# COUNTERMEASURE ALLOCATION AND LOAD-OUT OPTIMISATION

by

**Nicholas Robert Osner**

Submitted in partial fulfilment of the requirements for the degree
Philosophiae Doctor (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

May 2019

## SUMMARY

---

### COUNTERMEASURE ALLOCATION AND LOAD-OUT OPTIMISATION

by

**Nicholas Robert Osner**

| | |
|---|---|
| Supervisor: | Prof. W.P. du Plessis |
| Department: | Electrical, Electronic and Computer Engineering |
| University: | University of Pretoria |
| Degree: | Philosophiae Doctor (Electronic Engineering) |
| Keywords: | Electronic warfare (EW), decision support systems, electronic countermeasures (ECM), radar countermeasures, genetic algorithms |

Currently, a number of systems exist for the purpose of electromagnetic countermeasure strategy optimisation. However, these systems simply allocate jamming resources as a whole, rather than specific jamming techniques. Further, these systems do not account for important interactions in the electromagnetic spectrum (EMS) such as the constructive and destructive interactions between different countermeasures. The future effects of countermeasure actions are also not taken into account, along with their interactions with the radar modes of threats as they progress from search stages through to guidance. Lastly, and importantly, these systems do not generate an optimal cartridge load-out for a platform.

In this work, a high-level decision-support system is proposed that aims at addressing these shortcomings. This is achieved using a process of threat evaluation and countermeasure allocation that is run prior to mission commencement based on available intelligence information. On a time-interval-by-time-interval basis, threats are first prioritised according to their characteristics and overall level of danger they present to the platform, before countermeasures are allocated so as to minimise this danger. Competing strategies are compared and optimised using a genetic algorithm according to various metrics such as risk, cost and levels of emission control (EMCON). The developed strategy can then be used in a decision-support role to inform, guide, and train mission

planners. Alternatively, the strategy can also be used to directly program the electromagnetic countermeasure system of the platform, and load the platform with the optimal cartridge load-out.

The major contribution of this work is that it determines an optimal cartridge load-out for a platform prior to mission commencement by taking into account the strong interaction between the use of expendables and active jamming techniques over the course of an entire mission. Due to the limited cartridge capacity of a platform, these are rationed throughout the mission so as to balance the competing objectives of cost and safety, without expending these resources prematurely and leaving the platform defenceless in the latter parts of the mission. Further, this mission-level optimisation allows for the prioritisation of different overall countermeasure strategy characteristics such as cost, safety, and levels of EMCON. This is achieved while taking into account the effects of, and interactions between, different countermeasures depending on technique, relative operating frequencies and bandwidths, antenna gain patterns, and the radar cross section of the platform. The effects of radar modes and their progression, as well as threat uncertainties are also accounted for. Importantly, this is all achieved in a high-level model that is simple and computationally efficient enough to allow for the rapid solution of the problem.

# LIST OF ABBREVIATIONS

| | |
|---|---|
| A | Acquisition stage |
| ARC | Adaptive radar countermeasures |
| B/W | Bandwidth |
| B | Frequency band of operation |
| CFAR | Constant false alarm rate |
| CMDS | Countermeasures-dispensing system |
| CP | Cover pulse |
| CSIR | Council for Scientific and Industrial Research |
| DARPA | Defence Advanced Research Projects Agency |
| DASS | Defensive aids subsystem |
| DIL | Dilution chaff |
| DIRCM | Directional infrared countermeasures |
| DIS | Distraction chaff |
| DRFM | Digital radio-frequency memory |
| ECM | Electronic countermeasures |
| EMCON | Emissions control |
| EMS | Electromagnetic spectrum |
| ES | Electronic support |
| EW | Electronic Warfare |
| G | Guidance stage |
| ID | Identification |
| IR | Infrared |
| IRCM | Infrared countermeasures |
| JSR | Jamming-to-signal ratio |
| M | Medium-band |

| | |
|---|---|
| MAW | Missile approach warner |
| MFT | Multiple false targets |
| MLW | Missile launch warner |
| MN | Medium-narrow band |
| MW | Medium-wide band |
| N | Narrow-band |
| NJ | Noise jamming |
| RCS | Radar cross section |
| RF | Radio frequency |
| RGPO | Range gate pull off |
| RWR | Radar warning receiver |
| S | Search stage |
| SADM | Ship Air Defence Model |
| SEWES | Sensors and Electronic Warfare Engagement Simulation |
| SNR | Signal-to-noise ratio |
| T | Tracking |
| TECA | Threat evaluation and countermeasure allocation |
| TEWA | Threat evaluation and weapon allocation |
| U | Undetected stage |
| UV | Ultraviolet |
| VGPO | Velocity gate pull off |
| W | Wide-band |

# TABLE OF CONTENTS

# CHAPTER 1 INTRODUCTION

## 1.1 PROBLEM STATEMENT

### 1.1.1 Context of the problem

In the theatre of battle, due to the large number of friendly and adversary platforms competing for superiority over one another, the electromagnetic spectrum (EMS) has become an exceedingly complex realm. The situation is further complicated by the fact that these platforms are often fitted with a whole array of systems including radars, communication systems, and jammers – often more than one of each. On top of this, there are also numerous interactions between the various countermeasures and counter-countermeasures implemented by all parties. Such interactions include the illumination of platforms, making them easier for adversaries to detect and track, as well as both constructive and destructive interactions between jamming strategies, which can either increase or decrease jammer performance. In light of this, it is clear that the optimisation of platform countermeasure strategy and load-out is necessary in order to maximise the probability of platform survival.

