5.7);
        Generate a random number $r \in (0, 1)$;
        if $r < \min\{1, \exp\left(\frac{-\triangle_E(\boldsymbol{x}', \boldsymbol{x})}{q_c}\right)\}$ **then**
            $\boldsymbol{x} \leftarrow \boldsymbol{x}'$;
            if $|\mathcal{A}_{\boldsymbol{x}}| = 0$ **then**
                $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}\}$;
                for $\boldsymbol{y} \in \mathcal{A}$ **do**
                    if $\boldsymbol{x} \prec \boldsymbol{y}$ **then**
                        $\mathcal{A} \leftarrow \mathcal{A} \backslash \{\boldsymbol{y}\}$;
                        $A \leftarrow A + 1$;
        $t \leftarrow t + 1$;
    $c \leftarrow c + 1$;
    Update the system temperature q_c;
    if $A = 0$ **then**
        $\xi = \xi + 1$;
$\mathcal{P}_F \leftarrow \mathcal{A}$;

---

neighbouring solution, and the temperature q is low, then the probability $P(\boldsymbol{x}')$ of accepting $\boldsymbol{x}'$ will be small. The probability of acceptance therefore does not depend on an *a priori* weighting of the objectives — the probability of acceptance will remain unchanged if the objective functions were to be rescaled. If the neighbouring solution is accepted, it becomes the new current solution from which a new neighbouring solution is generated.

The DMBOSA is also executed iteratively in stages known as epochs (similar to those incorporated in the simulated annealing method for single-objective optimisation problems). The temperature q of the search remains constant during an epoch. The length of an epoch is again determined by the success of the search, as described in §5.3.1. An epoch is either terminated by performing a *cooling* or *reheating* operator. Cooling is aimed at making it harder to accept worsening solutions — this typically happens when too many neighbouring solutions are being

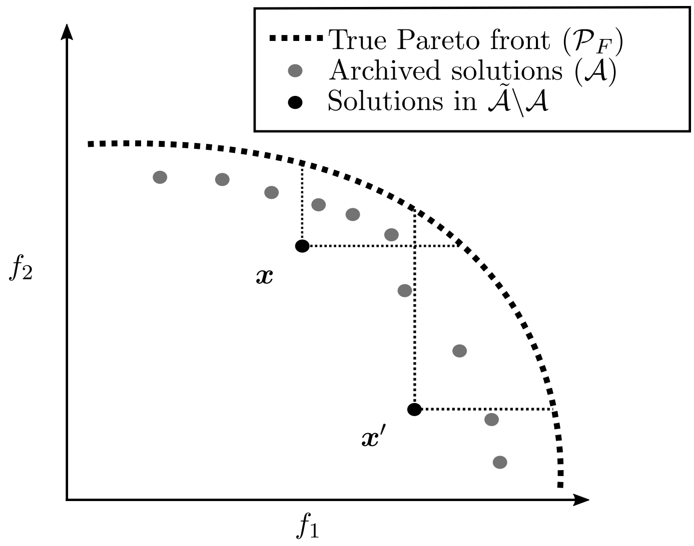


FIGURE 5.5: *The difference in energy of a current solution* $\boldsymbol{x}$ *and its neighbouring solution* $\boldsymbol{x}'$ *for a bi-objective optimisation problem in which both objectives* f_1 *and* f_2 *are to be maximised.*

accepted. A cooling schedule is employed to achieve this and any one of the cooling schedules described earlier may be employed in the DBMOSA. On the other hand, reheating is aimed at making it easier to accept worsening solutions — this typically happens when too few solutions are being accepted.

The algorithm is iterated in this fashion until a stopping criterion is met. A popular stopping criterion is to terminate the search when a specified number of successive epochs have elapsed without accepting a single solution. Once the algorithm terminates, the archive $\mathcal{A}$ containing the approximate Pareto front is returned by the algorithm as output.

5.4.2 The nondominated sorting genetic algorithm II

The NSGA II was put forward in 2002 by Agarwal *et al.* [2] and aims to find a good approximation of Pareto optimal solutions to a multi-objective optimisation problem of the form (4.1)–(4.4). Its working is based on that of the genetic algorithm described in §5.3.2 and therefore shares a number of features of the genetic algorithm. The main difference between the algorithms is the way in which fitness values are assigned to solutions and, consequently, the way in which parent solutions are selected for crossover.

A pseudocode description of the working of the NSGA II is given in Algorithm 5.4 and the working of the algorithm is discussed briefly in this section. The NSGA II takes as input a vector $\boldsymbol{z}$ containing values for each of the M objective functions, the size of the population N of candidate solutions to be maintained during the search, and the maximum number of generations $G_{\max}$ for which the algorithm should be executed. An initial population $\boldsymbol{P}_0$ of candidate solutions of size N is generated randomly. These solutions are then ranked and sorted by using the FNSA [51] (discussed in §4.7). For each solution $\boldsymbol{i}$, a dominance count $d^c_{\boldsymbol{i}}$ (the number of solutions that dominate $\boldsymbol{i}$), and that subset $S_{\boldsymbol{i}}$ of solutions within the current population dominated by $\boldsymbol{i}$, are computed. All solutions achieving a dominance count of $d^c_{\boldsymbol{i}} = 0$ are placed in a separate set $\mathcal{F}_1$, called the *first nondominated front*, and are assigned rank 1. For each solution $\boldsymbol{i}$ in $\mathcal{F}_1$, the algorithm cycles through each solution $\boldsymbol{j}$ in $S_{\boldsymbol{i}}$, and decrements its

$d^c_{\boldsymbol{j}}$-value by one, thus discounting the effect of solution $\boldsymbol{i}$ on solution $\boldsymbol{j}$'s dominance count. All solutions of rank 1 are then removed from the current population of solutions. The solutions now achieving a dominance count of $d^c_{\boldsymbol{i}} = 0$ are placed in another set $\mathcal{F}_2$ for the algorithm to cycle through. These solutions are called the *second nondominated* front and are assigned rank 2. The algorithm continues in this fashion until all the solutions have been partitioned into nondominated fronts and have been assigned ranks.

Algorithm 5.4: Nondominated Sorting Genetic Algorithm II [2].

Input: A multi-objective maximisation problem instance of the form (4.1)–(4.4), the size of the population N, and the maximum number of generations G_{max}.

Output: A set $\mathcal{W}^*$ of approximately Pareto optimal solutions to the instance of (4.1)–(4.4).

Generate an initial solution $\boldsymbol{P}_0$ of size N randomly
Rank and sort $\boldsymbol{P}_0$ by using the FNSA [Algorithm 4.7]
Calculate the crowding distance for each solution in $\boldsymbol{P}_0$ by using the crowding distance assignment algorithm [Algorithm 5.5]
Create child population $\boldsymbol{Q}_0$ of size N by using binary tournament selection based on the crowding distance operator $\prec_c$ from $\boldsymbol{P}_0$, and performing crossover and mutation
$t \leftarrow 0$ **while** $t < G_{\max}$ **do**
    $\boldsymbol{R}_t \leftarrow \boldsymbol{P}_t \cup \boldsymbol{Q}_t$
    Rank and sort $\boldsymbol{R}_t$ into nondominated fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$ by using FNSA
    $\boldsymbol{P}_{t+1} \leftarrow \emptyset$ and $m \leftarrow 1$
    while $|\boldsymbol{P}_{t+1}| < N$ **do**
        if $|\mathcal{F}_m| + |\boldsymbol{P}_{t+1}| \leq N$ **then**
            $\boldsymbol{P}_{t+1} \leftarrow \boldsymbol{P}_{t+1} \cup \mathcal{F}_m$
        else
            Calculate the crowding distance for each solution in $\mathcal{F}_m$
            Sort the solutions in $\mathcal{F}_m$ in descending order, based on crowding distance
            $\boldsymbol{P}_{t+1} \leftarrow \boldsymbol{P}_{t+1} \cup$ [the first $(N - |\boldsymbol{P}_{t+1}|)$ solutions in $\mathcal{F}_m$]
        $m \leftarrow m + 1$
    Calculate the crowding distance for each solution in $\boldsymbol{P}_{t+1}$.
    Create child population $\boldsymbol{Q}_{t+1}$ by using binary tournament selection based on the crowding distance operator $\prec_c$, crossover and mutation.
    $t \leftarrow t + 1$
return $\mathcal{W}^* = \boldsymbol{P}_{G_{max}}$

Next, a *crowding distance density measure* is calculated for each candidate solution in a nondominated front. This measure is used to quantify the solution density in the sense that a higher value indicates that a solution is more isolated, while a lower value implies that a solution is more crowded by other solutions. The crowding distance algorithm is given in pseudocode as Algorithm 5.5.

The crowding distance density measure requires that the solutions in the population be sorted in ascending order of magnitude along each objective axis. Denote the objective function value of the i^{th} candidate solution for the h^{th} objective (in the sorted list) by $X[i]|h$. The crowding distance of the i^{th} solution for the h^{th} objective is then denoted by $i_{dist}|h$. Furthermore, let k denote the number of solutions in the population. An infinite crowding distance is assigned to the boundary solutions, that is the solutions $X[1]|h$ and $X[k]|h$, so as to ensure that they are selected for crossover. The crowding distances of the intermediate solutions i are incremented by the normalised distance between their closest neighbouring solutions. More specifically,

Algorithm 5.5: Crowding Distance Assignment Algorithm [2].

Input: A population $\boldsymbol{P}$ of candidate solutions to an instance of the multi-objective maximisation problem (4.1)–(4.4), and a collection of vectors $\boldsymbol{z}$ containing the objective functions' values for each solution.

Output: The crowding distance $\boldsymbol{P}[i]_{\text{dist}}$ for each solution in $\boldsymbol{P}$

$k = |\boldsymbol{P}|$
forall $i \in \boldsymbol{P}$ **do**
  $\boldsymbol{P}[i]_{\text{dist}} \leftarrow 0$
forall M *objectives* **do**
  $\boldsymbol{P} = \text{sort}(\boldsymbol{P}, h)$
  $\boldsymbol{P}[1]_{\text{dist}}|h \leftarrow \infty$
  $\boldsymbol{P}[k]_{\text{dist}}|h \leftarrow \infty$
  forall $i = 2$ *to* $(k-1)$ **do**
    $\boldsymbol{P}[i]_{\text{dist}}|h \leftarrow \boldsymbol{P}[i]_{\text{dist}}|h + (\boldsymbol{P}[i+1]|h - \boldsymbol{P}[i-1]|h)/(h_{max} - h_{min})$
return $\boldsymbol{P}[i]_{\text{dist}}$ for each solution in $\boldsymbol{P}$

the normalised distance value is calculated as $(X[i+1]|h - X[i-1]|h)/(h_{max} - h_{min})$, where h_{max} and h_{min} represent the maximum and minimum values of the h^{th} objective function, respectively. The overall crowding distance for each objective is taken as the accumulated value of the crowding distances of the individual solutions. An example of the crowding distance measure of a solution in objective space for a bi-objective optimisation problem in which both objective functions f_1 and f_2 have to be minimised is illustrated in Figure 5.6.

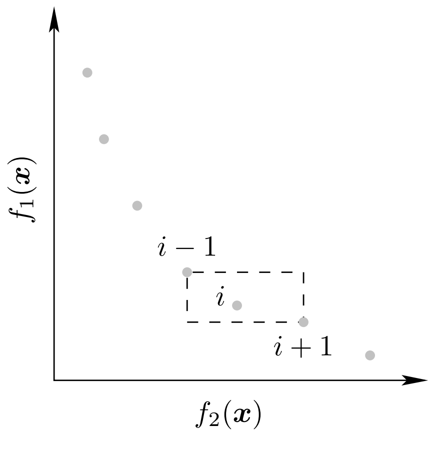


FIGURE 5.6: *Cuboid formed around a solution in the calculation of its crowding distance in objective space for a bi-objective problem in which both objectives are to be minimised.*

The selection procedure in the NSGA II employs two selection criteria. The rank assigned to a solution serves as the first criterion by which to determine superiority between solutions. A solution achieving a lower rank value is considered the superior solution, *i.e.* if rank $\boldsymbol{i} <$ rank $\boldsymbol{j}$, then $\boldsymbol{i} \prec \boldsymbol{j}$ (denoting that solution $\boldsymbol{i}$ is superior to solution $\boldsymbol{j}$). If, however, a pair of solutions achieves the same rank value, then the crowding distance operator, denoted by $\prec_c$, is employed as a second selection criterion. The solution achieving the highest crowding distance is considered the superior solution as a means to break the tie, *i.e.* if rank $\boldsymbol{i}$ = rank $\boldsymbol{j}$ and $\boldsymbol{i}_{dist}|h > \boldsymbol{j}_{dist}|h$, then $\boldsymbol{i} \prec_c \boldsymbol{j}$. In this way, more solutions are explored in less crowded regions of the solution space, which leads to a more uniformly spaced Pareto front approximation.

The population of candidate solutions $\boldsymbol{P}_t$ during iteration t of the NSGA II is sorted and ranked into fronts by applying the FNSA. Crowding distances are computed for each solution in $\boldsymbol{P}_t$ and an intermediate population $\boldsymbol{Q}_t$ of offspring is generated by performing the binary tournament selection procedure described in §5.3.2, as well as the crossover and mutation operators, to the solutions in $\boldsymbol{P}_t$. The parent population $\boldsymbol{P}_t$ and offspring population $\boldsymbol{Q}_t$ are combined to form a larger intermediate population $\boldsymbol{R}_t = \boldsymbol{P}_t \cup \boldsymbol{Q}_t$ of size $2N$. This larger population $\boldsymbol{R}_t$ is then sorted and ranked into nondominated fronts by again applying the FNSA. The next population of candidate solutions $\boldsymbol{P}_{t+1}$ is generated by taking the solutions in the first nondominated front $\mathcal{F}_1$, then the solutions in the second nondominated front $\mathcal{F}_2$, and so forth, until the population size N is reached. If all the solutions in a particular front cannot be included in $\boldsymbol{P}_{t+1}$, the solutions in that front are sorted in descending order with respect to their crowding distance and solutions are added to $\boldsymbol{P}_{t+1}$ in this order until a population size of N is reached. Once the initial population $\boldsymbol{P}_0$ has been generated, the algorithm is iterated in the fashion described above until a stopping criterion is reached, such as the generation counter t reaching a prespecified maximum value. Once the stopping criterion has been reached, the set of approximately Pareto optimal solutions are returned as output by the algorithm. A graphical illustration of the formation of a new population $\boldsymbol{P}_{t+1}$ from the current population $\boldsymbol{P}_t$ in the NSGA II may be found in Figure 5.7.

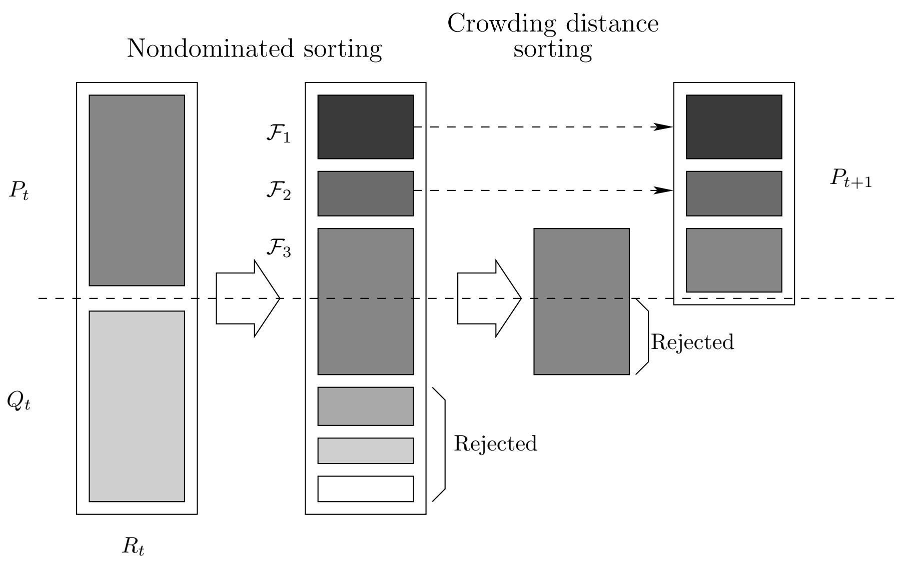


FIGURE 5.7: *High-level schematic of the formation of a new population $\boldsymbol{P}_{t+1}$ from the current population $\boldsymbol{P}_t$ in the NSGA II [39].*

In 2010, Bullinaria and Garcia-Najera [26] developed a multi-objective evolutionary algorithm that is based on the NSGA II, but is specifically aimed at solving a multi-objective optimisation problem which is based on the vehicle routing problem with time windows. The reason for including a discussion on this solution approach here is that it is used to solve one of the WAMs proposed later in this chapter.

The algorithm developed by Bullinaria and Garcia-Najera [26] follows the same steps as those of the NSGA II described earlier. The main difference is the way in which solutions are selected for recombination. The selection procedure proposed in the algorithm employs two selection criteria. The first parent is selected based on its fitness value (*i.e.* the number of the nondominated front in objective space in which the solution resides) while the second parent is chosen based on a diversity preservation measure known as a *similarity measure*. This measure evaluates how similar a solution is to the rest of the solutions in the current population. Solutions with small similarity values are preferred to solutions with large similarity values in a bid to explore a more diverse region of the solution space. The similarity measure adopted in the algorithm is based on Jaccard's similarity coefficient [95], which measures the similarity between two sets X and Y as the ratio

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}. \tag{5.8}$$

If both the sets X and Y share the same elements, then $|X \cap Y| = |X \cup Y|$ and so $J(X,Y) = 1$. If, at the other extreme, the sets share no elements, then $J(X,Y) = |X \cap Y| = 0$. In all other cases, $J(X,Y) \in (0,1)$. The similarities of the solutions in the population are similarly calculated as the ratio of the number of assignments shared by both solutions to the total number of assignments present in both solutions combined. Hence, if both solutions are exactly the same, they will achieve a similarity value of 1, whereas if they are distinct in all their assignments, they will achieve a similarity value of 0. The similarity value of a single solution with respect to the rest of the solutions in the population is taken as the average of the similarities achieved by the solution with respect to all the other solutions in the population, considered one at a time [26].

5.5 Chapter summary

This chapter was dedicated to a discussion on solution methodologies for solving WAMs. The solution approaches were discussed in three groups. The chapter opened with descriptions of the working of two exact optimisation approaches in §5.1, *i.e.* the (explicit) method of total enumeration and the celebrated (implicit) branch-and-bound method due to Doig and Land [57]. Although exact solution approaches are able to find globally optimal solutions to optimisation problems, they are typically computationally expensive and are often abandoned in favour of computationally less expensive solution approaches which are able to find good solutions that are not necessarily optimal. Such solution approaches fall within the realms of heuristics and metaheuristics.

In §5.2, the focus shifted to a brief review on heuristics. Iterative and constructive heuristics were outlined briefly and various greedy algorithms for solving the WAP were also mentioned.

In §5.3, two well-known metaheuristics for solving single-objective optimisation problems were discussed in general. The first was the trajectory-based method of simulated annealing due to Kirkpatrick *et al.* [104] and the second was a population-based solution approach, namely a genetic algorithm due to Holland [88]. Pseudocode descriptions of the working of all these algorithms were also provided.

Since some of the WAMs proposed later in this dissertation are multi-objective optimisation problems, §5.4 was finally devoted to descriptions of multi-objective optimisation incarnations of the aforementioned two metaheuristics. These algorithmic extensions were the DBMOSA proposed by Smith *et al.* [173] and the NSGA II developed by Agarwal *et al.* [2]. Pseudocode descriptions of the working of these algorithmic extensions were also provided.

Part III

Proposed decision support system

CHAPTER 6

A generic weapon assignment subsystem architecture

Contents

In this chapter, a generic WA design architecture is put forward for inclusion as a part of an integrated GBAD TEWA DSS. This architecture is considered the principal contribution of the research presented in this dissertation.

The chapter opens in §6.1 with a brief discussion on three general methods that may be followed when developing a DSS. In §6.2, the focus shifts to a brief discussion on the design of a generic WA subsystem architecture in particular. The proposed architecture contains two smaller subsystems, *i.e.* an EQ subsystem and a WA subsystem. Each of these subsystems comprises two components which are discussed in detail.

The first component of the EQ subsystem is called the *Physical Element Filter* (PEF) component and is described in detail in §6.3. A method for discretising SSHP information and filtering this information for environmental conditions and terrain obstructions is discussed and illustrated by means of worked examples. The second component of the EQ subsystem is called the EEM component and is described in §6.4. A straight flight path prediction model and a probabilistic area prediction model are described for predicting the future flight paths of threats. An example is also provided of how the probabilistic area prediction model may be used to predict the future SSHP values that WSs can achieve with respect to threats. A brief discussion follows on how the EEM may be constructed.

The WA subsystem architecture proposed in this chapter requires input from the TE subsystem as well as information preprocessed in the EQ subsystem. The scope of this dissertation is, however, limited to the design of a WA subsystem architecture and the assumption is therefore made that the threats have been identified, classified and evaluated, and that threat values have been assigned to them accordingly — this value also takes into account the importance of the DAs. In 2008, Potgieter [145] laid the ground work for methods of discretising SSHP information and filtering this information for environmental conditions and terrain obstructions. He also touched upon methods for predicting the future flight paths of threats. Therefore, the methods and worked examples presented in the early part of this chapter on discretising and filtering SSHP information follows the general exposition of Potgieter and include descriptions of the examples used by him, rather than constructing new examples illustrating the working of these methods.

The principal component of the WA subsystem is called the WAM component and is described in detail in §6.5. Four classes of WAMs ranging in different levels of complexity are proposed for inclusion in the WAM component. These classes are single-objective static WAMs, multi-objective static WAMs, single-objective dynamic WAMs and multi-objective dynamic WAMs. A prototype from each of these classes is selected and discussed in detail. The four WAM prototypes are presented in increasing order of complexity. To the best knowledge of the author, there are no WAMs in the most complex class of WAMs (*i.e.* the class of multi-objective dynamic WAMs). A new tri-objective dynamic WAM is therefore formulated for use in the final class of the WAM component proposed here — this is also considered one of the novel contributions of this dissertation.

Another WA subsystem component is the WASS component and this component is briefly discussed in §6.6. It is proposed that a number of solution methodologies be employed concurrently to solve the WAM configured by the FCO within the WAM component and that the solutions obtained from all these methodologies be combined and sorted based on the notion of solution dominance as described in §4.3 so as to obtain a high-quality set of nondominated solutions for presentation to the FCO as real time DS.

The chronological order of events envisaged to occur within a single TEWA cycle is next discussed and elucidated in detail in §6.7. The chapter finally closes in §6.8 with a brief summary of the chapter contents.

6.1 Decision support system design approaches

Doucet [60] and Kimble [103] described the function of a DSS as a tool that is able to support a decision maker (or a group of decision makers) within an organisation by providing them with information that would otherwise be difficult to obtain without the help of the system. The aim in developing an effective DSS is that it should be well designed and constructed by taking the skill level of the end-user into consideration, since the success of the system is determined by the way in which the end-user is able to use the system comfortably and understand the results provided by the system [103]. Doucet [60] defined three essential elements of a DSS.

These elements are a *database*, a *graphical user interface* and a *model base*. The database of a DSS allows for the structured storage of all information relevant to the decision making process of the organisation. It forms a central part of the system and typically provides the link between all the components contained in the DSS. A well-designed database may lead to a DSS that is able to perform well [127]. A number of database models exist in the open literature which affect the way in which data are stored in a DSS. Obbayi [138] mentions five main types of database models, namely *flat file-based* (such databases contain binary or *human-readable* text formats), *relational-based* (such databases normalise the data and store the results thereof in tables), *hierarchical* (such databases are tree-like in nature and operate in a parent-child way — the relationship is typically one-to-many), *network* (such databases are similar to hierarchical databases, but the relationships are typically of the kind many-to-one) and *object-orientated* (such databases are typically constructed with the intention of being linked with object-orientated programming languages and the data are typically partitioned into objects from where it is accessed). French [70] explained that although there are a number of database types that can be used in the development of DSS, it should be noted that each database structure has its own strengths and limitations and that an appropriate database should be selected and tailored (should the need occur) to fit the needs of the organisation.

The graphical user interface forms an integral part of a DSS since it may be seen as the link between the human operating the DSS and the computerised DSS. It also allows for a human to interact with the DSS (typically via a computer display screen) in terms of providing various inputs and also viewing results returned by the DSS. Kendall and Kendall [101], as well as Stone *et al.* [179], accentuated the fact that a graphical user interface should be *usable* in the sense that the human interacting with the system should be able to engage effortlessly with the system and comfortably perform his duties. Benyon [16] further advocated that the design of a well-structured graphical user interface should aim to find a good balance between not overwhelming human operators with information and providing them with an unclear graphical user face which may cause confusion on the part of the operator. It is therefore important to understand the needs of the end-user as well as his capabilities when designing a graphical user interface — the end-user should continually be consulted during the development of a DSS. Mandel [124] provided three guidelines to be followed during the development stages of a graphical user interface: (1) give the user adequate authority over the graphical user interface, (2) decrease the user's cognitive load and (3) create a consistent graphical user interface so as to avoid additional confusion.

The model base element constitutes the component of the DSS in which models reside that are used to solve problems in order to obtain various alternatives during the decision making process — it forms the heart of the system since it is here where the actual DSS results (presented to the human operator) are computed. Models employed in the model component may take many forms such as, for example, a mathematical algorithm or a combination of various mathematical algorithms. When a combination of such mathematical algorithms is used, it should further be decided whether the models should be implemented in a competing manner (in which case the results from the model achieving the best results may be returned as output by the DSS) or whether the models should be implemented in a complementary manner (in which case the results from the models may be combined when formulating DS). When choosing to combine the output from all the models of the system, the problem of how these results should be combined arises.

In order to develop a well-structured DSS, a *systems development methodology* may be employed. Walters *et al.* [205] defined a systems development methodology as "an array of operations, tools, techniques and documentation methods that may assist system analysts in the creation

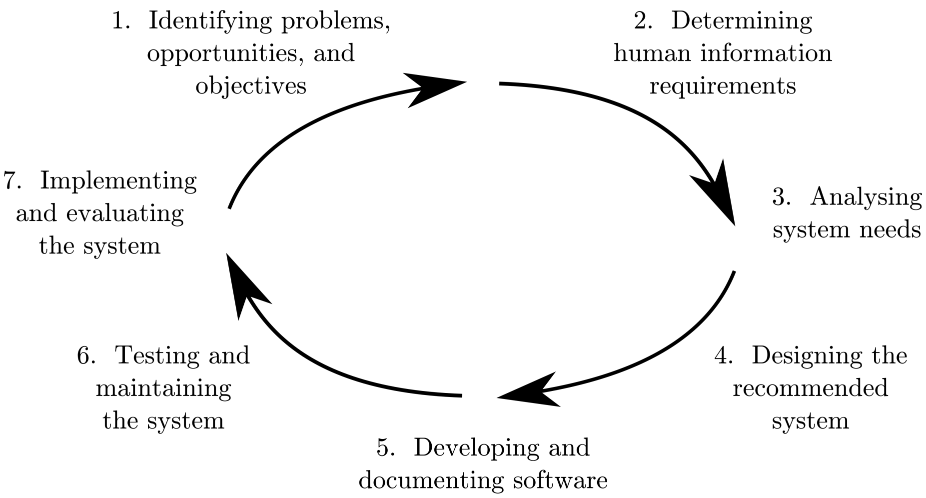


Figure 6.1: *Seven phases in the systems development life cycle of a system development methodology proposed by Kendall and Kendall [101].*

of DSS." Such systems development methodologies share a number of core components with the development process of a DSS. These components are known as the *systems development life cycle* and Kendall and Kendall [101] described seven of these phases, which are illustrated graphically in Figure 6.1. These seven phases merely serve as a foundation from which a DSS may be developed and tailor-made phases may be developed for different types of DS. Three popular systems development methodologies are the *waterfall* method, the *agile* method and the *object-orientated* method:

Waterfall method. This method may be described as a *top-down* (and precise) approach in the sense that the phases in the method are completed sequentially as the method is documented [55]. Furthermore, the method is structured in such a way that one cannot return to a previous phase as one progresses through the phases. One is only allowed to proceed forward — similar to water moving over a series of cliffs. As progression is made along the various phases, the approval of the organisation (or stakeholders) are sought first at each stage, before the next stage may commence — hence the name *waterfall*. Advantages of adopting the waterfall method include that it is simpler to manage the overall development process of a DSS since a number of milestones are typically set for each phase and are measured against regular development updates. Furthermore, proper documentation of the phases and system requirements are known in advance [55]. One of the major disadvantages of the waterfall method is that there is not much time to revise design work which makes the task of designing an entire system theoretically on paper very challenging. Also, the waterfall method typically involves a long development process which may lead to a DSS that successfully meets user requirements, but may not be of much use by the time that it is built due to possible changes in the environment of the organisation.

Agile method. This method was originally developed to speed up system development in order to counter the disadvantages associated with more traditional systems development methodologies. In addition, the agile method is more aimed at identifying and appropriately following user requirements, as well as developing a DSS that is practical to implement, rather than creating a DSS in theory. Advantages of the agile method include that a faster delivery of a DSS is possible and that developers of the DSS are able to amend and improve the DSS at any stage during the development process. Disadvantages associated with the agile method typically arise from the incorrect implementation thereof or from human error, as opposed to flaws in the methodology itself.

Object-orientated method. This method may be described as a bottom-up approach (as opposed to the top-down waterfall approach). It partitions the DSS into objects containing data, such as locations, events, people or actual components of the DSS. These objects are then grouped together into classes containing similar objects and share related characteristics.

The WA process architecture proposed in this chapter is put forward in order to provide a first-order "blue print" design which may be consulted for practical implementation purposes at a later stage. The architecture was designed as a theoretical concept in close conjunction with two military experts [146, 160] and, more importantly, two FCOs [160, 200] (who are the end-users in the context of GBAD) — the military experts and FCOs were consulted on a continual basis throughout the development process. They also approved various phases during the development of the generic WA DSS architecture put forward in this chapter. The working of the components of the architecture and their characteristics are outlined and carefully documented in this chapter and the following two chapters in accordance with the waterfall design approach described above.

Although a number of working examples are provided in order to illustrate the working of the concepts in the proposed theoretical design, the DS which emanates from these concepts may not be sufficient to use as is in a practical setting at some later stage if the generic WA architecture were to be considered for implementation. Developers of DSSs would have to be consulted in order to facilitate an effective practical implementation of the design and such implementation would require that the developers of the system be able to make changes and suggest improvements to the DSS design during any stage of the practical implementation process. In addition, the software developers working on a computerised incarnation of the theoretical concept put forward in this chapter would have to work in close conjunction with military experts and FCOs who would again have to approve all the various phases during the practical implementation of the system. The characteristics associated with such an implementation procedure conform to the characteristics associated with the waterfall and agile design methods. It is therefore advocated that a combination of these two methods be used in any eventual practical implementation of the WA process architecture design proposed in this chapter.

6.2 A generic WA subsystem architecture

A generic WA process architecture is proposed in this section for use within a larger, integrated TEWA DSS. It is proposed that the processes involved in WA be subdivided into activities performed by two smaller subsystems which should be activated consecutively within each cycle of the TEWA DSS working. The first subsystem is the *EQ subsystem*. Recall, from §2.5.1, that each GBAD WS is associated with a measure of effectiveness (an SSHP value) according to which it can achieve success in respect of the engagement of threats. The aim of the EQ subsystem is to utilise the SSHP information of WSs (in conjunction with other information) so as to obtain a single SSHP value for each WS-threat pair for each time stage in the decision period time continuum under consideration. It is further proposed that the EQ subsystem comprise two components (which should also be activated consecutively within each cycle of the TEWA DSS working), known as the PEF component and the EEM component.

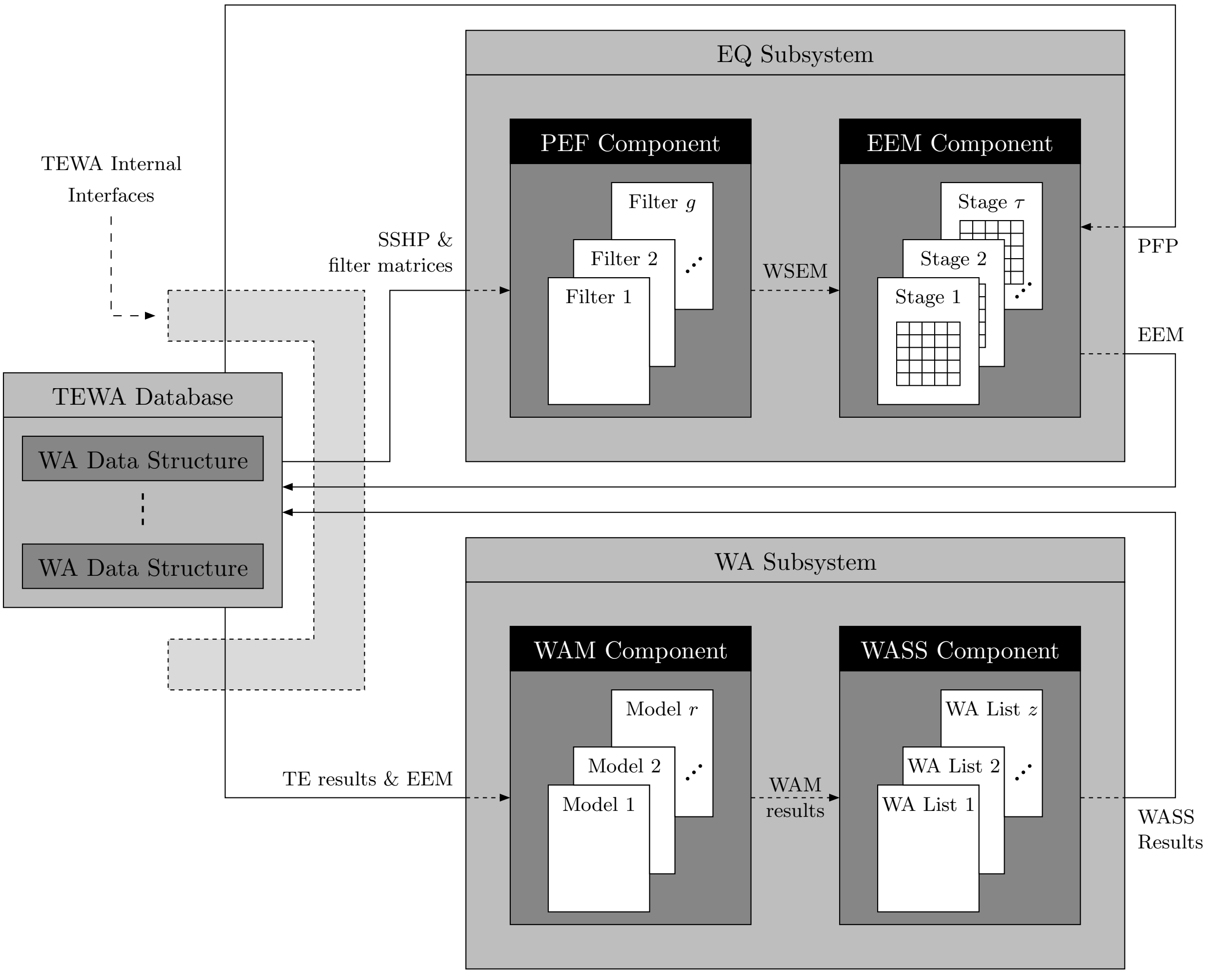


FIGURE 6.2: *The EQ and WA subsystems which form part of the larger WA part of an integrated TEWA DSS.*

The second subsystem of the proposed architecture is the *WA subsystem*, which forms an essential part of the real-time DS provided to the FCO, since it is here where proposed assignments of WSs to threats are determined. It is further proposed that the WA subsystem also contains two components, known as the WAM component and the *Weapon Assignment Solution Selection* (WASS) component.

The EQ and WA subsystems, as well as their components, are illustrated graphically in Figure 6.2. Each of the four aforementioned components, as well as the interaction between them, is discussed in detail in the remaining sections of this chapter.

6.3 The physical element filter component

The first component activated within each working cycle of the proposed WA architecture of Figure 6.2 is the PEF component. The main purpose of the PEF component is to fuse a number of data sets into a single data set in the format required by the EEM component and other components of the WA subsystem. This is achieved in two phases within the PEF component. The first phase involves the discretisation of the estimated effectiveness values achieved by the

WSs when they are assigned to engage aerial threats, and the second phase involves predicting the future flight paths of threats by means of a mathematical prediction model. These two phases are discussed in some detail in this section.

6.3.1 The method of WS effectiveness information discretisation

The SSHP values achievable by WSs with respect to aerial threats are functions of the three-dimensional volume surrounding the WS. In order for the WA subsystem to utilise these SSHP values, a method of discretisation of the SSHP information is required. Potgieter [145] proposed that a three-dimensional grid be superimposed over the SSHP volume of WSs in order to discretise the information. By superimposing such a grid over the SSHP volume, each cell in the grid is assigned a single SSHP value which corresponds to the SSHP partition in which the cell predominantly lies. Care should, however, be taken in selecting the coarseness of this grid which determines the eventual size of the EQ matrix. Selecting too coarse a grid may result in an unacceptable loss of valuable SSHP information whereas selecting too fine a grid may result in an unacceptable increase in the complexity of subsequent computations which have to be carried out during actual WA. The output of the discretisation procedure is a three-dimensional SSHP matrix containing an entry in each cell which represents the anticipated hit probability (or the effectiveness) value of a particular WS with respect to eliminating a threat finding itself in that cell. Such an SSHP matrix has to be constructed for each WS.

The orientations of GBAD WSs are not necessarily fixed. WS ammunition may be fired in different directions within a WS's fire arc, which means that the shape of the SSHP volume surrounding a WSs is not fixed, but rather differs for each one of the possible orientations of the WS. This implies that the SSHP matrix is a function of WS orientation. It is therefore advocated that the SSHP volume of a WS (as specified for a fixed orientation by its manufacturer) be discretised for each of a finite set of representative orientations that a WS may adopt within its fire arc. In this way an SSHP matrix is obtained for each of the permissible orientations of the WS within its fire arc. Furthermore, it is assumed that each of these different orientations are mutually exclusive — a WS may therefore adopt only one orientation at a given time stage τ.

Let $\mathbb{W}$ denote a set of n_w GBAD WS types available to the own force, let $\mathbb{V}$ denote a set of n_v aircraft available to the opposing force, let $\mathbb{E}$ be a set of n_e weapons that may be carried by the aircraft in $\mathbb{V}$, and let $\mathbb{A}$ be a set of n_a attack techniques that can be flown by the aircraft in $\mathbb{V}$. Furthermore, let $\boldsymbol{M}_\tau(w, v, e, a)$ denote the resulting SSHP matrix for time stage τ corresponding to a WS of type $w \in \mathbb{W}$ for the following formative threat combination: an aircraft of type $v \in \mathbb{V}$ carrying weapons of type $e \in \mathbb{E}$ and executing an attack technique $a \in \mathbb{A}$. In order to illustrate the process involved in the discretisation of the SSHP volume in order to populate the matrix $\boldsymbol{M}_\tau(w, v, e, a)$, consider the following two-dimensional example taken (in a slightly adapted form) from Potgieter [145].

Example 6.1 (Discretising SSHPs) *Suppose that the opposing force has only two types of aircraft at its disposal, that these aircraft may be equipped with one of three possible weapon types, and that these types of weapons may be delivered by executing one of three possible attack techniques according to the doctrine of the opposing force. Suppose, furthermore, that only one GBAD WS type is available to the own force for engaging enemy aircraft. Let these GBAD elements be captured as follows using the set notation introduced above:*

$\mathbb{W} = \{1[\textit{Man-portable SAM}]\}$, *with* $n_w = 1$;

$\mathbb{V} = \{1[\textit{Hawk}], 2[\textit{Grippen}]\}$, *with* $n_v = 2$;

$\mathbb{E} = \{1[\textit{Guns and Cannons}], 2[\textit{Rockets}], 3[\textit{Free falling bombs}]\}$, *with* $n_e = 3$; *and*

$\mathbb{A} = \{1[\textit{Combat hump dive}], 2[\textit{Toss bomb}], 3[\textit{Combat turn dive}]\}$, *with* $n_a = 3$.

Assume that an aircraft of the opposing force is of type 2 [Grippen], is carrying type 2 [Rockets] weapons and is executing a type 1 [Combat hump dive] attack technique. Suppose that an SSHP matrix is sought for time stage τ_0 *in respect of a WS of type 1 [Man-portable SAM]. The SSHP matrix* $\boldsymbol{M}_{\tau_0}(1, 2, 2, 1)$ *is therefore sought.*

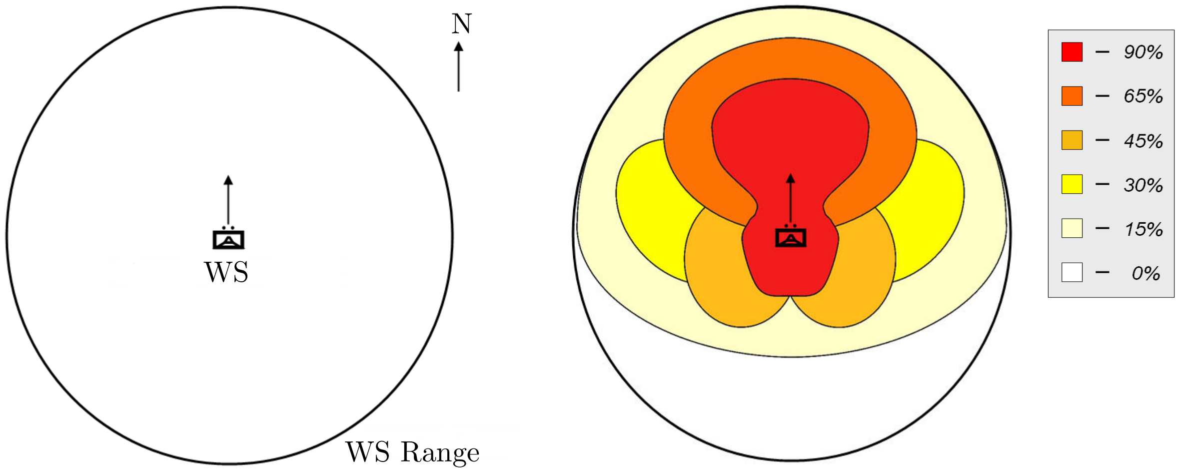
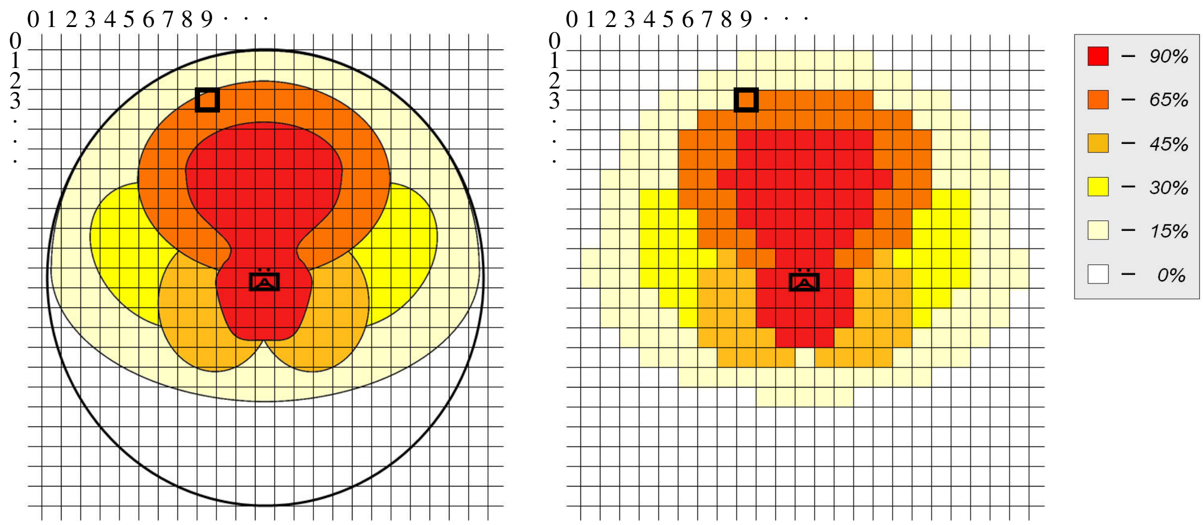


FIGURE 6.3: *(a) Top view of a WS with a fixed range assuming a northerly orientation and (b) SSHP values at a fixed height in the surrounding area of the WS, as supplied by the WS manufacturer [145].*

Next, suppose the WS of type 1 [Man-portable SAM] has a fixed range, assumes a northerly orientation and shoots projectiles at an angle. Suppose further that the WS achieves a maximum SSHP value of 90% in areas of space above it which lie within close proximity of the WS and that the SSHP values generally diminish as the distance from the SSHP increases. The SSHP values may, for example, be distributed in two dimensions as shown in Figure 6.3 for a fixed aircraft height.