Currently, there are many threat evaluation and weapon assignment (TEWA) systems that allocate weapon systems to adversary platforms according to the level of threat they pose (e.g. [1], [2]). These systems represent a similar problem in that they require a process of threat evaluation and resource allocation, but they do not take into account the specific characteristics associated with actions in the EMS.

To account for these differences, a few threat evaluation and countermeasure allocation (TECA) systems have been proposed (e.g. [3] - [7] ). Most of these systems prioritise threats according to the level of danger they pose and then use either a jamming factor, or a probability of jamming success to determine the optimal allocation of countermeasure resources. However, these systems

only allocate jamming resources to be applied to the threats rather than the specific optimal jamming techniques to be used, with the exception of the system developed by Noh and Jeong [3]. Even then, this system only chooses between either active or passive countermeasures against either radio frequency (RF) or infrared (IR) threats, rather than specific techniques. Secondly, the constructive and destructive interactions between different jamming techniques and their signals are not taken into account, despite the fact that such interactions are inherent in all actions in the EMS. This is one of the major differences to TEWA systems, in that jamming strategies will work effectively for some threats, but illuminate the platform for others depending on the frequencies and bandwidths used, as well as on threat radar modes. Existing TECA systems also do not take into account the future effects of current countermeasure actions which result from the progression of radar modes of the threats from search to guidance, or the effect of threat counter-countermeasures such as flare rejection capabilities. Further, these existing systems do not take into account the interaction between active and passive countermeasures, hence not allowing for effects such as the active illumination of chaff. Finally, the inherent operational and positional uncertainty involved in the threat environment are also not taken into account by these systems.

Moreover, a platform has a limited capacity of passive countermeasure cartridges due to weight and space restrictions. This limited capacity requires that the use of cartridges be rationed correctly throughout the mission so as to balance the competing objectives of cost and platform safety, without prematurely expending these resources and leaving the platform defenceless. Further, this limited capacity requires the optimal load-out of cartridges to be determined prior to mission commencement, where there is a strong interaction between the use of expendables and the use of jammer channels. This requires the jamming and expendable strategies to be determined simultaneously.

Further, there are a number of competing objectives in the development of countermeasure strategies for a platform. Some objectives include the financial cost of the strategy, the levels of emissions control (EMCON), and the level of risk likely to be encountered by the platform. Obviously, a cheaper mission strategy is preferred. The use of jamming techniques should also be minimised in order to implement EMCON, and reduce the likelihood of platform detection. Whilst on the other hand, more aggressive countermeasure strategies reduce risk for the platform. Therefore, a balance between these objectives must be achieved in order for such a system to be effective.

Ideally, this optimisation should be performed using a physics-based or parameter-level approach as it would be the most accurate. However, such systems are far too computationally expensive to be of substantial use, with systems like SEWES and SADM taking multiple days to complete similar or simpler simulations on powerful computer clusters [8]. This is simply far too long for time-sensitive missions, where the threat environment would have changed by the time an optimal strategy is determined, rendering it useless.

### 1.1.2 Research gap

In light of the above context, it is clear that computer-based optimisation is necessary in the process of threat evaluation, countermeasure allocation and cartridge load-out determination. Current approaches fall short by either being too simplified, too computationally expensive, or in the case of cartridge load-out optimisation, non-existent altogether. Therefore, due to the inter-relatedness of these tasks, there is a gap for a high-level decision-support system that performs the entire task whilst accounting for all of the factors listed above, including the effects of different jamming techniques on each threat, interactions between jammers, radar modes and their progression, the effects of different operating frequencies, and threat positional and operational uncertainties. Further, such a system should include a number of user-definable parameters to allow a wide range of systems to be modelled, as well as a user-weighted objective function that will allow for the prioritisation of different mission strategy characteristics.