Suppose a coarse, two-dimensional grid is superimposed over the SSHP information of Figure 6.3 as illustrated in Figure 6.4(a). Each cell in the grid is assigned the colour of the cell in which it predominantly lies. Consider, for example, row 3 and column 9 in the discretisation in Figure 6.4. The colour which predominantly covers this cell corresponds to a 65% SSHP in Figure 6.3 — hence this cell is assigned a SSHP value of 65%. This procedure may be repeated for each cell in the grid in order to obtain a SSHP value for each cell, resulting in the two-dimensional SSHP matrix for the specific height in question above the WS shown in Figure 6.5. This procedure has to be repeated for a range of possible aircraft heights in order to obtain a full three-dimensional SSHP matrix. It is, however, assumed here that this matrix is invariant as a function of aircraft height (i.e. that each horizontal slice of the matrix is the same). Although this assumption is not realistic, it is made for the purposes of simplicity and brevity in all the numerical examples presented as mere concept demonstrations throughout this chapter. ■



(a) (b)

FIGURE 6.4: *(a) Top view of a coarse two-dimensional grid placed over the SSHP information of Figure 6.3 and (b) the corresponding cells relating to the partition in which it predominantly lies [145].*

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&15&15&15&15&15&15&15&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&15&15&15&15&15&15&15&15&15&15&15&0&0&0&0&0&0\\
0&0&0&0&15&15&15&15&\boxed{65}&65&65&65&65&65&65&15&15&15&15&0&0&0&0\\
0&0&0&15&15&15&65&65&65&65&65&65&65&65&65&65&65&15&15&15&0&0&0\\
0&0&15&15&15&65&65&65&90&90&90&90&90&90&90&65&65&65&15&15&15&0&0\\
0&0&15&15&15&65&65&65&90&90&90&90&90&90&90&65&65&65&15&15&15&0&0\\
0&15&15&15&15&65&65&90&90&90&90&90&90&90&90&90&65&65&15&15&15&15&0\\
0&15&15&15&15&65&65&65&90&90&90&90&90&90&90&65&65&65&15&15&15&15&0\\
0&15&15&30&30&30&65&65&90&90&90&90&90&90&90&65&65&30&30&30&15&15&0\\
0&15&15&30&30&30&65&65&65&90&90&90&90&90&65&65&65&30&30&30&15&15&0\\
15&15&15&30&30&30&30&45&65&65&90&90&90&65&65&45&30&30&30&30&15&15&15\\
15&15&15&30&30&30&45&45&45&90&90&90&90&90&45&45&45&30&30&30&15&15&15\\
0&15&15&15&30&30&45&45&45&90&90&90&90&90&45&45&45&30&30&15&15&15&0\\
0&15&15&15&15&30&45&45&45&90&90&90&90&90&45&45&45&30&15&15&15&15&0\\
0&0&15&15&15&15&45&45&45&45&90&90&90&45&45&45&45&15&15&15&15&0&0\\
0&0&0&15&15&15&15&45&45&45&45&15&45&45&45&45&15&15&15&15&0&0&0\\
0&0&0&0&0&15&15&15&15&15&15&15&15&15&15&15&15&15&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&15&15&15&15&15&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

FIGURE 6.5: *The SSHP matrix* $\boldsymbol{M}_{\tau_0}(1,2,2,1)$ *obtained from Figure 6.4(b) in Example 6.1 [145].*

6.3.2 Filtering of discretised single shot hit probabilities

The SSHP values of WSs, estimated by their manufacturers, typically do not take into account the effects that external elements in the tactical environment have on the effectiveness of WSs in real time, such as meteorological conditions (*e.g.* precipitation, wind strength and direction, or cloud cover) or terrain obstacles (*i.e.* line-of-sight restrictions). In order to make provision for constraints or masking effects imposed by these elements on the battlefield, the SSHP values supplied by WS manufactures should be filtered appropriately, discounting these values for such elements.

Furthermore, the aforementioned discounting process should take into account the effects of different levels of intensities of the external element (*e.g.* wind may be divided into levels of no wind, light breeze, moderate breeze, light gale, moderate gale and strong gale, which may have the following respective effects on the WS: none, little effect, a notable effect, a large effect, a significant effect, a substantial effect). Let $\mathbb{U}$ denote a set of n_u environmental conditions which may affect the effectiveness of WSs and let $\mathbb{L}_u$ denote a set of n_ℓ intensity levels associated with environmental condition u. Again, it is assumed here that the various levels involved in a particular environmental condition are mutually exclusive events in order to ensure that only one level may be active at a given time stage τ. Furthermore, let $\tilde{\boldsymbol{M}}_\tau^{u_\ell}(w, v, e, a)$ denote the efficiency matrix containing the discounted SSHP value entry at time stage τ for environmental condition $u \in \mathbb{U}$ at intensity level $(u_\ell) \in \mathbb{L}_u$ for a WS of type $w \in \mathbb{W}$ and for the following aircraft (threat) combination: an aircraft of type $v \in \mathbb{V}$, carrying weapons of type $e \in \mathbb{E}$ and executing an attack technique $a \in \mathbb{A}$. Also, let $\tilde{\boldsymbol{M}}_\tau^{u_\ell}(w, v, e, a)_{x,y,z}$ denote the entry of the efficiency matrix $\tilde{\boldsymbol{M}}_\tau^{u_\ell}(w, v, e, a)$ at position (x, y, z)

As in the case of the original SSHP values of WSs, information with respect to the effects that different environmental conditions may have on the SSHP values of WSs should be quantified during the pre-deployment stages of a mission. A simulation model may, for example, be used to run a number of simulated replications for *each* discretised cell in the surrounding area of the WSs in order to determine the effects that a given environmental condition may have on the effectiveness of the WSs. The reason for having to run a simulation experiment for each cell in the surrounding area of the WS is that the SSHP values that WS are able to achieve (see Figure 6.3) are affected differently by environmental conditions in different regions surrounding the WS. An estimation of the SSHP values of the WSs may then be made for different levels of intensities of the environmental condition in question by fitting a regression curve through the simulation results — the method of least squares may, for example, be used for this purpose. A regression would therefore have to be executed for each specific cell in the discretised SSHP volume of Figure 6.3 surrounding the WS. The expected WS SSHP values may be obtained by mapping the mid-point of the range of the intensity level to the regression curve for the environmental condition under consideration at the specific intensity level. The discounted SSHP value $\tilde{\boldsymbol{M}}_\tau(w, v, e, a)^{u_\ell}_{x,y,z}$ may then be obtained by reading out the value of the regression curve or by calculating it using the regression function obtained. The aforementioned process of discounting SSHP values for environmental conditions is illustrated by means of a numerical example.

Example 6.2 (Filtering SSHP information for weather conditions) *This example follows on Example 6.1; the SSHP matrix of Figure 6.5 is assumed as a point of departure. Suppose that WSs in the battlefield may be subject to one of two environmental conditions in the set:*

$\mathbb{U} = \{1[\mathit{Rain}],\ 2[\mathit{Wind}]\}$, *with* $n_u = 2$.

Suppose further that the same configuration of WS type and aircraft type as considered in Example 6.1 is again applicable and that a simulation model has been executed subject to environmental condition $1[\mathit{Rain}]$. *Assume that the output of the simulation model yielded the results presented in Table 6.1 for row 3 and column 9 of the SSHP matrix (involving an entry of 65%) in Figure 6.4. When these results are plotted, the corresponding graph in Figure 6.6 is obtained and it is clear from the figure that the relation between SSHP and rain may be modelled as an approxi-*

mately linear relationship. The regression function $Y = a + bX + \epsilon$ *may therefore be fitted to the data points of the simulation, where* Y *denotes the effectiveness of the WS when* X *amount of rain is falling. The residual* ϵ *is a random variable with mean zero, while* a *and* b *are parameters to be estimated. Let* y_i *denote the* i^{th} *measured SSHP value when an amount* x_i *of rain falls. The parameters* a *and* b *of the regression function may be estimated by minimising the sum of the squared errors*

$$R^2 = \sum_i [y_i - (a + bx_i)]^2. \tag{6.1}$$

In order to achieve a minimum value for R^2*, the first partial derivative of* R^2 *with respect to both the parameters* a *and* b *must be zero, i.e.*

$$\frac{\partial R^2}{\partial a} = -2\sum_{i=1}^{n}[y_i - (a + bx_i)] = 0 \tag{6.2}$$

and

$$\frac{\partial R^2}{\partial b} = -2\sum_{i=1}^{n}[y_i - (a + bx_i)]x_i = 0. \tag{6.3}$$

When simplifying (6.2) and (6.3), the equations

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \tag{6.4}$$

and

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \tag{6.5}$$

are obtained. By solving (6.4) and (6.5) simultaneously for the data in Table 6.1, it is found that $a = 0.6372$ *and* $b = -0.0609$*. Since the value of the parameter* a *lies very close to the original SSHP value of 65%, it may be taken as 65% instead of 0.6372 — recall the assumption that the SSHP is the WS effectiveness without the effect of any external elements.*

With $a = 0.65$*, it follows instead that* $b = -0.0640$*. The resulting regression function* $y = 0.65 - 0.0640x$ *may be used to calculate the expected WS effectiveness for each level of rainfall. Suppose that the environmental condition* rain *may be divided into the levels* none, light, moderate *and* heavy*, and that each of these levels is associated with the following range mid-points: 0mm/h, 1mm/h, 3mm/h and 5mm/h, respectively. The expected SSHP values for each of these levels may then be taken from the regression graph in Figure 6.6, calculated as illustrated in Table 6.2.*

In the examples presented thus far, the effects of physical terrain obstacles on WS efficiency were ignored. It is, however, possible that a deployment of WSs is required in areas which

Rain (mm/h)	2.32	3.48	0.58	2.90	1.74	1.16	4.64	5.80	0.00	4.06	5.22
Efficiency (%)	48.21	41.67	57.44	52.83	49.95	59.72	43.41	25.54	65.00	28.90	34.06

TABLE 6.1: *Simulated WS efficiencies at different intensities of rainfall.*

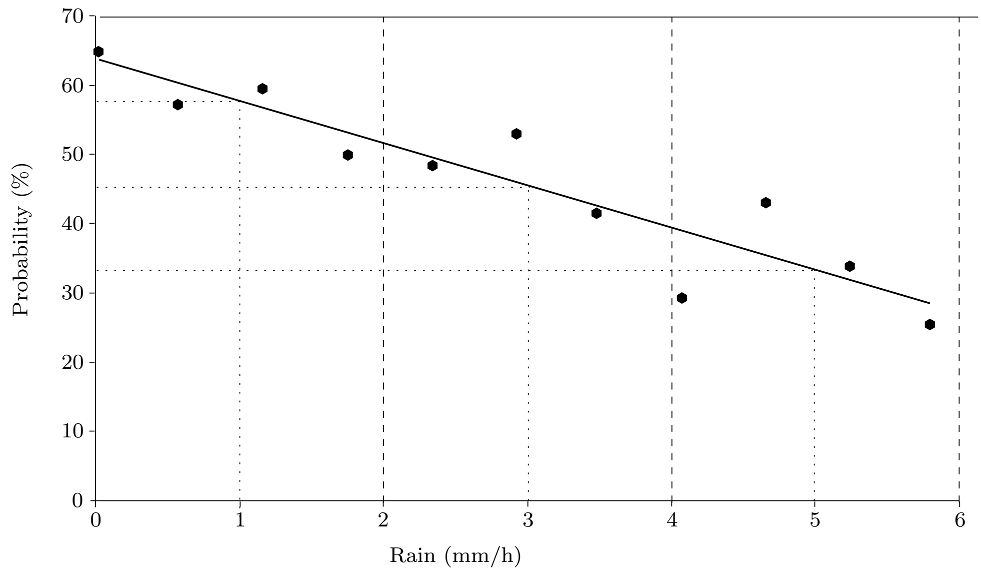


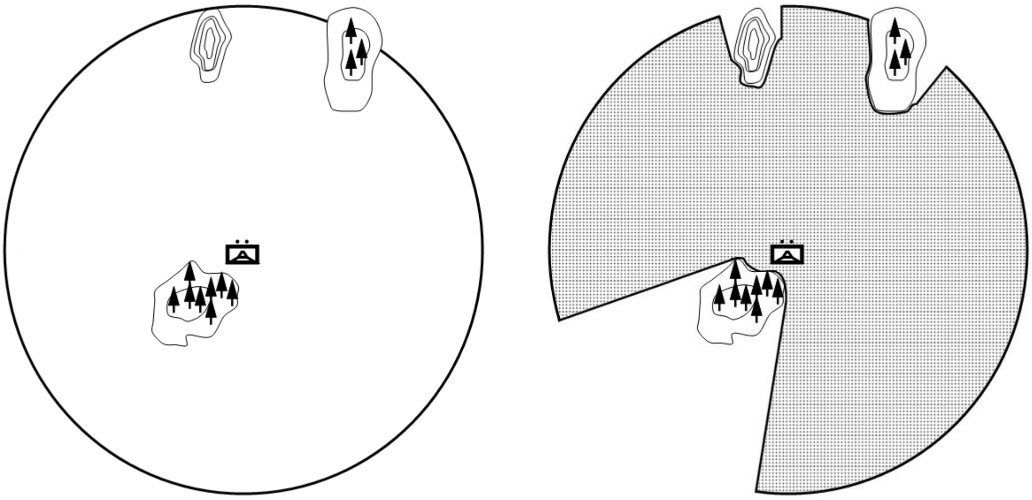
FIGURE 6.6: *Graphical representation of the data in Table 6.1 and a regression line fitted to the data [145].*

$$
\begin{aligned}
\tilde{\boldsymbol{M}}_\tau^{1,0}(1,2,2,1)_{3,9} &= 0.65 - (0.0640 \times 0) = 0.65\\
\tilde{\boldsymbol{M}}_\tau^{1,1}(1,2,2,1)_{3,9} &= 0.65 - (0.0640 \times 1) = 0.5860,\\
\tilde{\boldsymbol{M}}_\tau^{1,2}(1,2,2,1)_{3,9} &= 0.65 - (0.0640 \times 3) = 0.4579,\\
\tilde{\boldsymbol{M}}_\tau^{1,3}(1,2,2,1)_{3,9} &= 0.65 - (0.0640 \times 5) = 0.3299,
\end{aligned}
$$

TABLE 6.2: *The expected effectiveness values of the WS in Example 6.2 for row 3 and column 9 of the discounted SSHP matrix $\tilde{\boldsymbol{M}}_\tau^{u_\ell}(w,v,e,a)$ for the different levels of rain: none, light, moderate and heavy.*

pose restrictions on the effectiveness that WSs are able to achieve due to uneven terrain. These restrictions typically result in occlusion of certain regions in the tactical environment and WS efficiencies have to be adjusted for these restrictions accordingly. Examples of such restrictions include mountain ranges and large hills. A terrain filter matrix having the same dimensions as the efficiency matrix $\tilde{\boldsymbol{M}}_\tau^{u_\ell}(w,v,e,a)$ may be constructed for this purpose which discounts the SSHP values for terrain restrictions. Let $\overline{\boldsymbol{M}}_\tau(w,v,e,a)$ denote a binary terrain filter matrix at time stage τ for a WS of type $w \in \mathbb{W}$ and for the following aircraft (threat) combination: an aircraft of type $v \in \mathbb{V}$, carrying weapons of type $e \in \mathbb{E}$ and executing an attack technique $a \in \mathbb{A}$. The efficiency matrix $\tilde{\boldsymbol{M}}_\tau^{u_\ell}(w,v,e,a)$ may the be multiplied in an element-by-element fashion by the terrain filter matrix $\overline{\boldsymbol{M}}_\tau(w,v,e,a)$ to obtain a so-called *Weapon System Effectiveness Matrix* (WSEM) used as input to the WA subsystem. Let $\hat{\boldsymbol{M}}_\tau(w,v,e,a)$ denote the resulting WSEM containing the SSHP values discounted for environmental conditions and terrain restrictions at time stage τ for a WS of type $w \in \mathbb{W}$ and for the following aircraft (threat) combination: an aircraft of type $v \in \mathbb{V}$, carrying weapons of type $e \in \mathbb{E}$ and executing an attack technique $a \in \mathbb{A}$. The filtering process for terrain features is illustrated by means of a numerical example.

Example 6.3 (Filtering SSHP information for terrain obstruction) *Suppose that a WS is deployed in an area with hills to the north east and south west and a small mountain to the north. Suppose furthermore that these obstacles completely restrict WS operation in certain areas of a WS fire arc. The deployment of the WS and the aforementioned terrain obstacles are illustrated graphically in Figure 6.7(a) and the areas in which the WS have restricted efficiencies* *are illustrated in Figure 6.7(b). The shaded region represents the area in which the WS can achieve positive SSHP values. The efficiency values of the WS now have to be adjusted for the restriction posed by the obstacles in Figure 6.7.*



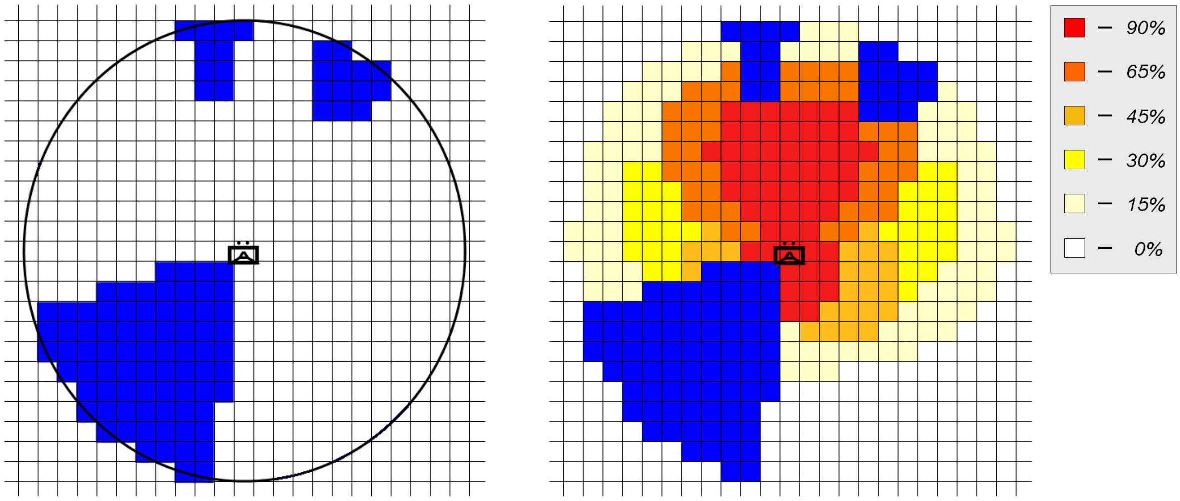
(a) (b)

FIGURE 6.7: *(a) A two-dimensional top view of a WS deployed in an area where terrain obstacles pose an obstruction within the fire arc of the WS in some regions and (b) the region in which the WS can achieve positive SSHP values [145].*

The partitioned SSHP information in Figure 6.3 of Example 6.1 have to be adjusted to take into account the terrain obstacles in Figure 6.7. The terrain features may be discretised in a similar fashion as the SSHP volume of Example 6.1. A grid (with the same coarseness as that employed in the SSHP example) may be placed over the terrain features. Cells in the grid which correspond to areas of the terrain where WS operation is restricted partially or fully may be coloured blue, while cells which correspond to areas where WS operation is not influenced at all may be coloured white. This procedure is illustrated graphically in Figure 6.8.

The colours in the grid placed over the terrain may now be replaced by binary values in order to populate a terrain filter matrix. Blue cells are replaced by the value zero while white cells are replaced by the value one. Assume that the terrain filter matrix for time stage τ_0 *is calculated for the same configuration of WS type and aircraft type as those considered in Example 6.1. By using the same grid coarseness as in Figure 6.4, the resulting terrain filter matrix* $\overline{\boldsymbol{M}}_{\tau_0}(1,2,2,1)$ *in Figure 6.9 is obtained.*

This terrain filter matrix may now be used to calculate the WSEM for time stage τ_0*. The assumption is made that terrain is the only external element which has an effect on WS efficiency in this case — no environmental conditions are included. The terrain filter matrix in Figure 6.9 is therefore applied to the SSHP matrix in Figure 6.5 (obtained in Example 6.1) by multiplying each cell in the terrain filter matrix by the corresponding cell in the SSHP matrix — the values in the SSHP matrix which lie in areas of restricted WS operation are thus in effect eliminated by multiplying them by zero. The resulting WSEM* $\hat{\boldsymbol{M}}_{\tau_0}(1,2,2,1)$*, containing the filtered SSHP values for time stage* τ_0*, is given in Figure 6.10.* ■



(a) (b)

FIGURE 6.8: *(a) A two-dimensional top view of a terrain filter and (b) the corresponding SSHP information adjusted according to terrain obstacles.*

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

FIGURE 6.9: *The terrain filter matrix* $\overline{\boldsymbol{M}}_{\tau_0}(1,2,2,1)$ *of Example 6.3.*

6.4 The engagement efficiency matrix component

The second component in the EQ subsystem of the proposed architecture in Figure 6.2 is the EEM component. The main purpose of the EEM component is to predict future flight paths for aircraft in the defended airspace and to use this information in conjunction with the information

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 15 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 15 & 15 & 15 & 0 & 0 & 15 & 15 & 15 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 15 & 15 & 15 & 15 & 65 & 0 & 0 & 65 & 65 & 65 & 65 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 15 & 15 & 15 & 65 & 65 & 65 & 0 & 0 & 65 & 65 & 65 & 65 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 \\
0 & 0 & 15 & 15 & 15 & 65 & 65 & 65 & 90 & 90 & 90 & 90 & 90 & 90 & 90 & 0 & 0 & 0 & 15 & 15 & 15 & 0 & 0 \\
0 & 0 & 15 & 15 & 15 & 65 & 65 & 65 & 90 & 90 & 90 & 90 & 90 & 90 & 90 & 65 & 65 & 65 & 15 & 15 & 15 & 0 & 0 \\
0 & 15 & 15 & 15 & 15 & 65 & 65 & 90 & 90 & 90 & 90 & 90 & 90 & 90 & 90 & 90 & 65 & 65 & 15 & 15 & 15 & 15 & 0 \\
0 & 15 & 15 & 15 & 15 & 65 & 65 & 65 & 90 & 90 & 90 & 90 & 90 & 90 & 90 & 65 & 65 & 65 & 15 & 15 & 15 & 15 & 0 \\
0 & 15 & 15 & 30 & 30 & 30 & 65 & 65 & 90 & 90 & 90 & 90 & 90 & 90 & 90 & 65 & 65 & 30 & 30 & 30 & 15 & 15 & 0 \\
0 & 15 & 15 & 30 & 30 & 30 & 65 & 65 & 65 & 90 & 90 & 90 & 90 & 90 & 65 & 65 & 65 & 30 & 30 & 30 & 15 & 15 & 0 \\
15 & 15 & 15 & 30 & 30 & 30 & 30 & 45 & 65 & 65 & 90 & 90 & 90 & 65 & 65 & 45 & 30 & 30 & 30 & 30 & 15 & 15 & 15 \\
15 & 15 & 15 & 30 & 30 & 30 & 45 & 45 & 45 & 90 & 90 & 90 & 90 & 90 & 45 & 45 & 45 & 30 & 30 & 30 & 15 & 15 & 15 \\
0 & 15 & 15 & 15 & 30 & 30 & 45 & 0 & 0 & 0 & 0 & 90 & 90 & 90 & 45 & 45 & 45 & 30 & 30 & 15 & 15 & 15 & 0 \\
0 & 15 & 15 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 90 & 90 & 90 & 45 & 45 & 45 & 30 & 15 & 15 & 15 & 15 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 90 & 90 & 45 & 45 & 45 & 45 & 15 & 15 & 15 & 15 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 45 & 45 & 45 & 45 & 15 & 15 & 15 & 15 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 15 & 15 & 15 & 15 & 15 & 15 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 15 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

FIGURE 6.10: *The resulting WSEM* $\hat{\boldsymbol{M}}_{\tau_0}(1,2,2,1)$ *after having filtered the SSHP matrix* $\boldsymbol{M}_{\tau_0}(1,2,2,1)$ *for terrain obstruction.*

contained in the WSEM generated by the PEF component in order to populate the EEM. The EEM contains efficiency values of all the WSs deployed with respect to all the threats in the defended airspace for each time stage in the temporal decision window under consideration. The models involved in the FPP module and the process of constructing the EEM are discussed in some detail in this section.

6.4.1 Flight path prediction models

Recall from §2.5 that a FPP module residing within the TM may be used to predict the future flight paths of enemy threats detected in the defended airspace over a pre-defined number of future time stages. In this section, two FPP models ranging in different levels of complexity are described briefly. The first is presented as three different incarnations of a basic *straight line* prediction model and the second is a more robust *probabilistic* prediction model.

A future SSHP prediction of a WS with respect to a hostile aircraft is only possible if the aircraft is detected within range of the specific WS for the future time stage in question — the SSHP value may simply be taken from the WSEM corresponding to the specific WS-threat pair for the future time stage. If an enemy aircraft is detected, but is outside the WSs fire arc range during the future time stage in question, the SSHP value is simply taken as zero. The estimation of SSHP values of WS-threat pairs for future time stages therefore depends on the future flight paths that enemy aircraft will follow.

Perhaps the simplest approach towards predicting the future flight path of an aircraft is to adopt a so-called *straight line* prediction model. According to this approach, the kinematic data of aircraft are extrapolated to form the future flight paths of threats under the strong assumption that the aircraft will continue to travel at their current velocities in straight lines (*e.g.* if an aircraft is travelling at 150 m/s, then it is assumed that the aircraft will continue travelling at this speed and in the same direction for any future time stage). The straight line prediction model is illustrated in Figure 6.11(a) [158].

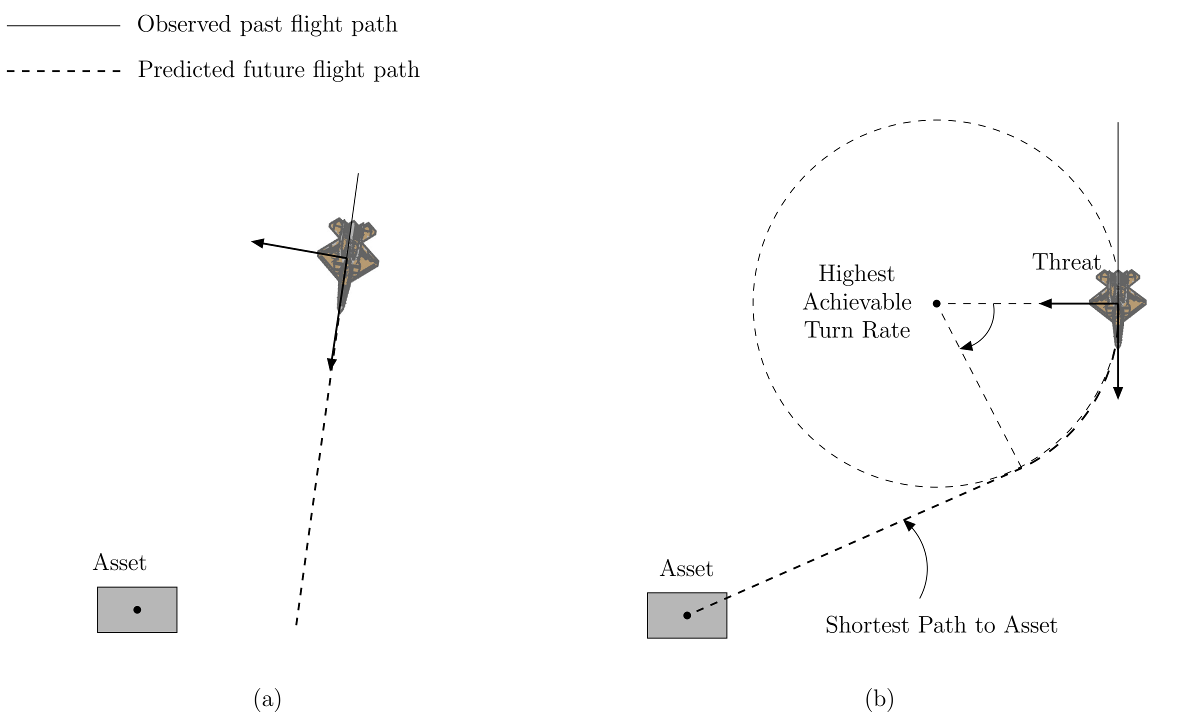


FIGURE 6.11: *(a) A straight-line FPP model based on current aircraft velocity and (b) an FPP model based on imminent highest achievable turn rate and a straight path to the asset [158].*

A more complex (albeit perhaps overly cautious) method for anticipating the future flight path is to make the assumption that an aircraft has to face the DA when delivering its weapons — in a worst-case scenario this will involve the aircraft achieving its highest possible turn rate until it faces the DA [158]. From there on it may be assumed that the threat will follow a straight-line path until it reaches the DA. This implies that the aircraft will follow the shortest path to the DA. This approach is illustrated graphically in Figure 6.11(b).

Another, more realistic, approach may be to assume that the aircraft will follow one of the attack techniques described in §2.5, rather than just continuing in a straight line. Aircraft attributes and the position of the aircraft at a given time stage may then be used to assign a probability to each attack technique that the aircraft can follow, based on its past and present kinematic data. The predicted position of the aircraft at the next time stage may subsequently be taken as a projected position along the attack technique that the aircraft is most likely to follow. The position of an aircraft during the next time stage may alternatively be calculated by taking a weighted average of the projected positions along each possible attack technique — the probabilities associated with the various attack techniques are used as weights in the calculation.

Yet another approach is to predict an area within which an aircraft may likely be during future time stages, rather than attempting to predict a specific point at which an aircraft may be. It is envisaged that such a prediction would involve a three-dimensional volume containing different areas in which the aircraft may find itself. Each of these areas may be associated with a probability that the aircraft will be in that specific area. A top view of a horizontal slice of an aircraft area prediction is illustrated graphically in Figure 6.12. Similar to the way in which the SSHP volume was discretised in Figure 6.3, the volume of air space may be discretised by employing a three-dimensional grid (having the same coarseness as the SSHP matrix), thus facilitating computation of a so-called *Predicted Position Matrix* (PPM). If a threat is predicted to be within range of a WS fire arc, a future efficiency value for the WS-threat pair may be calculated by multiplying each entry in the PPM with the corresponding entry in the SSHP matrix and adding all these multiplications together [145].

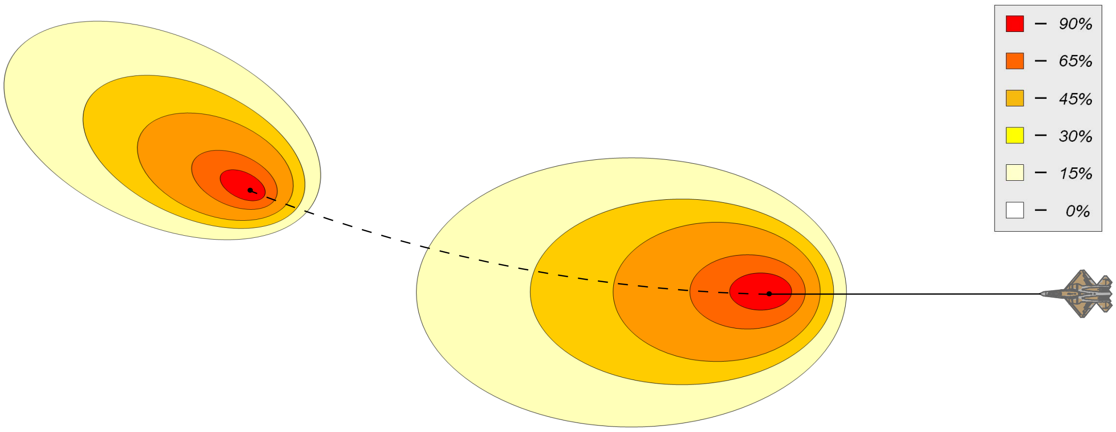


FIGURE 6.12: *Prediction of the positions of a threat at two future time steps by means of a probabilistic area prediction model.*

Let $\boldsymbol{P}_{t+T_\ell}(w, v, e, a)$ denote the PPM at time stage τ predicted for time $\tau + T_\ell$ and let $\hat{\boldsymbol{M}}_\tau(w, v, e, a)$ be the WSEM at time τ for a WS of type $w \in \mathbb{W}$ and for the following aircraft (threat) combination: an aircraft of type $v \in \mathbb{V}$ carrying a weapon of type $e \in \mathbb{E}$ and executing an attack technique of type $a \in \mathbb{A}$ [145]. The efficiency $E_\tau(i, j, T_\ell)$ of WS i predicted at time stage τ for future time stage $\tau + T_\ell$ in respect of threat j may then be calculated as

$$E_\tau(i, j, T_\ell) = \sum_x \sum_y \sum_z \big((\boldsymbol{P}_{t+T_\ell}(w, v, e, a))_{x,y,z} \times (\hat{\boldsymbol{M}}_{t+T_\ell}(w, v, e, a))_{x,y,z}\big),$$

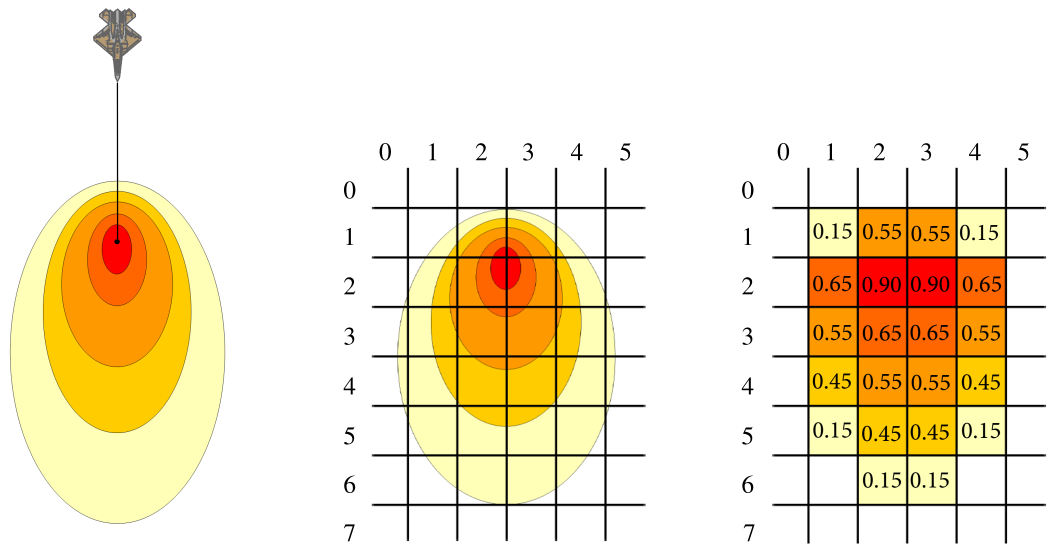
subject to the requirement that

$$\sum_x \sum_y \sum_z (\boldsymbol{P}_{t+T_\ell}(w, v, e, a))_{x,y,z} = 1.$$

The working of the probabilistic area prediction model described above is illustrated in the following example.

Example 6.4 (Probabilistic area prediction model for a single time stage) *Suppose a WS is deployed with a northerly orientation. Assume that the WS's SSHP volume has been discretised as in Example 6.4 and filtered to produce the WSEM for time stage τ_0 presented in Figure 6.10 — the same WS type and aircraft configuration as in Example 6.1 is assumed.*

Assume further that the aircraft approaches from the north and that a probabilistic prediction model is used to predict the future area of the threat, as illustrated graphically in two dimensions in Figure 6.13(a). This area may be discretised in a similar fashion as the SSHP information in Example 6.1 (by employing the same coarseness as that adopted in the discretisation of the



(a) (b) (c)

FIGURE 6.13: *A two-dimensional top view of (a) an area prediction of an aerial threat approaching from the north, (b) a grid placed over the area prediction and (c) the SSHP values corresponding to cells of the grid.*

SSHP information). The discretised grid is illustrated graphically in Figure 6.13(b). The cells in each block of the discretised grid may be coloured in the same way as the cells in the grid of the discretised SSHP information, as illustrated in Figure 6.13(c).

Suppose the PPM matrix in Figure 6.14(a) is predicted for time step $\tau_0 + T_0$ as output from the FPP model at time τ_0. Suppose further that these cells correspond to the cells of rows 1 to 6 and columns 12 to 15 of the WSEM at time τ_0, as illustrated in Figure 6.14(b).

The efficiency value that WS 3 achieves with respect to the threat at time step $\tau_0 + T_0$ may therefore be calculated by weighing each entry in the WSEM of Figure 6.14(b) with its corresponding entry in the PPM of Figure 6.14(a) and adding the results together. The efficiency value

$$\begin{aligned} E_{\tau_0}(3,4,T_0) &= 2\% \times 9\% + 5\% \times 11\% + 5\% \times 9\% + 2\% \times 0\% \\ &\quad + 7\% \times 50\% + 9\% \times 50\% + 9\% \times 41\% + 7\% \times 0\% \\ &\quad \vdots \\ &\quad + 0\% \times 80\% + 2\% \times 80\% + 2\% \times 69\% + 0\% \times 69\% \\ &= 39\% \end{aligned}$$

is thus obtained. ■

Although the stochastic prediction approach illustrated above may be computationally more expensive than a simple straight line prediction model, it may provide a more realistic estimation of the efficiency of WSs with respect to threats at future time stages [145]. The approach may also absorb certain deficiencies, including sensor measurement errors and variations in aircraft dynamics.

Caution should, however, be taken not to attempt to predict too far into the future, as predicting aircraft positions too far ahead may yield a distorted picture of the efficiency that WSs are able

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} 12 & 13 & 14 & 15 \end{array} \\ \begin{bmatrix} 2 & 5 & 5 & 2 \\ 7 & 9 & 9 & 7 \\ 5 & 7 & 7 & 5 \\ 3 & 5 & 5 & 3 \\ 2 & 3 & 3 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix} \end{array}$$

(a)

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} 12 & 13 & 14 & 15 \end{array} \\ \begin{bmatrix} 9 & 11 & 9 & 0 \\ 50 & 50 & 41 & 0 \\ 50 & 50 & 41 & 0 \\ 69 & 69 & 57 & 0 \\ 69 & 69 & 57 & 41 \\ 80 & 80 & 69 & 69 \end{bmatrix} \end{array}$$

(b)

FIGURE 6.14: *(a) The PPM for time stage $\tau_0 + T_0$ and (b) the entries in rows 1 to 6 and columns 12 to 15 in the WSEM at time τ_0.*

to achieve with respect to threats, which may behave in an unforseen manner, resulting in the TEWA system providing misleading results as DS.

6.4.2 Constructing the engagement efficiency component

The resulting WSEM produced by the PEF component may be used in conjunction with the predicted future flight paths of threats to calculate SSHP values for the WSs with respect to all the threats over the entire prediction window for a given scenario. If a particular threat is predicted to be within range of a specific WS during a specific future time stage, the SSHP of the WS with respect to that threat for the specific time stage may be taken as the value in the WSEM corresponding to the position of the threat during the specific time stage. This is, of course, only applicable if the given threat is predicted to be within range of the WS. If the predicted future position of the threat falls outside of the range of the WS, the efficiency value of the WS with respect to that threat is taken as zero, as mentioned.

The EEM is therefore a three-dimensional $m \times n \times \tau$ matrix, where m denotes the number of WSs in the deployment, n denotes the number of threats in the defended airspace, and τ denotes the pre-determined number of future time stages over which the flight paths of the threats are predicted. The EEM is illustrated graphically in Figure 6.15. Once the EEM component has been populated, it is stored in the TEWA database for subsequent use by the WA subsystem.

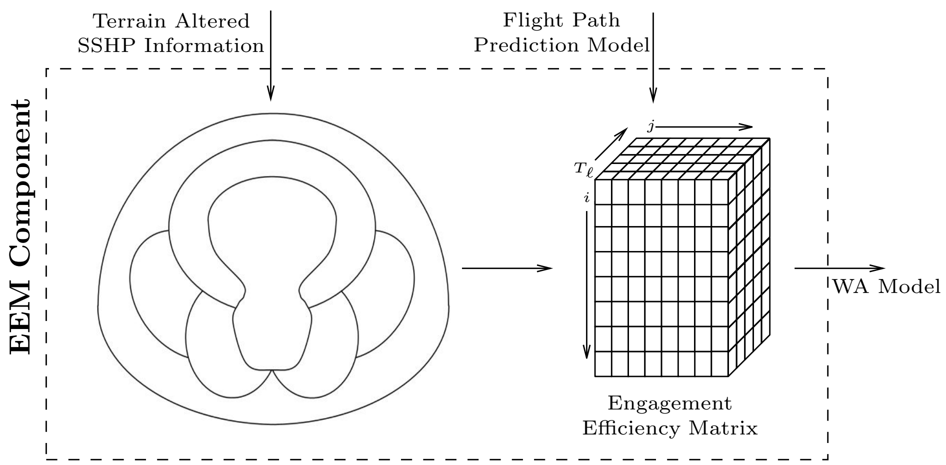


FIGURE 6.15: *The three-dimensional EEM constructed from terrain altered SSHP information.*

6.5 The weapon assignment model component

The WAM component contains a collection of mathematical assignment models, each responsible for solving a special variant of the WA problem and proposing assignments of WSs to threats in real-time. These assignments are typically based on their desirability with respect to some pre-specified objective function or set of objective functions, as dictated by the relevant WA problem variant. It is proposed that the WAM component contains four classes of WAMs, as illustrated in Figure 6.16. The four classes of WAMs are single-objective static WAMs, multi-objective static WAMs, single-objective dynamic WAMs and finally multi-objective dynamic WAMs. The four classes of models range in different levels of complexity in terms of the solutions that they yield as well as the computational time and effort expended to solve the WAMs in each class. The complexities of the four classes of WAMS are illustrated using a grey-scale shading scheme in Figure 6.16. The WAMs contained in the lighter coloured quadrants are considered the least complex while the WAMs contained in the darker coloured quadrants are considered to be more complex. Furthermore, Figure 6.16 also contains references to authors who have contributed to the formulations of the four WA prototypes over the years (*i.e.* the models reviewed in §3).

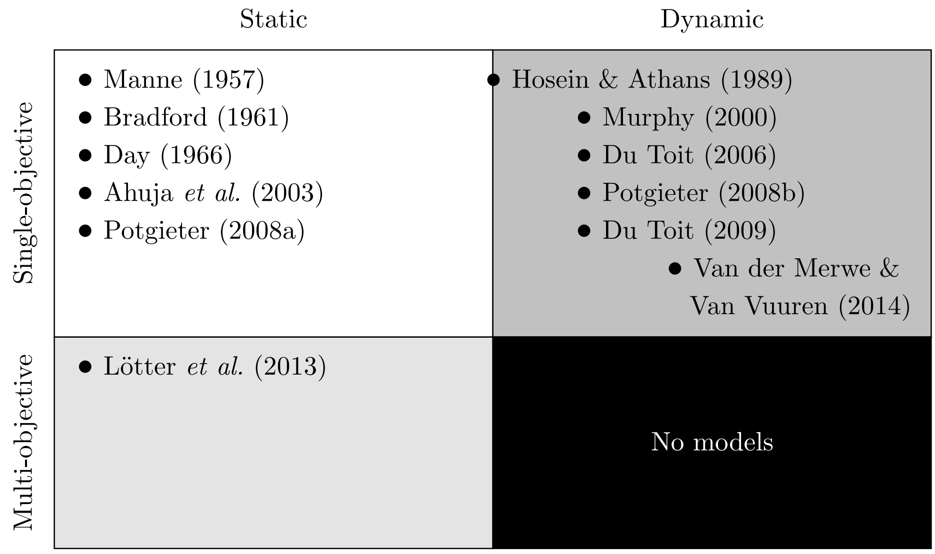


FIGURE 6.16: *Four classes of WAMs together with WAM instances in each class (where applicable).*

As of the beginning of 2014, the author was only aware of the multi-objective static WAM proposed by Lötter *et al.* [120] in 2013 (*i.e.* the model in the third quadrant of Figure 6.16). Later, in 2014, this formulation was extended by the author with the addition of a third objective function to the bi-objective model [123]. This additional objective aims to maximise the number of times that the least re-engagable WS in the set of WSs may be considered again for future assignments *after* the assignment of the current time stage under consideration has been completed. These models are, to the best knowledge of the author, the only two multi-objective static WAMs.

Furthermore, the author is not aware of any model in the final class of Figure 6.16 (*i.e.* the multi-objective dynamic WAM class). The author therefore formulated a tri-objective dynamic WAM in order to complete the four classes of available WAMs in the WA literature [121]. The development of this model is one of the main contributions of this dissertation and is discussed in some detail later in this chapter.

In the remainder of this section, a WAM prototype is selected within each of the model classes in Figure 6.16 and discussed in detail. These four prototypes are arguably the most realistic models in each of the four classes of Figure 6.16, and it is advocated that these prototypes should be included in the WAMS component of Figure 6.2 as default WAMs in case the FCO chooses not to configure a WAM for inclusion in the WA subsystem. The four prototypes are presented in increasing order of complexity as indicated by the grey-scale colour scheme in Figure 6.16 and not in the order of their chronology.

6.5.1 Single-objective, static WAM prototype

The first class of WAMs considered is the single-objective, static class which is considered to be the least complex of the four model classes of Figure 6.16. WAMs in this class are static in a temporal sense, *i.e.* they are solved over a single time stage at a time — the WA problem is typically solved only for the current time stage or over a specified number of time stages in the future, consecutively and independently, save for updating parametric model information (so as, for example, to account for threats eliminated). A further characteristic of models in this class is that they involve single-objective optimisation (*i.e.* only one objective function is included in each model within this class, which either has to be minimised or maximised).

The single-objective static WAM chosen for inclusion as default formulation in the WAM component is the k-WAP of Potgieter [145]. In this model, a single objective is pursued and the problem is limited to a single time stage only in a discretisation of the temporal decision period under consideration. The model functions under the assumption that the events of a threat surviving engagements by two different WSs are independent. Hence, the probability of survival of a specific threat may be calculated as the product of the probabilities of surviving engagements by the separate WSs assigned to engage it — the value is therefore a product of the complements of all the SSHP values of all the WSs assigned to engage it. The survival probability of the threat is then weighted by the priority of eliminating the threat, as determined by the TE subsystem. The quality of a feasible solution to the model may therefore be expressed in terms of the accumulated survival probabilities of all the threats weighted by the priorities of eliminating the respective threats. Furthermore, it is assumed that the hit probability of a WS with respect to a specific threat depends only on the WS and the threat involved.

Suppose there are $m(\tau)$ WSs available for assignment to engage any of the $n(\tau)$ observed aircraft classified as threats during time stage τ. Let $V_j(\tau)$ denote the priority[1] of eliminating threat j during time stage τ. Also, let $p_{ij}(\tau)$ denote the probability of eliminating threat j if it is engaged by WS i during time stage τ (*i.e.* the SSHP value of WS i with respect to threat j during time stage τ). Then $q_{ij}(\tau) = 1 - p_{ij}(\tau)$ denotes the survival probability of threat j if it is engaged by WS i during time stage τ. In addition, let $A_i(\tau)$ denote the number of ammunition units available to WS i during time stage τ — these units may, for example, be bursts of bullets when a cannon is used in burst mode or the number of missiles launched by a missile system. Furthermore, let κ be the maximum number of WSs that may be assigned to any threat. The decision variables adopted in the formulation are binary in nature and are denoted by $x_{ij}(\tau)$, taking a value of 1 if WS i is assigned to engage threat j during time stage τ, or the value 0 otherwise. The objective in the κ-WA problem is to

$$\text{minimise} \quad \sum_{j=1}^{n(\tau)} V_j(\tau) \prod_{i=1}^{m(\tau)} q_{ij}(\tau)^{x_{ij}(\tau)} \tag{6.6}$$

[1]Recall from §6.7 that this value is determined during the process of TE.

subject to the constraints
$$\sum_{j=1}^{n(\tau)} x_{ij}(\tau) \leq A_i(\tau), \quad i = 1, \ldots, m(\tau), \tag{6.7}$$
$$\sum_{i=1}^{m(\tau)} x_{ij}(\tau) \leq \kappa, \quad j = 1, \ldots, n(\tau), \tag{6.8}$$
$$x_{ij}(\tau) \in \{0, 1\}, \quad i = 1, \ldots, m(\tau), \; j = 1, \ldots, n(\tau). \tag{6.9}$$

Constraint set (6.7) ensures that WS i is considered for assignment at most $A_i(\tau)$ times during time stage τ, while constraint set (6.8) ensures that no more than κ WSs are considered for assignment to engage any single threat. Finally, constraint set (6.9) enforces the binary nature of the decision variables.

A detailed description of how the model (6.6)–(6.9) may be solved efficiently (albeit approximately) is provided in a later chapter of this dissertation where the model is applied in a concept demonstration to a realistic GBAD scenario.

6.5.2 Multi-objective, static WAM prototype

At the next level of complexity is the class of multi-objective static WAMs. WAMs in this class are similar to single-objective static WAMs in the sense that they are temporally static in nature. Multi-objective, static models are, however, distinguished from the previous class in the sense that they involve multi-objective optimisation (*i.e.* more than one objective function has to be minimised or maximised simultaneously). The objectives in such models are typically conflicting in nature, which requires that suitable trade-offs be sought between objective function values. Multi-objective optimisation therefore naturally gives rise to the notion of Pareto optimality (as described in Chapter 4) instead of the simpler and more intuitive notion of optimality.

The multi-objective static WAM selected as the prototype of its class is the tri-objective, static WAM of the author [123]. This model was formulated by selecting three objectives from the list of objectives in Table 3.1. Again, the model functions under the assumption that the events of a threat surviving engagements by two different WSs are independent and hence that the probability of survival of a specific threat may be calculated as the product of the probabilities of surviving engagements by the separate WSs assigned to engage it. It is also, again, assumed that the hit probability of a WS with respect to a specific threat depends only on the WS and the threat involved. Furthermore, the model is static, implying that it has to be re-solved for each time stage in the time continuum under consideration.

The first objective in the model is similar to the objective function in the k-WA model (6.6)–(6.9) which aims to minimise the accumulated survival probabilities of the threats in the system, weighted by the priorities of eliminating each of these threats, respectively. The second objective in the model is to minimise the accumulated cost of the WS-threat assignment pairs — the cost of a WS-threat assignment pair is taken as the monetary (Rand) value of a single burst of the ammunition used in the assignment. Finally, the third objective in the model is to maximise the number of times that a WS is available for re-engagement *after* the proposed assignment during future time stages — the objective therefore encourages the assignment of WSs having the largest quantity of ammunition available first, rather than assigning WSs with lower quantities of available ammunition first in a bid to ensure that all weapons are reusable (in terms of ammunition availability) after the assignment. Adopting the same notation as in §6.5.1, the objectives in the static tri-objective WA model are therefore to

$$\text{minimise} \quad \sum_{j=1}^{n(\tau)} V_j(\tau) \prod_{i=1}^{m(\tau)} q_{ij}(\tau)^{x_{ij}(\tau)}, \tag{6.10}$$

$$\text{minimise} \quad \sum_{i=1}^{m(\tau)} C_i \sum_{j=1}^{n(\tau)} x_{ij}(\tau), \text{ and} \tag{6.11}$$

$$\text{maximise} \quad \min_i \left\{ A_i(\tau) - \sum_{j=1}^{n(\tau)} x_{ij}(\tau), \right\} \tag{6.12}$$

subject to the constraints (6.7)–(6.9), where C_i denotes the monetary cost (in Rand value) of firing a single unit of WS i ammunition.

The working of the model (6.10)–(6.12) is also illustrated in detail in a later chapter in this dissertation where it is solved approximately in the context of a realistic GBAD scenario.

6.5.3 Single-objective, dynamic WAM prototype

The next class of WAMs is the class of single-objective dynamic WAMs. Models in this class are similar to the single-objective static class of WAMs in that they involve single-objective optimisation, but they differ from the former class of models in the sense that they take into account the change in the kinematic behaviour of aircraft over a pre-specified temporal interval or time horison, which classifies them as dynamic WAMs. The dynamic nature of these models imply that the WA problem is solved for the current time stage as well as for a number of pre-specified future time stages (*i.e.* the number of time stages specified in the FPP model described in §6.4.1) in the time continuum. These models therefore also involve a scheduling element of *when* WSs should engage threats in addition to the desirability or effectiveness of matching WSs to threats (the assignment element).

The single-objective dynamic WAM selected as prototype within this class is the single-objective, dynamic WA scheduling model of Van der Merwe and Van Vuuren [196], described in more detail in §3.5.

As in the VRPTW, a virtual depot is included in the prototype single-objective, dynamic WAM which may be seen as an artificial construct representing an idle state during which no WS engages any threats. Let the virtual depot be indexed by threat 0 and threat $n+1$, where n is the number of threats in the system. The depot has a demand of zero and the system is required to start from this state and to return to it again after all WS-threat pair engagements have been carried out. Let d_i denote the WS setup time of WS i. Define a FTTF e_{ijk} and a LTTF ℓ_{ijk} for WS i when engaging threat j during the pair's k^{th} FW. Let s_{ijk} denote the engagement time duration[2] of threat j by WS i during the pair's k^{th} FW, where $s_{ijk} = \ell_{ijk} - e_{ijk} + 1$, and let f_{ij} be the number of distinct FWs for WS-threat pair (ij). Furthermore, let p_{ijk} denote

[2]The engagement time duration refers to the number of time stages required by a WS to guide its ammunition in the direction of the threat from the time stage during which the ammunition is launched (the time stage during which the trigger is pulled) until the time stage during which the ammunition hits the threat. This is only applicable to WSs which require operator guidance, such as cannons which are able to fire multiple bursts with the pull of the trigger, but require manual WS operator assistance to guide the ammunition in the direction of the threat. In some instances, WSs have on-board guidance systems, such as certain missiles, and hence require no guidance from the WS operator, in which case the value s_{ijk} is assumed to be zero. The value s_{ijk} is included in the model formulation for the sake of flexibility.

the SSHP value associated with threat j if engaged by WS i during the pair's k^{th} FW, and let $q_{ijk} = 1 - p_{ijk}$.

A fixed mean approach is adopted in the formulation in order to associate a single survival value with the k^{th} FW of WS-threat pair (i, j)[3]. The fixed mean approach entails considering a subset (of fixed, prespecified cardinality) of contiguous time stages within a FW and calculating the mean of the survival probabilities associated with the time stages in this subset. The time stages chosen for inclusion in the subset should be consecutive and should also include the time stage achieving the minimum survival probability in the entire set of time stages. This procedure is carried out for each such subset in the FW [196]. Finally, the smallest mean obtained over all the subsets is selected and taken as the "survival probability" of the threat for the FW under consideration. Let M_{ijk} denote the set of consecutive time stages to be used in calculating the fixed mean of the k^{th} FW for WS-threat pair (i, j). Then the fixed mean value is

$$\mu_{ijk} = \frac{1}{|M_{ijk}|} \sum_{\tau \in M_{ijk}} q_{ij\tau}. \tag{6.13}$$

A binary decision variable x_{ijk} is adopted in the formulation which takes the value 1 if WS i engages threat j during the k^{th} FW associated with the WS-threat pair, or a value of 0 otherwise. A binary auxilliary variable y_{ihj} is also incorporated, which may be interpreted as a flow variable, and takes a value of 1 if threat h directly precedes threat j in a sequence of engagements by WS i, or a value of 0 otherwise. Adopting the same notation as used in the previous WAMs discussed in this section, the objective in the single-objective dynamic WAM is to

$$\text{minimise} \quad \sum_{j=1}^{n} V_j \prod_{i=1}^{m} \prod_{k=1}^{f_{ij}} (\mu_{ijk})^{x_{ijk}} \tag{6.14}$$

subject to the constraints

$$\sum_{i=1}^{m} \sum_{h=0}^{n+1} y_{ihj} \leq \kappa, \qquad j = 1, \ldots, n, \tag{6.15}$$

$$\sum_{h=1}^{n+1} y_{i0h} = 1, \qquad i = 1, \ldots, m, \tag{6.16}$$

$$\sum_{h=1}^{n+1} y_{i,h,n+1} = 1, \qquad i = 1, \ldots, m, \tag{6.17}$$

$$\sum_{h=0}^{n+1} y_{ihj} - \sum_{h=0}^{n+1} y_{ijh} = 0, \qquad i = 1, \ldots, m, \quad j = 0, \ldots, n+1, \tag{6.18}$$

$$\sum_{k=1}^{f_{ij}} x_{ijk} = \sum_{\substack{h=1 \\ h \neq j}}^{n+1} y_{ihj}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n, \tag{6.19}$$

$$\sum_{k=1}^{f_{ih}} (e_{ihk} + s_{ihk}) x_{ihk} - \sum_{k=1}^{f_{ij}} e_{ijk} x_{ijk} + d_i < (1 - y_{ihj}) L, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n, \quad h = 1, \ldots, n, \tag{6.20}$$

[3]Note that the objective function formulated for inclusion as default WAM in the single-objective dynamic class of WAMs is formulated in such a way that it conforms to the same objective used in the multi-objective dynamic WAM formulated later in this section.

$$\sum_{j=1}^{n}\sum_{k=1}^{f_{ij}} x_{ijk} \leq A_i, \qquad i = 1,\ldots,m, \tag{6.21}$$

$$y_{ihj} \in \{0,1\}, \qquad i = 1,\ldots,m, \quad j = 0,\ldots,n+1, \quad h = 0,\ldots,n+1, \tag{6.22}$$

$$x_{ijk} \in \{0,1\}, \qquad i = 1,\ldots,m, \quad j = 0,\ldots,n+1, \quad k = 1,\ldots,f_{ij}. \tag{6.23}$$

Constraint set (6.15) ensures that at most κ WSs are assigned to engage any threat over the scheduling horizon. Constraint set (6.16) ensures that WS i "leaves the depot" (idle state) exactly once if it is assigned to engage threats at all, while constraint set (6.17) ensures that WS i "returns to the depot" exactly once after being used to engage threats. Constraint set (6.18) ensures, if a threat is engaged by WS i, that the WS "leaves the threat" again in order to "move on" to engage the next threat assigned to it. Constraint set (6.19) ensures, if threat h precedes threat j for engagement by WS i, that threat h is engaged during exactly one stage. If threat h is engaged by WS i directly before threat j, constraint set (6.20) ensures that the time stage during which the engagement of threat h starts plus the time it takes to engage threat h plus the time it takes for WS i to "travel" from threat h to threat j does not exceed the time stage during which engagement of threat j starts, where L is a large number. Furthermore, constraint set (6.20) also ensures that the stage during which WS i engages threat j is within a FW associated with the (WS, threat)-pair. Constraint set (6.21) ensures that the capacity of WS i is not exceeded and constraint sets (6.22)–(6.23) finally ensure the binary nature of the decision and auxilliary variables.

A method for solving the model (6.14)–(6.23) is discussed in detail later in this dissertation where the model is solved in the context of a realistic GBAD scenario.

6.5.4 Multi-objective dynamic WAM prototype

The final class of WAMs is the class of multi-objective dynamic WAMs which is also considered to be to most complex of the four classes of WAMs proposed. These models involve solving the WA problem in a multi-objective context over a number of pre-specified, predicted time stages and also include a scheduling element similar to models in the class of single-objective, dynamic WA models (*i.e.* requiring decisions as to *when* WSs should engage threats in addition to *which* WSs should engage which threats).

After conducting an extensive literature survey on the WAMs available in the operations research literature, the author could not find any existing multi-objective dynamic WAMs for use in the fourth quadrant of Figure 6.16. The author therefore formulated a tri-objective dynamic WAM prototype, published by Lötter and Van Vuuren [121], in order to complete the various classes of WAMs in the proposed WAM component.

The formulation of this WAM is also based on the classical VRPTW and was formulated in the same way as its single-objective dynamic WAM counterpart by Van der Merwe and Van Vuuren [196], described in §6.5.3. Apart from being able to schedule future engagement FWs during which WS-threat assignment pairs should occur, the tri-objective dynamic WAM also incorporates three objectives simultaneously. The first objective is to minimise the accumulated survival probabilities of the threats weighted by their respective elimination priorities as in the model of §6.5.3. The second objective is to minimise the accumulated cost of the assignment proposed and the third objective is to maximise the number of times that the least re-engagable WS in the set of WSs can be used for future assignments after the current assignments have occurred, as in the model of §6.5.2. The model functions under the same assumptions as those of the single-objective, dynamic model (6.14)–(6.23).

Following the same notation as in the single-objective static WAM (6.6)–(6.9), the bi-objective static WAM (6.10)–(6.12) and the single-objective dynamic WAM (6.14)–(6.23), the objectives in the tri-objective dynamic WAM are to

$$\text{minimise} \quad \sum_{j=1}^{n} V_{j\tau} \prod_{i=1}^{m} \prod_{k=1}^{f_{ij}} (\mu_{ijk})^{x_{ijk}}, \tag{6.24}$$

$$\text{minimise} \quad \sum_{i=1}^{m} C_i \sum_{j=1}^{n} \sum_{k=1}^{f_{ij}} x_{ijk}, \text{ and} \tag{6.25}$$

$$\text{maximise} \quad \min_i \left\{ A_i - \sum_{j=1}^{n} \sum_{k=1}^{f_{ij}} x_{ijk} \right\} \tag{6.26}$$

subject to the constraints (6.15)–(6.23).

The working of the model (6.24)–(6.26) is also illustrated in the context of a simulated, but realistic, GBAD test scenario later in this dissertation.

6.5.5 WAM configuration

It is proposed that the FCO configures one WAM from one of the four classes of proposed WAMs during the pre-deployment stages of a mission for use during the mission. It is further proposed that for each pre-configured WAM there should be a collection of suitable solution methodologies available for solving the WA problem. The FCO's preferred solution methodologies should also be configurable during the pre-deployment stage. The idea is that when the FCO chooses a specific WAM, the WA problem should be solved (approximately) by a collection of solution methodologies simultaneously within the context of that WAM. Existing solution methodologies from the operations research literature which may be used to solve WAMs in each of the four classes of WAMs were described in §6.5. The results obtained by the various solution methodologies should then be relayed to the WASS component for further analysis.

Finally, it is acknowledged that time is a critical factor in providing efficient TEWA DS to the FCO. It is therefore important that the solution methodologies employed in solving the models in the WAM component should be able to solve the models almost instantaneously, albeit only approximately. It is proposed that the solution methodologies employed to solve the WAM configured by the FCO should run in parallel so as to ensure that they solve the configured WAM within a pre-specified time budget (*e.g.* 1 second).

6.6 The weapon assignment solution selection component

The final component in the proposed WA architecture design in Figure 6.2 is the WASS component. The purpose of this component is to present the results obtained by solving the WAM configured by the FCO in the WAM component in an uncluttered fashion as decision support to the FCO. This may be achieved by combining the results obtained from all the solution methodologies in the WAM component so as to form a possibly large collection of candidate solutions to the WA problem instance in question. Solutions of poor quality (*i.e.* those dominated by other in objective space) should then be filtered out in order to retain only the nondominated solutions (*i.e.* to obtain a collection of approximately Pareto optimal solutions only). This may be achieved by using any of the methods described in §4.6. The front of approximately Pareto optimal solutions should then be stored in the TEWA database and relayed to the FCO *via* the HMI as real-time DS. The FCO may then choose a solution which best corresponds to his intuition based on his experience and subjective judgement.

6.7 The envisaged chronological order of TEWA events

The components described in the previous sections of this chapter all function as part of a so-called *TEWA cycle* of actions or events. It is envisaged that a trigger of such a cycle initiates consecutive calls to these components in an orderly fashion so as to provide real-time DS to the FCO. For the purposes of the proposed WA subsystem in this dissertation, it is assumed that a TEWA cycle is initiated either by a change in any of the data fields[4] in the TEWA database, which is significant enough to justify the termination of a current TEWA cycle and to start a next TEWA cycle, or by the natural continuation of an implementation clock cycle. The sequential order in which the components function within a TEWA cycle is illustrated graphically in Figure 6.17 and is discussed in some detail in this section.

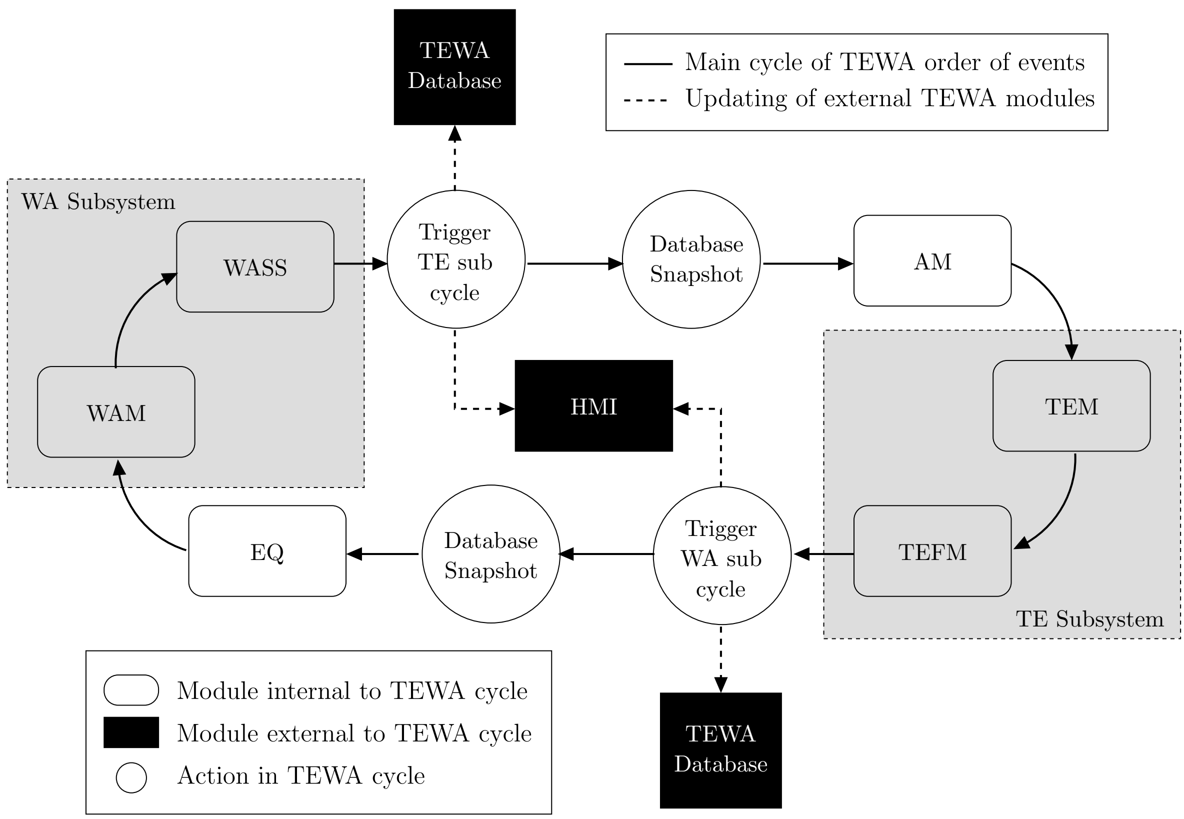


FIGURE 6.17: *Detailed sequential order of events in a TEWA implementation clock cycle.*

[4]This includes changes in any of the entries in the TEWA database (*e.g.* sensor, target, WS, or DA track changes).

Once a TEWA cycle is triggered, a snapshot of the TEWA database is taken at that specific time stage. This snapshot contains the data to be used during the current TEWA computation cycle. After the snapshot of the database has been taken, the AM commences operation and the measured attributes obtained from the sensor systems are fused by the TM, as described in §2.5.2, after which derived attributes are computed for each of the observed threats.

Upon completion of the computations by the AM, the TEM process is initiated, during which the TEMs described in §2.5.2 evaluate the perceived level of threat posed by the observed aircraft with respect to the DAs. Threat lists are computed containing the prioritised threat values and these lists are fused together by a multi-criteria decision analysis method[5] within the TEFM component to obtain a single prioritised list of threats. This single list is stored in the TEWA database and presented to the FCO via the HMI. Once the results from the TE subsystem have been stored in the TEWA database, WA subsystem operation commences.

Another snapshot of the TEWA database is taken to acquire the additional data required by the WA components. The efficiency values that WSs are expected to achieve with respect to threats during future time stages are discretised and discounted for meteorological conditions and terrain obstacles within the EQ component, as described in §6.3. The output (*i.e.* an EEM) is stored in the TEWA database for each future time stage. Thereafter, the WAM component employs the threat list produced by the TEFM component and the EEM produced by the EQ component to propose assignments of available WSs to engage the aerial threats. The outputs of the WAM component (*i.e.* a number of proposed WA lists) are stored in the TEWA database and presented to the WASS component in which the WA lists are filtered and presented to the FCO via the HMI in a manner configured by the operator during pre-deployment. The presentation of these results to the FCO marks the end of a full TEWA cycle.

Although the TEWA system may seem to function as a fully automated software system, it is important to acknowledge that the FCO should have the authority to override assignment decisions suggested by the system. He should, for example, be able to configure the TEMs actually used in the threat evaluation process, alter the prioritised threat list obtained from the TEFM component, and alter the WA lists proposed by the WAM component, based on his experience and subjective judgement.

6.8 Chapter summary

A generic WA subsystem architecture was put forward in this chapter for use in a GBAD TEWA DSS. The chapter opened in §6.1 with a brief discussion on DS design approaches in general. In §6.2, the design of a novel WA subsystem architecture was proposed and discussed briefly. The remainder of the chapter was devoted to detailed descriptions of the components contained in the proposed WA architecture.

The WA architecture comprises two subsystems, *i.e.* an EQ subsystems and a WA subsystem. Furthermore, the EQ subsystem contains two components, *i.e.* the PEF component and the EEM component. The working of the PEF component was discussed in §6.3 and a method was reviewed for discretising the effectiveness values that WSs achieve with respect to threats. In addition, a process for filtering effectiveness values of WSs for constraints posed by extreme environmental conditions and terrain features was also presented. In §6.4, the working of the EEM component was discussed. Two methods for predicting the future flight paths of threats

[5]Typical multi-criteria decision analysis methods which may be used within the TEFM component include the analytic hierarchy process of Saaty [165], goal-orientated models (such as goal programming and aspiration level models) or outranking models (such as ELECTRE) [14].

were discussed and a method for constructing the EEM was presented using the information returned by the FPP models.

Next, the focus shifted to a discussion on the components contained in the WA subsystem. The WAM component, which forms the heart of the WA architecture, was considered in §6.5. Four classes of WAMs ranging in different levels of complexity were proposed for inclusion in the WAM component. These classes were single-objective static WAMs, multi-objective static WAMs, single-objective dynamic WAMs and multi-objective dynamic WAMs. A prototype within each of these classes was selected for default inclusion in the WAM component and discussed in detail. These prototypes were presented in increasing order of complexity. A novel multi-objective static WAM was also formulated for inclusion in the second most complex class of WAM. The author was (to the best of his knowledge) unaware of any WAMs residing within the most complex class of WAMs (*i.e.* multi-objective dynamic WAMs). A novel contribution of this dissertation is therefore the formulation of a tri-objective dynamic WAM for inclusion in the final, hitherto empty quadrant of Figure 6.2.

In §6.6, the final component of the WA architecture, *i.e.* the proposed working of the WASS component, was discussed briefly. It was proposed that a number of solution methodologies be used concurrently to solve the WAM configured by the FCO and that the candidate solutions thus obtained be sorted to obtain a set of nondominated solutions for presentation as real-time DS to the FCO.

The chronological order in which the events within a single TEWA computational cycle occur was finally described in §6.7.

CHAPTER 7

Weapon assignment implementation suggestions

Contents

In this chapter, eight practical suggestions are put forward which may prove useful when implementing a WA subsystem (such as the one proposed in Chapter 6) in practice. The chapter opens in §7.1 with a discussion on the automation of TEWA systems. Five levels of automation are discussed briefly and it is advocated that a TEWA system should not be a fully automated system. It should aim to assist FCOs in their decisions related to the execution of complex computations rather than replacing the FCO or contradicting the decisions that the FCO would have made in the absence of a TEWA DSS. Next, the focus shifts in §7.2 to a discussion on how the quality and quantity of TEWA-related data may be improved in order to facilitate high-quality results from a TEWA DSS.

The importance of providing the FCO with only the information that he needs is discussed briefly in §7.3. Since the HMI serves as the link between the FCO and the results provided by the TEWA DSS, a number of suggestions are provided for an efficient layout of the HMI. The problem of providing the FCO with unnecessary information is related to the problem of information overload. The former problem is discussed in §7.4. It is important not to overwhelm the FCO with unnecessary information, and this problem is alleviated to some extent by proposing a decision tree which may aid the operator in configuring WAM models (during the pre-deployment stages of a mission) for customised use in the TEWA DSS.

Next, the focus shifts in §7.5 towards incorporating FCO preferences and biases. A potential problem emanating from multi-objective models in the WAM component of §6.16 is that the operator may be presented with too many Pareto optimal solutions from which he must choose one for implementation purposes. Four suggestions for solving this problem are provided and discussed briefly.

The undesirable phenomenon of switching (which refers to rapid changing of assignment suggestions by the DSS during a small subset of consecutive time stages) is considered in §7.6. It is suggested that threshold values be implemented in the system in order to allow only changes in the suggestions from one time stage to another once a variation in the results equivalent to the specified threshold value is reached.

In §7.7, the importance of properly testing the various components of a DSS before putting it to use is discussed, with a focus on the importance of testing military DSSs. Although the aim should be to perform testing in the context of realistic combat scenarios, such testing procedures are expensive in terms of maintenance and operational costs. Simulation methods are therefore typically employed for testing purposes and three simulation methods for testing the constituent parts of a TEWA DSS are mentioned briefly. Furthermore, the importance of evaluating the performance of a TEWA system as a whole (*i.e.* in an integrated fashion within the context of an ADC environment) is discussed in §7.8. A simulation environment is suggested for achieving this and the performance evaluation framework of Truter [187] is proposed for the evaluation of a TEWA DSS as part of an integrated ADC environment. The chapter finally closes in §7.9 with a brief summary of its contents.

7.1 Level of automation of the system

The automation of DSSs within the military domain is a widely discussed subject with many experts arguing that automation is typically employed in the designs of various military DSS in order to improve performance and reduce FCO workload — in essence, that automation is employed to achieve a faster OPTEMPO of the FCO's OODA loop. On the other hand, there are also experts who argue that when the workload experienced by an FCO is at its highest, an automated system is often of least assistance, since such systems can typically only handle routine tasks [67].

There are a number of ways in which a system may be automated. According to Endsley [68], the five main levels of automation include (1) manual automation (*i.e.* the task is executed manually by the FCO), (2) advisory automation (*i.e.* a computer programme suggests various alternatives to the FCO from which he may choose one to implement himself), (3) consensual (*i.e.* a computer programme suggests the best alternative and implements the option if the FCO gives consent), (4) monitored (*i.e.* a computer programme generates alternatives and automatically implements the best alternative, unless vetoed by the FCO), and (5) fully automated (*i.e.* a computer programme executes the entire task automatically without any human interaction).

In the context of a South African GBAD environment, doctrinal rules require that there be an operator to execute fire control [159]. This implies that a WS-threat assignment proposal by the TEWA system will only be executed once the FCO *manually* sends an engagement order to the WS in question. The FCO will, of course, only send the engagement order if he is satisfied with the results provided by the TEWA system. The rationale behind the TEWA system providing real-time DS is that it should not replace the FCO nor should it contradict the decisions that the FCO would have made in the absence of a TEWA system. The aim is that TEWA results should rather assist FCOs in their decisions by the execution of complex computations. Furthermore, it is vital that the FCO should have a high level of confidence in the results presented to him by the TEWA system, for otherwise he may loose trust in the system, choosing rather to make his decisions based solely on doctrinal rules, training and knowledge.

7.2 The quality and quantity of TEWA related input data

A TEWA DSS will only be able to perform to its full potential if sufficient quality and quantity of input data are available. These data, obtained from sensor systems and intelligence reports as described in §2.5.1, should be analysed and preprocessed thoroughly in order to provide high-quality data to the TEWA DSS and ultimately create an enhanced state of SA for the FCO. A number of ways in which the quantity and quality of these input data may be improved are outlined in this section.

The scope of this dissertation is limited to only include fixed wing aircraft. It may, however, be beneficial to expand the scope of the aerial threats considered by the WA subsystem proposed in this dissertation by additionally including models for other platform types, such as rotary wing aircraft [162, 198]. By expanding the range of platform types, a more robust TE subsystem may be obtained, which may, in turn, result in an improvement in the quality and quantity of TE-related input data.

Furthermore, the identification of influential measured and derived aircraft attributes obtainable from sensors and intelligence reports may also result in a more accurate and reliable classification of aircraft, as well as a more appropriate estimation of the level of threat posed by aircraft. However, the availability and classification of such data are typically restricted [162]. If such information were available, existing data mining procedures could be employed for extracting significant measured attributes or discovering derived attributes from the data which influence aircraft threat values significantly. If the required data were available, *discovery-orientated* data mining techniques could be used to identify recurring patterns in the data to achieve this.

7.3 Design of an effective HMI

The objective of a TEWA system is ultimately the provision of high-quality TEWA solution suggestions at any given time stage. It is, however, important to provide the FCO with only the information that he needs rather than providing him with excessive amounts of information which may overwhelm him and may compromise his ability to make effective WA decisions. On the other hand, a sufficient store of information should be available in case the FCO may wish to access more detailed information than that provided by the TEWA system in order to motivate decisions. When designing a TEWA HMI display, careful consideration should therefore be given as to what information is deemed important to provide to the FCO. Gruhn [79] suggests that an effective HMI should be based on a user-centered design which integrates information in ways that fit the tasks and needs of the user. This implies that the FCO should ideally be included in the design of an effective HMI and that he should be able to configure or customise the HMI during the pre-deployment stages of a mission (or even during a mission if the need arises).

One way of minimising clutter when designing an HMI is to hide excess information by employing pop-up windows on the HMI display screen. The FCO may then use a computer mouse to hover over a particular solution in objective space, resulting in the opening of a pop-up window which displays the corresponding solution in decision space (*i.e.* the proposed assignment list and appropriate first and last times to fire in each case, as well as the various objective function values). For example, in the case where a list or graph of approximately Pareto optimal solutions (in multiple-objective space) is presented to the FCO, a pop-up window may display the actual assignments of WSs to threats (in solution space) when hovering over one of the solutions. When the FCO then chooses one of these solutions, the progress of unfolding assignments may then be displayed on the screen over time.

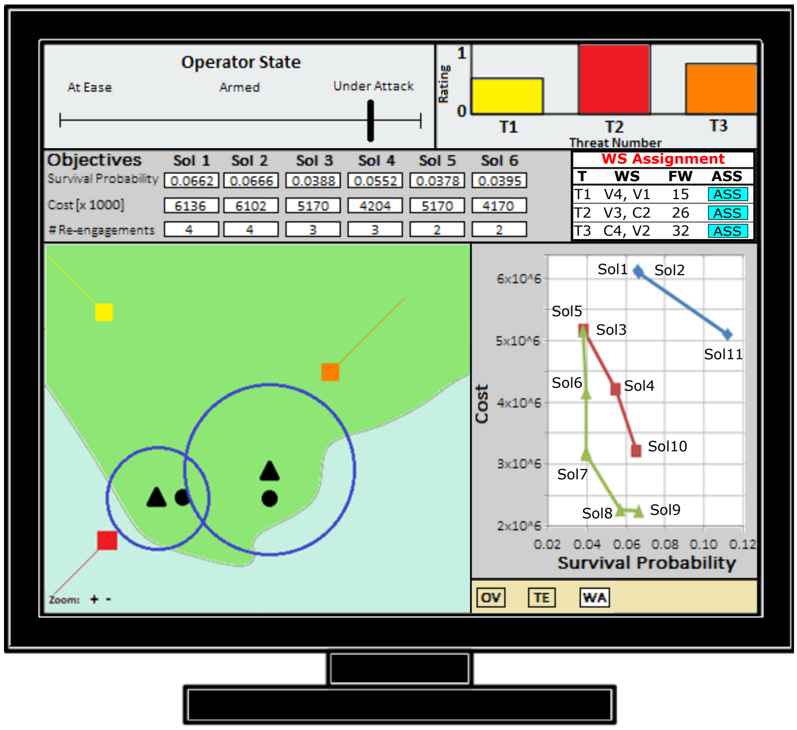


Figure 7.1: *High-detail WA DS screen of a TEWA HMI design by Heuer [85].*

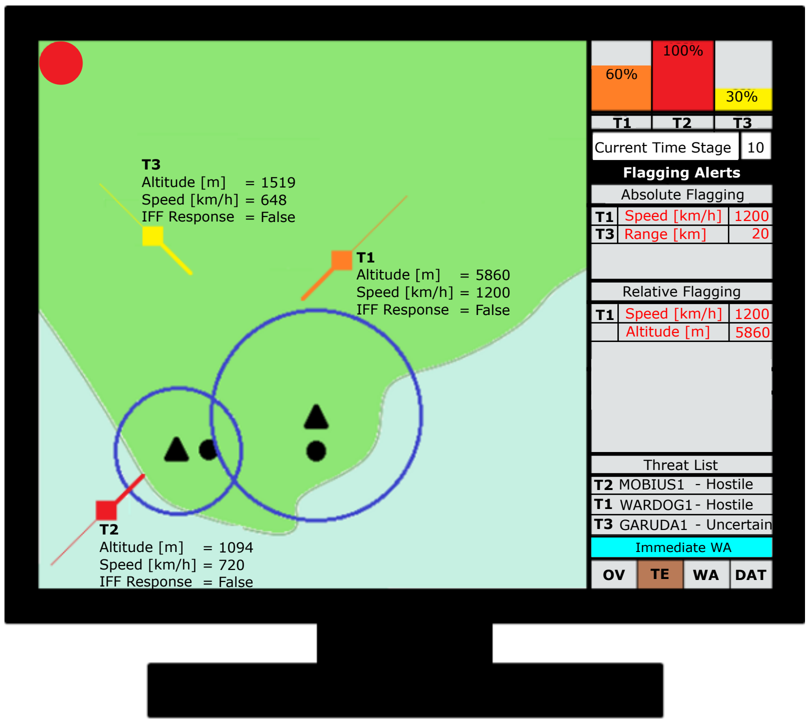


Figure 7.2: *Low-detail TE DS screen of a TEWA HMI design by Heuer [85].*

In 2016, Heuer [85] investigated the design of an effective TEWA HMI as part of a final year engineering research project under the supervision of the author. He conducted interviews with a military expert and an expert in the field of TE subsystem design in order to obtain real-life practical advice on important elements to include in the effective design of a TEWA HMI. He subsequently put forward a suite of ergonomically suitable and uncluttered HMIs for use in a TEWA DSS. These HMIs may serve as a first-order stepping stone for further HMI designs which may be used in the implementation of a TEWA DSS.

One of Heuer's HMI layout designs for WA DS is presented in Figure 7.1. In this layout, a tri-objective WAM has been configured for use by the FCO and the Pareto optimal WA solutions are presented to the FCO graphically in the bottom right-hand portion of the screen. The majority of the screen is, however, reserved for providing the FCO with an air picture aimed at enhancing his SA. Figure 7.2 contains a lower level of detail in the same HMI layout design in which aircraft attribute values, classifications and TEM results are visible.

7.4 Overwhelming the FCO with information

As mentioned, it is important not to overwhelm the FCO with information when providing WA DS during combat situations, since unnecessary information may cause confusion on the part of the FCO when important WA engagement decisions have to be made under severely stressful conditions within very short timeframes. It is therefore proposed that during the pre-deployment stages of a mission, the FCO uses a decision tree, such as the one shown in Figure 7.3, to configure the actual WAMs for inclusion in the system, so as to achieve personally customised DS.