## 1.2 RESEARCH OBJECTIVE AND QUESTIONS

The main research question to be addressed by this work is whether it is possible to attain a sufficient compromise between computational speed and accuracy in an EW model, such that an EW load-out and tactics optimisation system would be effective. In other words, is it possible to achieve reasonable results from such a system in a reasonable amount of time? The reason for this is that military missions are often only devised and planned relatively close to their intended commencement in order to prevent the adversary from becoming aware of these plans. This also prevents too many changes occurring on the battlefield before such a plan is put into action. In either case, the optimal EW load-out and tactics must be determined rapidly enough to prevent the adversary from making changes (either intentional or not) that could render these strategic plans ineffective, no matter how accurate and effective they were to begin with. On the other hand, if

insufficient accuracy is achieved, the solution chosen may not be close enough to the optimal solution, resulting in an unnecessarily high probability of the platform not surviving the mission or the mission failing, with both having potentially dire consequences in financial cost, human life, and ultimately state security.

The second question to be addressed is as to whether the information regarding the optimum EW load-out and tactics for a platform can be succinctly, accurately and effectively relayed to the human operator so as to convince them that the chosen tactics are indeed close to the optimum.

## 1.3 APPROACH

The overall approach to the problem is one of threat evaluation and countermeasure allocation. As such, threats must be allocated danger values according to their characteristics for the purpose of prioritisation. Thereafter, the effects of possible countermeasure strategies must be determined and compared using a jamming factor and an objective function. These strategies must then be optimised in order to improve strategy performance, before being validated through analysis. These individual processes are broken down below.

### 1.3.1 Danger values

The first objective is to determine a measure of the level of danger each threat presents to the platform for prioritisation purposes. This is achieved using a danger value calculated from each threat's characteristics including: its radar stage, the time before it will progress to the next radar stage, the time required for its projectile to reach the platform, the probability of the platform encountering it, and its range-adjusted accuracy. Current methods, such as those used in TEWA systems, typically take into account a number of these characteristics, but importantly do not take into account the progression of a threat's radar stages.

### 1.3.2 Jamming factor

The next objective is to develop a method of taking into account the effect of jamming on the danger value presented by each threat, where the goal is for the platform to use jamming to reduce the amount of danger it experiences over the mission. As stated previously, current approaches do

not take into account the various interactions inherent in the EMS. These include: interference between jamming techniques and their effect on different radar modes, interactions between jammers, the progression and jamming of radar modes, and the effects of different operating frequencies. The core of this work is to take all of these factors into account when determining the effect of countermeasures.

### 1.3.3 Objective function

In order to optimise the platform's countermeasure strategy, an objective function is required to calculate the relative strength of alternative strategies according to their characteristics. These characteristics include the level of risk posed to the platform, the cost of the strategy, and the levels of EMCON, where user-defined weights modify the prioritisation of these characteristics according to user requirements. Current systems focus on reducing the risk to the platform and do not take into account the cost of the strategy or prioritise any control of emissions.

### 1.3.4 Optimisation algorithm

In order compare the extremely large number of possible strategies, an efficient combinatorial optimisation method is required. This is can be achieved using a genetic algorithm due to its robust nature. However, a standard genetic algorithm is not a very fast method of optimisation, and as a result, this work requires specialised genetic operators that take into account the unique characteristics of the problem. These include seeding the problem with an intuitive approach to the problem, and mutation operators that modify poor strategies to jam successful threats at the last opportunity.

### 1.3.5 Validation of results

Due to the lack of comparable systems in the published literature, the classified nature of the platforms and systems involved, and the inability to truly quantify soft characteristics like danger, the reasonableness of the results of this work will be validated by the ability to dissect and explain the developed strategies in the context of the mission. The system as a whole, including its approach and results, will then be further validated by the peer review process required for publication in a respected scientific journal.

## 1.4 RESEARCH GOALS

Since the goal of this research is to develop an effective system with a sufficient compromise between computational speed and accuracy, its performance must be assessed by using the generation of a reasonable countermeasure strategy and load-out in a reasonable amount of time as a proxy for this compromise. However, due to the lack of similar systems in the literature, there is no work with which such a system can be quantitatively compared. Instead, a reasonable strategy is defined as one that makes sense in the context of the mission, where analysis of the inputs and outputs of the system will be used to validate the reasonableness of the system. The goal of developing such a strategy in a reasonable amount of time is defined as achieving a result in a time significantly less than what a physics-based or parameter-level approach would require. The overall appropriateness of the model would then be validated through the publication of the work in reputable peer-reviewed journals.

## 1.5 RESEARCH CONTRIBUTION

The original contributions of this work are broken down and summarised below. Since the process of threat evaluation using a calculated danger value has been widely used in TEWA and other TECA systems, the contributions pertain to the process of countermeasure allocation and the process of using this to determine an optimal cartridge load-out. In the case of the first three contributions, these pertain to the calculation of the jamming effect value ($E$) that forms a part of the countermeasure allocation process by accounting for the effect of jamming on the danger value of a threat.