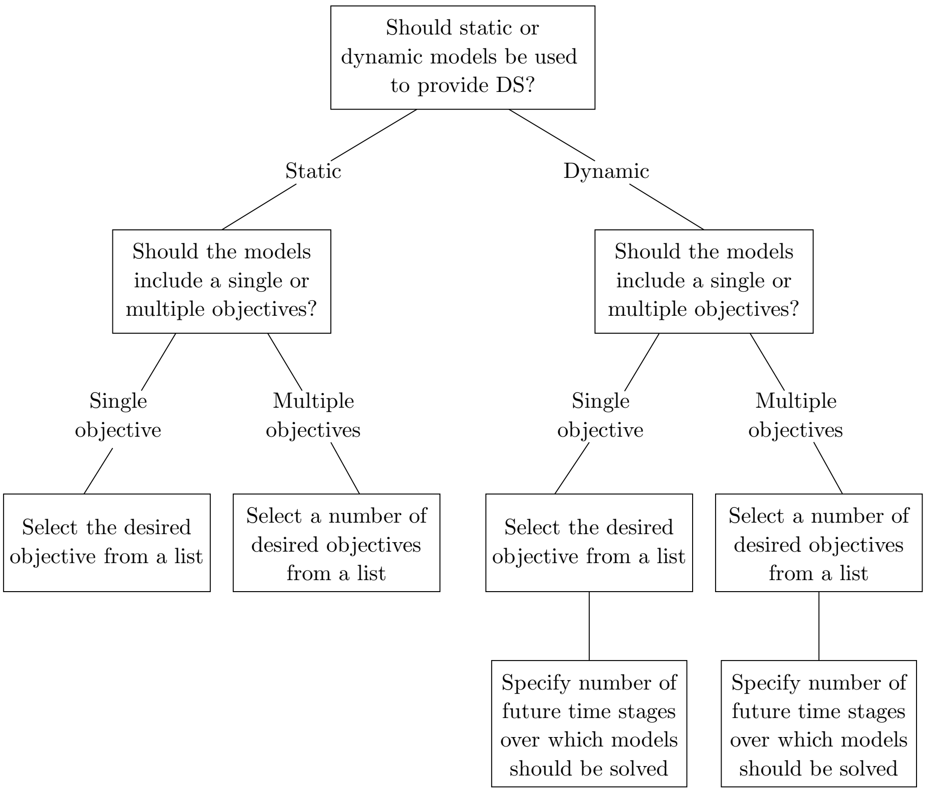


FIGURE 7.3: *Decision tree for aiding an FCO in the configuration of WAMs for inclusion in the WAM component in Figure 6.2.*

In particular, it is proposed that the FCO should be able to decide whether the models to be included in the WAM component should be of a static or dynamic nature first, after which he should then be able to pursue a single WA objective or multiple WA objectives in the models. The FCO should then be able to configure one or more WAMs (from the same class) by choosing objectives from a list of possible objectives. The objectives available to the FCO to choose from should be dictated by the availability and quality of system input data. The FCO should be able to perform (and alter) these configuration decisions easily (*e.g.* by means of radio buttons via the HMI). A default WAM should also be specified during the design phase of the DSS in order to accommodate the case where the FCO does not choose any WAM.

In the event of the FCO choosing to include dynamic WAMs in the WAM component, it is further proposed that the FCO should be able to specify the number of time stages to include in the prediction window of the threats. A default number of such time stages should also be set during the design phase of the DSS so as to accommodate the case where the FCO does not specify a prediction window length. The implication that the prediction window has on the WAM component is that, for the static classes of WAMs, the WA problem is solved separately for each stage in the specified prediction window, taking into account the successful elimination of threats during previous stages, whereas for the dynamic class of WAMs, the WA problem is solved once-off over the entire prediction window remainder.

7.5 Incorporating FCO preferences and biases

Another implementation suggestion involves the number of solutions collected by the WASS component of §6.6 for presentation to the FCO from which he should choose one for implementation purposes. Care should be taken in respect of the number of solutions to present to the FCO since presenting too many approximately Pareto optimal solutions may cause indecision on the part of the FCO when picking one of these solutions. It is suggested that the HMI through which the WA recommendations are communicated to the FCO should be configured in such a way that the FCO is presented with only the necessary information actually required — as mentioned, the FCO should also have the option to view any additional WA-related information, should he choose to.

One way of reducing the number of solutions presented by the DSS to the FCO is to employ operator-specified threshold values for each objective function in such a way that the system automatically filters out solutions that exceed these threshold values. This notion is illustrated in Figure 7.4 within the context of a bi-objective optimisation problem in which both objectives are to be minimised. Suppose the FCO specifies the threshold value for objective 2 represented by the line segment AB and the threshold value for objective 1 represented by the line segment CD. The TEWA DSS should then only provide Solutions 2, 3 and 4 to the FCO as real-time DS.

Another way of reducing the number of solutions presented to the FCO, is to eliminate some solutions involving objective function values that are very close to one another in all the objectives (*i.e.* solutions lying very close together in solution space, such as Solutions 3 and 4 in Figure 7.4). The operator may well be indifferent between pairs of solutions that are so close together in solution space, in which case only one solution of each pair may actually be presented to the FCO in order to reduce decision support clutter.

This filtering approach may be achieved in an automated fashion by superimposing an *indifference grid* with an indifference granularity of ϵ_i in the direction of the i-th objective over the solutions in objective space, where the values $\epsilon_1, \ldots, \epsilon_m$ are user-specified parameters. Two solutions residing within the same cell in such a grid may then be considered to be indistin-

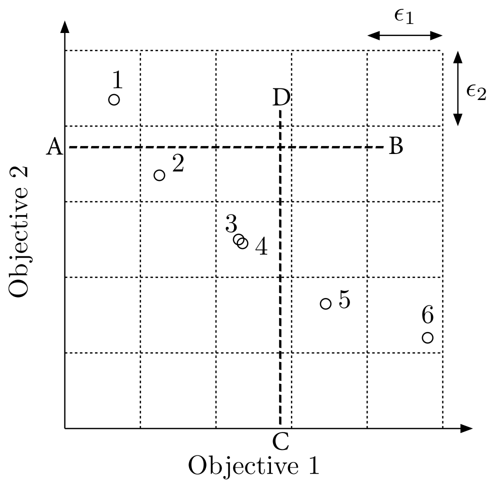


FIGURE 7.4: *An (ϵ_1, ϵ_2)-grid for a bi-objective problem in which both objectives have to be minimised.*

guishable in terms of their solution quality, with the result that only one of these solutions may be presented to the FCO. Consider again the example in Figure 7.4. There are six solutions in the Pareto optimal front labelled $1, \ldots, 6$ in the figure, and Solutions 3 and 4 have objective function values that are very close to one another. Suppose an indifference grid with indifference granularity values of ϵ_1 and ϵ_2 has been specified by the FCO during system configuration, as shown in the figure. Then Solutions 3 and 4 reside within the same grid cell and hence only one of them may be presented as decision support to the FCO (together with Solutions 1, 2, 5 and 6).

The criteria used in the decision as to which solution of a number of alternatives in the same indifference grid cell to present to the FCO should be implemented as a default setting in the TEWA DSS. An example of such a criterion may be to choose the solution which achieves the smallest accumulated survival probability of the threats. The FCO should be able to configure these default selection criteria pre-deployment and should also have the option of overriding the default criteria in real time. Suppose, for the example in Figure 7.4, that the system is configured in such a way that the solution which achieves the smallest value in objective function 1 should be chosen. Solution 3 should then be presented to the FCO from the alternative Solutions 3 and 4.

Care should, of course, be taken in selecting appropriate values for the indifference granularities $\epsilon_1, \ldots, \epsilon_m$ since values that are too large may result in unnecessarily discarding solutions, while values that are too small may result in removing no or very few solutions from the recommended set and hence, again, overwhelming the FCO with information.

Although the indifference grid procedure may seem to be a good approach in attempting to avoid overwhelming the FCO with decision support alternatives, it is sensitive to the way in which the indifference granularity values $\epsilon_1, \ldots, \epsilon_m$ are chosen. Two solutions which lie close to one another in objective space, but lie in different cells in the grid are not considered as lying close to one another in the indifference grid method, and hence, both solutions will be presented to the FCO and, again, may unnecessarily overwhelm him with decision support alternatives.

Another approach that may be considered to resolve this problem is to use an operator-specified distance measure between solutions to filter away unnecessary solutions. The FCO should specify an allowable distance for each of the objectives (similar to the indifference granularity values

$\epsilon_1, \ldots, \epsilon_m$ for each objective in the indifference grid method). Solutions should be compared with one another and if the distance between two solutions exceeds the specified value in all the objectives, both solutions should remain in the set of (approximately) Pareto optimal solutions presented to the FCO. Otherwise, only one of the two solutions should remain in the (approximately) Pareto set of solutions. The FCO should specify the objective function that he prefers and the solution of a pair in close proximity achieving the highest value in this objective function should be provided to the FCO as DS.

A final suggestion for reducing the number of solutions presented to the FCO is to filter out approximately Pareto optimal solutions from the suggested list according to the operator's biases and preferences, by employing a pre-determined FCO utility function.

7.6 Switching of DSS results between consecutive time stages

An undesirable phenomenon which may occur when WS assignment suggestions are reported to the FCO during a combat situation is called *switching*. Switching refers to the excessively rapid changing of WA suggestions presented to the FCO by the TEWA DSS during a small subset of consecutive time stages. This kind of behaviour may be ascribed to small changes in the SSHP-values that WSs are capable of achieving with respect to threats during these time stages. Switching may, of course, cause confusion and compromise the FCO's confidence in the results produced by the TEWA system and may, in turn, lead to the FCO making suboptimal decisions when choosing to rely on his own judgement rather than trusting the seeming indecision of the DSS. The problem of switching may be solved by implementing threshold values in the system in such a way that assignment recommendations are only altered from one time stage to another once variations in the results equivalent to the threshold value are reached (*i.e.* if the two solutions in question are deemed significantly different).

7.7 Testing the components of the system

The importance of properly testing the various components of a system before implementation is often underestimated. Testing involves testing the successful operation of a system under controlled conditions, including normal and abnormal conditions. A number of methods exist in the open literature for testing systems and some of these methods include black box testing (*i.e.* testing the functionality of a system without knowing the internal structural design of the system and without knowing how the system operates internally), white box testing (*i.e.* testing the functionality of a system while knowing the internal structure design of the system and knowing how the system operates internally), unit testing (*i.e.* testing the smallest testable parts of a system in an individual and independent fashion — the units are scrutinised individually for proper operation), failover testing (*i.e.* validating the ability of a system to allocate additional resources and to switch operations to a back-up system in the event of a server failure), *ad-hoc* testing (*i.e.* testing the system only once without any planning or documentation related to the testing procedures), user-acceptance testing (*i.e.* testing a system in which a user is employed to ensure that the system can handle required tasks in real-world scenarios) and exploratory testing (*i.e.* testing a system in a "hands-on" fashion in which testers have personal freedom to test the system with minimum planning), to name but a few [90].

Testing the constituent parts of a TEWA system in the military domain is of vital importance since human lives are at stake when it is implemented in practice. The parts of a TEWA system should therefore not only be tested in isolation (or at unit level), but rather in conjunction with the relevant elements of a fully functional ADC system in the context of diverse combat scenarios. This makes the aforementioned testing procedures difficult to implement due to the high operational and maintenance costs (and risks) involved — certain functional elements, such as fighter aircraft and WS armoury, are typically used in testing, training and maintenance procedures.

Due to the high cost and risks involved in conventional testing procedures, they are typically abandoned in favour of testing by means of simulators which are able to imitate ADC physical elements in controlled military testing environments. According to Mepham [129] there are three overarching categories of simulation systems which may be used for testing ADC elements in a military environment. These systems are constructive simulation (*i.e.* a simulation in which simulated people are involved in operating simulated systems — in this type of simulation, real people are involved in providing certain inputs, but they do not determine any outcomes of the system), virtual simulation (*i.e.* a simulation in which real people are involved in operating simulated systems) and live simulation (*i.e.* a simulation in which real people are involved in operating real systems in real-world scenarios — these scenarios are called simulations since they are not conducted against a real enemy).

7.8 Evaluating the performance of the system as a whole

Although the design of the TE subsystem by Roux and Van Vuuren [162] and the WA subsystem proposed in Chapter 6, seem, in principle, to be able to provide acceptable quality decision support to FCOs, the combined performance of these subsystems has not yet been tested in an integrated manner. Such an integrated testing approach is very important in the development of a TEWA DSS where the various subsystem elements interact with one another to yield the characteristic emergent properties of the TEWA system as a whole. These emergent properties cannot be revealed by following a reductionistic evaluation approach only — involving testing the TE and WA subsystems separately. Adopting such a reductionistic approach may result in a failure to assess whether the TEWA DSS as a whole will function as expected.

One way of evaluating the performance of the integrated system is to consult a military expert or a group of military experts (*e.g.* in the form of a workshop) with a view to evaluate and analyse the results produced by these subsystems based on doctrinal rules and their knowledge and experience. An expert validation of the system may be achieved in this way. This method of evaluating system performance is, however, subjective in nature and may vary if a different military expert or group of military experts is employed in the evaluation procedure. Therefore, a more robust, scientific way of evaluating system performance should also be employed.

Roux [159] suggested that a simulation environment be used to test and evaluate a TEWA system's performance and that testing procedures be performed in an incremental manner. First-order examples of evaluating the performance of TEWA systems in this manner were put forward independently by Kok [107] and by Johansson and Falkman [97]. It is, of course, also important that this method of evaluation be generic in the sense that should future changes be made to any of the components in the TEWA subsystems, the method should be easily adaptable to incorporate these changes and to re-evaluate the overarching system's performance.

In 2016, Truter [187] developed and demonstrated the working of a performance evaluation framework for TEWA systems adopting a simulation approach. He embraced a *system of systems*[1] approach in his design and proposed four metrics for use within the developed simulation

[1]A system of systems approach involves considering a number of small, dispersed, independent systems together

paradigm. These metrics are survivability (*i.e.* a measurement of the ability of WSs to protect DAs from enemy aircraft attacks — the metric is taken as the ratio between the protection value of surviving DAs to the total protection value of all DAs), economy (*i.e.* a measurement accounting for the cost of WS ammunition used for engagement strategies), engagement effectiveness (*i.e.* a measurement of the ability of WSs to successfully cause considerable damage to or destroy high-value threats) and adaptability (*i.e.* a measurement of the ability of engagement strategies to be adaptable in terms of future assignments — the metric measures the propensity of an engagement to maximise the number of times that a WS can be available for future engagements). The framework proposed by Truter [187] may serve as a first-order method for testing TEWA DSS results in an integrated fashion.

7.9 Chapter summary

This chapter was dedicated to a discussion on practical suggestions for implementing a TEWA DSS. Eight suggestions were proposed. The chapter opened in §7.1 with a discussion on the automation of systems in general. Five levels of automation were discussed and the suggestion was made that a TEWA DSS should aim to support and conform to FCO thought when an operator has to make difficult decisions in short time frames, rather than pursuing a fully-automated TEWA DSS. Furthermore, a suggestion was made in §7.2 for improving certain input data in a bid to facilitate higher quality results from the TEWA DSS.

Next, the importance of providing the FCO with only the necessary information that he needs was discussed in §7.3. A number of suggestions for an efficient layout of the HMI was also provided. In §7.4, the focus shifted to a common problem experienced in military combat situations — that of overwhelming FCOs with unnecessary information. This may cause confusion when FCOs have to make important WA engagement decisions. A decision tree was proposed which the FCO may use to configure WAMs during the pre-deployment stages of a mission so as to alleviate the stress experienced during combat situations. Next, in §7.5, four suggestions for incorporating FCO preferences and biases into a TEWA system were discussed in detail with a view to reduce WA alternatives reported as DSS recommendations to an FCO. In §7.6, the undesirable notion of switching was described and a suggestion of incorporating threshold values to counter this problem was highlighted.

The importance of testing the various components of the system was briefly discussed in §7.7 and specific mention was made of the vital importance of thoroughly testing the constituent parts of military DSSs due the high-stake involvement of human lives. In §7.8, the importance of evaluating a TEWA DSS as a whole before implementation was elaborated upon. It was suggested that a TEWA DSS should be evaluated as an integrated system within the context of an ADC environment in order to account for possible emergent behaviour. The use of a simulation environment was suggested for this purpose and the performance evaluation framework of Truter [187] was suggested for testing a TEWA DSS in such a fashion.

as part of a larger, more complex system.

CHAPTER 8

Decision support system validation

Contents

In this chapter, the working of the four WAM prototypes selected for default inclusion in the WAMS component (proposed in §6.5) is illustrated by solving each of the models in the context of a simulated GBAD scenario according to the solution methodologies described in Chapter 5. The chapter opens in §8.1 with a detailed description of the hypothetical, yet realistic, GBAD scenario that forms the context within which the WAMs are solved.

Section 8.2 contains four subsections dedicated to detailed descriptions of the numerical results obtained by each of the models of §6.5.1–§6.5.4. Each subsection is devoted to one of the four WAMs and contains a detailed description of the way in which the relevant solution methodology of §5.2 was implemented to solve the model, as well as a detailed presentation and interpretation of the numerical results obtained. A conventional genetic algorithm (as described in §5.3.2) is used to solve the single-objective static WAM (6.6)–(6.9) and the implementation thereof, as well as the numerical results thus obtained within the aforementioned hypothetical GBAD scenario, are presented in §8.2.1. This is followed in §8.2.2 by a similarly structured description of the implementation of the NSGA II (as described in §5.4.2) used to solve the multi-objective static WAM (6.10)–(6.12) and the numerical results obtained for the same GBAD scenario. The method of simulated annealing (as described in §5.3.1) is used to solve the single-objective dynamic WA model (6.14)–(6.23) and the numerical results obtained are presented in §8.2.3. Finally, an extension of the NSGA II (as discussed in §5.4.2) is used to solve the multi-objective dynamic WA model (6.24)–(6.26) and the results thus obtained are presented in §8.2.4.

The results obtained by each of the WAMs are validated in §8.3. Three validation methods are employed for this purpose. In §8.3.1, a face validation is conducted, resulting in a subjective opinion on the realism and quality of the results obtained in §8.2. The second method of validation, applied in §8.3.2, involves a random benchmark validation in which thirty candidate solutions are generated randomly for each WAM, ascertaining that the results returned by the algorithms are, in fact, (significantly) superior to or dominate those generated randomly. The final method of validation is carried out in §8.3.3 and entails an analysis of the feedback obtained during consultations with two military experts in which they were first asked to solve (by hand) each of the WAMs in the context of the simulated scenario of §8.1 and then to comment on the suitability and practicality of the results returned by the algorithms employed to solve each of the four WAMs. The chapter finally closes in §8.4 with a brief summary of the chapter contents.

8.1 A simulated ground-based air defence scenario

The GBAD test scenario adopted for validation purposes in this chapter was originally designed by Roux [159] and mimics a real-life GBAD deployment in which two DAs (called DA_1 and DA_2) are afforded protection by two sets of GBAD WSs deployed in concentric circles around DA_1. The first set comprises four CIWSs, labelled C_1, C_2, C_3 and C_4 in Figure 8.1, which are spaced evenly and deployed at approximately half the distance of their ranges from DA_1 (they are deployed closest to the DAs). The second set comprises eight VSHORAD WSs, labelled V_1, V_2, ..., V_8, which are deployed further away from the DAs in such a way that their primary fire arcs cross. A top view of the locations of the DAs and the deployment of the two sets of WSs is shown in Figure 8.1.

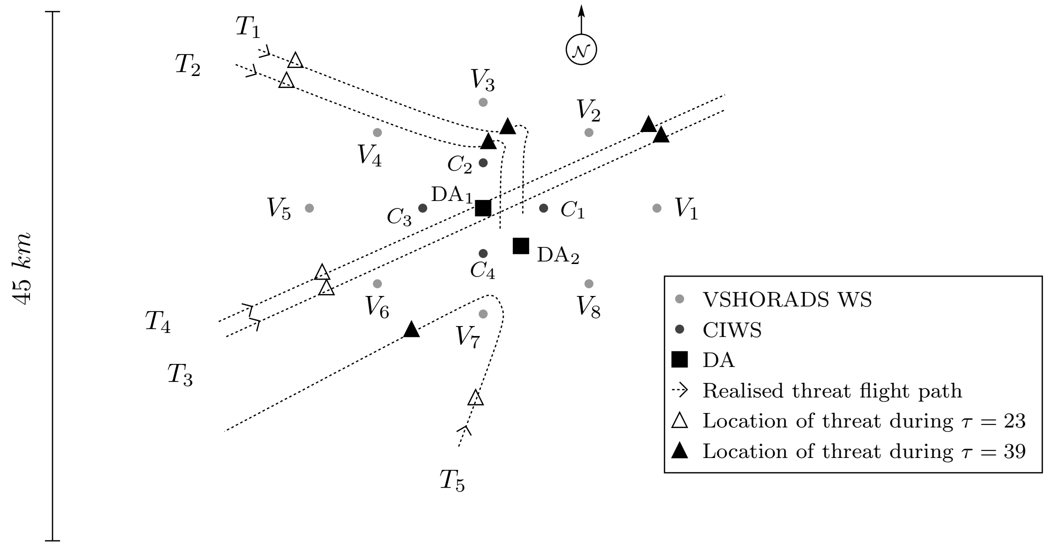


FIGURE 8.1: *Top view of a simulated GBAD scenario [158].*

An attack scenario is considered in which five aerial threats, labelled T_1, T_2, T_3, T_4 and T_5 in Figure 8.1, are detected in the defended airspace. These threats enter the defended airspace in three groupings, offset at different time stages. The first group comprises threats T_1 and T_2, and they approach from a north-westerly direction at an average speed of 251 km/h and at an altitude of 430 m. They aim to attack DA_1 by executing a *combat hump dive attack technique*, as described in §2.5.1. The second group comprises threats T_3 and T_4, and they approach from a south-westerly direction during a later stage at an average speed of 246 km/h and at an altitude of 4 400 m. They are considered as fly-overs and act as decoys to the system in the sense that they do not aim to attack any of the DAs. Finally, the third group consists of threat T_5 only which enters from a southerly direction during the same stage as the second group, but travels at an average speed of 249 km/h and at an altitude of 90 m. It aims to attack DA_2 by executing a *toss bomb attack technique*, also described in §2.5.1. A top view of the flight profiles[1] executed by the five threats is also shown in Figure 8.1.

The time continuum of the scenario is partitioned into 120 time stages of four seconds each. During each of these time stages the future flight path of each threat is predicted for the remaining stages by employing a straight line prediction model, as described in §6.4.1 (*i.e.* it is predicted that each threat will continue travelling in a straight line, while maintaining its current speed, direction and altitude). The efficiency values achievable by the WSs with respect to engaging the threats during each of these time stages are discretised into a 3-dimensional EEM (as described in §6.4). The EEM used in the scenario is not filtered for any environmental conditions such as inclement weather or terrain obstructions (*i.e.* ideal engagement conditions and a flat earth are assumed for simplicity). The priority values associated with eliminating each threat (typically obtained from a TE subsystem) during each of the 120 time stages, as well as the future predicted EEM values for each time stage, may be found online [119].

Since the static WAMs described in §6.5.1 and §6.5.2 involve WA decisions during a single time stage at each time instant only, these models are solved here for a single time stage for the sake of brevity. Time stage 39 is chosen for illustrative purposes during which to solve the single objective static WAM (6.6)–(6.7) and the multi-objective static WAM (6.10)–(6.12), because of its diversity and the close proximity of the threat locations to the DAs when $\tau = 39$. The approximate locations of threats during this time stage are denoted by black triangles in Figure 8.1. The EEM and the threat list corresponding to this time stage are shown in Table 8.1.

	Threat				
WS	T_1	T_2	T_3	T_4	T_5
V_1	0	0	0.1	0.1	0
V_2	0.7	0	0.5	0.5	0
V_3	0.1	0	0	0	0
V_4	0	0	0	0	0
V_5	0	0	0	0	0
V_6	0	0	0	0	0.1
V_7	0	0	0	0	0.5
V_8	0	0	0	0	0
C_1	0.2	0.4	0	0	0
C_2	0.5	0.9	0	0	0
C_3	0	0	0	0	0
C_4	0	0	0	0	0

(a)

Threat	V_j
T_1	0.99
T_2	1.00
T_3	0.76
T_4	0.74
T_5	0.50

(b)

TABLE 8.1: *(a) An extract from the EEM and (b) the threat list entries corresponding to time stage $\tau = 39$ for the hypothetical GBAD scenario depicted in Figure 8.1.*

[1]These flight paths represent the actual locations of the threats as a function of time, as opposed to the predicted flight paths of the threats estimated by the FPP module from the EEM component. Neither the former flight paths nor the intended targets of the aerial threats are, of course, known in advance by the ground force TEWA system.

Since the dynamic WAMs described in §6.5.3 and §6.5.4 involve WA decisions over a pre-specified number of future time stages at once, the entire predicted future time continuum in the scenario is considered in these models. Time stage 23 in the scenario of Figure 8.1 is chosen as the time stage during which to solve the single-objective dynamic WAM (6.14)–(6.23) and the multi-objective dynamic WAM (6.24)–(6.26) over the predicted remaining future time stages due to the diversity in the locations of the threats during this time stage. The approximate locations of the threats during this time stage are illustrated graphically by open triangles in Figure 8.1. In addition, priority values associated with eliminating each threat during time stage 23 are given in Table 8.2. Furthermore, it is assumed that the setup time of each WS is three time stages, that the minimum length of a FW is four time stages and that the number of time stages included in the fixed-mean calculation in (6.13) are fixed to the minimum length of a FW. The earliest and latest times during which a WS may engage a threat (*i.e.* the values of e_{ij} and ℓ_{ij}) for time stage 23 are given in Table 8.3.

Threat	V_j
T_1	0.20
T_2	0.22
T_3	1.00
T_4	1.00
T_5	0.30

TABLE 8.2: *The threat values corresponding to time stage $\tau = 23$ for the remaining time stages of the hypothetical GBAD scenario depicted in Figure 8.1.*

	Threat				
WS	T_1	T_2	T_3	T_4	T_5
V_1	72	72	32	33	86
V_2	0	0	35	33	98
V_3	34	36	0	0	0
V_4	19	18	0	0	0
V_5	0	0	0	0	0
V_6	0	0	0	0	0
V_7	0	0	0	0	34
V_8	83	80	0	0	57
C_1	66	64	19	17	85
C_2	53	52	0	19	0
C_3	0	0	3	1	0
C_4	0	0	5	4	0

(a) Earliest engagement stages

	Threat				
WS	T_1	T_2	T_3	T_4	T_5
V_1	102	100	63	60	110
V_2	0	0	67	67	116
V_3	56	54	0	0	0
V_4	50	49	0	0	0
V_5	0	0	0	0	0
V_6	0	0	0	0	0
V_7	0	0	0	0	73
V_8	103	105	0	0	85
C_1	81	81	40	37	103
C_2	70	69	0	28	0
C_3	0	0	19	19	0
C_4	0	0	24	23	0

(b) Latest engagement stages

TABLE 8.3: *(a) The earliest and (b) the latest time stages during which WSs may engage threats for engagement decisions when viewed from the perspective of time stage 23 for the remaining time stages of the hypothetical GBAD scenario depicted in Figure 8.1.*

The multi-objective WAMs of §6.5.2 and §6.5.4 also require the levels of ammunition available to each WS during each time stage of the scenario in Figure 8.1 as well as the unit cost of the WS ammunition used in the scenario. The levels of ammunition available for use by each WS during time stage 23 are shown in Table 8.4 and the levels of ammunition available for use by each WS during time stage 39 are given in Table 8.5.

WS	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	C_1	C_2	C_3	C_4
A_i	7	6	5	5	5	7	8	9	5	6	4	6

TABLE 8.4: *The levels of ammunition available to each WS at time stage 23 of the hypothetical GBAD scenario depicted in Figure 8.1.*

Finally, the cost of the ammunition (in South African Rand value) for a single burst of a CIWS is assumed to be $R\,34\,000$ and the cost of a single missile (in South African Rand value) for a VSHORAD WS is assumed to be $R\,1\,000\,000$. These values were realistic approximations of the costs of a burst of ammunition for a 35 mm cannon and an Umkhonto missile, respectively, at the time of writing.

WS	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	C_1	C_2	C_3	C_4
A_i	4	4	5	5	5	4	3	5	3	4	5	5

TABLE 8.5: *The levels of ammunition available to each WS at time stage 39 of the hypothetical GBAD scenario depicted in Figure 8.1.*

8.2 Numerical results

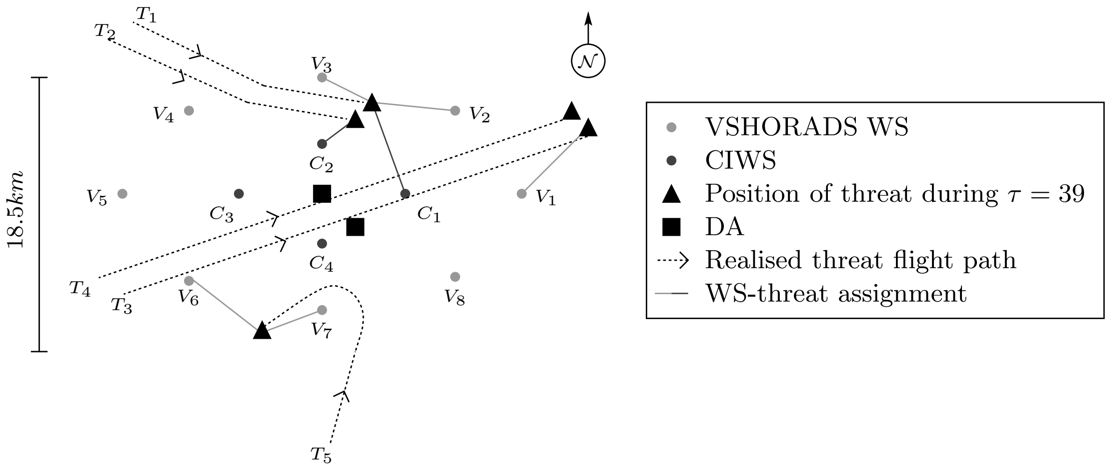
This section contains detailed descriptions of the way in which each of the WAM prototypes of Chapter 6 was implemented within the context of the simulated GBAD scenario of Figure 8.1. Numerical results obtained are also presented and discussed briefly. The section contains four subsections devoted to discussions on the implementations of the various solution methodologies for each of the WAMs, respectively, as well as the numerical results returned by these WAMs.

8.2.1 Numerical results for the single-objective static model prototype

Due to the nonconvexity of the models in Chapter 6, it was decided to solve the single-objective static WAM (6.6)–(6.9) approximately by means of a genetic algorithm, as described in §5.3.2. Candidate solutions to this model were encoded as binary matrices having dimensions $m(\tau) \times n(\tau)$ (recall that $m(\tau)$ represents the number of WSs during stage τ and that $n(\tau)$ represents the number of threats in the system during stage τ), where a value of 1 in row i and column j of the matrix indicates an assignment of WS i to threat j, while a value of 0 indicates no assignment. The algorithm was initiated by generating an initial population of 200 candidate solutions randomly. The current population of solutions was then used to populate the next generation of solutions during each iteration. This was achieved by implementing a tournament selection procedure, as described in §5.3.2, with pool and tour sizes of 2 and 100, respectively, in order to select parent solutions in an elitist manner from the current generation and to create offspring solutions by applying conventional crossover and mutation operators (as described in §5.3.2) to the selected parent solutions. A mutation probability of 0.025 was employed and the algorithm was iterated until a stopping criterion of having performed 400 iterations in total was satisfied. Finally, the value $\kappa = 3$ was assumed (that is, each aerial threat may be engaged by at most three WSs).

Threat	Threat Priority $V_j(\tau)$	WSs assigned	Survival values $q_{ij}(\tau)$	Prioritised survival probability
T_1	0.99	C_1, V_2, V_3	$0.8, 0.3, 0.9$	$0.99 \times 0.8 \times 0.3 \times 0.9$
T_2	1.00	C_2	0.1	1.00×0.1
T_3	0.76	V_1	0.9	0.76×0.9
T_4	0.74	None	—	0.74×1
T_5	0.50	V_6, V_7	$0.9, 0.5$	$0.5 \times 0.9 \times 0.5$
			Total:	1.9628

(a)



(b)

FIGURE 8.2: *(a) WS to threat assignment list and (b) top view graphical illustration of the assignments of WSs to threats proposed by the single-objective static WA model (6.6)–(6.9) during stage $\tau = 39$ of the simulated GBAD scenario described in §8.1.*

The algorithm was implemented on an Intel Core i7-4770 processor with 8GB of random access memory operating at 3.40 GHz, which was able to solve the model almost instantaneously in the manner described above. The best solution obtained for the single-objective static WA model (6.6)–(6.9) is shown in Figure 8.2. In this solution, WSs V_2, V_3, and C_1 are assigned to threat T_1, WS C_2 is assigned to threat T_2, WS V_1 is assigned to threat T_3, and WSs V_6 and V_7 are assigned to threat T_5. Note that threat T_4 is assigned no WSs. This assignment yields a threat priority-weighted accumulated survival probability value for the threats of 1.9628 at an assignment cost of $R\,5\,068\,000$.

8.2.2 Numerical results for the multi-objective static model prototype

The multi-objective static WA model (6.10)–(6.12) was again solved for $\kappa = 3$ and for time stage $\tau = 39$ of the simulated GBAD scenario depicted in Figure 8.1, but this time using the NSGA II of Agarwal *et al.* [2], as described in §5.4.2. Solutions were again encoded as binary matrices, as with the implementation of the single-objective static WAM (6.6)–(6.9). The current population of solutions were ranked based on the nondominated statuses of solutions and partitioned into nondominated fronts of solutions based on their rank values. This was achieved by implementing the FNSA of Deb *et al.* [51], as described in §4.7.

The algorithm was initialised by randomly generating an initial population of 200 candidate solutions. The current population was used during each iteration to generate an intermediate population by adopting a tournament selection procedure as described in §5.3.2, again with pool and tour sizes of 2 and 100, respectively, in order to select parent solutions and performing conventional crossover and mutation operators on the selected parent solutions, as described in §5.3.2. The solutions from the current and intermediate populations were then combined to form a larger population. The larger population was ranked and sorted, again using the FNSA [51], and the next generation of candidate solutions was populated by first including the solutions in nondominated front $\mathcal{F}_1$, followed by the solutions in nondominated front $\mathcal{F}_2$, and so on until the size of the initial population was reached, as described in §5.4.2. The algorithm was again executed until a stopping criterion of 400 iterations was reached. The algorithm was able to solve the model in 13.4 seconds on an Intel Core i7-4770 processor with 8GB of random access memory operating at 3.40 GHz.

Thirteen approximately Pareto optimal solutions were thus obtained. These solutions are listed in Table 8.6, which contains the WS-threat pair assignments as well as the corresponding objective function values of the solutions (the threat priority-weighted accumulated survival probability, the ammunition cost incurred by the assignments and the number of units of ammunition available at the least re-engagable WS *after* the assignment). The assignment for each WS-threat pair is given as the WS(s), followed by a right-arrow and the threat to which it is assigned (*e.g.* $V_3, C_1 \to T_1$ implies that WSs V_3 and C_1 should engage threat T_1).

Sol	WS-threat pair assignments	Obj 1	Obj 2	Obj 3
1	$C_1,\ V_2,\ V_3 \to T_1;\ C_2 \to T_2;\ V_1 \to T_3;\ V_6, V_7 \to T_5$	1.9628	$R\,5\,068\,000$	2
2	$C_1,\ V_2,\ V_3 \to T_1;\ C_2 \to T_2;\ V_1 \to T_3;\ V_7 \to T_5$	1.9878	$R\,4\,068\,000$	2
3	$C_1,\ V_2 \to T_1;\ C_2 \to T_2;\ V_1 \to T_3;\ V_7 \to T_5$	2.0116	$R\,3\,068\,000$	2
4	$V_2 \to T_1;\ C_2 \to T_2;\ V_1 \to T_3;\ V_7 \to T_5$	2.0710	$R\,3\,034\,000$	2
5	$V_2 \to T_1;\ C_1, C_2 \to T_2;\ V_7 \to T_5$	2.1070	$R\,2\,068\,000$	2
6	$V_2 \to T_1;\ C_2 \to T_2;\ V_7 \to T_5$	2.1470	$R\,2\,034\,000$	2
7	$C_1, V_2 \to T_1;\ C_2 \to T_2;$	2.3376	$R\,1\,068\,000$	2
8	$C_1 \to T_1;\ C_2 \to T_2$	2.8920	$R\,68\,000$	2
9	$V_2, V_3 \to T_1;\ C_2 \to T_2;\ V_1 \to T_3;\ V_6 \to T_5$	2.2413	$R\,4\,034\,000$	3
10	$V_2 \to T_1;\ C_2 \to T_2;\ V_1 \to T_3;\ V_6 \to T_5$	2.2710	$R\,3\,034\,000$	3
11	$V_2 \to T_1;\ C_2 \to T_2;\ V_1 \to T_3$	2.3210	$R\,2\,034\,000$	3
12	$V_2 \to T_1;\ C_2 \to T_2;$	2.3970	$R\,1\,034\,000$	3
13	$C_2 \to T_2$	3.0900	$R\,34\,000$	3

TABLE 8.6: *The set of approximately Pareto optimal solutions together with their WS-threat pair assignments and their respective objective function values for the multi-objective static WAM (6.10)–(6.12) at time stage $\tau = 39$ within the context of the GBAD scenario depicted in Figure 8.1. Obj 1 represents the accumulated threat-priority weighted survival objective function values, Obj 2 represents the ammunition cost objective function values and Obj 3 represents the least re-engagable WS objective function values. The assignment pairs are given as the WS, followed by a right-arrow indicating to which threat the WS should be assigned.*

The thirteen approximately Pareto optimal solutions are also presented graphically in objective function space in Figure 8.3. These solutions achieve objective function values in the priority-weighted accumulated survival probabilities objective (6.10) and assignment cost objective (6.11) as indicated on the horizontal and vertical axes of the figure, respectively. The objective function values achieved for the re-engagability objective (6.12) may be interpreted as follows. The solutions are numbered $1, \ldots, 13$. For solutions $1, \ldots, 8$ (represented by solid triangles) the least re-engagable WS can engage twice after the assignment, while for solutions $9, \ldots, 13$ (represented by solid squares) the least re-engagable WS can engage three times after the assignment. Figure 8.3 clearly illustrates how much of one objective is compromised for a gain in another — the trade-offs between the objective function values may be observed when moving from one solution to an adjacent solution.

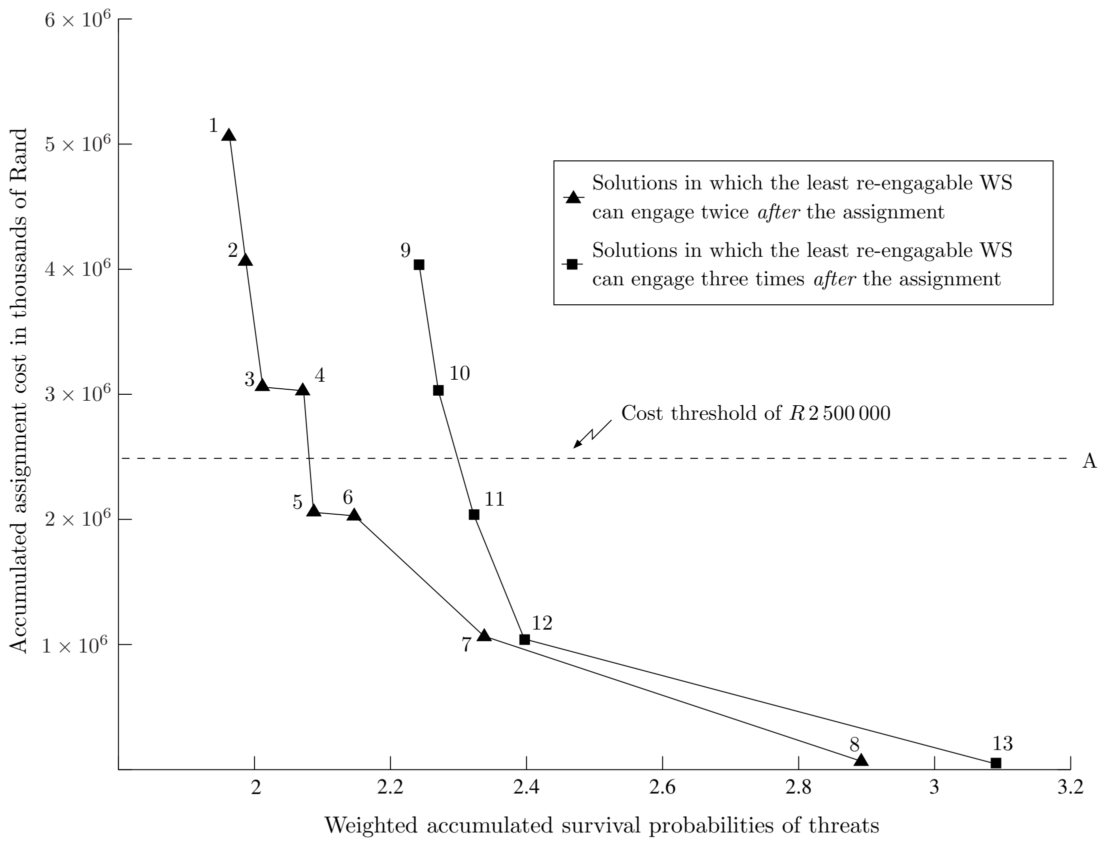


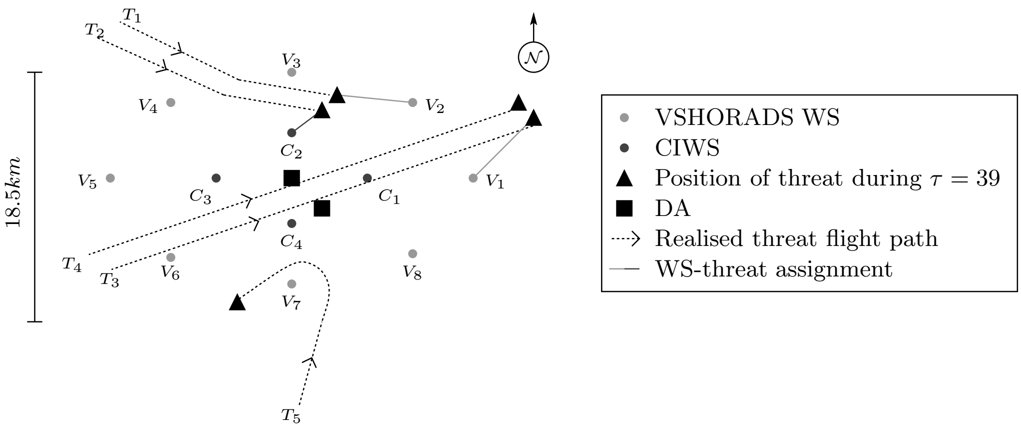
FIGURE 8.3: *Approximately Pareto optimal solutions to the multi-objective static WA model (6.10)–(6.12) in objective function space within the context of the GBAD scenario of §8.1.*

The solutions in Table 8.6 and Figure 8.3 may be presented to the FCO after which he may choose one of them for implementation (based on his subjective preference and expertise). Suppose, for illustrative purposes, that an assignment is sought in which the least re-engagable WS can engage at least three times after the engagement and that a restriction of $R\,2\,500\,000$ is placed on the accumulated assignment cost (as illustrated by the horizontal line labelled A in Figure 8.3). The best solution satisfying both these criteria is Solution 11 in the figure. In this solution, WS V_2 is assigned to threat T_1, WS C_2 is assigned to threat T_2, and WS V_1 is assigned to threat T_3. No WSs are assigned to threats T_4 or T_5. These assignments yield a priority-weighted accumulated survival probability value for the threats of 2.321 at an assignment cost of $R\,2\,034\,000$. A graphical illustration of the assignments embodied in Solution 11 is shown in Figure 8.4.

Note that the solution to the single-objective static WA model obtained in the previous section (see Figure 8.2) is also (approximately) Pareto-optimal for the multi-objective, static WA model; this is Solution 1 in Figure 8.3.

Threat	Threat Priority $V_j(\tau)$	WSs assigned	Survival values $q_{ij}(\tau)$	Prioritised survival probability	Accumulated cost in Rand
T_1	0.99	V_2	0.3	0.99×0.3	1 000 000
T_2	1.00	C_2	0.1	1.00×0.1	34 000
T_3	0.76	V_1	0.9	0.76×0.9	1 000 000
T_4	0.74	None	—	0.74×1	—
T_5	0.50	None	—	0.5×1	—
			Totals:	2.321	2 034 000

(a)



(b)

FIGURE 8.4: *(a) WS to threat assignment list and (b) top view graphical illustration of the assignments of WSs to threats in Solution 11 in Figure 8.3 to the multi-objective static WA model (6.10)–(6.12).*

8.2.3 Numerical results for the single-objective dynamic model prototype

Van der Merwe and Van Vuuren [196] experimented with solving the single-objective dynamic WAM (6.14)–(6.23) by means of a conventional simulated annealing algorithm, as described in §5.3.1. The model requires the survival probability values of the threats for the FWs in Figure 8.3 and these values were calculated using the fixed-mean approach described in §6.5.3. The calculated fixed-mean values for each FW are shown in Table 8.7. A value $\kappa = 2$ was assumed when solving the model. Solutions to the single-objective, dynamic WA model (6.14)–(6.23) were encoded as integer matrices of dimensions $m(\tau) \times n(\tau)$ this time. If a WS is assigned to a threat, the corresponding entry in the solution matrix is an integer value representing the FTTF for the WS-threat pair in question. Otherwise, the entry is zero, indicating that the WS-threat pair is not assigned. The algorithm was initiated by generating an initial feasible solution in a greedy fashion. This solution was stored as the *incumbent solution*. During each iteration of the algorithm, a neighbourhood move operator performed a number of intelligent, random perturbations in respect of the current solution in order to obtain neighbouring solutions. The temperature which controls the randomness of the search in the algorithm was lowered by implementing a geometric cooling schedule (as described §5.3.1) with an initial temperature of 0.1 and a cooling factor of 0.01. The algorithm was iterated for nine iterations and the incumbent solution was reported as an approximate solution to the single-objective dynamic WAM (6.14)–(6.23).

	Threat				
WS	T_1	T_2	T_3	T_4	T_5
V_1	0.451	0.451	0.367	0.369	0.481
V_2	1.000	1.000	0.374	0.369	0.435
V_3	0.371	0.376	1.000	1.000	1.000
V_4	0.340	0.338	1.000	1.000	1.000
V_5	1.000	1.000	1.000	1.000	1.000
V_6	1.000	1.000	0.701	1.000	1.000
V_7	1.000	1.000	1.000	1.000	0.371
V_8	0.474	0.468	1.000	1.000	0.420
C_1	0.328	0.201	0.057	0.052	0.460
C_2	0.192	0.156	1.000	0.560	1.000
C_3	1.000	1.000	0.390	0.238	1.000
C_4	1.000	1.000	0.075	0.207	1.000

TABLE 8.7: *Fixed-mean survival values of threats for FWs in Table 8.3*

The algorithm was also implemented on an Intel Core i7-4770 processor with 8GB of random access memory operating at 3.40 GHz and was able to solve the model (approximately) in 0.002 seconds. The best solution obtained for the WA model (6.14)–(6.23) is given in Table 8.8. The following assignments are proposed: WSs V_3 and C_2 are assigned to Threat T_1 during time stages 34 and 66, respectively, while WSs C_1 and C_2 are assigned to threat T_2 during time stages 64 and 52, respectively. Similarly, WSs V_1 and C_1 are assigned to threat T_3 during time stages 32 and 19, respectively, while WSs C_3 and C_4 are assigned to threat T_4 during time stages 1 and 4, respectively. Finally, WSs V_8 and C_1 are assigned to threat T_5 during time stages 57 and 85, respectively. These assignments yield a priority-weighted accumulated survival probability value for the threats of 0.1493 (at an accumulated assignment cost of $R\,3\,238\,000$). The proposed assignments are also illustrated graphically in Figure 8.5. The dotted lines in the figure represent the predicted flight paths of the threats, the bold lines represent the FWs for each threat and the thin solid lines represent WS-threat assignments. The underlying SSHP values of each WS are denoted by means of a grey-scaled colour scheme.

Threat	WSs assigned	FTTF	LTTF	Accumulated cost in Rand
T_1	V_3	34	56	$R\,1\,000\,000$
	C_2	66	70	$R\,34\,000$
T_2	C_1	64	81	$R\,34\,000$
	C_2	52	61	$R\,34\,000$
T_3	V_1	32	63	$R\,1\,000\,000$
	C_1	19	40	$R\,34\,000$
T_4	C_3	1	19	$R\,34\,000$
	C_4	4	23	$R\,34\,000$
T_5	V_8	57	85	$R\,1\,000\,000$
	C_1	85	103	$R\,34\,000$
			Total:	$R\,3\,238\,000$

TABLE 8.8: *WS to threat assignment list as well as the FTTF and LTTF values proposed by the single-objective dynamic WA model (6.14)–(6.23).*

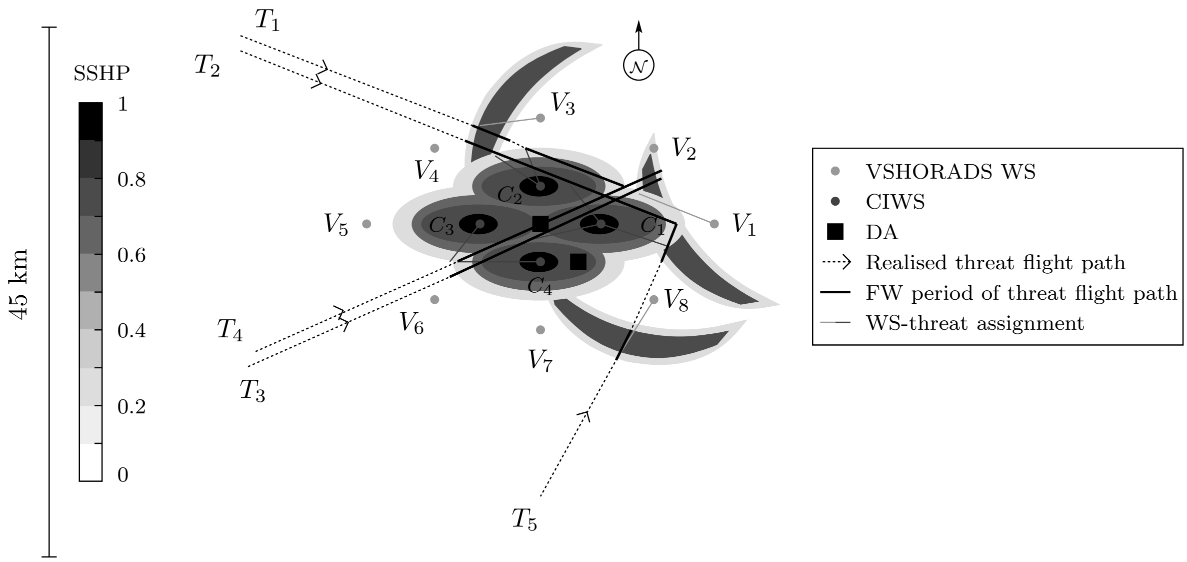


FIGURE 8.5: *Top view graphical illustration of the assignments of WSs to threats proposed by the single-objective, dynamic WA model (6.14)–(6.23) for the future predicted time stages from time stage $\tau = 23$ and onwards involving a straight line prediction period of 120 seconds.*

8.2.4 Numerical results for the multi-objective dynamic model prototype

The multi-objective dynamic WA model (6.24)–(6.26) was again solved for $\kappa = 2$ in the context of the simulated scenario depicted in Figure 8.1 by implementing the variation of the NSGA II described in §5.4.2. A population size of 100 was selected and a tour size of 2 was implemented in the tournament selection procedure. A further diversity preservation measure, known as a *multistart procedure*, was also implemented in the algorithm. This procedure allows for the algorithm to be executed in a pre-specifed number of parallel runs in order to obtain a number of approximately Pareto optimal sets of solutions. These sets are then combined to form a larger set of solutions and the FNSA of Agarwal *et al.* [2] is again used to filter out the dominated solutions so as to obtain a single new, approximately Pareto optimal set of solutions. The idea is that the procedure allows for a better and more diverse set of approximately Pareto optimal sets of solutions emanating from different randomly generated initial solutions. In this way a wider exploration of the solution space is promoted. A multistart procedure of 20 runs was implemented.

Furthermore, a WS was allowed to be assigned to a specific threat at most once during the scheduling window. A mutation probability of 0.07 and a stopping criterion of 100 generations were implemented. This set of parameter values was decided upon after extensive experimentation since they produce an acceptable spread of solutions along the approximate Pareto front in objective function space. The algorithm was able to solve the model instance in 139 seconds on an Intel Core i7-4770 processor with 8GB of random access memory operating at 3.40 GHz.

Twelve approximately Pareto optimal solutions were thus obtained. These solutions are listed in Table 8.9. The table contains the various WS-threat pair assignment schedules (*i.e.* the FWs during which WA assignments may occur) as well their corresponding objective function values (the priority-weighted accumulated survival probability, the ammunition costs incurred by the assignments and the numbers of units of ammunition available at the least re-engagable WSs after the assignments). The assignment schedules for each WS-threat pair is given as the WS, followed by the FW during which the assignment should occur in brackets, followed by a right-

Sol	WS-threat pair assignment schedules	Obj 1	Obj 2	Obj 3
1	V_3 [40,55], C_2 [58,64] $\rightarrow T_1$; V_4 [21,40], V_1 [78,88] $\rightarrow T_2$; C_4 [14,20], V_2 [39,67] $\rightarrow T_3$; C_4 [6,10], C_1 [22,33] $\rightarrow T_4$; V_7 [38,61], V_8 [60,85] $\rightarrow T_5$	0.0662	*R* 6 136 000	4
2	V_3 [35,53], C_2 [58,64] $\rightarrow T_1$; V_4 [22,28], V_8 [81,103] $\rightarrow T_2$; C_4 [8,19], V_2 [39,47] $\rightarrow T_3$; C_1 [22,33] $\rightarrow T_4$; V_7 [37,55], V_8 [57,67] $\rightarrow T_5$	0.0666	*R* 6 102 000	4
3	V_3 [35,53], C_2 [58,64] $\rightarrow T_1$; V_4 [23,40] $\rightarrow T_2$; C_4 [9,19], V_2 [39,47] $\rightarrow T_3$; C_1 [22,33]$\rightarrow T_4$; V_7 [35,47], V_8 [60,85] $\rightarrow T_5$	0.1120	*R* 5 102 000	4
4	C_2 [56,64], V_8 [83,101] $\rightarrow T_1$; V_4 [20,27], C_1 [69,76] $\rightarrow T_2$; C_4 [13,23], V_2 [38,45] $\rightarrow T_3$; C_3 [8,12], C_1 [20,34]$\rightarrow T_4$; V_8 [61,74],V_2 [98,112] $\rightarrow T_5$	0.0388	*R* 5 170 000	3
5	C_1 [71,76], V_8 [85,100] $\rightarrow T_1$; C_2 [63,67], V_1 [78,92] $\rightarrow T_2$; C_3 [7,16], C_4 [9,21] $\rightarrow T_3$; C_1 [21,33], C_2 [22,28] $\rightarrow T_4$; V_8 [61,71], V_2 [107,113] $\rightarrow T_5$	0.0552	*R* 4 204 000	3
6	C_1 [66,77], V_8 [85,100] $\rightarrow T_1$; C_2 [63,67] $\rightarrow T_2$; C_3 [7,16], C_4 [10,23] $\rightarrow T_3$; C_1 [20,32], C_2 [22,28]$\rightarrow T_4$; V_8 [61,71], V_2 [103,113] $\rightarrow T_5$	0.0656	*R* 3 204 000	3
7	V_3 [42,55], C_2 [57,69] $\rightarrow T_1$; C_1 [71,81], V_1 [75,86] $\rightarrow T_2$; C_4 [6,17], V_2 [39,64] $\rightarrow T_3$; C_1 [23,34], V_1 [35,38] $\rightarrow T_4$; C_1 [90,99], V_2 [100,116] $\rightarrow T_5$	0.0378	*R* 5 170 000	2
8	V_3 [42,55], C_2 [57,69] $\rightarrow T_1$; C_1 [71,81], V_8 [93,101] $\rightarrow T_2$; C_4 [6,17], V_2 [39,64] $\rightarrow T_3$; C_1 [23,34] $\rightarrow T_4$; C_1 [90,99], V_2 [100,116] $\rightarrow T_5$	0.0395	*R* 4 170 000	2
9	V_3 [42,55], C_2 [57,66] $\rightarrow T_1$; C_1 [71,81] $\rightarrow T_2$; C_4 [6,17], V_2 [39,64] $\rightarrow T_3$; C_2 [19,27], C_1 [23,34] $\rightarrow T_4$; C_1 [90,99], V_2 [100,116] $\rightarrow T_5$	0.0396	*R* 3 204 000	2
10	V_3 [42,55], C_2 [57,69] $\rightarrow T_1$; C_1 [71,81] $\rightarrow T_2$; C_4 [6,17], V_2 [38,65] $\rightarrow T_3$; C_1 [23,34]$\rightarrow T_4$; C_1 [90,99], V_2 [104,114] $\rightarrow T_5$	0.0405	*R* 3 170 000	2
11	C_2 [64,67], C_1 [66,75] $\rightarrow T_1$; C_2 [54,59], V_8 [86,102] $\rightarrow T_2$; C_3 [4,14], C_4 [13,24] $\rightarrow T_3$; C_2 [19,24], C_1 [21,36] $\rightarrow T_4$; V_8 [58,64],C_1 [85,98] $\rightarrow T_5$	0.0573	*R* 2 272 000	2
12	C_2 [63,68], C_1 [67,76] $\rightarrow T_1$; C_2 [53,57], V_8 [83,101] $\rightarrow T_2$; C_4 [6,18], C_3 [7,18] $\rightarrow T_3$; C_1 [23,36] $\rightarrow T_4$; V_8 [57,77], C_1 [90,101] $\rightarrow T_5$	0.0665	*R* 2 238 000	2

TABLE 8.9: *The set of approximately Pareto optimal solutions obtained when solving the multi-objective dynamic WAM (6.24)–(6.26) within the context of the GBAD scenario §8.1, WS-threat pair assignment schedules are given, together with their respective objective function values as illustrated in Figure 8.6. Obj 1 represents the priority-weighted accumulated survival objective function values, Obj 2 represents the ammunition cost objective function values and Obj 3 represents the least re-engagable WS objective function values. The assignment schedules are given as the WS, followed by the FW in brackets, followed by a right-arrow indicating to which threat the WS should be assigned.*

arrow and the threat to which it is assigned (*e.g.* V_3 [40,55] $\rightarrow T_1$ means that WS V_3 should engage threat T_1 during any time stage from time stage 40 to time stage 55, inclusive).

The twelve approximately Pareto optimal solutions are also presented graphically in objective function space in Figure 8.6. These solutions again achieve objective function values in the priority-weighted accumulated survival probabilities objective (6.24) and assignment cost objective (6.25) as indicated on the horizontal and vertical axes of the figure, respectively. The objective function values achieved for the re-engagability objective (6.26) may be interpreted as follows. The solutions are numbered $1, \ldots, 12$. For solutions 1, 2 and 3 (represented by solid circles) the least re-engagable WS can engage four times after the assignment, for solutions 4, 5 and 6 (represented by solid squares) the least re-engagable WS can engage three times after the

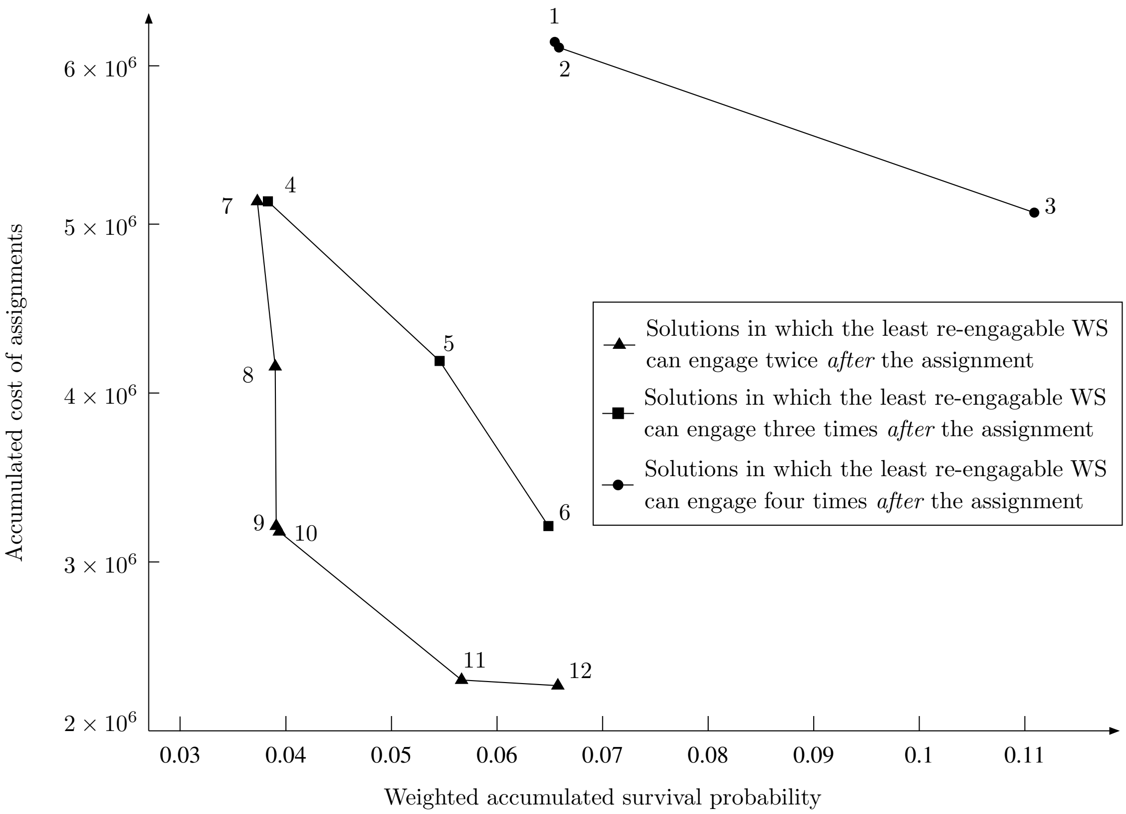


Figure 8.6: *The approximately Pareto optimal front in objective function space obtained when solving the solving the multi-objective dynamic WAM (6.24)–(6.26) within the context of the GBAD scenario of §8.1.*

assignment, and for solutions $7, \ldots, 12$ (represented by solid triangles) the least re-engagable WS can engage twice after the assignment. Figure 8.6 again clearly illustrates how much of one objective is compromised for a gain in another. These solutions may again be presented to the FCO after which he may choose one of them for implementation (based on his subjective preference and expertise).

In order to illustrate the WS-threat pair assignment schedules in Table 8.9, consider Solution[2] 10. This solution contains eight WS-threat assignment pairs with an accumulated survival probability of 0.0405 at an assignment cost of $R\,3\,170\,000$. The least re-engagable WS in this solution (WS C_1) can re-engage twice after the assignment. WS V_3 should engage Threat T_1 during time stages 42 to 55 (inclusive). The same threat should again be engaged by WS C_2 during time stages 57 to 69 (inclusive). WS C_1 should engage Threat T_2 during time stages 71 to 81 (inclusive). WS C_4 should engage Threat T_3 during time stages 6 to 17 (inclusive) and the threat should again be engaged by WS V_2 during time stages 38 to 65 (inclusive). WS C_1 should furthermore engage Threat T_4 during time stages 23 to 34 (inclusive). Finally, WS C_1 should engage Threat T_5 during time stages 90 to 99 (inclusive), and the threat should again be engaged by WS V_2 during time stages 104 to 114 (inclusive). A graphical representation of this assignment schedule is shown in Figure 8.7. The predicted threat flight paths are denoted by dotted lines, the fire windows are denoted by thick solid lines and the WS-threat pair assignments are denoted by thin solid lines.

[2] Only Solution 10 is discussed in detail here in order to avoid undue repetition.

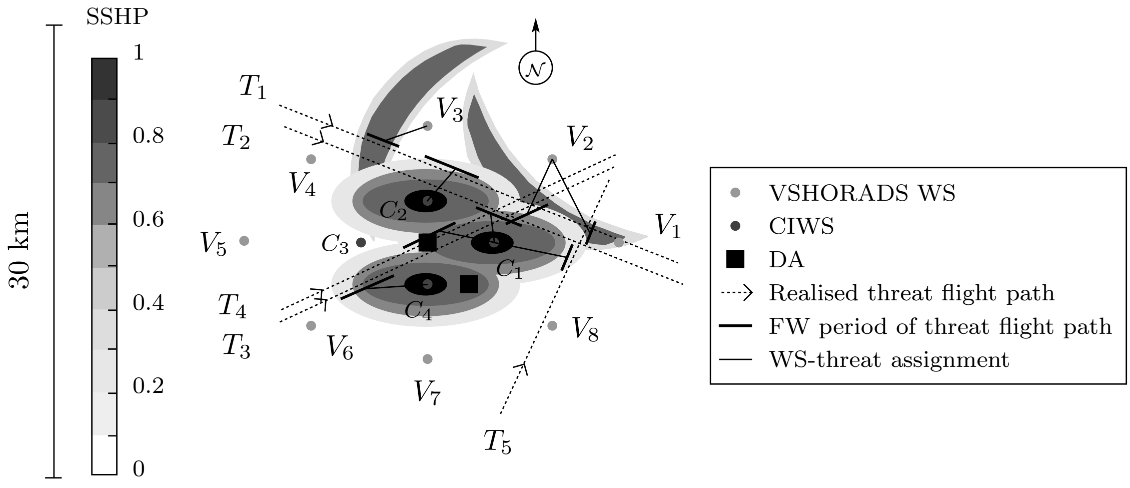


FIGURE 8.7: *Top view graphical illustration of the assignment schedule corresponding to Solution 10 in Table 8.9.*

8.3 Validation of model results

In this section, three methods are employed to validate the numerical results obtained in §8.2 for each of the WAM prototypes of §6.5. The first method is a face validation, which involves a subjective evaluation of the quality and realism of the WA proposals obtained for each WAM prototype. The second method is a random benchmark validation procedure and the third method involves consulting two military experts for validations of the results obtained.

8.3.1 Face validation

Face validations are carried out in this section for each of the four models of §6.5 separately for which results were reported in §8.2.

The single-objective static WAM (6.6)–(6.9)

The results returned by the genetic algorithm for the single-objective static WAM (6.6)–(6.9), presented in Figure 8.2, seem plausible in the sense that high-quality WS-threat assignment pairs are proposed. The solution proposes engagements of the two most threatening aircraft, T_1 and T_2, which have threat priority values of 0.99 and 1.00, respectively. Upon assigning WSs C_1, V_2 and V_3 (with respective SSHP values of 0.2, 0.7 and 0.1) to threat T_1, its priority-weighted survival probability drops from 0.9900 to $0.99 \times (1-0.2) \times (1-0.7) \times (1-0.1) = 0.2138$. Similarly, upon assigning WS C_2 (with an SSHP value of 0.9) to threat T_2, its priority-weighted survival probability drops from 1.0000 to $1.00 \times (1-0.9) = 0.1000$. These survival probabilities are suitably low for threats with such large elimination priorities. Note that no further WSs could have been assigned to these two threats during time stage 39 — the closest unassigned[3] WSs (WSs C_3, C_4, V_4 and V_8) are all out of range of threats T_1 and T_2 during time stage 39 (*i.e.* their SSHP values with respect to these threats are smaller than 0.1).

[3]Recall that each WS can be assigned at most once to engage aerial threats single-objective static WAM (6.6)–(6.9).

The next most urgent threats are T_3 and T_4, with elimination priorities of 0.76 and 0.74, respectively. Upon assigning the only WS within range of threat T_3 to this threat (namely WS V_1 with an SSHP value of 0.1), its priority-weighted survival probability drops from 0.7600 to $0.76 \times (1 - 0.1) = 0.6840$. Although this large value is a concern, there are no further WSs in range of T_3 during time stage 39 by which this value could have been lowered. Furthermore, since there are no unassigned WSs within range of threat T_4 during time stage 39, no WSs are assigned to this threat. This means that its priority-weighted survival probability remains at a very high value of 0.7400, but again the WS deployment is such that no assignment could have remedied this situation during time stage 39.

Moreover, threat T_5 has a threat or elimination priority of 0.5000. The only two WSs within range of T_5 during time stage 39 are WSs V_6 and V_7. Upon assigning these WSs (with respective SSHP values of 0.1 and 0.5) to threat T_5, its priority-weighted survival probability drops to $0.50 \times (1 - 0.1) \times (1 - 0.5) = 0.2250$, which is acceptably low.

The accumulated priority-weighted survival probability of the solution is therefore $0.2138 + 0.1000 + 0.6840 + 0.7400 + 0.2250 = 1.9628$. It is concluded that those threats with the largest elimination priority values are assigned WSs which achieve relatively large SSHP values with respect to these threats when, in fact, there are WSs within range to justify such assignments. Although the solution returned by the genetic algorithm seems plausible, the disadvantage of employing a static WAM is abundantly clear: There seem to be much better times at which to engage the threats than during time stage 39. For instance, it would be beneficial to engage threats T_1 and T_2 much earlier than at time $\tau = 39$ in view of the locations of WSs C_2, V_3 and V_4. Similar conclusions may be drawn for the remaining three threats.

Finally, the genetic algorithm employed to solve the model is computationally efficient, being able to solve the model almost instantaneously (in less than 0.01 second).

The multi-objective static WAM (6.10)–(6.12)

A total of thirteen approximately Pareto optimal solutions were returned by the NSGA II for the multi-objective static WAM (6.10)–(6.12), as reported in Table 8.6 and illustrated in objective space in Figure 8.3. For the sake of brevity, only one of these solutions is subjected to detailed face validation in this section. Solution 11, highlighted in Figure 8.4, is selected for this purpose.

Solution 11 is the best alternative amongst the thirteen approximately Pareto optimal solutions under a budgetary constraint of R 2 500 000 on ammunition cost incurred by the WA. Note that this budget allows for at most two VSHORADS WSs to engage threats. From among the eight available VSHORADS WSs, the two most effective in terms of engaging threats are indeed the two assigned in the solution, namely WS V_2 to threat T_1 and WS V_1 to threat T_3. Upon assigning WS V_2 (with an SSHP value of 0.7) to threat T_1, its priority-weighted survival probability drops to $0.99 \times (1 - 0.7) = 0.2970$, which is small compared to the value of 0.9900 that would have resulted if no WS were assigned to T_1. Similarly, upon assigning WS V_1 (with an SSHP value of 0.1) to threat T_3, its priority-weighted survival probability drops to $0.76 \times (1 - 0.1) = 0.6840$. This value is not particularly small compared to the value of 0.7600 that would have resulted if no WS were assigned to T_3. Other assignments, which would seem intuitively to be good alternatives are, however, not more effective than the solution returned by the NSGA II. WS V_2 cannot, for example, be assigned to threat T_2 during time step 39, because T_2 is out of range of V_1 when $\tau = 39$ (from Table 8.1 it may be seen that the WS achieves an SSHP value of less than 0.1 with respect to this threat). Furthermore, assigning WS V_2 to threat T_4 instead of to threat T_1 would also yield a worse result, because V_2 achieves a smaller SSHP value (of 0.5) with respect to T_4 than with respect to T_1 (a value of 0.7).

It would seem that the assignment of WS V_3 to threat T_1 represents a viable alternative to the solution returned by the NSGA II, but this is not the case. If WS V_3 were to be assigned to threat T_1 (achieving an SSHP value of 0.1) instead of assigning WS V_2 to the threat (achieving an SSHP value of 0.7), it would have resulted in a priority-weighted survival probability $0.99 \times (1-0.1) = 0.8910$ for T_1 instead of the current value of $0.99 \times (1 - 0.7) = 0.2970$ in view of the particular orientation of the WSs (both orientated in an easterly direction).

The budget of $R500\,000$ that remains after having assigned two VSHORADS WSs to engage threats is used to assign CIWS C_2 to threat T_2 (with an SSHP value of 0.9). This assignment results in the priority-weighted survival probability of T_2 dropping to $1.00 \times (1 - 0.9) = 0.1000$, which is very small compared to the value of 1.0000 that would have resulted if no WS were to be assigned to T_2. No other CIWSs are within range of any threat at time stage 39. The solution therefore yields an accumulated priority-weighted survival probability of $0.2970 + 0.1000 + 0.6840 + 0.7400 + 0.5000 = 2.2310$ at an assignment cost of $(1 \times R34\,000) + (2 \times R1\,000\,000) = R2\,034\,000$. The least re-engageable WSs after the assignment are WSs C_2, V_1 and V_2, which all have three units of ammunition available.

It is again concluded that threats with the largest threat priority values are assigned WSs that are relatively effective in terms of their SSHP values. Better assignments would, of course, have been possible under more generous budgetary constraints, as illustrated in Figure 8.3. It is particularly heartening to note that the solution returned by the genetic algorithm for the single-objective static WAM (6.6)–(6.9) in Figure 8.2 is indeed also returned by the NSGA II for the multi-objective static WAM (6.10)–(6.12).

Finally, the NSGA II employed to solve the model does not seem to be sufficiently computationally efficient. It required a computation time of 13.4 seconds to produce the solutions in Figure 8.3, while solutions would realistically be required in under one second, because radar refresh rates are typically between one and four seconds, resulting in updated air pictures.

The single-objective dynamic WAM (6.14)–(6.23)

When considering the results retuned by the simulated annealing algorithm (implemented by Van der Merwe and Van Vuuren [196]), the following analysis may be made in respect of the solution. Upon assigning WS V_3 (with a fixed-mean SSHP[4] value of 0.629 during the window $[34, 56]$) and WS C_2 (with a fixed-mean SSHP value of 0.808 during the window $[66, 70]$) to threat T_1, its priority-weighted survival probability drops to $0.2 \times (1 - 0.629) \times (1 - 0.808) = 0.0142$, which is quite small compared to the value of 0.2 that would have resulted had no WSs been assigned to the threat. In a similar fashion, upon assigning WS C_1 (with a fixed-mean SSHP value of 0.799 during the window $[64, 81]$) and WS C_2 (with a fixed-mean SSHP value of 0.844 during the window $[52, 61]$) to threat T_2, its priority-weighted survival probability drops from 0.2200 to $0.22 \times (1 - 0.799) \times (1 - 0.844) = 0.0068$, which is suitably small. Similarly, upon assigning WS V_1 (with a fixed-mean SSHP value of 0.633 during the window $[32, 63]$) and WS C_1 (with a fixed-mean SSHP value of 0.943 during the window $[19, 40]$) to threat T_3, its priority-weighted survival probability drops from 1.000 to $1 \times (1 - 0.633) \times (1 - 0.943) = 0.0209$, which is again very small. Furthermore, upon assigning WS C_3 (with a fixed-mean SSHP value of 0.762 during the window $[1, 19]$) and WS C_4 (with a fixed-mean SSHP value of 0.793 during the window $[4, 23]$) to threat T_4, its priority-weighted survival probability drops to $1 \times (1 - 0.762) \times (1 - 0.793) = 0.0495$, which is very small compared to the value of 1 that would have resulted from assigning no WSs to the threat. Finally, upon assigning WS V_8 (with a fixed-mean SSHP value of 0.580 during the window $[57, 85]$) and WS C_1 (with a fixed-mean SSHP value

[4]Note that the fixed-mean values are obtained from Figure 8.7.

of 0.540 during the window $[85, 103]$) to threat T_5, its priority-weighted survival probability drops from 0.3000 to $0.3 \times (1 - 0.580) \times (1 - 0.540) = 0.0579$, which is again comparatively small. The accumulated priority-weighted survival probability of the assignment is therefore $0.0142 + 0.0068 + 0.0209 + 0.0495 + 0.0579 = 0.1493$.

When analysing the results described above, it seems that the single-objective dynamic WAM (6.14)–(6.23) is able to propose very good WS-threat assignment pair proposals. The dynamic complexity of the WAM, however, makes it difficult to comment on the quality of the solution relative to other alternatives. A significant disadvantage of the model is that it does not take into consideration the possibility that a threat may be destroyed before coming up for engagement by a WS scheduled later. It is therefore proposed that the model be solved for each time stage in the time continuum of a given GBAD scenario in order to make provision for the possibility that a threat may be eliminated from the scenario at some stage. In such a case, the threat should be removed from the threat list and the remainder of the threats should be considered for future engagements.

Finally, the simulated annealing algorithm employed to solve the model is computationally efficient, being able to solve the model almost instantaneously (in under 0.01 seconds). Although the model is able to schedule FWs for its assignment proposals, it considers only a single objective within a single computation run of the model.

The multi-objective dynamic WAM (6.24)–(6.26)

A total of twelve approximately Pareto optimal solutions were returned by the NSGA II for the multi-objective dynamic WAM (6.24)–(6.26), as reported in Table 8.9 and illustrated in objective space in Figure 8.6. For the sake of brevity, only one solution is subjected to detailed face validation in this section. Solution 10, highlighted towards the end of §8.2.4, is selected for this purpose.

Upon assigning WSs V_3 and C_2 (with respective fixed-mean SSHP values of 0.65 over the window $[42, 55]$ and 0.96085 over the window $[57, 69]$) to threat T_1, its priority-weighted survival probability drops to $0.2 \times (1 - 0.65) \times (1 - 0.96085) = 0.0027$, which is small compared to the value of 0.2000 that would have resulted had no WS been assigned to T_1. Similarly, upon assigning WS C_1 (with a fixed-mean SSHP value of 0.9883 over the window $[71, 81]$) to threat T_2, its priority-weighted survival probability drops to $0.2200 \times (1 - 0.9883) = 0.0026$, which is also small compared to the value of 0.2200 that would have resulted had no WS been assigned to T_2. In a similar vein, the priority-weighted survival probability of threat T_3 drops to $1 \times (1 - 0.9393) \times (1 - 0.7) = 0.0182$ upon assigning WSs C_4 (with a fixed-mean SSHP value of 0.9393 over the window $[6, 17]$) and V_2 (with a fixed-mean SSHP value of 0.7 over the window $[38, 65]$) to threat T_3. This value is again small compared to the value of 1 that would have resulted had no WS been assigned to T_3. Furthermore, the priority-weighted survival probability of threat T_4 drops to $1 \times (1 - 0.9994) = 0.0006$ upon assigning WSs C_1 (with a fixed-mean SSHP value of 0.9994 over the window $[23, 34]$) to the threat — this is again small compared to the value of 1 that would have resulted had no WS been assigned to the threat. Finally, the priority-weighted survival probability of threat T_5 drops to $0.3 \times (1 - 0.7253) \times (1 - 0.8001) = 0.0164$ upon assigning WS C_1 (with a fixed-mean SSHP value of 0.7253 over the window $[90, 99]$) and WS V_2 (with a fixed-mean SSHP value of 0.8001 during the window $[104, 114]$) to threat T_5. This is also small compared to the value of 0.3 had no WS been assigned to threat T_5.

The accumulated priority-weighted survival probability of the assignment is therefore $0.0027 + 0.0026 + 0.0182 + 0.0006 + 0.0164 = 0.0405$. Since this WA schedule involves five CIWS assignments and three VSHORADS assignments, the total ammunition cost incurred by the schedule is

$(5 \times R\,34\,000) + (3 \times R\,1\,000\,000) = R\,3\,170\,000$. The least re-engageable WS after the assignment is WS C_1, which has two units of ammunition available.

While the aforementioned model solution seems good, it is difficult to comment on the relative quality of alternative assignments in view of the complexity of the dynamic WA situation. It is noted, with satisfaction however, that there is a clear trade-off between the three conflicting objectives of the model, as demonstrated in Figure 8.6.

As with the single-objective dynamic WAM (6.14)–(6.23), the multi-objective dynamic WAM (6.24)–(6.26) exhibits the disadvantage of not taking into consideration the possibility that a threat might be destroyed during an earlier scheduled FW before it can come up for re-engagement during a later FW. It is therefore again proposed that the model be solved during each time stage in the time continuum for all future time stages of a given GBAD scenario in order to be able to take cognisance of the temporal unfolding of the air picture.

Finally, the NSGA II employed to solve the model does not seem to be computationally efficient. It required a computation time of 139 seconds to produce the twelve approximately Pareto optimal solutions in Table 8.9, while an approximate Pareto front would realistically be required in under one second, because of the radar refresh rates typically available, as explained above.

8.3.2 Random benchmark validation

Random benchmark validations are carried out separately in this section for each of the four WAMs for which results were reported in §8.2. This involves generating thirty random feasible solutions for each WAM prototype and comparing these solutions with the results returned by the relevant algorithms used in §8.2 in to solve the WAMs.

Single-objective static random benchmark validation

The thirty random solutions generated for the single-objective static WAM (6.6)–(6.9) are given in Table 8.10. Since these random solutions may, in fact, also be used to validate the multi-objective static WAM (6.10)–(6.12), the table therefore contains columns for all three objective function values considered in the multi-objective static WAM (6.10)–(6.12).

In order to validate the solution returned by the genetic algorithm for the single-objective static WAM (6.6)–(6.9) within the context of the GBAD scenario of §8.1, the solutions are plotted in objective function space in Figure 8.8 together with the solution returned by the algorithm. From the figure it is clear that the solution returned by the algorithm performs better than the best solution in the set of random solutions. The random solution closest in objective space to the solution returned by the algorithm is Solution 16 of Table 8.10, which achieves a priority-weighted accumulated survival probability of 2.1448. The genetic algorithm is therefore able to achieve a solution that is approximately 8% better than the best random solution found. This may seem a small difference percentage. The low level of complexity of the single-objective static WAM (6.6)–(6.9) (when compared to the WAM prototypes in the other three classes), however, makes it relatively easy to solve for small problem instances (such as the one in §8.1). Solution 16 involves assigning WSs C_1 and V_3 to threat T_1, WS C_2 to threat T_2, WSs V_1 and V_2 to threat T_3 and WS V_7 to threat T_5. This assignment is similar to the assignment proposed by the algorithm (see Figure 8.2), except that the random solution rather assigns WS V_2 to threat T_3 than to threat T_2 and only assigns WS V_6 to threat T_5.

Furthermore, the average priority-weighted accumulated survival probability of the thirty random solutions (the column labelled Obj 1 in Table 8.10) is 2.6279. When comparing this value with that of the solution obtained in §8.2.1 (*i.e.* 1.9628) it is found that the genetic algorithm is able to achieve an improvement of approximately 25% in the objective function value of the single-objective static WAM (6.6)–(6.9) over the average of the set of random benchmark solutions. Based on these findings it may be concluded that the genetic algorithm is able to provide significantly better solutions to the single-objective static WAM (6.6)–(6.9) than proposing random feasible assignments of WSs to threats.