### 1.5.1 Stage effectiveness

One of the major contributions of this work is to model a threat's engagement procedure as a series of radar mode progressions or stages, in order to use this information in the prioritisation of threats, as well as to take these modes into account during the allocation of jamming. A second major contribution is the allocation of specific jamming techniques to the jamming of threats, rather than simply allocating jamming resources. The interaction between these two factors forms an integral part of the process of countermeasure allocation, and this is accounted for in the stage effectiveness factor (SE). This factor, as its name suggests, accounts for the effectiveness of a specific

countermeasure technique on the examined threat's radar stage in the examined time interval, by either increasing or decreasing the jamming effect. This accounts for the fact that some countermeasure techniques will effectively jam certain radar stages, whilst illuminating the platform to others.

### 1.5.2 Technique interaction

Another advantage of allocating specific jamming techniques is that the interactions between different techniques can be accounted for in the case of a platform with multiple jamming channels. These interactions can be either positive or negative and depend on the relative frequency bands and bandwidths of each technique. This is achieved using the technique interference factor ($I$) that either increases or decreases the jamming effect accordingly.

### 1.5.3 Cross effect

The jamming effect is calculated on a threat-by-threat basis for each individual jamming channel. The effect of each jamming channel on a threat is heavily dependent on the relative frequency bands used by both the threat and the jamming technique. If the jamming is being performed in a different part of the frequency spectrum, it will have a reduced or no effect. The cross-effect factor (CE) is used to modify the jamming effect of a particular jamming channel on a particular threat according to their relative frequency and bandwidths.

### 1.5.4 Cartridge load-out

The problem of determining an optimal cartridge load-out for a platform according to an optimum countermeasure strategy has not been previously considered in the literature. This is a significant contribution because a platform has limited capacity for passive countermeasure cartridges due to weight and space restrictions. As a result of this, cartridge use must be rationed correctly so as to balance the competing objectives of cost and platform safety, without expending these resources and leaving the platform defenceless. Importantly, there is a strong interaction between expendable strategy and active jamming strategy, thus requiring these to be optimised simultaneously in order to determine the cartridge load-out.

### 1.5.5 Active and passive countermeasure interaction

The next major contribution of this work follows on from the previous one. This is due to the fact that the interaction between active and passive countermeasures needs to be accounted for in order to optimise their simultaneous use and in turn to develop optimal strategies for both. Further, accounting for such interactions allows for the implementation additional techniques such as illuminated chaff that require a combination of active and passive jamming techniques. The interference effect of chaff on the active jamming channels is taken into account using a multiplicative chaff interference factor (CI) that either increases or decreases the active jamming effect on a threat according to the type of chaff countermeasure being used. The illumination of chaff by the two active jamming channels is taken into account using a chaff illumination factor ($U$) for each active channel. These factors are calculated according to the specific chaff technique and active jamming techniques being considered, and the resultant interference. Further, this interference is adjusted for the relative operating frequencies and bandwidth of the threat being examined and the active jamming technique using the cross-effect factor. This frequency adjusted chaff illumination factor is then used to either increase or decrease the jamming effect of the chaff accordingly.

### 1.5.6 Weighted objective function

Current systems focus solely on reducing platform danger levels or maximising jamming resource effectiveness. However, this is not the sole countermeasure strategy characteristic that is of importance. Other factors such as the financial cost of a strategy and the levels of EMCON are extremely important when choosing between strategies of similar risk levels. As a result, a weighted objective function is implemented in this work to allow the user to determine the levels of prioritisation of these different characteristics.

### 1.5.7 Antenna gain pattern and platform RCS

A very important characteristic of modelling interactions in the EMS is a platform's radar cross section (RCS) which forms a vital part of the radar equation, and hence the signal energy received by a threat's radar, and as a result its danger to the platform. Further, a platform's antenna direction and gain pattern have a large effect on the jamming signal power emitted by a platform in each direction, and hence the jamming signal power received by a threat's radar. The combination of

both these effects results in a very large effect on the jamming-to-signal ratio for every threat, and hence the effect of jamming and illumination in general. Essentially, the effect of the platform's RCS is accounted for using a multiplicative factor on the danger value of a threat according to its position relative to the platform. On the other hand, the antenna gain pattern is treated as a multiplicative factor on the jamming effect value for each channel and each threat, according to its relative position to the platform and the antenna direction. Note that the contribution of this work is the use of RCS and antenna gain patterns in this countermeasure optimisation process, rather than the actual RCS and antenna gain pattern models themselves.

## 1.6 RESEARCH OUTPUTS

This work resulted in the generation of two journal papers – one published, and one that is currently under review. A summary of this work and its training applications were also presented at the Aardvark Roost of the Association of Old Crows' (AOC) 14th Little Crow Conference at the University of Pretoria [9], and the 19th Symposium of Operational Applications in Areas of Defence (XIX SIGE) in Brazil [10]. Further, a co-authored paper was presented at the AOC's fifth International Conference on Electronic Warfare (EWCI) in India [11]. These are summarised below.