Sol	WS-threat pair assignments	Obj 1	Obj 2	Obj 3
1	C_1, C_2, $V_2 \rightarrow T_1$; $V_1 \rightarrow T_3$; $V_6 \rightarrow T_5$	2.9928	$R\,3\,068\,000$	2
2	$C_2 \rightarrow T_1$; $C_1 \rightarrow T_2$; $V_1 \rightarrow T_3$; $V_2 \rightarrow T_4$; V_6, $V_7 \rightarrow T_5$	2.3740	$R\,4\,068\,000$	2
3	$V_3 \rightarrow T_1$; $C_2 \rightarrow T_2$; $V_2 \rightarrow T_3$; $V_1 \rightarrow T_4$ $V_6 \rightarrow T_5$	2.4870	$R\,4\,034\,000$	3
4	C_2, $V_3 \rightarrow T_1$; $C_1 \rightarrow T_2$; $V_2 \rightarrow T_3$; $V_1 \rightarrow T_4$; $V_7 \rightarrow T_5$	2.3415	$R\,4\,068\,000$	2
5	V_2, $V_3 \rightarrow T_1$; $C_1 \rightarrow T_2$; $V_1 \rightarrow T_3$; V_6, $V_7 \rightarrow T_5$	2.5163	$R\,5\,034\,000$	2
6	$C_1 \rightarrow T_1$; $C_2 \rightarrow T_2$; V_1, $V_2 \rightarrow T_3$; V_6, $V_7 \rightarrow T_5$	2.1990	$R\,4\,068\,000$	2
7	$V_3 \rightarrow T_1$; $V_1 \rightarrow T_3$; $V_6 \rightarrow T_5$	3.7650	$R\,3\,000\,000$	3
8	$C_1 \rightarrow T_1$; $C_2 \rightarrow T_2$; V_1, $V_2 \rightarrow T_4$; $V_7 \rightarrow T_5$	2.2350	$R\,3\,068\,000$	2
9	$V_3 \rightarrow T_1$; C_1, $C_2 \rightarrow T_2$; V_1, $V_2 \rightarrow T_3$; V_6, $V_7 \rightarrow T_5$	2.2580	$R\,5\,068\,000$	2
10	V_2, $V_3 \rightarrow T_1$; $C_1 \rightarrow T_2$; $V_1 \rightarrow T_3$; $V_7 \rightarrow T_5$	2.5413	$R\,4\,034\,000$	2
11	$V_3 \rightarrow T_1$; $C_2 \rightarrow T_2$; $V_1 \rightarrow T_4$; $V_6 \rightarrow T_5$	2.8670	$R\,3\,034\,000$	3
12	$C_1 \rightarrow T_1$; $C_2 \rightarrow T_2$; $V_2 \rightarrow T_3$; $V_1 \rightarrow T_4$; $V_7 \rightarrow T_5$	2.1880	$R\,3\,068\,000$	2
13	C_1, $C_2 \rightarrow T_1$; V_1, $V_2 \rightarrow T_4$; V_6, $V_7 \rightarrow T_5$	2.7140	$R\,4\,068\,000$	2
14	$V_3 \rightarrow T_1$; $C_1 \rightarrow T_2$; $V_1 \rightarrow T_3$; $V_2 \rightarrow T_4$; $V_6 \rightarrow T_5$	2.9950	$R\,4\,034\,000$	2
15	$V_2 \rightarrow T_1$; $C_1 \rightarrow T_2$; $V_1 \rightarrow T_3$; $V_6 \rightarrow T_5$	2.7710	$R\,3\,034\,000$	2
16	C_1, $V_3 \rightarrow T_1$; $C_2 \rightarrow T_2$; V_1, $V_2 \rightarrow T_3$; $V_7 \rightarrow T_5$	2.1448	$R\,4\,068\,000$	2
17	C_1,$V_3 \rightarrow T_1$; $C_2 \rightarrow T_2$; $V_2 \rightarrow T_4$; $V_6 \rightarrow T_5$	2.7928	$R\,3\,068\,000$	2
18	$V_3 \rightarrow T_1$; C_1, $C_2 \rightarrow T_2$; V_1, $V_2 \rightarrow T_4$; $V_7 \rightarrow T_5$	2.2940	$R\,4\,068\,000$	2
19	C_1, C_2, $V_3 \rightarrow T_1$; V_1, $V_2 \rightarrow T_3$; V_6, $V_7 \rightarrow T_5$	2.6634	$R\,5\,068\,000$	2
20	C_1, V_2, $V_3 \rightarrow T_1$; $C_2 \rightarrow T_2$; $V_1 \rightarrow T_3$	2.2379	$R\,4\,064\,000$	2
21	C_1, C_2, $V_2 \rightarrow T_1$; $V_1 \rightarrow T_4$; V_6, $V_7 \rightarrow T_5$	2.7698	$R\,4\,068\,000$	2
22	C_1, V_2, $V_3 \rightarrow T_1$; $C_2 \rightarrow T_2$; $V_1 \rightarrow T_3$; $V_6 \rightarrow T_5$	2.1878	$R\,4\,068\,000$	2
23	C_1, $C_2 \rightarrow T_2$; $V_2 \rightarrow T_3$; $V_1 \rightarrow T_4$; $V_6 \rightarrow T_5$	2.5460	$R\,3\,068\,000$	2
24	V_1, $V_2 \rightarrow T_3$; V_6, $V_7 \rightarrow T_5$	3.2970	$R\,4\,000\,000$	2
25	C_2, $V_2 \rightarrow T_1$; $C_1 \rightarrow T_2$; $V_1 \rightarrow T_3$	2.6725	$R\,2\,068\,000$	2
26	C_2, V_2, $V_3 \rightarrow T_1$; $V_1 \rightarrow T_3$; V_6, $V_7 \rightarrow T_5$	2.7827	$R\,5\,034\,000$	2
27	$C_2 \rightarrow T_1$; $C_1 \rightarrow T_2$; V_1, $V_2 \rightarrow T_3$; V_6, $V_7 \rightarrow T_5$	2.4020	$R\,4\,068\,000$	2
28	$V_3 \rightarrow T_1$; V_1, $V_2 \rightarrow T_4$	3.4840	$R\,3\,000\,000$	3
29	C_1, $C_2 \rightarrow T_2$; $V_1 \rightarrow T_3$; $V_2 \rightarrow T_4$	2.6040	$R\,2\,068\,000$	2
30	C_1, $C_2 \rightarrow T_1$; V_1, $V_2 \rightarrow T_4$; V_6, $V_7 \rightarrow T_5$	2.7140	$R\,4\,068\,000$	2
	Mean:	2.6279	$R\,3\,752\,133$	2.1

TABLE 8.10: *Thirty randomly generated solutions to the single-objective static WAM (6.6)–(6.9) and the multi-objective static WAM (6.10)–(6.12) for time stage 39 of the GBAD scenario of Figure 8.1. Obj 1 represents the priority-weighted accumulated survival probability objective function values for the single-objective static WAM (6.6)–(6.9) and the multi-objective static WAM (6.10)–(6.12). Obj 2 furthermore represents the cost objective function values of the multi-objective static WAM (6.10)–(6.12) and Obj 3 represents the least re-engagable WS objective function values of the multi-objective static WAM (6.10)–(6.12). The assignment pairs are given as WSs, followed by a right-arrow indicating to which threat the WSs should be assigned.*

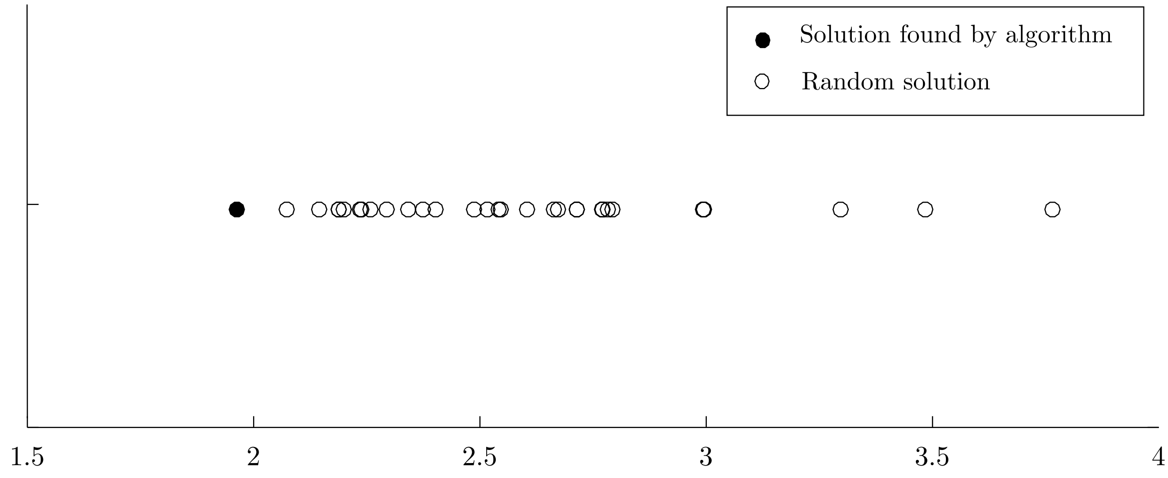


FIGURE 8.8: *The objective function value of the solution to the single-objective static WAM prototype (6.6)–(6.9) returned by the genetic algorithm, together with the objective function values of the thirty random solutions in Table 8.10.*

Multi-objective static random benchmark validation

The same thirty random solutions in Table 8.10 are used to validate the numerical results returned by the NSGA II for the multi-objective static WAM (6.10)–(6.12) in §8.2.2. The objective function values of each solution are presented in Table 8.10. Sorting the thirty random solutions in Table 8.10 by means of the FNSA (described in §4.6), in order to obtain the nondominated solutions among the set of random solutions, yields six nondominated solutions. These solutions together with the remaining 24 randomly generated solutions are illustrated graphically in objective function space in Figure 8.9. Solutions in which the least re-engageable WSs can re-engage twice after the assignment are denoted by open triangles, while the solutions in which the least re-engageable WS can re-engage three times after the assignment are denoted by open squares — the nondominated randomly generated solutions are joined by broken line segments.

The approximately Pareto optimal solutions returned by the NSGA II are also plotted in objective function space in Figure 8.9. Solutions in which the least re-engageable WSs can re-engage twice after the assignment are denoted by solid triangles (they are joined by solid line segments), while solutions in which the least re-engageable WSs can re-engage three times after the assignment are denoted by solid squares (again joined by solid line segments).

When comparing the approximately Pareto optimal front achieved by the NSGA II with the nondominated front of randomly generated solutions, it is clear that the NSGA II is able to achieve a much better and wide-spread nondominated front than that formed by the randomly generated solutions. It may seem worrying that the nondominated front of the randomly generated solutions does not seem to be very far removed from the approximate Pareto front achieved by the NSGA II — there is a small gap between these fronts of solutions in Figure 8.9. Although the multi-objective static WAM (6.10)–(6.12) may be considered more complex than the single-objective static WAM (6.6)–(6.9) due to the presence of multiple objectives in the former, the model is still static in nature which makes it considerably easier to find solutions lying close to the approximate Pareto optimal solutions for small problem instances (such as the one in §8.1).

The hypervolume technique discussed in §4.4 was used to calculate the hypervolume of the ap-

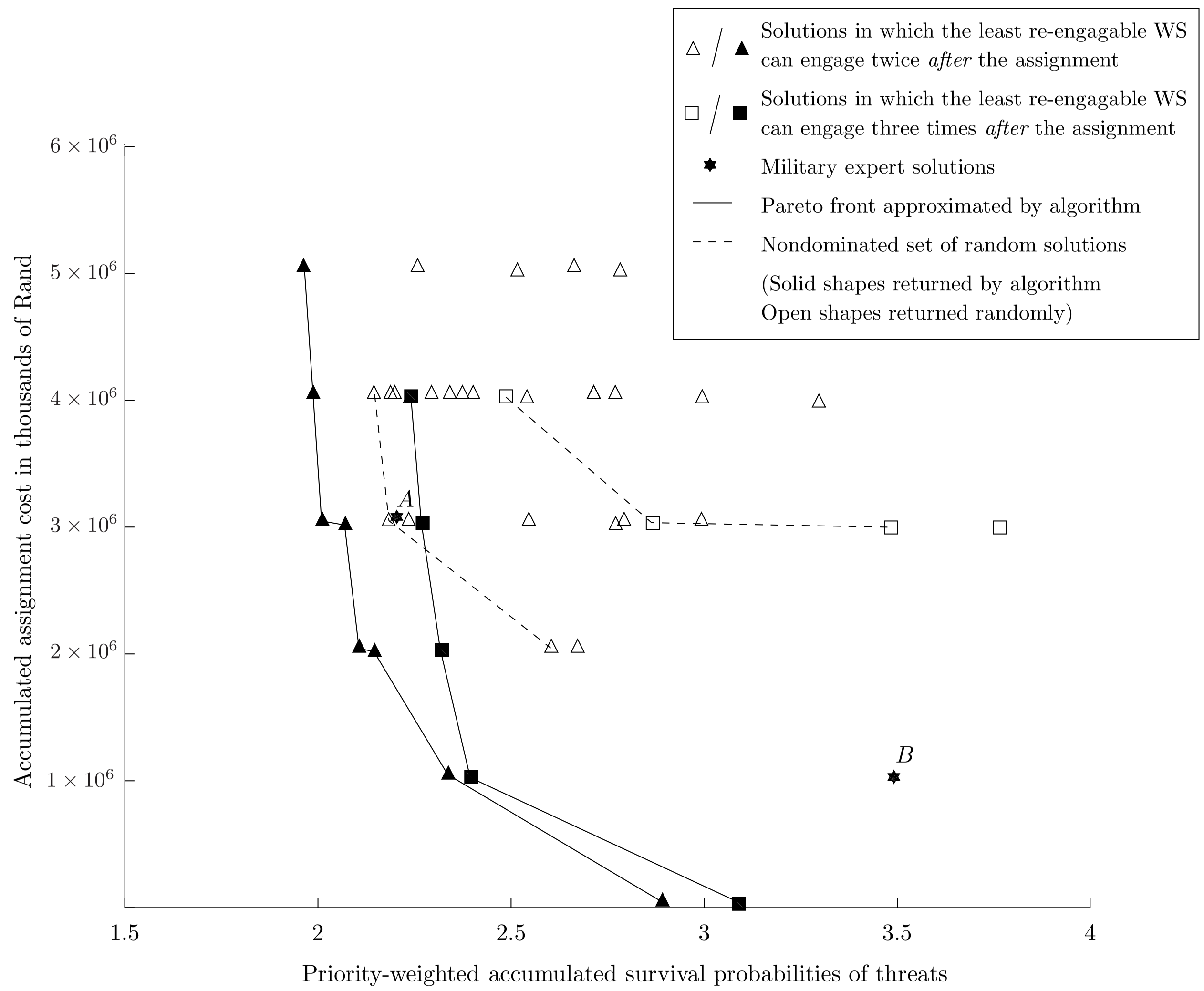


FIGURE 8.9: *Approximately Pareto optimal solutions returned by the NSGA II for the multi-objective static WAM (6.10)–(6.12) in objective function space for the GBAD scenario of §8.1, as well as the thirty randomly generated solutions in Table 8.10 together with two solutions returned by military experts.*

proximately Pareto optimal solutions achieved by the NSGA II in Table 8.4 and the hypervolume of the nondominated front of randomly generated solutions in Table 8.10 — a larger hypervolume measure value is better than a smaller value. A reference point for the hypervolume calculation was chosen to contain a value of 3.5 in the priority-weighted accumulated survival probability objective, a value of $R\,5\,100\,000$ in the cost objective and a value of 1 in the re-engageability objective was used. A hypervolume measure of 1.1514×10^7 was obtained for the approximately Pareto optimal solutions achieved by the NSGA II in Table 8.4, while a hypervolume measure of 5.2754×10^6 was obtained for the nondominated front of randomly generated solutions in Table 8.10. This confirms that the solutions returned by the NSGA II dominate a significantly larger portion of the solution space than the thirty randomly generated solutions.

Based on the discussion above, it may be concluded that the multi-objective static WAM (6.10)–(6.12) is able to provide a good spread of high-quality nondominated solutions in objective function space compared to a randomly generated set of solutions.

Single-objective dynamic random benchmark validation

The thirty random solutions generated for the single-objective dynamic WAM (6.14)–(6.23) are given in Table 8.11.

In order to validate the solution returned by the simulated annealing algorithm for the single-objective dynamic WAM (6.14)–(6.23) within the context of the GBAD scenario of §8.1, the random solutions are again plotted in objective function space in Figure 8.10 together with the solution returned by the algorithm. From the figure it is again clear that the solution returned by

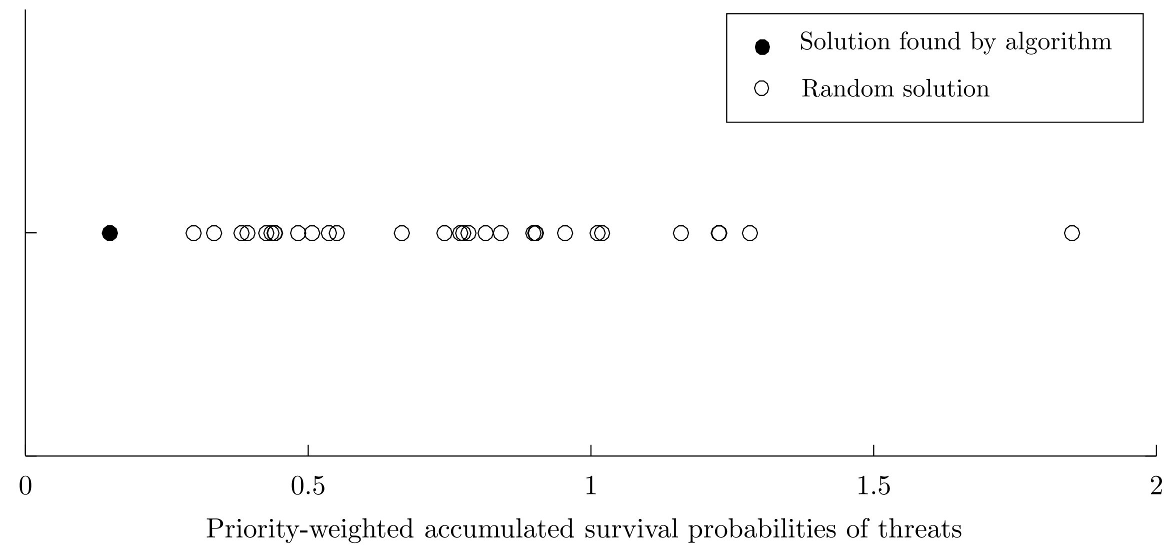


FIGURE 8.10: *The objective function value returned by the simulated annealing algorithm in solving the single-objective dynamic WAM prototype (6.14)–(6.23) in conjunction with the objective function values of the thirty random solutions in Table 8.12.*

the algorithm outperforms the randomly generated solutions. The solution closest in objective space to the solution returned by the algorithm is Solution 26 in Table 8.11, which achieves a priority-weighted accumulated survival probability of 0.2974. The solution returned by the simulated annealing algorithm is therefore able to achieve a solution that is approximately 50% better than the best randomly generated solution. This is a significant improvement.

Sol	WS-threat pair assignment schedules	Objective
1	V_3 [34,56]$\rightarrow T_1$; C_3 [3,19] $\rightarrow T_3$; V_1 [33,60] $\rightarrow T_4$; V_8 [57,80] $\rightarrow T_5$	1.1592
2	V_8 [80,105]$\rightarrow T_2$; C_1 [19,40] $\rightarrow T_3$; V_2 [33,67] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	0.8406
3	C_1 [66,81]$\rightarrow T_1$; V_1 [72,100] $\rightarrow T_2$; C_2 [19,28] $\rightarrow T_4$; V_8 [57,85] $\rightarrow T_5$	1.8508
4	V_1 [72,102]$\rightarrow T_1$; V_8 [80,105] $\rightarrow T_2$; C_3 [3,19] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; C_1 [85,103] $\rightarrow T_5$	1.2812
5	V_8 [83,103]$\rightarrow T_1$; V_4 [18,49] $\rightarrow T_2$; C_4 [5,24] $\rightarrow T_3$; C_1 [17,37] $\rightarrow T_4$; V_1 [86,110] $\rightarrow T_5$	0.4405
6	C_1 [66,81]$\rightarrow T_1$; V_1 [72,100] $\rightarrow T_2$; C_3 [3,19] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	1.2261
7	V_4 [19,50]$\rightarrow T_1$; V_3 [36,54] $\rightarrow T_2$; C_1 [19,81] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; V_2 [98,116] $\rightarrow T_5$	0.8982
8	V_1 [72,102]$\rightarrow T_1$; V_2 [36,54] $\rightarrow T_2$; C_4 [5,24] $\rightarrow T_3$; C_1 [17,37] $\rightarrow T_4$; V_8 [57,85] $\rightarrow T_5$	0.4259
9	C_2 [53,70]$\rightarrow T_1$; V_8 [80,105] $\rightarrow T_2$; C_3 [3,19] $\rightarrow T_3$; V_2 [33,67] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	1.0117
10	C_2 [53,70]$\rightarrow T_1$; C_1 [64,81] $\rightarrow T_2$; C_4 [5,24] $\rightarrow T_3$; C_3 [4,19] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	0.5069
11	V_1 [72,102], V_8 [83,103] $\rightarrow T_1$; V_3 [36,54], C_1 [64,81] $\rightarrow T_2$; C_3 [3,19], V_1 [32,63] $\rightarrow T_3$; C_2 [19,28]$\rightarrow T_4$; V_7 [34,73], V_8 [57,85] $\rightarrow T_5$	0.8137
12	C_1 [66,81], C_2 [53,70] $\rightarrow T_1$; V_1 [72,100], V_4 [18,49] $\rightarrow T_2$; C_3 [3,19] $\rightarrow T_3$; V_2 [33,67], C_2 [19,28] $\rightarrow T_4$; V_8 [57,85] $\rightarrow T_5$	0.7688
13	V_1 [72,102] $\rightarrow T_1$; V_3 [36,54], V_4 [18,49] $\rightarrow T_2$; V_1 [32,63], C_1 [19,40] $\rightarrow T_3$; V_2 [33,67], C_2 [19,28] $\rightarrow T_4$; V_7 [34,73], V_8 [57,85] $\rightarrow T_5$	0.3925
14	V_1 [72,102], V_3 [34,56] $\rightarrow T_1$; C_2 [52,69], C_1 [64,81] $\rightarrow T_2$; C_3 [3,19], C_4 [5,24] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	0.7409
15	V_3 [36,54], V_4 [18,49] $\rightarrow T_2$; V_1 [32,63], V_2 [35,67] $\rightarrow T_3$; C_1 [17,37], C_2 [19,28] $\rightarrow T_4$; V_7 [57,85], V_8 [85,103] $\rightarrow T_5$	0.4414
16	V_1 [73,102], C_2 [53,70] $\rightarrow T_1$; V_4 [18,49], C_1 [64,81] $\rightarrow T_2$; V_2 [35,67] $\rightarrow T_3$; C_3 [3,19], C_4 [5,24] $\rightarrow T_4$; V_7 [57,85], V_8 [85,103] $\rightarrow T_5$	0.4822
17	V_8 [83,103], C_1 [66,81] $\rightarrow T_1$; V_3 [36,54], C_2 [52,69] $\rightarrow T_2$; V_1 [32,63], V_2 [35,67] $\rightarrow T_3$; C_4 [4,23] $\rightarrow T_4$; V_7 [57,85], V_8 [85,103] $\rightarrow T_5$	0.4350
18	V_3 [34,56], V_8 [83,103] $\rightarrow T_1$; V_4 [18,49], C_1 [64,81] $\rightarrow T_2$; V_1 [32,63], V_2 [35,67] $\rightarrow T_3$; C_3 [33,19], C_4 [4,23] $\rightarrow T_4$	0.5366
19	C_1 [66,81], C_2 [53,70] $\rightarrow T_1$; V_3 [36,54], V_4 [18,49] $\rightarrow T_2$; C_3 [3,19], C_4 [5,24] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; V_1 [86,110] $\rightarrow T_5$	0.7741
20	V_8 [83,103], C_1 [66,81] $\rightarrow T_1$; V_1 [72,100], C_2 [52,69] $\rightarrow T_2$; C_3 [3,19], V_2 [35,67] $\rightarrow T_3$; C_2 [19,28], C_1 [17,37] $\rightarrow T_4$; V_7 [34,73]$\rightarrow T_5$	0.3335
21	V_3 [34,56] $\rightarrow T_1$; V_1 [72,100] $\rightarrow T_2$; C_1 [19,40] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	0.9017
22	V_1 [72,102] $\rightarrow T_1$; C_2 [52,69] $\rightarrow T_2$; C_4 [5,24] $\rightarrow T_3$; C_1 [17,37] $\rightarrow T_4$; V_2 [98,116] $\rightarrow T_5$	0.3820
23	V_4 [19,50] $\rightarrow T_1$; V_3 [36,54] $\rightarrow T_2$; V_2 [35,67] $\rightarrow T_3$; V_1 [33,60] $\rightarrow T_4$; V_8 [57,85] $\rightarrow T_5$	1.0197
24	C_1 [66,81] $\rightarrow T_1$; C_2 [52,69] $\rightarrow T_2$; V_2 [65,67] $\rightarrow T_3$; V_1 [33,60] $\rightarrow T_4$; V_7 [37,73] $\rightarrow T_5$	0.9542
25	V_3 [34,56] $\rightarrow T_1$; V_4 [18,49] $\rightarrow T_2$; V_2 [65,67] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; V_1 [86,110] $\rightarrow T_5$	1.2269
26	V_3 [34,56], V_4 [19,50] $\rightarrow T_1$; V_1 [72,100] $\rightarrow T_2$; C_1 [19,40] $\rightarrow T_3$; C_2 [19,28], C_4 [4,23] $\rightarrow T_4$; V_8 [57,85] $\rightarrow T_5$	0.2974
27	V_8 [83,85] $\rightarrow T_1$; V_1 [72,100], V_4 [18,49] $\rightarrow T_2$; V_2 [35,67] $\rightarrow T_3$; C_1 [17,37] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	0.6656
28	C_1 [66,81], C_2 [53,70] $\rightarrow T_1$; V_3 [36,54], V_4 [18,49] $\rightarrow T_2$; V_1 [32,63] $\rightarrow T_3$; V_2 [33,67] $\rightarrow T_4$; V_8 [57,85] $\rightarrow T_5$	0.9026
29	V_1 [72,102], V_3 [34,56] $\rightarrow T_1$; V_4 [18,49] $\rightarrow T_2$; C_1 [19,40], C_4 [5,24] $\rightarrow T_3$; C_2 [19,28] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	0.7834
30	V_8 [83,103], C_2 [53,70] $\rightarrow T_1$; V_3 [36,54], V_4 [18,49] $\rightarrow T_2$; V_2 [35,67] $\rightarrow T_3$; V_1 [33,60], C_1 [17,37] $\rightarrow T_4$; V_7 [34,73] $\rightarrow T_5$	0.5506
	Mean:	0.2974

TABLE 8.11: *Thirty randomly generated solutions employed to validate the results returned by the simulated annealing algorithm for the single-objective dynamic WAM (6.14)–(6.23) during time stage 23 and onwards of the GBAD scenario of Figure 8.1. The objective represents the priority-weighted accumulated survival probability of threats in each solution. The assignment schedules are given as the WSs, followed by the FW in brackets, followed by a right-arrow indicating to which threat the WSs should be assigned.*

Furthermore, the average priority-weighted accumulated survival probability over the thirty random solutions is 0.7681. When comparing this value to the value of 0.1493 returned by the solution obtained in §8.2.3, it is found that the simulated annealing algorithm is able to achieve an improvement of approximately 81% over the randomly generated set of solutions. Based on these findings it may be concluded that the simulated annealing algorithm is able to provide significantly better solutions to the single-objective dynamic WAM (6.14)–(6.23) than proposing random feasible assignments.

Multi-objective dynamic random benchmark validation

The thirty random solutions generated for the multi-objective dynamic WAM (6.24)–(6.26) are given in Table 8.12. The objective function values for each solution are also given in the table.

Sol	WS-threat pair assignment schedules	Obj 1	Obj 2	Obj 3
1	C_1 [67,74], C_2 [55,59] $\rightarrow T_1$; V_1 [72,84], C_2 [63,69] $\rightarrow T_2$; V_2 [35,65], C_4 [5,8] $\rightarrow T_3$; C_2 [20,24], C_4 [15,20] $\rightarrow T_4$; V_1 [98,102], V_7 [48,69] $\rightarrow T_5$	0.3449	R 4 204 000	3
2	C_4 [23,48], V_8 [86,90] $\rightarrow T_1$; V_1 [88,97], V_8 [96,100] $\rightarrow T_2$; C_1 [19,27], V_1 [45,58] $\rightarrow T_3$; C_2 [19,24], V_1 [34,41] $\rightarrow T_4$; V_1 [102,106], V_7 [45,51] $\rightarrow T_5$	0.4165	R 8 068 000	3
3	C_2 [60,69], V_8 [83,101] $\rightarrow T_1$; V_3 [37,54], V_4 [29,38] $\rightarrow T_2$; V_1 [48,61], V_2 [35,54] $\rightarrow T_3$; C_2 [23,33],C_4 [11,21] $\rightarrow T_4$; C_1 [93,100], V_2 [103,108] $\rightarrow T_5$	0.3021	R 6 136 000	3
4	C_1 [69,78], V_4 [30,33] $\rightarrow T_1$; V_4 [18,25], V_8 [98,102] $\rightarrow T_2$; C_1 [36,39], V_2 [54,57] $\rightarrow T_3$; C_2 [19,27], V_1 [53,58]$\rightarrow T_4$; V_1 [89,99] $\rightarrow T_5$	0.7674	R 7 102 000	3
5	V_1 [90,93], V_3 [35,41] $\rightarrow T_1$; C_1 [69,81], C_2 [66,69] $\rightarrow T_2$; C_3 [14,19], V_1 [34,50] $\rightarrow T_3$; C_1 [18,26], V_1 [57,60] $\rightarrow T_4$; C_1 [93,98], V_1 [100,107] $\rightarrow T_5$	0.2573	R 5 170 000	2
6	V_1 [73,83], V_8 [88,94] $\rightarrow T_1$; V_3 [44,53], V_8 [80,83] $\rightarrow T_2$; C_3 [6,13], C_4 [10,16] $\rightarrow T_3$; C_4 [20,23], V_2 [35,39]$\rightarrow T_4$; V_1 [87,107], V_2 [101,105] $\rightarrow T_5$	0.3017	R 7 102 000	3
7	C_2 [57,68], V_3 [36,55] $\rightarrow T_1$; V_4 [28,45], V_8 [82,92] $\rightarrow T_2$; C_4 [21,24], V_1 [51,54] $\rightarrow T_3$; C_2 [20,23], C_4 [7,17] $\rightarrow T_4$; C_1 [88,96], V_2 [102,110] $\rightarrow T_5$	0.5393	R 5 170 000	4
8	C_2 [53,68], V_1 [84,88] $\rightarrow T_1$; V_1 [92,95], V_8 [84,101] $\rightarrow T_2$; C_1 [21,26], C_3 [5,14] $\rightarrow T_3$; C_4 [9,13], V_2 [44,55] $\rightarrow T_4$; V_7 [51,71], V_8 [58,68] $\rightarrow T_5$	0.2162	R 6 136 000	3
9	C_2 [62,68], V_4 [34,42] $\rightarrow T_1$; V_1 [77,91], V_8 [90,98] $\rightarrow T_2$; C_4 [8,19], V_1 [42,46] $\rightarrow T_3$; C_3 [12,15], V_1 [56,60] $\rightarrow T_4$; C_1 [88,102], V_2 [98,108] $\rightarrow T_5$	0.2666	R 6 136 000	3
10	V_3 [34,55], V_4 [28,38] $\rightarrow T_1$; V_4 [43,47], V_8 [82,94] $\rightarrow T_2$; V_1 [35,48], V_2 [48,60] $\rightarrow T_3$; C_2 [29,34], V_2 [37,42] $\rightarrow T_4$; C_1 [95,101], V_1 [86,93] $\rightarrow T_5$	0.2782	R 8 068 000	3
11	C_2 [54,64], V_3 [49,54] $\rightarrow T_1$; V_1 [72,99], V_4 [36,47] $\rightarrow T_2$; C_1 [32,39], C_4 [10,21] $\rightarrow T_3$; V_1 [50,54], V_2 [43,62] $\rightarrow T_4$; V_7 [60,70],V_8 [67,73] $\rightarrow T_5$	0.2900	R 7 102 000	4
12	C_2 [59,63], V_1 [79,90] $\rightarrow T_1$; V_1 [72,75], V_4 [42,47] $\rightarrow T_2$; C_1 [20,24], V_1 [33,58] $\rightarrow T_3$; C_1 [29,37], C_2 [22,27] $\rightarrow T_4$; V_2 [99,115], V_8 [69,85] $\rightarrow T_5$	0.1892	R 6 136 000	3
13	C_1 [66,71], C_2 [57,63] $\rightarrow T_1$; V_3 [45,54], V_4 [19,26] $\rightarrow T_2$; C_3 [8,12], V_1 [36,41] $\rightarrow T_3$; C_1 [25,34], V_1 [46,53] $\rightarrow T_4$; C_1 [86,99], V_2 [111,116] $\rightarrow T_5$	0.1699	R 5 170 000	2
14	C_1 [72,77], V_3 [44,50] $\rightarrow T_1$; V_4 [40,49], V_8 [82,105] $\rightarrow T_2$; C_1 [19,24], V_2 [35,42] $\rightarrow T_3$; V_1 [100,104], V_2 [47,62] $\rightarrow T_4$; C_1 [92,97] $\rightarrow T_5$	0.3941	R 7 102 000	2
15	V_1 [77,94], V_4 [35,44] $\rightarrow T_1$; C_1 [65,69], C_2 [52,56] $\rightarrow T_2$; C_3 [6,12], C_4 [6,10] $\rightarrow T_3$; C_1 [20,36], V_2 [56,59] $\rightarrow T_4$; C_1 [90,95],	0.2188	R 4 204 000	2

Sol	WS-threat pair assignment schedules	Obj 1	Obj 2	Obj 3
	V_2 [100,107] $\to T_5$			
16	C_1 [68,73], V_8 [99,102] $\to T_1$; V_1 [82,97], V_8 [91,95] $\to T_2$; V_1 [33,50], V_2 [52,66] $\to T_3$; C_1 [31,36], C_4 [14,21] $\to T_4$; V_7 [40,46], V_8 [64,78] $\to T_5$	0.3515	R 7 102 000	3
17	C_2 [53,67], V_8 [90,93] $\to T_1$; V_3 [36,46], V_4 [19,36] $\to T_2$; V_1 [38,57], V_2 [57,60] $\to T_3$; C_1 [19,34], C_3 [6,19] $\to T_4$; V_7 [51,70], V_8 [57,78] $\to T_5$	0.3309	R 7 102 000	3
18	C_2 [61,69], V_4 [23,32] $\to T_1$; C_1 [64,74], V_8 [86,96] $\to T_2$; C_3 [6,15], C_4 [12,18] $\to T_3$; C_4 [4,8], V_2 [45,55] $\to T_4$; V_7 [54,65], V_8 [64,72] $\to T_5$	0.3133	R 5 170 000	3
19	C_2 [56,62], V_1 [80,93] $\to T_1$; V_3 [42,45], V_8 [89,101] $\to T_2$; C_1 [34,38], C_3 [7,11] $\to T_3$; C_1 [27,30], V_1 [36,56] $\to T_4$; V_7 [40,51], V_8 [64,79] $\to T_5$	0.1184	R 6 136 000	3
20	V_1 [77,93], V_3 [38,45] $\to T_1$; C_1 [64,77], V_4 [27,33] $\to T_2$; C_1 [23,27], V_1 [41,59] $\to T_3$; C_1 [32,35], C_3 [8,17] $\to T_4$; C_1 [92,103], V_8 [64,79] $\to T_5$	0.1012	R 5 170 000	1
21	V_1 [20,28], V_8 [84,101] $\to T_1$; C_1 [68,71], V_4 [36,41] $\to T_2$; C_3 [6,17], C_4 [19,24] $\to T_3$; C_2 [19,25], C_12 [11,15] $\to T_4$; V_2 [104,112], V_8 [65,76] $\to T_5$	0.2747	R 5 170 000	3
22	V_4 [20,28], V_8 [86,100] $\to T_1$; C_1 [66,74], V_1 [76,98] $\to T_2$; C_1 [34,37], V_1 [56,59] $\to T_3$; C_1 [21,30], V_2 [44,58] $\to T_4$; V_2 [109,116], V_8 [61,68] $\to T_5$	0.1727	R 7 102 000	2
23	C_2 [63,66], V_4 [26,48] $\to T_1$; V_3 [38,44], V_8 [80,102] $\to T_2$; C_1 [22,26], V_1 [42,50] $\to T_3$; C_2 [20,27], C_4 [9,19] $\to T_4$; V_1 [90,104], V_2 [99,112] $\to T_5$	0.1757	R 6 136 000	4
24	C_2 [57,67], V_4 [33,43] $\to T_1$; V_1 [78,83], V_4 [25,28] $\to T_2$; C_3 [5,12], V_2 [42,60] $\to T_3$; C_2 [20,27], C_4 [12,23] $\to T_4$; C_1 [89,96], V_1 [88,102] $\to T_5$	0.2709	R 5 170 000	3
25	C_2 [64,67], V_3 [34,47] $\to T_1$; C_1 [66,73], C_2 [52,59] $\to T_2$; C_1 [20,27], C_4 [18,23] $\to T_3$; C_1 [31,37], C_4 [4,14] $\to T_4$; V_2 [109,113], V_7 [36,47] $\to T_5$	0.0642	R 3 238 000	2
26	V_3 [39,51], V_4 [23,43] $\to T_1$; C_2 [64,67], V_1 [76,97] $\to T_2$; C_3 [9,16], V_1 [48,55] $\to T_3$; C_1 [31,36], C_2 [23,28] $\to T_4$; V_2 [108,111], V_7 [42,71] $\to T_5$	0.4167	R 6 136 000	3
27	V_3 [48,52], V_8 [86,99] $\to T_1$; C_1 [69,76], V_1 [93,98] $\to T_2$; C_4 [12,22], V_2 [36,46] $\to T_3$; C_4 [4,8], V_2 [52,62] $\to T_4$; V_1 [105,108], V_7 [49,68] $\to T_5$	0.5724	R 7 102 000	4
28	C_1 [68,72], V_4 [20,24] $\to T_1$; V_3 [36,46], V_4 [34,39] $\to T_2$; C_1 [35,38], C_4 [8,13] $\to T_3$; C_3 [2,8], V_2 [42,52] $\to T_4$; V_1 [87,110], V_2 [100,107] $\to T_5$	0.2336	R 6 136 000	3
29	C_4 [27,35], V_3 [49,56] $\to T_1$; C_2 [52,59], V_8 [88,99] $\to T_2$; C_4 [14,17], V_2 [37,43] $\to T_3$; C_2 [24,27], C_4 [20,25] $\to T_4$; V_1 [96,100], C_1 [90,100] $\to T_5$	0.0901	R 5 170 000	3
30	V_8 [85,93], C_2 [53,66] $\to T_1$; V_1 [78,81], V_8 [100,104] $\to T_2$; C_4 [7,11], V_2 [42,62] $\to T_3$; C_2 [20,25], C_4 [15,21] $\to T_4$; C_1 [90,100], V_7 [37,68] $\to T_5$	0.2778	R 5 170 000	4
	Mean:	0.2906	R 5 534 207	2.9

Table 8.12: *Thirty randomly generated solutions employed to validate the results obtained by the NSGA II (described in §5.4.2) for the multi-objective dynamic WAM (6.24)–(6.26) during time stage 23 of the GBAD scenario of Figure 8.1. Obj 1 represents the threat-priority weighted accumulated survival probability objective function values, Obj 2 represents the cost objective function values and Obj 3 represents the least re-engagable WS objective function values. The assignment schedules are given as the WS, followed by the FW in brackets, followed by a right-arrow indicating to which threat the WSs should be assigned.*

Upon sorting the thirty solutions in Table 8.12 by means of the FNSA described in §4.4, five nondominated solutions are obtained. These solutions, together with the remaining 25 randomly generated solutions, are illustrated graphically in objective function space in Figure 8.11. In addition to the notations described in legend used in the validation of the multi-objective static WAM (6.10)–(6.12), solutions in which the least re-engageable WSs can re-engage once after the assignment are denoted by open diamonds, while solutions in which the least re-engageable WSs can re-engage four times after the assignment are denoted by open circles — the nondominated randomly generated solutions are joined by broken line segments.

The approximately Pareto optimal solutions returned by the evolutionary algorithm are also plotted in objective function space in Figure 8.11. The same symbol conventions defined in the legend used during the validation of the multi-objective static WAM (6.10)–(6.12) are again adopted here. In addition, solutions in which the least re-engageable WSs can re-engage once after the assignment are denoted by solid diamonds, while solutions in which the least re-engageable WSs can re-engage four times after the assignment are denoted by solid circles (again joined by solid lines).

When comparing the approximately Pareto front achieved by the evolutionary algorithm with the nondominated front of randomly generated solutions, it is clear that the evolutionary algorithm is able to provide a much better and wide-spread Pareto front than the nondominated front of the randomly generated solutions. The gap achieved between the two fronts also seems satisfactorily wide.

The hypervolume technique described in §4.4 was also used to compare the hypervolume of the approximately Pareto optimal solutions obtained in Figure 8.6 with the hypervolume of the set of nondominated randomly generated solutions. A reference point involving a 0.35 value in the priority-weighted accumulated survival probability, a value of $R6\,150\,000$ in the cost objective and a value of 1 in the re-engageability objective was adopted for hypervolume calculation purposes. A hypervolume measure of 2.3251×10^6 was obtained for the approximately Pareto optimal solutions in Table 8.9 (the solutions returned by the evolutionary algorithm for the WAM (6.24)–(6.26)) and a corresponding measure of 1.1641×10^6 was obtained for the nondominated set of randomly generated solutions. This indicates that the solutions returned by the evolutionary algorithm again dominate a significantly larger portion of the objective function space than the nondominated set of randomly generated solutions. It may therefore again be concluded that the evolutionary algorithm is able to provide significantly better solutions to the multi-objective dynamic WAM (6.24)–(6.26) than proposing random feasible assignments.

8.3.3 Military expert validation

The final method of validation involved having individual consultations with two military experts who were asked to solve (by hand) each of the four WAM prototypes of §6.5 in the context of the simulated GBAD scenario described in §8.1. The results of their solutions were then compared with the results returned by the algorithms used in §8.2 to solve the WAM prototypes.

The first military expert is Dr Jaco Roux [160] who worked for Reutech Radar Systems at the time when this project was started. Dr Roux is a software engineer who was part of the team responsible for developing a South African-based GBAD DSS for the South African National Defence Force. In 2010, Dr Roux obtained his PhD in which he proposed the design of a TE subsystem for use in a GBAD environment — the counterpart of the subsystem proposed in this dissertation. Since he started working at Reutech Radar Systems, Dr Roux worked on the South African GBAD DSS programme for eight years and was involved in the integration and testing of the TEWA DSS of the *European Aeronautic Defence and Space Company*, configuring this DSS for the South African National Defence Force and developing a South African GBAD TEWA system as a concept demonstrator. He also collaborated with the Council for Scientific

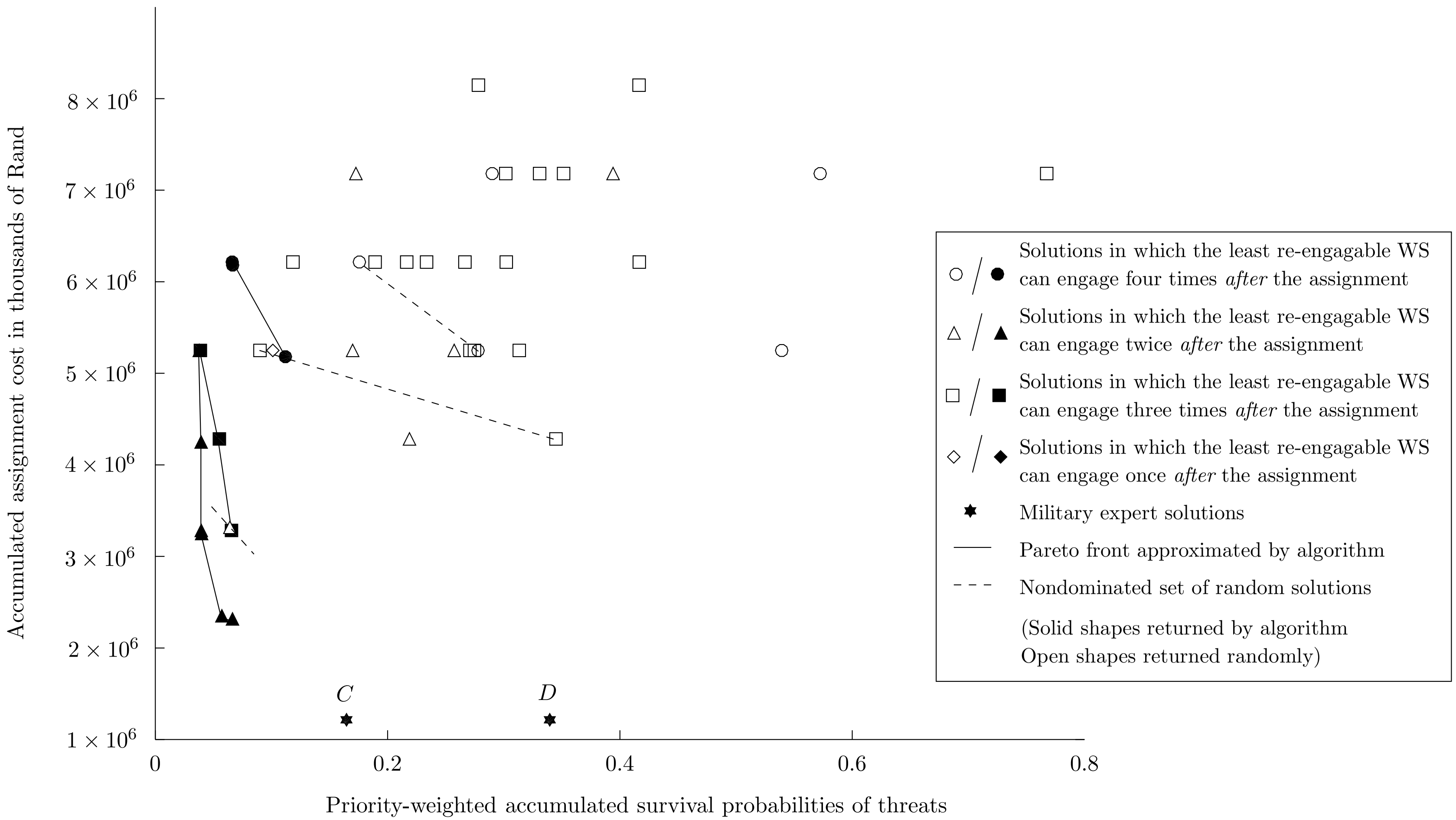


FIGURE 8.11: *Approximately Pareto optimal solutions returned by the evolutionary algorithm for the multi-objective dynamic WAM (6.24)–(6.26) in objective function space for the GBAD scenario of §8.1, as well as the thirty randomly generated solutions in Table 8.10 together with two solutions returned by military experts.*

and Industrial Research by integrating the TEWA system into their military simulator, conducted various workshops with military experts and advised the AD user doctrine development committee in Kimberley — he also studied all the air defence artillery doctrinal notes and documentation for GBADS. Furthermore, he completed an FCO training course and attended various military exercises at Roodewal Bombaan (in Polokwane), at the airforce base in Bredasdorp, and at the Council for Scientific and Industrial Research headquarters (in Pretoria).

The second military expert is Lieutenant Colonel Bob Visser [200], a former FCO within the South African National Defence force, who currently leads the Cape Garrison Artillery, a reserve force regiment of the South African Army AD artillery formation. In addition to leading the regiment, Lieutenant Colonel Visser is also employed at Reutech Radar Systems in the capacity of systems engineer.

During the period in which Dr Roux worked at Reutech Radar systems, Lieutenant Colonel Visser also worked at the same institution. He worked in conjunction with Dr Roux on the integration and testing of the European Aeronautic Defence and Space Company's TEWA DSS within a South African GBAD context. In addition, he provided valuable input in the design of the South African GBAD DSS programme. During the development of the South African GBAD programme, Lieutenant Colonel Visser also assisted Dr Roux in setting up and attending workshops with other military experts.

During each of the consultations, the hypothetical GBAD scenario was described to a military expert in detail. This included the layout of the WS and DA deployment, the groupings of the incoming threats as well as the attack techniques that the threats are anticipated to execute. Furthermore, all the data involved in the scenario were provided to the expert. These data included the threat priority values of the threats during time stages $\tau = 23$ and 39, the EEMs for these time stages, the cost of a single burst of ammunition for CIWSs and VSHORADs, as well as the ammunition available at each WS in the deployment. It was also mentioned that for the dynamic WAMs, a straight-line flight path prediction model is assumed and the EEMs for the 120 future time stages from time stage $\tau = 23$ onwards were shown to the military expert. Each of the four WAM prototypes was finally described to the military expert in detail, including the rationale behind each of the objective functions as well as the constraints of each model.

Single-objective static military expert validation

For the single-objective static WAM (6.6)–(6.9), military expert Dr Roux explained that he would assign the WS achieving the highest SSHP value with respect to the threat involving the highest priority value first, followed by assigning the WS achieving the second highest SSHP with respect to the threat involving the second highest priority value, and so forth. He also added that in this modelling paradigm, he would assign the maximum number of WSs in order to inflict as much damage to the threats as possible — in order to achieve the maximum accumulated priority-weigthed survival probability. He therefore proposed the following WS-threat assignment pairs: WS C_1 to threat T_1, WS C_2 to threat T_2, WS V_1 to threat T_3, WS V_2 to threat T_4, and WSs V_6 and V_7 to threat T_5. This assignment yields an accumulated priority-weighted survival probability of 2.171, as shown in Figure 8.13. When comparing this result with the results returned by the genetic algorithm in §8.2.1 (*i.e.* an accumulated priority-weighted survival probability of 1.9628), it may be concluded that the genetic algorithm is able to achieve an approximately 9.5% improvement over the solution proposed by the military expert.

Dr Roux was also asked to comment on the results returned by the genetic algorithm in §8.2.1. He concluded that the results are good in terms of the quality of the WS-threat assignment pairs

Threat	Threat Priority $V_j(\tau)$	WSs assigned	Survival values $q_{ij}(\tau)$	Prioritised survival probability
T_1	0.99	C_1	0.8	0.99×0.8
T_2	1.00	C_2	0.1	1.00×0.1
T_3	0.76	V_1	0.9	0.76×0.9
T_4	0.74	V_2	0.5	0.74×0.5
T_5	0.50	V_6, V_7	$0.9, 0.5$	$0.5 \times 0.9 \times 0.5$
			Total:	2.171

TABLE 8.13: *WS to threat assignment list proposed by military expert Roux [160] during a consultation session in which he was asked to solve the single-objective WAM prototype (6.6)–(6.9) by hand within the context of the GBAD scenario presented in Figure 8.1 for time stage* $\tau = 39$.

proposed by the algorithm. He noticed that a number of solutions proposed by the algorithm coincided exactly with his own assignment proposals. Furthermore, he was not surprised that the algorithm was able to outperform him. He added that this should be the case and that the aim of using a WAM is to provide an FCO with DS rather than taking over the role of the FCO — the results returned by the WAM should confirm operator judgement and be able to improve upon it.

Next, Lieutenant Colonel Bob Visser was consulted to perform the same exercise. He commented on the hypothetical scenario of §8.1 and highlighted that from a practical point of view, threats T_3, T_4 and T_5 have already advanced to such a stage in the GBAD scenario at time stage $\tau = 39$ that they would already have executed their attack manoeuvres in order to release their weapons towards the DAs — the threats had already passed the DAs at that point in time. He therefore did not consider these aircraft as threats any more. According to him, the only threats remaining in the system at that point were threats T_1 and T_2 — they are at the point of readying themselves to attack the DAs. He noted that he would assign WS V_3 to threat T_1 and WS C_1 to threat T_2. The reason for assigning WS V_3 to threat T_1 is that the threat may then be attacked from behind (should a fire order be sent) and the reason for assigning WS C_1 to threat T_2 is that it is further away from threat T_2 than WS C_2 at $\tau = 39$, which will give the operator enough time to react to the situation. The assignment proposal of Lieutenant Colonel Bob Visser is summarised in Table 8.14.

Threat	Threat Priority $V_j(\tau)$	WSs assigned	Survival values $q_{ij}(\tau)$	Prioritised survival probability
T_1	0.99	V_3	0.9	0.99×0.9
T_2	1.00	C_1	0.6	1.00×0.6
T_3	0.76	None	1	0.76×1
T_4	0.74	None	1	0.74×1
T_5	0.50	None	1	0.5×1
			Total:	3.491

TABLE 8.14: *WS to threat assignment list proposed by military expert Visser [200] during a consultation session in which he was asked to solve the single-objective WAM prototype (6.6)–(6.9) by hand within the context of the GBAD scenario presented in Figure 8.1 for time stage* $\tau = 39$.

When comparing the solution in Table 8.14 with the solution returned by the genetic algorithm in §8.2.1, it is found that the algorithmic solution is significantly better than the solution of Lieutenant Colonel Visser. The reason for this finding is that Lieutenant Colonel Visser did not consider aircraft T_3, T_4 and T_5 as threats and hence their accumulated priority-weighted survival probabilities are rather high (no WSs are assigned to them). Lieutenant Colonel Visser was also asked to comment on the results returned by the genetic algorithm. He was quite impressed with the small accumulated priority-weighted survival probability achieved by the algorithm and understood why the algorithm was able to outperform him — the model takes into account the threat priority values and EEM for the time stage, ignoring the fact that the threats had already passed the DAs at time stage $\tau = 39$. Lieutenant Colonel Visser also commented that the WS-threat assignment pairs proposed by the algorithm are of a high quality.

Multi-objective static military expert validation

After the working of the multi-objective static WAM was explained to the military experts during their individual consultation sessions, they found the concept of multiple simultaneous objectives in a WAM rather strange. Their first reaction was that in a combat situation, an FCO would typically not consider the cost of an assignment important at all — FCOs are typically trained to assign WSs so as to inflict the maximum damage (*i.e.* minimising survival probability). They did, however, acknowledge the importance of including the re-engageability objective — a WS that runs out of ammunition during a combat situation can result in a gap in the own force's defence which may have disastrous consequences should a WS not be available for later assignments due to such a lack of ammunition. Both the military experts mentioned that they deem the accumulated priority-weighted survival probability objective as the most important, followed by the re-engageability as the second-most important and lastly (if a must), the cost objective. The author then explained to the military experts that the aim in multi-objective optimisation problems is to find a number of Pareto optimal solutions, illustrating the trade-offs between the various solutions. These solutions may then be presented to the FCO from which he may choose one, based on subjective judgement, for implementation purposes. It was also explained to the experts that no weights are associated with the objective functions and that the multi-objective WAMs are able to illustrate the trade-offs between the objective functions clearly. They remained sceptical about the multi-objective WAM prototypes, but they nevertheless proceeded to solve the multi-objective static WAM prototype.

Dr Roux proposed similar assignments for the multi-objective static WAM prototype (6.10)–(6.12) as for the single-objective static WAM prototype (6.6)–(6.9). The WS-threat assignment pairs he selected are as follows: WS C_1 should be assigned to engage threat T_1, WS C_2 should be assigned to engage threat T_2, WS V_1 should be assigned to threat T_3, WS V_2 should be assigned to threat T_4 and only WS V_7 should be assigned to threat T_5 (not WS V_6 as was the case in the single-objective static WAM prototype (6.6)–(6.9)). His assignment yielded a priority-weighted accumulated survival probability of 2.196 at an assignment cost of $R\,3\,068\,000$, as illustrated in Table 8.15. The least re-engageable WSs after the assignment are WSs C_1 and V_7, which each has 2 units of ammunition left *after* the assignment. This solution (labelled A) is illustrated graphically in objective function space in Figure 8.9. From the figure it may be deduced that the solution does not lie on the Pareto front approximated by the NSGA II, implying that the solution is dominated.

Dr Roux explained that he would follow the same procedure in assigning WSs to threats as in the single-objective static WAM prototype (6.6)–(6.9), *i.e.* assigning WS achieving high SSHP to high priority threats first, and that he would then consider WS re-engageability when making

Threat	Threat Priority $V_j(\tau)$	WSs assigned	Survival values $q_{ij}(\tau)$	Prioritised survival probability	Accumulated cost in Rand	Re-engage-ability after the assignment
T_1	0.99	C_1	0.8	0.99×0.8	34 000	**2**
T_2	1.00	C_2	0.1	1.00×0.1	34 000	3
T_3	0.76	V_1	0.9	0.76×0.9	1 000 000	3
T_4	0.74	V_2	0.5	0.74×0.5	1 000 000	3
T_5	0.50	V_7	0.5	0.50×0.5	1 000 000	**2**
			Totals:	2.196	3 068 000	2

TABLE 8.15: *WS to threat assignment list proposed by military expert Roux [160] during a consultation session in which he was asked to solve the multi-objective static WAM prototype (6.10)–(6.12) by hand within the context of the GBAD scenario presented in Figure 8.1 for time stage $\tau = 39$.*

assignments (*i.e.* assigning WSs with more ammunition first). He would consider the cost of assignments last.

When asked to comment on the results returned by the algorithm, Dr Roux was very impressed by the way in which the model is able to illustrate the trade-offs between the objective functions. He liked the idea that small decreases in one objective may lead to large improvements in another objective function. Consider, for example, Solution 2 in Table 8.6 achieving an accumulated priority-weighted survival probability objective value of 1.9878 at a cost of $R\,4\,068\,000$ and Solution 3 achieving an accumulated priority-weighted survival probability value of 2.0116 at a cost of $R\,3\,068\,000$ in Figure 8.4. An approximately 1% decrease in the accumulated priority-weighted survival probability leads to an approximately 25% decrease in the cost objective. In this way more cost-effective decisions may be made.

Although Dr Roux genuinely seemed to be impressed by the results returned by the NSGA II, he did, however, question the practicality of the implementation of the multi-objective static WAM prototype (6.10)–(6.12), explaining that he was worried the operator may be overwhelmed by all the solutions from which he was required to choose one. It was then explained to him that sufficient implementation suggestions (as discussed in Chapter 7) had, in fact, been proposed to counter this concern and that, if implemented correctly, would not have a negative effect on operator judgement.

Lieutenant Colonel Visser proposed the exact same assignments for the multi-objective static WAM prototype (6.10)–(6.12) as for the single-objective static WAM prototype (6.6)–(6.9). He therefore proposed that WS V_3 should be assigned to threat T_1 and that WS C_1 be assigned to threat T_2. His assignment again yielded an accumulated priority-weighted survival probability value of 3.4910 at an assignment cost of $R1\,034\,000$. The least re-engageable WS after the assignment is WS C_1, which can engage twice after the assignment. This assignment is summarised in Table 8.16. Lieutenant Colonel Visser again reiterated that he does not consider cost to be important in assigning WSs to threats. He did, however, mention that he would not use WS C_1, unless he absolutely had to, because WS C_1 only has three bursts of ammunition left. In this case, however, he felt it necessary to assign the WS to the threat since the WS would still have two bursts of ammunition left over after the assignment. Furthermore, Lieutenant Colonel Visser mentioned again that he did not consider aircraft T_3, T_4 and T_5 as threats since at time stage $\tau = 39$ they would have already attacked the DAs if they had in the first place intended to do so.

Threat	Threat Priority $V_j(\tau)$	WSs assigned	Survival values $q_{ij}(\tau)$	Prioritised survival probability	Accumulated cost in Rand	Re-engage-ability after the assignment
T_1	0.99	V_3	0.9	0.99×0.9	1 000 000	4
T_2	1.00	C_1	0.6	1.00×0.6	34 000	**2**
T_3	0.76	None	—	0.76×1	—	—
T_4	0.74	None	—	0.74×1	—	—
T_5	0.50	None	—	0.50×1	—	—
			Totals:	3.4910	1 034 000	2

TABLE 8.16: *WS to threat assignment list proposed by military expert Visser [200] during a consultation session in which he was asked to solve the multi-objective static WAM prototype (6.10)–(6.12) by hand within the context of the GBAD scenario presented in Figure 8.1 by hand for time stage $\tau = 39$.*

Lieutenant Colonel Visser's solution (labelled B) is also plotted in objective function space in Figure 8.9. When comparing his solution with the remaining solutions returned by the NSGA II, it is clear that Lieutenant Colonel Visser's solution also does not lie on the approximate Pareto front — this is again due to the fact that he did not consider aircraft T_3, T_4 and T_5 as threats, resulting in a higher accumulated priority-weighted survival probability for these threats.

Upon showing the results returned by the NSGA II to Lieutenant Colonel Visser, he also liked the idea of being able to quantify the trade-offs between the objective functions as one moves along the approximate Pareto front. He also raised the concern of not overwhelming the FCO with information. The implementation suggestions (as discussed in §7) to counter this concern were explained to Mr Visser and he seemed satisfied with this.

Single-objective dynamic military expert validation

Upon explaining the working of the dynamic WAMs to the military experts, they both commented that it is nearly impossible for a human simply to look at the EEMs (there are 120 EEMs from time stage $\tau = 23$ onwards in the GBAD scenario of Figure 8.1) and to propose WS-threat assignment pairs — the combinations of WS-threat pairs in conjunction with the future time stage during which these assignments should occur are too many for a human to consider in the short time frames available to an FCO.

Dr Roux explained that rather than having to work through 120 EEMs for the predicted future time continuum from time stage $\tau = 23$ onwards, he would give the author a clear strategy that he would have followed. He explained that for a specific WS-threat pair, he would wait until the WS achieves an SSHP value of at least 0.6 with respect to the threat before he assigns the WS to the threat. Once the threat is assigned a WS, he would keep the assignment until the SSHP value drops to 0.2. The author therefore analysed the 120 EEMs produced for time stage $\tau = 23$ in order to find WS-threat assignment pairs based on the information provided by Dr Roux. The following WS-threat assignment pairs were thus constructed (summarised in Table 8.17): WS C_2 should be assigned to threat T_1 during the FW $[57, 69]$, while WS C_1 should be assigned to threat T_2 during the FW $[68, 80]$. WSs V_1 and C_4 should be assigned to threat T_3 during the FWs $[37, 45]$ and $[9, 24]$, respectively. Furthermore, WSs C_1 and C_3 should be assigned to threat T_4 during the FWs $[21, 28]$ and $[7, 19]$, respectively. Finally, WSs V_2 and V_8 should be assigned to threat T_5 during the FWs $[103, 144]$ and $[61, 73]$, respectively. Utilising the threat priority values in Table 8.2 and the fixed-mean values in Table 8.7, an accumulated weighted-priority survival probability value of 0.1773 was calculated for this assignment. This yielded a worse objective function value than the value 0.1493 returned by the simulated annealing algorithm. The improvement achieved by the algorithm seems to be small in view of the complexity of the problem at hand. The solution constructed on the advice of a military expert, however, had the advantage of fixed mean values which had already been computed. In a real-life scenario, these values would have to be computed by the FCO.

Threat	WS-threat pair assignement schedules	Prioritised survival probability
T_1	$C_2[57, 69]$	0.2×0.192
T_2	$C_1[68, 80]$	0.22×0.201
T_3	$V_1[37, 45]$,$C_4[9, 24]$	$1 \times 0.367 \times 0.075$
T_4	$C_1[21, 28]$,$C_3[7, 19]$	$1 \times 0.052 \times 0.238$
T_5	$V_2[103, 114]$,$V_8[61, 73]$	$0.3 \times 0.435 \times 0.420$
	Total:	0.1773

TABLE 8.17: *WS-to-threat assignment list proposed by military expert Roux [160] during a consultation session in which he was asked to solve the single-objective dynamic WAM prototype (6.14)–(6.23) by hand within the context of the GBAD scenario presented in Figure 8.1 for time stage $\tau = 23$. The assignment pairs are given as the threat, followed by the WS and the time stages during which the assignment should occur.*

When asked to comment on the results returned by the simulated annealing algorithm for the single-objective dynamic WAM (6.14)–(6.23), Dr Roux commented that the results seem to be of a high quality and that he expected the model to provide a better solution than the one he proposed. Furthermore, he liked the idea of the scheduling element involved in the model, mentioning that this is a considerable advantage over the static WAMs.

In a similar fashion, Lieutenant Colonel Visser also gave the author a strategy that he would have followed rather than proposing actual WS-threat assignment schedules. He explained that, in addition to using SSHP values for proposing WS-threat assignment pairs, he would also have considered the distance between the WS and the threat. If a threat is too close to a WS, he would not assign it to the threat, since the FCO may typically not have enough time to engage the threat. He appreciated the choice of time stage $\tau = 23$ for the dynamic models since all the aircraft may now be considered as threats to the DAs. He concluded that he would have proposed assignments in such a way that once a threat is in range of a WS, the WS should be assigned to the threat, keeping the assignment until the threat is out of range of the WS. Based on this information, the following WS-threat assignment schedule was constructed: WSs C_1 and V_3 should be assigned to threat T_1 during the FWs $[66, 81]$ and $[34, 56]$, respectively. WSs C_2 and V_4 should be assigned to threat T_2 during the FWs $[52, 69]$ and $[18, 49]$, respectively. Furthermore, WS C_1 should be assigned to threat T_3 during the FW $[19, 40]$ and WS C_3 should be assigned to threat T_4 during the FW $[5, 19]$. Finally, WS V_8 should be assigned to threat T_5 during the FW $[57, 85]$. This assignment schedule is summarised in Table 8.18. Again utilising the threat priority values in Table 8.2 and the fixed-mean values in Table 8.7, an accumulated weighted-priority survival probability value of 0.4569 was calculated for the assignment. This solution yielded a considerably worse objective function value than the value of 0.1493 returned by the simulated annealing algorithm.

When asked to comment on the solution returned by the simulated annealing algorithm, Lieutenant Colonel Visser commented that the solution returned by the algorithm is of a good quality and that he was not at all surprised that the algorithm outperformed him — he expected this

Threat	WS-threat pair assignement schedules	Prioritised survival probability
T_1	$V_3[34, 56]$,$C_1[66, 81]$	$0.2 \times 0.328 \times 0.371$
T_2	$C_2[52, 69]$,$V_4[18, 49]$	$0.22 \times 0.156 \times 0.338$
T_3	$C_1[19, 40]$	1×0.057
T_4	$C_3[5, 19]$	1×0.238
T_5	$V_8[57, 85]$	0.3×0.420
	Total:	0.4569

TABLE 8.18: *WS-to-threat assignment list proposed by military expert Visser [200] during a consultation session in which he was asked to solve the single-objective dynamic WAM prototype (6.14)–(6.23) by hand within the context of the GBAD scenario presented in Figure 8.1 for time stage $\tau = 23$. The assignment pairs are given as the threat, followed by the WS and the time stages during which the assignment should occur.*

due to the complexity of the WAM. He was particularly impressed by the fact that the algorithm was able to solve the model instance almost instantaneously.

Multi-objective dynamic military expert validation

The same procedure was followed in validating multi-objective dynamic WAM (6.24)–(6.26) as for the single-objective dynamic WAM (6.14)–(6.23). The military experts again raised their concerns about the multi-objective optimisation involved in the problem, as well as the anticipated difficulty of solving the model due to the complexity introduced by the dynamic element. Both military experts again provided strategies by which they would have found assignment schedules, rather than having to work through the EEMs from time stage $\tau = 23$ onwards.

Dr Roux stated that he would follow nearly the same strategy as for solving the single-objective dynamic WAM (6.14)–(6.23). In solving the multi-objective dynamic WAM (6.24)–(6.26), he would, however, first consider the survival probabilities of the threats. If two WSs achieve SSHP values that are very close to one another with respect to the same threat, he would rather assign the WS involving the largest ammunition ration. Then only would he consider the cost of the assignment (that is, if two WSs were to achieve SSHP values that are very close to one another with respect to the same threat and they have the same amount of ammunition available to them, he would assign the WS incurring the lowest cost).

In view of the strategy provided above, the following assignment schedule was constructed for the multi-objective dynamic WAM (6.24)–(6.26): WS C_2 should be assigned to threat T_1 during the FW $[59, 69]$, followed by the assignment of WS C_1 to threat T_2 during the FW $[68, 80]$. WS C_4 should be assigned to threat T_3 during the FW $[9, 24]$ and WS C_1 should be assigned to threat T_4 during the FW $[21, 28]$. Finally, WS V_8 should be assigned to threat T_5 during the FW $[61, 73]$. This assignment schedule is summarised in Table 8.19. The schedule yielded an accumulated priority-weighted survival probability of 0.1645 at a cost of $R1\,136\,000$. The least re-engageable WS after the assignment was WS C_1, which can re-engage three times *after* the assignment. This solution (labelled C) is illustrated graphically in objective function space in Figure 8.11. From the figure, it is clear that the solution is dominated by the solutions returned by the NSGA II.

Lieutenant Colonel Visser proposed almost exactly the same strategy as Dr Roux, resulting in the following WS-threat assignment schedule for the multi-objective dynamic WAM (6.24)–(6.26):

Threat	WS-threat pair assignement schedules	Prioritised survival probability	Accumulated cost in Rand value	Re-engage-ability after the assignment
T_1	$C_2[57, 69]$	0.0078	R 34 000	5
T_2	$C_1[68, 80]$	0.0026	R 34 000	4
T_3	$C_4[9, 24]$	0.0606	R 34 000	5
T_4	$C_1[21, 28]$	0.0035	R 34 000	**3**
T_5	$V_8[61, 73]$	0.0900	R 1 000 000	8
	Total:	0.1645	R 1 136 000	3

Table 8.19: *WS to threat assignment list proposed by military expert Roux [160] during a consultation session in which he was asked to solve the multi-objective dynamic WAM prototype (6.24)–(6.26) by hand within the context of the GBAD scenario presented in Figure 8.1 for time stage $\tau = 23$.*

WS C_2 should be assigned to threat T_1 during the FW $[58, 64]$, followed by the assignment of WS C_1 to threat T_2 during the FW $[69, 78]$. WS C_1 should again be assigned to threat T_3 during the FW $[23, 27]$ and WS C_4 should be assigned to threat T_4 during the FW $[13, 18]$. Finally, WS V_8 should be assigned to threat T_5 during the FW $[61, 69]$. These assignments are summarised in Table 8.20. The assignments yielded an accumulated priority-weighted survival probability of 0.3396 at a cost of R1 136 000. The least re-engageable WS after the assignment is WS C_1, which can re-engage three times *after* the assignment. This solution (labelled D) is also illustrated graphically in Figure 8.11. From the figure, it is clear that the solution is also dominated by the solutions returned by the evolutionary algorithm.

Threat	WS-threat pair assignement schedules	Prioritised survival probability	Accumulated cost in Rand value	Re-engage-ability after the assignment
T_1	$C_2[58, 64]$	0.0078	R 34 000	5
T_2	$C_1[69, 78]$	0.0026	R 34 000	4
T_3	$C_1[23, 27]$	0.0419	R 34 000	**3**
T_4	$C_4[13, 18]$	0.1973	R 34 000	5
T_5	$V_8[61, 69]$	0.0900	R 1 000 000	8
	Total:	0.3396	R 1 136 000	3

Table 8.20: *WS to threat assignment list proposed by military expert Visser [200] during a consultation session in which he was asked to solve the multi-objective dynamic WAM prototype (6.24)–(6.26) by hand within the context of the GBAD scenario presented in Figure 8.1 for time stage $\tau = 23$.*

Both the military experts reiterated that the aim of the DS provided to FCOs should be to confirm operator judgement, and not to replace the operator. They both also acknowledged the significance of the four WAM prototypes — especially the dynamic WAM prototypes, which are able to consider future time stages for WS-threat assignment proposals. If the WAM prototypes proposed in this dissertation can be implemented efficiently, they should be able to provide time for the FCOs to focus on other important WA-related decisions rather than having to solve incarnations of the WAP. The military experts also seemed to be more at ease with the multi-objective WAMs after the consultations, stating that the trade-offs between the objective functions that these WAMs are able to demonstrate may lead to more cost-effective WA decisions, since it has been shown that a small portion of the accumulated priority-weighted survival probability objective is sometimes given up for a large saving in the cost objective. Based on the findings above, it may be concluded that the military experts were relatively satisfied with the way in which the models work and the results produced by these models.

8.4 Chapter summary

This chapter was devoted to illustrations of the working of the WAM prototypes chosen for default inclusion in the WA design architecture proposed in Chapter 6. The chapter opened with a detailed description of a simulated GBAD scenario in §8.1. This scenario was adopted as the context within which each of the four WAM prototypes of §6.5 was solved by appropriate solution methodologies, as described in §5. Section 8.2 was dedicated to a presentation of the numerical results returned by the various algorithms used to solve the four WAM prototypes. The algorithmic parameter values employed in each case were also documented in this section. First, the numerical results obtained for the single-objective static WAM prototype were provided in §8.2.1, and this was followed by a presentation of the the numerical results obtained for the multi-objective static WAM prototype in §8.2.2. Next, the results obtained for the single-objective dynamic WAM prototype were provided in §8.2.3, and this was followed by a presentation of the numerical results obtained for the multi-objective dynamic WAM prototype in §8.2.4.

The remainder of the chapter was concerned with a validation of the numerical results reported in §8.2. Three methods of validation were employed for this purpose in §8.3. First, a face validation was carried out in §8.3.1 with a view to commenting on the quality and realism of the results obtained for each WAM prototype. Next, a random benchmark validation was performed in §8.3.2 by generating a series of random solutions for each WAM prototype and comparing the quality of the randomly generated solutions with that of the results returned by the various algorithms for the WAM prototypes. The final validation process involved a consultation with two military experts in which they were asked to solve each of the WAMs within the context of the simulated scenario in Figure 8.1. Their WA proposals were compared in §8.3.3 with the results obtained by the algorithms for the WAM prototypes.

Part IV

Conclusion

CHAPTER 9

Dissertation summary

Contents

This summative chapter comprises two sections. A detailed summary of the contents of the dissertation is provided in a chapter-by-chapter overview fashion in §9.1, while the various novel contributions of the dissertation are highlighted in §9.2, which also contains an appraisal of the value and significance of these contributions.

9.1 Dissertation contents

Apart from the last part, Part IV, this dissertation comprises a total of eight chapters, the last seven of which have been partitioned into three parts. The exception is the stand-alone introductory chapter, Chapter 1, which was devoted to providing the reader with a background context for the work to follow in the document. After an opening background section in this first chapter, highlighting the need for carefully designed and tested TEWA DSSs from a naval perspective, an informal description was given of the problem to be considered throughout the dissertation, namely the design and application of a practical, generic WA subsystem which can function as part of such an integrated TEWA DSS. This was followed by a presentation of the nine research objectives to be pursued in the dissertation and a delimitation of the dissertation scope to a GBAD surface domain in which own force assets are assumed to have fixed positions and opposing force threats are assumed to be fixed-wing aircraft. The research methodology to be adopted was also discussed in detail and the chapter closed with an explanation of how the material presented later in the dissertation is organised into various chapters forming coherent dissertation parts.

The first part of the dissertation, entitled *Literature review*, comprised two chapters. The first of these, Chapter 2, was devoted to a comprehensive discussion on the basic elements of and factors related to GBAD in general, in fulfilment of Dissertation Objectives I(a) and I(b) §1.3. The chapter opened with a brief review of TEWA DS within a GBAD context, which included a description of IPB (which is central to the TE process), a coherent definition of TE, as well as various approaches that opposing force aerial threats may adopt when attacking own force assets, in fulfilment of Dissertation Objective I(d). The different platforms from which the process of WA may be approached were also discussed in some detail. It was furthermore described how three important fundamental theories of warfare, as well as the physical, functional and cognitive GBAD elements, inform or form a central part of an effective TEWA DSS. In particular, a description was given of the notion of C2 (which included a review of the well-known C2 model of Boyd [20]). Variations on this model by other authors were also touched upon. The importance of speed in terms of increasing the OPTEMPO of C2 was also briefly discussed before the review focus shifted to a discussion on SA. The importance of SA in military missions and a three-tiered hierarchy for achieving SA, due to Endsley [66], was highlighted. A three-level framework by Wallenius [204] for achieving complete SA in the context of C2 was also reviewed. The notion of NCW was considered next and its importance in the context of integrating sensors, WSs, decision makers and information was briefly discussed. This was followed by a detailed discussion on the social, cognitive, information and physical domains of NCW, as well as their interactions, culminating in the logical framework for NCW by Cebrowski [31]. The focus of the review then turned to the physical, functional and cognitive elements of a typical TEWA DSS, in fulfilment of Dissertation Objective I(c). It was explained how the physical (hardware) elements include DAs and how they may be prioritised. Sensor systems and their various capabilities with respect to detecting and tracking aerial threats were also discussed, as were various aircraft alternatives and their manoeuvring capabilities (including possible WS armament carried by them and attack profiles that they may execute) when attacking DAs. Ground-based WSs employed by the own force to counter aerial threats were also reviewed. The discussion on the functional (software) elements of an effective TEWA DSS was focussed on those functional elements pertaining to operator DS, testing and training, effecting and sensing, TM, data management, maintenance and testing, remote system processing, map processing, AM, and TE. The chapter finally closed with a discussion on the cognitive elements of a typical TEWA system, focussing on the roles and responsibilities of the commander and FCO.

The second chapter of Part I, Chapter 3, contained a review of existing WAMs in the military operations research literature since the inception of WA modelling during the early 1950s until the current state of the art of WAMs during the early 21$^{\text{st}}$ century, in fulfilment of Dissertation Objective I(e). These WAMs were presented in chronological order. The account of the evolution of WA modelling was, however, preceded by a brief review of the notion of **NP**-completeness since even the simplest WAMs in the literature are **NP**-complete. It was described how the starting point for WA modelling was the classical assignment problem of Votaw and Orden [202], formulated in 1952. This model formed the cornerstone for Manne [125] on which he built the first WAM in 1958. A detailed description of this first WAM was given and it was described how it paved the way for many authors to contribute towards the formulation of WAMs in many different incarnations. The ensuing history of WA modelling was partitioned into three parts representing natural WA modelling epochs. The first of these epochs was the period from the early 1960s to the mid 1980s during which various static WAMs were put forward. The second epoch was ushered in by Hosein *et al.* [89], who formulated the first dynamic WAM, and continued to the late 1990s. The last epoch was the early 21$^{\text{st}}$ century up to the current day, which saw the introduction of multi-objective WAMs.

The second part of the dissertation, entitled *Mathematical prerequisites*, also comprised two chapters. The first of these, Chapter 4, was devoted to a discussion on the basic principles involved in multi-objective optimisation, in fulfilment of Dissertation Objective I(f). The chapter opened with an introduction to the subject of multi-objective optimisation and this was followed by the presentation of a mathematical formulation for the general form of a multi-objective optimisation problem. The reader was also familiarised with the fundamental notions of decision space and objective space. The importance of the notion of convexity in multi-objective optimisation problems was discussed and it was pointed out that nonconvex multi-objective optimisation problems are considerably harder to solve than their convex counterparts. The review focus then shifted to the notions of solution dominance and of Pareto optimality. The notion of hypervolume was also described as a performance measure capable of measuring the quality of a nondominated set of solutions as an estimate of the true Pareto front of a multi-objective optimisation problem. Two sets of theoretical conditions for Pareto optimality were reviewed — a set of necessary conditions for a solution to be Pareto optimal and a set of sufficient conditions for a solution to be Pareto optimal. Three methods for identifying a nondominated set of solutions from among a given, finite set of candidate solutions to a multi-objective optimisation problem were reviewed and illustrated next, namely a naive and slow approach, a continuously updating approach and a more robust method due to Kung *et al.* [111]. Since many multi-objective optimisation techniques in the operations research literature require candidate solutions to be sorted into different classes based on their respective degrees of dominance, a description was given of the well-known fast nondominated sorting algorithm developed by Deb [51], designed for this purpose. The chapter finally closed with a discussion on problems that can arise when adopting the classical multi-objective optimisation method of weighting objective functions into a single objective instead of adopting a truly multi-objective trade-off paradigm.

The second chapter of Part II, Chapter 5, contained a review of three classes of solution methodologies that have been adopted in the literature for solving WAMs, in fulfilment of Dissertation Objective I(g). The first of these classes was exact solution approaches (in which the method of total enumeration and the celebrated branch-and-bound method by Doig and Land [57] were held up as examples). Although exact solution approaches are able to find globally optimal solutions to optimisation problems, they are typically computationally expensive and are often abandoned in favour of computationally less expensive solution approaches which are typically only able to find locally optimal solutions. Such solution approaches fall within the realm of heuristics (the second class of solution methodologies reviewed) and metaheuristics (the third class of solution methodologies). Within the class of metaheuristics, two optimisation paradigms were considered: trajectory-based solution approaches and population-based approaches. An example of a metaheuristic solution methodology was reviewed within each of these paradigms. These examples within the former paradigm were the method of simulated annealing due to Kirkpatrick *et al.* [104] (in the case of single-objective optimisation) and the DBMOSA developed by Smith *et al.* [173] (in the case of multi-objective optimisation). Within the latter paradigm, the example methodologies were the genetic algorithm due to Holland [88] (in the case of single-objective optimisation) and the NSGA II developed by Agarwal *et al.* [2] (in the case of multi-objective optimisation).

The third part of the dissertation, entitled *Proposed decision support subsystem*, was the heart of the dissertation and contained a number of novel contributions. This part comprised three chapters. The first of these, Chapter 6, contained a newly proposed WA subsystem architecture for use in a GBAD TEWA DSS, in fulfilment of Dissertation Objective II. After a brief discussion on DS design approaches in general, in fulfilment of Dissertation Objective I(h), the design of the WA subsystem architecture was put forward and discussed. The remainder of the chapter was devoted to detailed descriptions of the main components contained in the proposed WA subsystem architecture, as well as the working of another, prerequisite subsystem called the EQ subsystem. The EQ subsystem contains two components, a PEF component and an EEM component. The working of the PEF component was discussed in general and a method was reviewed for discretising the effectiveness values that WSs can achieve with respect to threats. In addition, a process for filtering the effectiveness values of WSs so as to respect constraints posed by extreme environmental conditions and terrain features was also described. Two constituent EEM-component methods for predicting the future flight paths of threats were furthermore described and a method for constructing the EEM was presented, using the information returned by the FPP models. The focus next shifted to a discussion on the components contained in the WA subsystem proper. The envisaged working of a WAM component, which forms the heart of the WA subsystem architecture, was described, and four classes of WAMs, ranging in different levels of complexity, were proposed for inclusion in the WAM component, in fulfilment of Dissertation Objective III. These classes were single-objective static WAMs, multi-objective static WAMs, single-objective dynamic WAMs and multi-objective dynamic WAMs. A model prototype within each of these classes was proposed for default inclusion in the WAM component. These model prototypes were presented in increasing order of complexity and each was discussed in detail. A novel multi-objective static WAM and a novel multi-objective dynamic WAM were also formulated for inclusion in the two most complex classes of WAMs, in fulfilment of Dissertation Objective IV. The final component of the WA subsystem architecture, a WASS component, was also discussed and it was proposed that a number of solution methodologies be used to solve a particular, preferred WAM configured by the FCO and that the candidate solutions thus obtained be sorted to obtain a set of high-quality nondominated solutions to present as real-time DS to the FCO. The chapter closed with an elucidation of the chronological order in which the events within a single TEWA computational cycle are envisaged to occur.

The second chapter of Part III, Chapter 7, was dedicated to a discussion on practical suggestions for implementing a TEWA DSS, in fulfilment of Dissertation Objective V. Eight suggestions were proposed in total. The chapter opened with a brief review on the five levels of automation of DSSs in general. It was advocated that a TEWA DSS should aim to support and conform to FCO thought when an operator has to make difficult decisions in short time frames, rather than pursuing a fully-automated TEWA DSS. A suggestion was also made for improving certain input data in a bid to facilitate higher quality results obtainable from a TEWA DSS. Next, the importance of providing the FCO with only the necessary information was discussed. A number of suggestions were made for efficient layouts of the HMI. Thereafter, the focus shifted to a common problem experienced in military combat situations — that of overwhelming FCOs with excessively many decision alternatives as DS. This may cause confusion when FCOs have to make important WA engagement decisions. A decision tree was proposed for use when the FCO configures WAMs during the pre-deployment stages of a mission so as to alleviate this kind of stress experienced during combat situations. This was followed by four suggestions for incorporating FCO preferences and biases into a TEWA system with a view to reduce the number of WA alternatives reported as DSS recommendations to an FCO. The undesirable notion of switching was described next and a suggestion was made to incorporate threshold values in order to counter this problem. The importance of testing the various components of a TEWA system was finally considered. Specific mention was made of the vital importance of thoroughly testing the constituent parts of military DSSs due the high-stake involvement of human lives in decisions supported by TEWA DSSs. The importance of evaluating a TEWA DSS as a whole before implementation was also elaborated upon. It was suggested that a TEWA DSS should be evaluated as an integrated system within the context of an ADC environment in order to account for possible emergent subsystem behaviour. The use of a simulation environment was proposed for this purpose and the performance evaluation framework of Truter [187] was suggested as a possible platform for such TEWA DSS testing.

The third chapter of Part III, Chapter 8, was devoted to providing an illustration and validation of the working of the various WAM prototypes chosen for default inclusion in the WA design architecture (as described in Chapter 6), in fulfilment of Dissertation Objective VII. The chapter opened with a detailed description of a simulated GBAD scenario adopted as context when solving each of the four WAM prototypes by means of the appropriate solution methodologies (as described in Chapter 5), in fulfilment of Dissertation Objective V. The numerical results returned for each model prototype by the various algorithms within the context of the aforementioned scenario were presented and interpreted. The algorithmic parameter values employed within each case were also given. Thereafter, a comprehensive validation followed of the numerical results obtained, in fulfilment of Dissertation Objective VIII. Three methods of validation were employed. First, a face validation was conducted in terms of the (subjectively judged) quality, realism and practicality of the WA recommendations returned by each WAM prototype. Next, a random benchmark validation was performed by generating a sequence of random solutions for each WAM prototype and ascertaining that these results were of considerably poorer quality than the approximately (Pareto) optimal solutions uncovered by the various algorithms. The final validation process involved a consultation with two military experts during which they were asked (i) to comment on the suitability of the WA alternatives produced by each model, and (ii) to solve each of the four WAMs in the context of the simulated scenario by hand. Their results were then compared to the results obtained *via* the WAM prototypes.

9.2 Appraisal of dissertation contributions

This section contains a brief appraisal of the contributions of this dissertation. The dissertation contains a total of eight novel contributions. In each case the contribution is noted formally and then discussed in terms of its value and significance.

Contribution 9.1 *A documentation of the history of WA modelling evolution from the 1950s up to the early years of the 21st century.*

An account was given in Chapter 3 of the development of WAMs in the military operations research literature. The various WAMs were presented chronologically since the inception of WA modelling during the 1950s and the discussion spanned the entire period of time up to the present. The exposition was partitioned into three parts representing natural WA modelling epochs. To the best knowledge of the author this review is the first historical account of the temporal evolution of WA modelling.

Contribution 9.2 *The formulation of a novel, tri-objective static WAM seeking to achieve Pareto optimal trade-offs between minimising accumulated threat survivability, minimising WS ammunition expenditure and maximising the ammunition stockpile of the least re-engagable WS after the WA decision.*

The tri-objective static WAM formulated in §6.5.2 is, to the best knowledge of the author, only the second multi-objective WAM to have been proposed to date in the literature [123], after an earlier bi-objective WAM put forward by the author during his masters study [118, 120] (which may be considered a forerunner study to the work contained in this dissertation). The expansion of WA modelling into the realm of multi-objective optimisation is considered a significant contribution which has the potential of changing the course of WA modelling efforts in the short to medium term.

Contribution 9.3 *The formulation, for the first time, of a multi-objective dynamic WAM seeking to schedule earliest and latest engagements of aerial threats by WSs in pursuit of Pareto optimal trade-offs between the same three scheduling objectives as the model of Contribution 9.2.*

The tri-objective dynamic WAM proposed in §6.5.4 was inspired by a combination of the single-objective dynamic WAM of Van der Merwe [195] and the tri-objective static WAM of Contribution 9.2. To the best knowledge of the author, this is the first model in the class of multi-objective dynamic WAMs to appear in the literature [121].

Contribution 9.4 *The proposal of a generic, practicable WA subsystem architecture that can function as part of an integrated TEWA DSS.*

The major contribution of this dissertation is the generic WA subsystem design put forward in Chapter 6. As mentioned in §1.2, the main aim in this dissertation was to fill the void in existing design attempts at a fully-fledged South African TEWA DSS, by designing a WA subsystem counterpart to the TE subsystem previously put forward by Roux and Van Vuuren [162]. The purpose of the WA subsystem proposed in this Dissertation is to provide real-time DS to FCOs with respect to high-quality WS-threat assignment alternatives. The WA subsystem design of Chapter 6 is generic in the sense that it facilitates population of the various subsystem components by customised models and other subcomponents. It is envisaged that this contribution may provide new momentum to current, local and international research efforts aimed at the design and establishment of effective TEWA DSSs, just as the TE subsystem design of Roux and Van Vuuren did directly after 2007 (that paper has been cited 79 times in the open literature).

Contribution 9.5 *A demonstration of the practical working of a concept demonstrator conforming to the WA subsystem architecture of Contribution 9.4 within the context of a realistic, but hypothetical, GBAD scenario.*

It was an express aim in this dissertation to put forward a WA subsystem design that is practicable instead of being merely of academic value. The above-mentioned demonstration of the possible practical working of a concept demonstrator of the WA subsystem design advocated in Chapter 6 (in the form of a series of worked numerical examples in Chapter 8) is therefore considered an important contribution of this dissertation. These numerical examples contained solutions to a number of WAM prototypes within the context of an existing, realistic GBAD scenario. Upon subjecting the solutions to these WAM prototypes to a face validation, it was found that the solutions are capable of forming a basis for WA DSS that makes sense from a logical point of view.

Contribution 9.6 *An expert validation of the WA recommendation results obtained in the worked examples of Contribution 9.5.*

The potential value of Contribution 9.5 was underlined by subjecting the numerical example results of Chapter 8 to expert validation. Two military experts [160, 200] confirmed the practical plausibility of the WA recommendations embodied in these numerical results and expressed appreciation for the potential benefit of having access to a TEWA DSS capable of generating such results.

Contribution 9.7 *The documentation of a number of practical suggestions with respect to the possible embedding of the WA subsystem architecture of Contribution 9.4 within a real TEWA DSS.*

Upon consultation with experts in the use and design of DSSs, the practical TEWA system implementation suggestions contained in Chapter 7 and published in [122] are considered to be potentially valuable guidelines for research consultants tasked with possible future end-user TEWA implementations.

Contribution 9.8 *The suggestion of a number of ideas for possible future work following on the contributions of this dissertation.*

The last contribution of this dissertation is proffered in the next chapter, Chapter 10. These suggestions are made in an effort to help orientate TEWA-related research in the immediate short term by documenting suitable avenues of investigation as possible follow-up work to the contributions of this dissertation.

CHAPTER 10

Suggestions for future work

Contents

This final chapter contains suggestions for nine avenues of further investigation as possible follow-up work on the contributions of this dissertation. In each case the suggestion is stated formally and then elucidated and motivated briefly.