### 1.6.1 Threat evaluation and jamming allocation

This paper was published in IET Radar, Sonar and Navigation in April 2017 [12]. It details the basic, overall concept of threat evaluation and jamming allocation, and excludes the allocation of passive countermeasures, as well as the genetic algorithm and its associated objective function. Hence, the focus of this paper is the contribution of this work to the process of jamming allocation – in that it takes into account the effects of different jamming techniques, interactions between jammers, radar modes and their progression, the effects of different operating frequencies and threat uncertainties.

### 1.6.2 Countermeasure allocation and expendable load-out automation

This paper is currently under review for publication in IEEE Transactions on Aerospace and Electronic Systems [13]. It details the additions to the work contained in the previous paper - the

addition of expendables in the form of chaff and flare countermeasures, and decoys in the form of a towed decoy. Hence, the focus of this paper is the contributions of cartridge load-out optimisation, the use of a weighted objective function in strategy selection, as well as the process of taking the interactions between active and passive countermeasures into account. Further, this work takes into account the effect of radar modes on passive countermeasure allocation, and the effect of threat counter-countermeasures such as flare rejection capabilities.

### 1.6.3 Aardvark roost of the AOC's 14th little crow conference

A summary of the work contained in the Threat Evaluation and Jamming Allocation paper was presented at this conference in Pretoria on 26 September 2016.

### 1.6.4 Electronic warfare training applications of decision-support systems

This paper was presented at the 19th Symposium of Operational Applications in Areas of Defence (XIX SIGE) in Sao Jose dos Campos, Brazil, in September 2017. It outlined the potential training applications of this work for EW operators and decision makers, as well as for military personnel in general. Hence, the focus of this paper is on how this work can be used to create a visual and interactive training tool that can help build an intuitive understanding of EW and the EMS as a whole in all personnel. It also demonstrates how this work can be used to create a benchmark against which EW operators and decision makers in training can be quantitatively evaluated and shown where their strategies are lacking. Lastly, it demonstrates how this work can be used to identify more effective approaches to scenarios that human strategists may not be aware of, where through practice these strategists can learn to identify such scenarios and hence apply these superior approaches.

### 1.6.5 Cognitive electronic warfare (EW) systems as a training aid

This paper was written and presented at the AOC's fifth International Conference on Electronic Warfare in Bangalore, India, in February 2018 by W.P. du Plessis. It also consisted of a summary of this work and how it, and cognitive EW systems as a whole, can be used for training purposes. This consisted of a discussion of how this system can be used for trainee solution evaluation, and how the analysis of a single solution can be used to generate greater insight into strategy generation, along with the use of intuitive graphical displays. Most importantly, this paper covered

how the comparative analysis of multiple solutions to the same scenario can be used to add even further value to the training process.

## 1.7 OVERVIEW OF STUDY

In Chapter 2, a literature study of the relevant areas of EW and optimisation is presented, along with current and future approaches to the problem. It is found that there is a distinct lack of published research in the field of EW countermeasure strategy optimisation. Current systems do not take into account a number of important interactions that are inherent to the EMS, or are simply too slow for the task. Further, there are no methods for determining optimal cartridge load-outs for platforms. As such, it is found that there is a need for a system that takes into account all of these interactions, and develops an optimised countermeasure load-out in a computationally efficient matter that can generate a useful result in a reasonable amount of time.

In Chapter 3, the approach used to model and optimise the problem is discussed. This begins with an overview of the approach and the assumptions made, before delving into the specifics of the model and the process of threat evaluation and countermeasure allocation itself. Thereafter, the optimisation process is detailed, along with the specifics of the specialised operators used to improve performance.

In Chapter 4, the results of the completed system are presented and analysed. This begins with a detailed description of the example scenarios used for demonstrating this work. Thereafter, an in-depth analysis of a generated countermeasure strategy is presented for both the challenging full scenario, and more-simple half scenario. The chapter is then concluded by analysing the performance of the optimisation algorithm itself and the implemented specialised operators, including a comparison of the different stop criteria, and Pareto solutions. It is found that the generated strategies indeed do make sense in the context of the mission. Further, it is found that the specialised operators do improve algorithm performance, the different stop criteria perform their intended functions, and the Pareto designs effectively offer the user useful alternative strategies.

In Chapter 5, the implications and applications of the results are discussed along with the observed shortcomings and how these may be overcome in future work. Overall, it is found that the generated strategies indeed do make sense in the context of the mission, are generated in a

reasonable amount of time, and can be succinctly, accurately and effectively relayed to a human operator, hence indicating that the research objectives were successfully achieved. However, it was found that the assumptions made in this work are a limiting factor, and more in-depth modelling is required in future work. Further, the implemented specialised operators have a number of unintended consequences on the generated results, which can potentially be overcome with additional operators in future work. Lastly, it was found that this work in its current form is primarily suited to training applications, but can also be applied to a number of other applications in future such as decision-support, direct strategy and load-out generation, and real-time operation. Further it could also be incorporated into other existing low-level systems as an initial coarse optimisation stage, or be used in route and manoeuvres optimisation.