10.1 Scope enlargement suggestions

This section contains two suggestions for future work related to natural scope enlargements of the work contained in this dissertation.

Suggestion 10.1 *Enlarge the scope of aerial threats accommodated in the TEWA system.*

The scope of aerial threats accommodated in both the TE and WA subsystems considered in this dissertation was limited to fixed-wing aircraft that are capable of attacking DAs according to a pre-defined set of attack profiles only (as described in §2.5.1). It is suggested that this scope be enlarged to consider additional attack techniques and even other classes of aerial threats (such as rotary-wing aircraft or long-range ballistic missiles). Such additional attack techniques may render the TEWA system more realistic and robust in the sense of being able to accommodate a larger variety of hostile behaviour. The inclusion of other classes of threats is furthermore expected to result in the exhibition of very different attack profiles and/or induce very different computation timescales. For example, a helicopter is capable of vertical ducking behind terrain features and emergence from outside line of sight, which renders successful engagement of such a threat by fixed ground WS difficult. The possibility of rapid changes in flight direction and/or elevation by a helicopter is furthermore expected to complicate the process of TE for such a threat class. Moreover, the speeds at which inter-continental ballistic missiles are capable of flying is such that faster TE and WA algorithms may be required and/or simpler model customisation configurations may be required.

Suggestion 10.2 *Enlarge the scope of ground-based WSs accommodated in the TEWA system.*

CIWSs and VSHORADS were the only types of WSs considered within the scope of GBAD systems in this dissertation. It is suggested that this scope be enlarged to consider additional WS types, such as long-range WSs, for example. This scope enlargement will be especially beneficial if additional threat classes, such as long-range ballistic missiles, are considered within the TEWA system, but is expected to have a dramatic effect on the required accuracy with which future hostile aircraft behaviour can be predicted. If accurate flightpath predictions are not available over long windows into the future, the inclusion of the possibility of long-range WSs in TE and WA models may be of limited benefit.

10.2 Modelling generalisation suggestions

This section contains a further three suggestions for future work related to natural paradigm generalisations of the models considered in this dissertation.

Suggestion 10.3 *Allow for the possibility of dynamism in the deployment of own-force GBAD elements.*

It was assumed in this dissertation that all own-force elements have fixed positions within the GBAD system. It is suggested that this limitation be removed, so that GBAD elements such as sensors, WSs and DAs, may be modelled as objects capable of movement within the area of responsibility. It may, for example, be beneficial to consider sensors with variable positions, so that they can be moved in order to be able to observe air volumes within current blind spots induced by terrain features. It may similarly be beneficial to move WSs so as to enhance their efficacy or DAs in order to afford them better terrain protection. Such an introduction of dynamism into the TEWA modelling paradigm will, of course, require modelling extensions of both TE and WA models into the realm of relative motion kinematics.

Suggestion 10.4 *Allow for the possibility of stochastic area-based WA modelling.*

A single-trajectory flight path prediction modelling approach was assumed in this dissertation. This gave rise to the consideration of WAMs that are essentially deterministic in terms presumed, single attack paths or scenarios adopted by threats once the EEM has been populated. It is suggested, however, that the flight path prediction modelling approach be generalised to a multi-trajectory paradigm in which probabilities are attached to multiple future flight path scenarios for each threat (related to the model described in §6.4.1). This generalisation may result in models capable of predicting future aircraft positions more robustly, but will require the simultaneous generalisation of the WA modelling approach to within the realm of stochastic attack paths or scenarios.

Suggestion 10.5 *Identify additional objectives for inclusion in multi-objective WA models.*

Only three WA criteria were included in the default multi-objective, dynamic model (6.24)–(6.26) proposed in Chapter 6. Although these WAM objectives are considered to be of practical relevance, as confirmed by two military experts [160, 200], it may be beneficial to engage with other military exerts to ascertain whether the inclusion of additional WA criteria in the model is expected to contribute to the model's realism.

10.3 Model solution suggestions

This section contains four final suggestions for future work related to WA model solution methodology, implementation and validation.

Suggestion 10.6 *Apply the WAM prototypes selected for default inclusion in the WAMS component in the contexts of other realistic simulated GBAD scenarios.*

A single hypothetical, albeit realistic, GBAD scenario was considered in the numerical WA model demonstration and validation process of Chapter 8. Although such GBAD scenarios are hard to come by (they typically have to be designed in careful consultation with military experts), it is obvious that an extended WA model demonstration and validation process, such as that in Chapter 8 but carried out within the contexts of other hypothetical GBAD scenarios, will contribute to the perceived practical plausibility and credibility of the WA modelling approach proposed in this dissertation.

Suggestion 10.7 *Reduce the computational time required to solve the multi-objective, dynamic model proposed in Chapter 6.*

It was found in §8.2.4 that an unrealistically long time is required in order to find a high-quality (approximate) Pareto front of trade-off solutions to the novel multi-objective, dynamic model (6.24)–(6.26); it required 139 seconds to solve the instance of this model related to the GBAD scenario considered in Chapter 8. This computational burden may be attributed, in part, to the high-level model solution implementation in Oracle's Java [141]. It is suggested that a lower-level implementation of the NSGA-II and the DBMOSA is pursued (in a programming language such as C++ [40], for example). It has been reported that high-level implementations of combinatorial algorithms in environments, such as Java or Wolfram's Mathematica [211], may run up to one hundred times longer than when implemented in low-level programming languages, such as C++ [27]. If a run time reduction of this order of magnitude is achievable in the case of the two algorithms mentioned above, then implementation of this suggestion may render the multi-objective, dynamic model (6.24)–(6.26) a more viable WAM from a realistic point of view.

Suggestion 10.8 *Increase the sophistication of the current WAM solution methodologies.*

Since the design of sophisticated WAM solution procedures was not the aim in this dissertation, rather rudimentary incarnations of the method of simulated annealing and a genetic algorithm were employed to solve the WAMs of Chapter 6 in the context of the GBAD scenario of Chapter 8, as described in some detail in Chapter 5. There is indeed considerable room for improvement of these implementations. Other simulated annealing move operators and other genetic algorithmic crossover procedures may, for example, be considered. Furthermore, only feasible solutions were considered in the methods described in Chapter 5 — perhaps the metaheuristic optimisation techniques may benefit (even significantly) from being allowed to search through infeasible regions of the decision space under suitable constraint violation penalty schemes (thus being able to reach disjoint pockets of the feasible domain). Also, very limited parameter optimisations were carried out in order to select suitable parameter values for the various algorithms (such as the cooling and reheating rates for the method of simulated annealing implementations, or the mutation and crossover rates of the genetic algorithm implementations); more thorough parameter evaluations may perhaps yield improved WAM solutions.

Suggestion 10.9 *Explore other WAM solution methodologies for possible inclusion in the WAMS component.*

Only two WAM solution methodologies were included in the Chapter 8 concept demonstration of the DSS architecture put forward in Chapter 6. These methods were the trajectory-based method of simulated annealing and the population-based genetic algorithm. It is suggested that other trajectory-based methods, such as tabu search or variable neighbourhood search, and other population-based methods such as particle swarm optimisation or ant colony optimisation, also be considered as possible WAM solution methodologies and included in an enhanced concept demonstration of the WASS proposed in §6.6.

References

[1] Aarts E, Korst J & Michiels W, 2005, *Simulated annealing*, pp. 187–210 in Burke EK & Kendall G (Eds), *Search methodologies*, Springer, New York (NY).

[2] Agarwal S, Deb K, Pratap A & Meyarivan T, 2002, *A fast and elitist multi-objective genetic algorithm: NSGA II*, IEEE Transactions on Evolutionary Computation, **6(2)**, pp. 182–197.

[3] Ahuja RK, Kumar A, Jha KC & Orlin JB, 2003, *Exact and heuristic algorithms for the weapon target assignment problem*, MIT Sloan, Working Paper No. 4464-03.

[4] Alberts DS, 2004, *Network centric warfare: Executive report*, [Online], [Cited May 2017], Available from `http://www.dod.mil/nii/NCW`.

[5] Alberts DS, Garstka JJ & Stein FP, 1999, *Network centric warfare: Developing and leveraging information superiority*, 2nd Edition, Command Control Research Program.

[6] Alberts DS & Hayes RE, 2003, *Power to the edge: Command and control in the information age*, (Unpublished) Technical Report Unpublished (ADA457861), Office of the Assistant Secretary of Defence, Washington (DC).

[7] Alves PG, 2000, *The transfer of dual-use outer space technologies: Confrontation or co-operation?*, PhD Dissertation, University of Geneve, Geneve.

[8] Argos Press Pty Ltd, 2009, *Radar glossary: Information on plan position indicator*, [Online], [Cited July 2012], Available from `http://www.argospress.com/Resources/radar/planpositiindica.htm`.

[9] Baal B, 2004, *Russian missile systems and space launchers*, [Online], [Cited June 2017], Available from `http://www.wonderland.org.nz/`.

[10] Back T, 1996, *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press, Oxford.

[11] Barone D, Hingston L, Huband P & While L, 2006, *A faster algorithm for calculating hypervolume*, IEEE Transactions on Evolutionary Computation, **10(1)**, pp. 29–38.

[12] Barone L, Bradstreet L, Hingston L & While L, 2005, *Heuristics for optimizing the calculation of hypervolume for multi-objective optimization problems*, 2005 IEEE Congress on Evolutionary Computation, pp. 2225–2232.

[13] Bell MR & Grubbs RA, 1993, *JEM modeling and measurement for radar target identification*, Aerospace and Electronic Systems, **29(1)**, pp. 73–87.

[14] Belton V & Stewart T, 2002, *Multiple criteria decision analysis: An integrated approach*, Kluwer Academic Publishers, Boston (MA).

[15] Benders JF, 1962, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik, **4**, pp. 238–252.

[16] Benyon D, 1987, *Towards a tool kit for the systems analyst*, The Computer Journal, **30(1)**, pp. 2–7.

[17] Bertsimas D & Tsitsiklis J, 1993, *Simulated annealing*, Statistical Science, **8(1)**, pp. 10–15.

[18] Booz A, 1999, *Measuring the effects of network-centric warfare*, 1st Edition, Office of the Secretary of Defence (Net Assessment), Washington (DC).

[19] Boyd JR, 1985, *A discourse on winning and losing*, Unpublished Lecture Notes: Maxwell AFB (Air University), Montgomery (AL).

[20] Boyd JR, 2013, *Destruction and creation*, [Online], [Cited May 2017], Available from `http://www.goalsys.com/books/documents/DESTRUCTION_AND_CREATION.pdf`.

[21] Bradford JC, 1961, *Determination of the optimal assignment of a weapon system to several threats*, Vought Aeronatics, AER-EITM-9, p. 14.

[22] Bradford WJ, 1992, *The theoretical layered air-defence capability of a ship engaged against multiple anti-ship capable missile attacks*, (Unpublished) Technical Report 007-086, Defence Science and Technology Organisation (DSTO), Melbourne.

[23] BrainyQuote, 2007, *George S Patton quotes*, [Online], [Cited July 2017], Available from `http://www.brainyquote.com`.

[24] Bringmann K & Friedrich T, 2013, *Approximation quality of the hypervolume indicator*, Artificial Intelligence, **195**, pp. 265–290.

[25] Brockhoff D, Thiele L & Zitzler E, 2000, *The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration*, The Journal of Evolutionary Computation, **8(2)**, pp. 173–195.

[26] Bullinaria JA & Garcia-Najera A, 2010, *An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows*, Computers and Operations Research, **38(2001)**, pp. 287–300.

[27] Burger AP, 2014, Researcher at *Department of Logistics, Stellenbosch University*, [Personal Communication], Contactable at `apburger@sun.ac.za`.

[28] Busetti F, 2003, *Simulated annealing overview*, [Online], [Cited August 2017], Available from `http://www.cs.ubbcluj.ro/~csatol/mestint/pdfs/Busetti_AnnealingIntro.pdf`.

[29] Cai X, Cui Z, Fang J & Zeng J, 2010, *A hybrid group search optimiser with Metropolis rule*, Proceedings of the 2010 International Conference on Modelling, Identification and Control, Okayama City, Japan, pp. 556–561.

[30] Cannon-Bowers JA, Johnston JH & Sales E, 1998, *Tactical decision making under stress (TADMUS): Mapping a program of research to a real-world incident — The USS Vinceness*, Proceedings of the Research and Technology Organisation Meeting Proceedings 4 (RTO-MP-4), pp. 1–3.

[31] Cebrowski AK, 1998, *Network centric warfare: Its origin and future*, Naval Institute, **124(1)**, pp. 232–235.

[32] Cebrowski AK, 2003, *Network centric warfare and information superiority*, [Online], [Cited May 2017], Available from `http://www.oft.osd.mil/`.

[33] Center DTI, 2004, *Department of defense dictionary of military associated terms*, [Online], [Cited May 2017], Available from `http://www.dtic.mil/dtic/`.

[34] CHASTAIN G, 2005, *Threat ordering in air defense weapon systems — An introduction*, [Online], [Cited June 2017], Available from `http://home.hiwaay.net/`.

[35] CHASTAIN G, 2006, *IFF Questions and answers*, [Online], [Cited June 2012], Available from `http://home.hiwaay.net/~georgech/WEAPONS/iff.htm`.

[36] CHECKMATE CRUISER, 2005, *USS Vinceness CG-49*, [Online], [Cited April 2017], Available from `https://web.archive.org/web/20050306212102/http://www.vincennes.navy.mil:80/index.html`.

[37] CHIEF OF NAVAL OPERATIONS, 2012, *Submarine frequently asked questions*, [Online], [Cited May 2012], Available from `http://www.navy.mil/navydata/cno/n87/faq.html`.

[38] CHRISTOFIDES N, MINGOZZI A & TOTH P, 1979, *The vehicle routing problem*, Wiley, Chichester.

[39] COELLO CAC, LAMONT GB & VAN VELDHUIZEN DA, 2007, *Evolutionary algorithms for solving multi-objective problems*, 2nd Edition, Springer Science, New York (NY).

[40] CPP INSTITUTE, 2017, *C++*, [Online], [Cited August 2017], Available from `http://cppinstitute.org/free-c-and-c-courses?gclid=EAIaIQobChMIpL-D9Kj31QIVy7_tCh21dQhJEAAYAiAAEgI7Y_D_BwE`.

[41] CRAMER ML, 1996, *Command and control warfare: OODA loop countermeasures*, 33rd Annal AOC International Electronic Warfare Technical Symposium and Convention, Washington (DC).

[42] CSIR, 2005, *Non-cooperative target recognition for search and track radars*, [Online], [Cited June 2017], Available from `http://www.csir.co.za/`.

[43] CUNHA NOD & POLAK E, 1967, *Constrained minimisation under vector evaluated criteria in finite dimensional spaces*, Mathematical Analysis and Applications, **19(1)**, pp. 103–124.

[44] DAHL EJ, 2002, *Network centric warfare and the death of operational art*, Defence Studies, **2(1)**, pp. 1–24.

[45] DANTZIG GB & WOLFE P, 1960, *Decomposition principle for linear programs*, Operations Research, **8**, pp. 101–111.

[46] DAY RH, 1966, *Allocating weapons to target complexes by means of nonlinear programming*, Operations Research, **14**, pp. 992–1013.

[47] DE VILLIERS AP, 2014, *Edge criticality in secure graph domination*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[48] DEB K, 1995, *Optimisation for engineering design: Algorithms and examples*, Prentice Hall, New Delhi.

[49] DEB K, 1999, *Multi-objective genetic algorithms: problem difficulties and construction of test problems*, Evolutionary Computation, **7(3)**, pp. 205–230.

[50] DEB K, 2001, *Multi-objective optimisation using evolutionary algorithms*, John Wiley & Sons, New York (NY).

[51] DEB K & SRINIVAS N, 1995, *Multi-objective optimisation using nondominated sorting genetic algorithms*, Journal of Evolutionary Computation, **2(3)**, pp. 221–248.

[52] DEFENCE UPDATE, 2004, *SHORAD transformation*, [Online], [Cited June 2017], Available from `http://www.defence-update.com`.

[53] DEN BROEDER GG, ELLISON RE & EMERLING L, 1959, *On optimum target assignments*, Operations Research, **7**, pp. 322–326.

[54] Dennehy K & Deighton C, 1966, *Development of an interactionist framework for operationalising situation awareness*, 1st International Conference in Engineering Psychology and Cognitive Ergonomics, Stratford-on-Avon.

[55] Dennis A, Wixom BH & Tegarden D, 2005, *Systems analysis and design with UML version 2.0: An object-orientated approach*, 2nd Edition, John Wiley and Sons, Hoboken (NJ).

[56] Departments of the Army and the Air Force, 1989, *Weather support for army tactical operations*, (Unpublished) Technical Report FM 34-81/AFM 105-4, Washington (DC).

[57] Doig AG & Land AH, 1960, *An automatic method for solving discrete programming problems*, Econometrica, **28(3)**, pp. 497–520.

[58] Dominguez C, Vidulich M, Vogel E & MacMillan G, 1994, *Situation awareness: Papers and annotated bibliography*, (Unpublished) Technical Report AL/CF-TR-1994-0085, Human System Center – Armstrong Laboratory, Dayton (OH).

[59] Dorigo M, 1992, *Optimisation, learning and natural algorithms*, PhD Dissertation, Politecnico di Milano, Milan.

[60] Doucet J, 2009, *Components, purpose and function of information systems*, [Online], [Cited August 2017], Available from `http://goo.gl/w9nZJg`.

[61] Dreo J, Petrowski A, Siarry P & Taillard E, 2003, *Metaheuristics for hard optimisation*, Springer-Verlag, Berlin.

[62] Dreo J, Petrowski A, Siarry P & Taillard E, 2006, *Metaheuristics for hard optimisation: Methods and case studies*, Springer, New York (NY).

[63] du Toit FJ, 2006, *Design of a weapon assignment system*, Honours Year Project, Stellenbosch University, Stellenbosch.

[64] du Toit FJ, 2009, *The dynamic weapon target assignment problem in a ground-based air defence environment*, MSc Thesis, Stellenbosch University, Stellenbosch.

[65] Ebrahimpour T, 2010, *The cavernous archive of US crimes*, [Online], [Cited November 2012], Available from `http://www.iranreview.org/content/Documents/Iran%5C_Air%5C_Flight%5C_655.htm`.

[66] Endsley MR, 1995, *Toward a theory of situation awareness in dynamic systems*, Human Factors, **37(1)**, pp. 32–64.

[67] Endsley MR & Kiris EO, 1983, *Ironies of automation*, Automatica, **19**, pp. 775–779.

[68] Endsley MR & Kiris EO, 1995, *The out-of-loop performance problem and level of control in automation*, Human Factors, **37(2)**, pp. 381–394.

[69] Flood M, 1957, *Target-assignment model*, Proceedings of the Princeton University Conference on Linear Programming, Princeton (NJ).

[70] French CD, 1995, *One size fits all database architectures do not work for DSS*, Proceedings of the 2nd ACM SIGMOD, San Jose (CA), pp. 449–450.

[71] Fuller JFC, 1925, *The foundations of the science of war*, Hutchington and Co Ltd, London.

[72] Geem ZW & Kim JH, 2001, *A new heuristic optimisation algorithm: Harmony search*, Simulation, **76(2)**, pp. 60–68.

[73] GlobalSecurity.org, 2005, *US Army Field Manual FM 1-100,* Chapter 3, [Online], [Cited May 2012], Available from `http://www.globalsecurity.org/military/library/policy/army/fm/1-100/ch3.htm`.

[74] GlobalSecurity.org, 2005, *US Army Field Manual FM 34-81-1,* Chapter 1, [Online], [Cited May 2012], Available from `http://www.globalsecurity.org/intell/library/policy/army/fm/34-81-1/ch1.htm`.

[75] GlobalSecurity.org, 2007, *Operation Earnest Will*, [Online], [Cited April 2017], Available from `http://www.globalsecurity.org/military/ops/earnest_will.htm`.

[76] GlobalSecurity.org, 2011, *Aegis Combat System (ACS)*, [Online], [Cited April 2017], Available from `http://www.globalsecurity.org/military/systems/ship/systems/aegis.htm`.

[77] Glover F, 1986, *Future paths for integer programming and links to artificial intelligence*, Computers and Operations Research, **13(5)**, pp. 533–549.

[78] Goebel G, 2004, *Dumb bombs (2): Cluster munitions and other bombs*, [Online], [Cited June 2017], Available from `http://www.vectorsite.net/twbomb2.html`.

[79] Gruhn P, 2001, *Human Machine Interface (HMI) design: The good, the bad and the ugly*, 66^{th} Annual Instrumentation Symposium for the Process Industries, Houston (TX).

[80] Guha M, 2010, *Reimagining war in the 21st century: From Clausewitz to network-centric warfare*, 1^{st} Edition, Routledge Critical Security Studies, Washington (DC).

[81] Guinn M, 2008, *Tactical Decision Making Under Stress (TADMUS)*, [Online], [Cited April 2017], Available from `http://mikeguinn.blogspot.co.za/2008/03/tactical-decision-making-under-stress.html`.

[82] Hamilton S, 2006, *Operational thread development*, [Online], [Cited May 2017], Available from `http://www.mors.org/meetings/bar/briefs/hamilton.pdf/`.

[83] Henning MA & van Vuuren JH, *Graph and network theory*, In process.

[84] Herman M, 1997, *Entropy-based warfare: A unified theory for modeling the revolution in military affairs*, Booz-Allen & Hamilton, San Diego (CA).

[85] Heuer HW, 2016, *Design of a threat evaluation and weapon assignment human machine interface*, BEng Year Project, Stellenbosch University, Stellenbosch.

[86] Hillier FS & Lieberman GJ, 2010, *Introduction to Operations Research*, 9^{th} Edition, McGraw-Hill, New York (NY).

[87] Ho CK, Robinson A, Miller DR & Davis MJ, 2005, *Overview of sensors and needs for environmental monitoring*, Sensors, **5**, pp. 4–37.

[88] Holland JH, 1975, *Adaptation in natural and artificial systems*, MIT Press, Massachusetts (MA).

[89] Hosein PA & Athans M, 1989, *The dynamic weapon-target assignment problem*, (Unpublished) Technical Report, Massachusetts Institute of Technology, Cambridge (MA).

[90] Hower R, 2012, *Software QA and testing frequently asked questions*, [Online], [Cited July 2012], Available from `http://www.softwareqatest.com/qatfaq1.html`.

[91] Huaiping C, Jingxu L, Yingwu C & Hao W, 2006, *Survey of the research on dynamic weapon-target assignment problem*, Journal of Systems Engineering and Electronics, **17(3)**, pp. 559–565.

[92] Huang MD, Romeo F & Sangiovani-Vincentelli AL, 1986, *An efficient general cooling schedule for simulated annealing*, Proceedings of the 1986 IEEE International Conference on Computer-Aided Design, Santa Carla (CA), pp. 381–384.

[93] Hutchins SG, Kelly RT, Moore RA & Morrison JG, 1991, *Tactical Decision Making Under Stress (Tadmus) decision support system*, [Online], [Cited April 2017], Available from `http://www.all.net/journal/deception/www-tadmus.spawar.navy.mil/www-tadmus.spawar.navy.mil/TADMUS_DSS.pdf`.

[94] Institute for National Strategic Studies, National Defence University, 2012, *Towards a theory of spacepower: Selected essays*, [Online], [Cited May 2012], Available from `http://www.ndu.edu/press/spacepower.html`.

[95] Jaccard P, 1901, *Distribution of the alpine flora in the Dranse's Basin and some neighbouring regions*, Bulletin de la Societe Vaudoise des Sciences Naturelles, **37**, pp. 241–272.

[96] James P, Kahan D, Worley R & Stasz C, 1989, *Understanding commanders' information needs*, (Unpublished) Technical Report, Rand Corporation, Santa Monica (CA).

[97] Johansson F & Falkman G, 2009, *Perfomance evaluation of TEWA systems for improved decision support*, (Unpublished) Technical Report, University of Skovde, Department of computer science, Skovde.

[98] Johnston JH & Paris J, 1999, *Towards assessing the impact of TADMUS decision support system and training on team decision making*, Command and Control Research and Technology Symposium, Orlando (FL).

[99] Joint Chiefs of Staff, 1996, *Joint doctrine command and control warfare (C2W)*, National Defense University Press, Washington (DC).

[100] Joint Doctrine and Concept Centre, 2003, *Joint warfare publication 3-63: Joint air defence*, (Unpublished) Technical Report 3-63, JDCC, Shrivenham.

[101] Kendall KE & Kendall JE, 2011, *Systems analysis and design*, 8$^{\text{th}}$ Edition, Pearson, Upper Saddle River (NJ).

[102] Kennedy J & Eberhart R, 1995, *Particle swarm optimisation*, Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, pp. 1942–1948.

[103] Kimble C, 2010, *Information systems and strategy*, [Online], [Cited August 2017], Available from `http://goo.gl/dXQJLJ`.

[104] Kirkpatrick S, Gelatt CD & Vecchi MP, 1983, *Optimisation by simulated annealing*, Research Report RC 9355, International Business Machines, Yorktown (NY).

[105] Klein GA, 1988, *Naturalistic models of C3 decision making*, pp. 86–92 in Johnson SE, Alexander H & Levis IS (Eds), *Science of command and control*, AFCEA International Press, Washington (DC).

[106] Klein GA, 1993, *Decision making in action: Models and methods*, Ablex Publishing Corporation, Norwood (CO).

[107] Kok BJ, 2009, *Performance evaluation of a threat evaluation and weapon assignment system*, MSc Thesis, Stellenbosch University, Stellenbosch.

[108] Kora P & Yadlapalli P, 2017, *Crossover operators in genetic algorithms: A review*, International Journal of Computer Applications, **162(10)**, pp. 34–36.

[109] Kramer O, 2017, *Genetic algorithm essentials*, Springer, New York (NY).

[110] Krulak CC, 1999, *The strategic corporal: Leadership in the three block war*, [Online], [Cited May 2017], Available from `http://www.au.af.mil/au/awc/awcgate/usmc/strategic_corporal.htm`.

[111] Kung HT, Luccio F & Preparata FP, 1975, *On finding the maxima of a set of vectors*, Journal of the Association for Computing Machinery, **22(4)**, pp. 469–476.

[112] Labuschagne PH, 2008, *Phase 1: Situation; Battle appreciation as a PS model at the lower tactical level*, Air Defence Artillery Planners Presentation at Reutech Radar Systems (RRS), Stellenbosch.

[113] Labuschagne PH, 2008, *Phase 2: The battle concepts; Battle appreciation as a PS model at the lower tactical level*, Air Defence Artillery Planners Presentation at Reutech Radar Systems (RRS), Stellenbosch.

[114] Labuschagne PH, 2008, *Phase 3: Options; Battle appreciation as a PS model at the lower tactical level*, Air Defence Artillery Planners Presentation at Reutech Radar Systems (RRS), Stellenbosch.

[115] Labuschagne PH, 2008, *Phase 4: ADA battle plan; Battle appreciation as a PS model at the lower tactical level*, Air Defence Artillery Planners Presentation at Reutech Radar Systems (RRS), Stellenbosch.

[116] Li H, Suganthan PN, Qu BY, Zhang Q, Zhao S & Zhou A, 2011, *Multi-objective evolutionary algorithms: A survey of the state of the art*, Swarm and Evolutionary Computation, **1(2011)**, pp. 32–49.

[117] Lizarraga G, Gomez MJ, Castanon MG, Acevedo-Davila J & Rionda SB, 2009, *A faster algorithm for calculating hypervolume*, LNAI, **5845**, pp. 647–657.

[118] Lötter DP, 2012, *Modelling weapon assignment as a multi-objective decision problem*, MComm Thesis, Stellenbosch University, Stellenbosch.

[119] Lötter DP, 2014, *A repository of data for a simulated GBAD scenario*, [Online], [Cited August 2014], Available from `http://www.vuuren.co.za`.

[120] Lötter DP, Nieuwoudt I & van Vuuren JH, 2013, *A multiobjective approach towards weapon assignment in a ground-based air defence environment*, ORiON, **29(1)**, pp. 31–54.

[121] Lötter DP & van Vuuren JH, 2014, *A tri-objective, dynamic weapon assignment model for surface-based air defence*, ORiON, **32(1)**, pp. 1–22.

[122] Lötter DP & van Vuuren JH, 2014, *Implementation challenges associated with a threat evaluation and weapon assignment system*, Proceedings of the 43rd Operations Research Society of South Africa Annual Conference, Parys, pp. 27–35.

[123] Lötter DP & van Vuuren JH, 2014, *Weapon assignment decision support in a surface-based air defence environment*, Submitted.

[124] Mandel T, 1997, *The elements of user interface design*, 1$^{\text{st}}$ Edition, John Wiley and Sons, New York (NY).

[125] Manne AS, 1958, *A target-assignment problem*, Operations Research, **6**, pp. 346–351.

[126] McCarthy JD, 1991, *USS Vinceness (CG 49) shootdown of Iran Air flight 655: A Comprehensive analysis of legal issues presented by the case concerning the aerial incident of 3 July 1988*, Master of Law Thesis, George Washington University, Washington (DC).

[127] Meloni JC, 2004, *Sams teach yourself PHP, MySQL and Apache all in one*, 2$^{\text{nd}}$ Edition, Sams Publishing, Canada (CA).

[128] Menezes AJ, van Oorschot PC & Vanstone SA, 1997, *Handbook of applied cryptography*, CRC Press, New York (NY).

[129] Mepham S, 1998, *Synthetic environments — Delivering real benefits to UK defence*, Advanced Simulation Technology and Training Conference (SimTecT), Adelaide.

[130] Miettinen K, 1999, *Nonlinear multi-objective optimisation*, Kluwer, Boston (MA).

[131] Military Factory, 2012, *United States navy ship hull classifications*, [Online], [Cited May 2012], Available from `http://www.militaryfactory.com/ships/us-navy-ship-classifications.asp`.

[132] Mladenovic N & Hansen P, 1997, *Variable neighbourhood search*, Computers and Operations Research, **24(11)**, pp. 1097–1100.

[133] Moore J & Chapman R, 1999, *Application of particle swarm to multi-objective optimisation*, Unpublished Manuscript.

[134] Mouat T, 2011, *Tom Mouat's mapsymbs.com*, [Online], [Cited July 2012], Available from `http://www.mapsymbs.com/maphome.html`.

[135] Murphey RA, 2000, *Target-based weapon target assignment problems*, pp. 39–53 in Pardalous PM & Pitsoulis LS (Eds), *Nonlinear assignment problems*, Kluwer Academic Publishers, Dordrecht.

[136] NASA, 2012, *What is a satellite?*, [Online], [Cited May 2012], Available from `http://www.nasa.gov`.

[137] National Reconnaissance Office, 2012, *Corona history*, [Online], [Cited May 2012], Available from `http://www.nro.gov/history/csnr/corona`.

[138] Obbayi SR, 2011, *Types of database management systems*, [Online], [Cited August 2017], Available from `http://www.brighthub.com/internet/web-development/articles/110654.aspx`.

[139] Oerlikon, 2002, *Aircraft tactics and helicopter missions*, (Unpublished) Technical Report TR 70308-1E, Oerlikon Military Products, Zurich.

[140] Oosthuizen R, 2007, *TEWA paper — Comments by R Oosthuizen*, (Unpublished) Technical Report, Monze Consulting, Pretoria.

[141] Oracle Software, 2014, *Java*, [Online], [Cited May 2013], Available from `https://www.oracle.com/java/index.html`.

[142] Orr GE, 1983, *USAF combat operations C3I: Fundamentals and interactions*, (Unpublished) Technical Report AU-ARI-82-5, Airpower Research Institute, Air University Press, Montgomery (AL).

[143] Pannone J, 2010, *Design considerations for effective human machine interface systems*, [Online], [Cited July 2012], Available from `http://www.eao.com`.

[144] Paradis S, Benaskeur A, Oxenham MG & Cutler P, 2005, *Threat evaluation and weapon allocation in network centric warfare*, Proceedings of the 7^{th} International Conference on Information Fusion (Fusion 2005), Stockholm, pp. 1078–1085.

[145] Potgieter JJ, 2008, *Real-time weapon assignment in a ground-based air defence environment*, MEng Thesis, Stellenbosch University, Stellenbosch.

[146] Potgieter JJ, 2012, Software engineer at *Reutech Radar Systems*, [Personal Communication], Contactable at `cobusp@reutech.co.za`.

[147] Publication MCD, 1996, *Chapter 2: Command and control theory*, [Online], [Cited May 2017], Available from `http://www.au.af.mil/au/awc/awcgate/mcdp6/ch2.htm/`.

[148] Raduege HD, 2004, *Net-centric warfare is changing the battlefield environment*, [Online], [Cited May 2017], Available from http://www.stsc.hill.af.mil/.

[149] Ramkumar P, 2012, *HMI: Human-Machine Interface*, [Online], [Cited July 2012], Available from http://www.birds-eye.net/definition/h/hmi-human_machine_interface.shtml.

[150] Rao SS, 1984, *Optimisation: Theory and applications*, John Wiley & Sons, New York (NY).

[151] Rardin RL, 1998, *Optimisation in Operations Research*, Prentice Hall, New Jersey (NJ).

[152] Rathgeber W, Baranes B, Schrogl KU & Venet C, 2010, *Yearbook on space policy 2008/2009*, SpringerWien, New York (NY).

[153] Reklaitis GV, Ravindran A & Ragsdell KM, 1983, *Engineering optimisation methods and applications*, John Wiley & Sons, New York (NY).

[154] Research D & Canada D, 2007, *Non-cooperative target recognition of air targets*, [Online], [Cited June 2017], Available from http://www.ottawa.drdc-rddc.gc.ca/.

[155] Riffenburgh RH, 1991, *Tactical Decision Making Under Stress (Tadmus) study*, [Online], [Cited April 2017], Available from http://www.dtic.mil/get-tr-doc/pdf?AD=ADA242766.

[156] Roberts NC, 1992, *Reconstructing combat decisions: Reflections on the shootdown of flight 655*, Naval Postgraduate School, Monterey (CA).

[157] Roodt JHS, Nel JJ & Oosthuizen R, 2007, *Framework and roadmap for DERI decision support to the Joint Command and Control and Intelligence (JC2 and Int) capability portfolio*, (Unpublished) Technical Report, CSIR, Pretoria.

[158] Roux JN, 2006, *Real-time threat evaluation of fixed wing aircraft in a ground-based air defence environment*, MScEng Thesis, Stellenbosch University, Stellenbosch.

[159] Roux JN, 2010, *Design of a threat evaluation subsystem in a ground-based air defence environment*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[160] Roux JN, 2017, Operations Director at *Wi Group*, [Personal Communication], Contactable at jaco@wigroup.co.za.

[161] Roux JN & van Vuuren JH, 2007, *Threat evaluation and weapon assignment decision support: A review of the state of the art*, ORiON, **23(2)**, pp. 151–187.

[162] Roux JN & van Vuuren JH, 2008, *Real-time threat evaluation in a ground-based air defence environment*, ORiON, **24(1)**, pp. 75–101.

[163] SAAB, 2013, *Battlefield command support system*, [Online], [Cited January 2010], Available from http://www.saabgroup.com/en/Markets/Saab-Australia/Land/BCSS/.

[164] SAAB, 2013, *Tactical Command and Control System*, [Online], [Cited January 2010], Available from http://www.saabgroup.com/en/Markets/Saab-Australia/Land/TACCS.

[165] Saaty T, 1990, *How to make a decision: The Analytic Hierarchy Process*, European Journal of Operational Research, **48**, pp. 9–26.

[166] Sarter N & Woods D, 1991, *Situation awareness: A critical but ill-defined phenomenon*, International Journal of Aviationi Psychology, **1**, pp. 45–57.

[167] Sastry K & Goldberg D, 2005, *Genetic algorithms*, pp. 97–125 in Burke EK & Kendall G (Eds), *Search methodologies*, Springer, New York (NY).

[168] Sen S & Sherali HD, 1984, *A branch-and-bound algorithm for extreme point mathematical programming problems*, Discrete Applied Mathematics, **11**, pp. 265–280.

[169] Simon HA, 1956, *Rational choice and the structure of the environment*, Psychological Review, **63(2)**, pp. 129–138.

[170] Sloan C, 2006, *Input-process-output model*, [Online], [Cited May 2017], Available from http://www.sloan-c-wiki.org/.

[171] Smith DJ, 2007, *Situation(al) awareness (SA) in effective command and control*, [Online], [Cited May 2017], Available from http://www.smithsrisca.demon.co.uk/.

[172] Smith EA, 2001, *Network centric warfare: What's the point?*, Naval War College Review, **54(1)**, pp. 59–75.

[173] Smith K, Everson RM, Fieldsend JE, Murphy C & Misra R, 2008, *Dominance-based multi-objective simulated annealing*, IEEE Transactions on Evolutionary Computation, **12(3)**, pp. 323–342.

[174] Solomon MM, 1987, *Algorithms for the vehicle routing problem and scheduling with time windows constraints*, Operations Research, **35**, pp. 45–65.

[175] South African Defence Force, 2006, *Joint warfare manual: Air defence operations*, UDDC, Kimberley.

[176] Stanton NA, Chambers PRG & Piggott J, 2001, *Situation awareness and safety*, Safety Science, **39**, pp. 189–204.

[177] Stewart TJ, 2007, *The essential multiobjectivity of linear programming*, ORiON, **23(1)**, pp. 1–15.

[178] Stewart V, 1981, *Business applications of the repertory grid*, McGraw-Hill Book Co., London.

[179] Stone D, Jarrett C, Woodroffe M & Minocha S, 2005, *User interface design and evaluation*, 1st Edition, Morgan Kaufmann Publishers, San Francisco (CA).

[180] Tadil J, 2000, *Introduction to tactical digital information link J and quick reference guide*, [Online], [Cited June 2012], Available from www.adtdl.army.mil.

[181] The Encyclopedia of Earth, 2016, *The Strait of Hormuz*, [Online], [Cited April 2017], Available from http://editors.eol.org/eoearth/wiki/Strait_of_Hormuz.

[182] The Royal Navy, 2012, *Submarine*, [Online], [Cited May 2012], Available from http://www.royalnavy.mod.uk.

[183] The Secretary of Defense, 1988, *Formal investigation into the circumstances surrounding the downing of Iran Air Flight 655 on 3 July 1988*, [Online], [Cited April 2017], Available from http://www.dtic.mil/get-tr-doc/pdf?AD=ADA203577.

[184] Thiele L & Zitzler E, 1999, *Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach*, IEEE Transactions on Evolutionary Computation, **3(4)**, pp. 257–271.

[185] Tighe T, Hill R & McIntyre G, 2006, *A decision for strategic effects: A conceptual approach towards effects based targeting*, [Online], [Cited May 2017], Available from http://www.airpower.maxwell.af.mil/airchronicles/cc/Hill.html/.

[186] Toth P & Vigo D, 2001, *The vehicle routing problem*, SIAM Monographs on Discrete Mathematics, Applications, Society for Industrial, and Applied Mathematics, Camden (PA).

[187] Truter ML, 2016, *Development and demonstration of a performance evaluation framework for threat evaluation and weapon assignment systems*, MEng Thesis, Stellenbosch University, Stellenbosch.

[188] UDDC, 2005, *Pamphlet 1: Air defence control concepts*, (Unpublished) ADA Doctrine Handbook, Volume 2, Book 4, Ver 1: Rev C (Restricted), UDDC, Kimberley.

[189] UDDC, 2005, *Pamphlet 2: Asset assessment*, (Unpublished) ADA Doctrine Handbook, Volume 2, Book 2 (Restricted), UDDC, Kimberley.

[190] UDDC, 2006, *ADA characteristics and limitations*, (Unpublished) ADA Doctrinal Note Serial No: ADA-06/0040 Rev 1 (Restricted), UDDC, Kimberley.

[191] Undersecretariat for Defence Industries Export Portal, 2013, *Genesis (Ship integrated combat management system)*, [Online], [Cited January 2011], Available from `http://defenceproducts.ssm.gov.tr/Pages/ProductDeta%5C%5Cils.aspx?pId=163`.

[192] US Department of Army, 2017, *US Army Field Manual FM 34-130*, [Online], [Cited January 1994], Available from `https://fas.org/irp/doddir/army/fm34-130.pdf`.

[193] US Department of State, 2001, *Outer space treaty*, [Online], [Cited May 2012], Available from `http://www.state.gov/www/global/arms/treaties/space1.html`.

[194] US Marine Corps, 2009, *CAC2S Track management capability solutions*, [Online], [Cited June 2012], Available from `www.neco.navy.mil`.

[195] van der Merwe M, 2013, *The weapon assignment scheduling problem in a ground-based air defence environment*, MComm Thesis, Stellenbosch University, Stellenbosch.

[196] van der Merwe M & van Vuuren JH, 2013, *The weapon assignment scheduling problem in a ground-based air defence environment*, In process.

[197] van Staden HE, 2012, Postgraduate student at *Stellenbosch University*, [Personal Communication], Contactable at `heletjevs@gmail.com`.

[198] van Staden HE, 2013, *Attack technique classification in a ground-based air defence environment*, MComm Thesis, Stellenbosch University, Stellenbosch.

[199] Vigeh A, 2011, *Investigation of a simulated annealing cooling schedule used to optimise the estimation of the fiber diameter distribution in a peripheral nerve trunk*, MSc Thesis, California Polytechnic State University, San Luis Obispo (CA).

[200] Visser B, 2017, Lieutenant Colonel at *the South African National Defence Force*, [Personal Communication], Contactable at `bobv@reutech.co.za`.

[201] von Clausewitz C, 2010, *On war: The complete translation by Colonel JJ Graham*, The Floating Press, London.

[202] Votaw DF & Orden A, 1952, *The personnel assignment problem*, Symposium on Linear Inequalities and Programming, Washington D.C., pp. 155–163.

[203] Wagenaar EA & van Vuuren JH, 1999, *Quantifying the role of personal management style in the success of investment portfolios*, ORiON, **17**, pp. 13–28.

[204] Wallenius K, 2004, *Support for situation awereness in command and control*, (Unpublished) Technical Report, Royal Institute of Technology, Stockholm.

[205] Walters SA, Broady JE & Hartley RJ, 1994, *A review of information systems development methodologies*, Library Management, **15(6)**, pp. 5–19.

[206] Wickens CD, 2008, *Situation awareness: Review of Mica Endsley's 1995 articles on situation awareness theory and measurement*, Human Factors, **50(3)**, pp. 397–403.

[207] Wilson C, 2007, *Network centric operations: Background and oversight issues for congress*, (Unpublished) Technical Report RL32411, Congressional Research Service, Washington (DC).

[208] Wilton D, 2000, *The application of simulation technology in military command and control decision support*, [Online], [Cited May 2017], Available from http://www.siaa.asn.au/get/2395362604.pdf.

[209] Winston WL, 2004, *Operations research: Applications and algorithms*, Brooks/Cole, Belmont (CA).

[210] Wolfaardt PJ, 2009, Technology executive at *Reutech Radar Systems*, [Personal Communication], Contactable at pjwolf@rrs.co.za.

[211] Wolfram, 2017, *Mathematica*, [Online], [Cited August 2017], Available from http://www.wolfram.com/mathematica/online/?src=google&420.

[212] Zyl DV, 1999, *Threat analysis done for the Ground Based Air Defence System (GBADS) project office*, (Unpublished) Technical Report RS98-D-0007 R Rev B, ARMSCOR, Pretoria.