In Chapter 6, some concluding remarks are made, and the research contributions of this work highlighted. This includes a summary of the work itself, the results attained, its potential applications, as well as its shortcomings and some areas of potential future improvement

# CHAPTER 2 LITERATURE STUDY

## 2.1 CHAPTER OVERVIEW

In this chapter a literature study of the relevant areas of electronic warfare and optimisation are presented. It begins with an overview of EW in Section 2.2 which covers EW modelling, as well as the weapon systems, radar stages, and countermeasure techniques that need to be modelled. Section 2.3 then presents a comparison of the different approaches to combinatorial optimisation, along with what approaches have been used in similar systems before selecting a technique accordingly. Section 2.4 then discusses the various specifics of the chosen optimisation approach including the design schema, seeding, reproduction, operators, population schemes, multi-objective optimisation, Pareto optimality, and stop criteria. Thereafter, the current approaches to this problem of countermeasure optimisation are discussed in Section 2.5, before finally covering potential future approaches in Section 2.6.

## 2.2 ELECTRONIC WARFARE

In this section a basic literature study of the relevant areas of EW is presented. This includes all areas necessary for the development of a basic EW countermeasure optimisation system including the various weapon systems, radar stages, countermeasure techniques, EW modelling, and modern EW systems. This serves to justify the design of the system, as well as many of the simplifying assumptions made in the process of modelling the vast field of EW.

### 2.2.1 EW modelling

A lot of information must be known about an engagement in order to accurately model it. The required information includes all player characteristics such as counter- and counter-countermeasures, interactions, and radar cross sections, amongst others. All this information must

be included in the model at the required level of fidelity that is the best possible compromise between speed and accuracy. Some parameters whose level of fidelity must be considered are: player locations, time increments of calculations, player velocity vectors, player RCS models, EW signal parameters as well as antenna orientations [14].

Due to the fact that this work is aimed at, amongst other things, developing an optimal cartridge load-out prior to mission commencement, such information would have to be gathered by intelligence gathering activities and programmed into the model. This is in contrast to real-time systems that would rely on electronic support (ES) systems in order to obtain such information. Further, since route planning is outside the scope of this work, it is assumed that the optimum route through the battlefield is known and programmed as a fixed path accordingly. The model then is required to replicate the entire mission and all associated interactions for various EW load-outs and tactics in order to determine their fitness according to a number of metrics, before the fittest solution is selected.

### 2.2.2 Weapon systems

1) *Artillery*

There are two main types of artillery, those with explosive rounds and those with non-explosive rounds. The main difference is that the non-explosive rounds require a greater level of accuracy as they need a direct hit in order to inflict damage. However, both types of systems are limited by their angular accuracy, which means that the accuracy of these weapon systems decreases linearly with range [15]. Therefore, when modelling these systems, accuracy must be constant at short range, before decreasing linearly at longer range. This is due to the fact that when a target is sufficiently close, the angular accuracy of the system is not the limiting factor, instead it is weapon ballistics.

2) *Guided missiles*

Guided missiles have the ability to change their trajectory after being fired, hence reducing the ability of a target to evade them. Missiles usually offer some trade-off between cost and performance, with some systems (including command missiles and beam-riding missiles) being guided from the initial platform that fired them and others (including active and semi-active missiles) being guided by a receiver built into the missile. This means that command and beam-

riding missiles are limited by the angular accuracy of the radar of the platform that fired them, and hence experience the same linearly decreasing accuracy as artillery systems [15]. This too needs to be accounted for in modelling along with the fact that these systems have an additional radar stage during which attempts to protect the platform can be made.

### 2.2.3 Radar stages

Due to the competing technical requirements of search and tracking radars, separate systems are usually used for each. This means that a search-radar system is used to initially detect, identify, and locate a target. Thereafter, the target is handed over to the tracking-radar system which must acquire the target using the location obtained by the search radar, before tracking the target accurately. This information is then processed by the fire control centre which uses velocity, location and direction information in conjunction with the known velocity of the projectile to determine a point of interception. The number of projectiles required to achieve a certain probability of kill are then launched once the target is within range [15]. In modelling, this means that the engagement procedure can be approximated by a sequence of radar stages beginning with a search stage, before acquisition, tracking and then finally a guidance stage. This also provides a method of threat prioritisation because the further along this sequence a weapon system is, the closer it is to inflicting harm to the platform, and hence the greater danger it presents.

### 2.2.4 Active jamming techniques

Due to the immense scale of the field of electronic warfare and the sheer scale of the number of implemented counter- and counter-countermeasures, it is infeasible to model every possibility in this work. As such, five major types of active countermeasure are considered for modelling in this work due to their widespread and pervasive appearance in elementary EW textbooks [14] [15]. These are detailed below.

*1) Range gate pull off*

A range gate pull off (RGPO) is a jamming technique, usually used against tracking radars, whereby a strong false target is generated with the same characteristics (both in range and velocity, relative to the adversary's radar) as the platform, before the apparent range of the false target is slowly altered. The effect of this is to lead automatic target tracking systems away from the

platform in range before removing the false target, thus causing a break-lock. Once break-lock has occurred, the radar must start the search and acquisition phases again before lock can be re-established. In high-speed encounters, such as in a missile-lock situation, this time can be sufficient for the missile to have already missed the target platform before lock can be re-established [15]. This means that due to its very nature, RGPO is only effective at breaking the lock of tracking radars, and in fact, can illuminate the platform to search radars because of the strong false target that is generated.

*2) Velocity gate pull off*

A velocity gate pull off (VGPO) is a jamming technique very similar to RGPO, except that the false target is altered in its perceived velocity (instead of range) by altering the angular frequency of its modulation function. The main advantage of this technique over a RGPO is that the false target velocity can be decreased in order to try and hook a missile onto stationary clutter [15]. As a result, VGPO is also effective against tracking radars, with a slightly greater effect on guided missiles than RGPO. However, it suffers the same issue of platform illumination for search radars due to its use of a false target.

*3) Noise jamming*

This technique is one where an EM disturbance is created in order to prevent the detection of the platform, where the disturbance is usually a noise of a given bandwidth centred around the carrier frequency of the attacking radar. Ideally, this noise should be very similar to the thermal noise experienced by the radar so that the jamming itself is not detected. Further, the noise needs to be generated over a bandwidth greater than or equal to that used by the pulse of the attacking radar in order to be most effective. However, most systems are only able to generate a certain amount of effective radiated power, which must be then spread over this bandwidth [15]. Due to the nature of noise jamming (NJ), the large amount of emitted signal energy can make the platform easier to track, as well as trigger track-on-jam counter-countermeasures. Therefore this technique should be modelled as being effective on search radars, but illuminate the platform to tracking radars. Further, due to the limited power of systems, there exists the possibility of a trade-off between bandwidth and noise power. This allows for the potential of either jamming a single target effectively, or multiple targets to be jammed less effectively over a wider bandwidth.

*4) Cover pulse*

A cover pulse (CP) is a technique whereby a very large false target is generated in and around the same location as the platform on the attacking radar's range-Doppler graph. The effect of this is to deceive automated radars that use a constant false alarm rate (CFAR) detector. This works because a CFAR detector uses the average local noise do determine the threshold of detection for each gate. This is then deceived by the cover pulse by it raising the local noise and hence detection threshold near the target, thus reducing the likelihood of its detection. A cover pulse is generated by creating a number of false targets in the immediate vicinity of the platform and if the exact CFAR method used by the attacking radar is known, then the placement of these false targets can be optimised to reduce likelihood of detection [15]. The main function of a cover pulse is to prevent detection by a search radar's CFAR detector. However, as a result of it creating a large target, it illuminates the platform to tracking radars, and should be modelled accordingly.

*5) Multiple false targets*

The multiple false target (MFT) jamming technique is one whereby the platform generates numerous false targets in its vicinity in order to over-load and confuse the target detection systems of the attacking radar. This technique is effective at preventing detection by search radars and can be made effective against even complex radar systems by creating realistic, moving targets with accurate target profiles. However, it is ineffective against tracking radars that have already detected the platform [15]. As such, this technique should be modelled as being effective against search radars, but have no effect on tracking radars.

### 2.2.5 Chaff

Chaff is one of the oldest forms of radar countermeasure and still remains a popular choice due to its simplicity and low cost [15]. It consists of a cloud of miniature, usually half-wave, dipoles that create a large radar return in the vicinity of the platform, masking it from an adversary's radar. However, due to Doppler processing in modern radars and the large velocities of platforms, the useful life of chaff after it has been launched is very short and lies in the region of half a second in airborne applications. In order to work effectively, the chaff cloud must be dispersed across the entire radar resolution cell (the combination of elevation, azimuth and range resolutions) of the adversary radar within its useful life period. This cannot be achieved with one large burst of chaff due to the time taken for the cloud to bloom, instead it is achieved with a number of bursts at very

short intervals, known as a salvo, which can then can be repeated at longer intervals. Therefore, the chaff dispersal strategy must be optimised individually for each threat radar in order to be effective [15].

There are two main techniques of chaff use: distraction and dilution. Distraction acts in a similar way to the multiple false targets jamming strategy in that it creates false targets at locations different to that of the platform, with the aim to confuse acquisition and search stages of an adversary radar [15]. Dilution on the other hand is aimed at breaking lock of tracking and guidance stages by dispensing chaff across the radar cell in which the platform is situated in an almost range or velocity gate pull off manner [15].

There are also two main methods of counteracting the use of Doppler processing in modern radar systems. The first is illuminated chaff, where the platform illuminates its own chaff with a noise or deception signal in order to create false targets for the adversary radar. This allows the platform to impose the necessary Doppler shift onto the chaff, thus making it more effective and turning it into a cheap, off-board decoy [15]. The second method requires the platform to perform a manoeuvre just prior to chaff launch so as to present a low Doppler profile to the adversary radar, thus allowing the platform to be masked by the decelerated chaff [15].

Therefore, chaff countermeasures should be modelled as salvos of cartridges that are optimised for different threat types, where the chaff can be dispensed in either a distraction or dilution manner depending on the radar mode of the threat. Threat counter-countermeasures such as Doppler processing should be taken into account along with the effects of necessary manoeuvres. Further, the effects of chaff illumination by active jamming techniques should also be accounted for so as to allow for the use of illuminated chaff countermeasures. Lastly, in order to make the most effective use of limited cartridge capacity onboard a platform, the chaff cartridge load-out of a platform should be optimised for a mission. This serves to both improve the likelihood of survival of a platform, and maximise the use of limited and costly resources.

### 2.2.6 Decoys

A decoy is an off-board device whose goal is to appear more attractive to an adversary than the platform, so as to lure the threat towards itself. This is achieved by the decoy transmitting a radar echo very similar to, but more attractive than, the echo of the platform. Decoys are most effective

against tracking and guidance radar stages where angular errors must be created in the adversary radar. This is assisted by the separation between the decoy and platform [15].

Decoys come in many forms and may be active or passive, towed or launched. Due to the sheer number of such systems in operation, these need to be narrowed down for implementation purposes. Thus, in order to be representative of modern systems, such as the Eurofighter Typhoon [16], fibre-optic towed decoys will be the focus in this implementation. Fibre-optic decoys operate by generating the required jamming signal in the ECM system of the main platform and transmitting this to the decoy via a fibre-optic cable. This means that these decoys are capable of more advanced jamming techniques and counter more advanced threats than repeater decoys, which simply repeat a radar signal back at a higher effective radiated power than the platform [15].

Therefore decoys should be modelled as another antenna through which active jamming techniques can be directed, where the use of the decoy results in greater effectiveness against tracking and guidance radar stages.

### 2.2.7 Flares

Infrared (IR) missiles, especially in the form of portable shoulder-mounted missiles, represent the greatest threat to airborne platforms. This is due to the fact that these weapons are relatively inexpensive, widely available and extremely portable, and secondly, they are passive systems [15]. IR guided missiles passively track a target by exploiting the IR radiation generated by the heat from engines, wind resistance effects, etcetera. This means that these systems are able to be fired without emitting any EM signal that could warn the platform of potential engagement. As a direct result of this, one of the only means of detecting these weapons is to use devices capable of detecting their IR or ultraviolet (UV) emissions. Such devices are called missile launch warners (MLW) and are used to give the platform sufficient warning to be able to perform countermeasures [15]. An alternative approach is to use a specialised radar capable of detecting an approaching missile. These are known as missile approach warners (MAW) and usually consist of a small continuous wave or pulse-Doppler radar capable of detecting fast approaching targets with small radar cross sections [15]. However, both systems are unable to accurately identify the type of approaching missile, requiring the type of missile and associated countermeasures to be known in advance if a non-generic countermeasure is to be used.

The main types of countermeasure for IR missiles are flares, laser directional infrared countermeasures (DIRCM), as well as general infrared countermeasures (IRCM). However, flares are the most commonly used type of IR countermeasure due to their relatively low cost and simplicity [15], and hence will be solely considered in this work. Flares are devices that are stored in cartridges that are then dispensed from the platform before generating IR energy at a far greater intensity than the platform, generating a far brighter target for the approaching missile, hence drawing it away. Flares are often discharged in salvos in order to improve performance, prevent the missile from re-acquiring the platform, and counter the various tracking technologies [15].

IR missiles are usually divided into three generations of guidance capabilities. Generation one missiles have no means of discriminating between flares and the target platform, generation two missiles include early flare rejection capabilities and generation three missiles make use of the latest technologies and hence have robust flare rejection [15]. As a result, flares are assumed to completely effective against generation one, whilst generation two and three missiles require increasing levels of optimised manoeuvres, flares and dispensing strategies for flares to be effective [15].

Therefore, IR systems must be modelled as threats that remain undetected until fired and they enter a guidance stage, where it is assumed the platform must be equipped with either a MLW or MAW. Thereafter, the lock of these missiles must be broken using salvos of flares, where the cartridge load-out of a platform should be optimised so as to maximise the use of limited space, and limited, costly resources. Finally, the generation of an IR system, and its associated flare rejection ability, should also be taken into account.

### 2.2.8 Modern EW systems

The EW capabilities of the platform being modelled should be both functional in the context of showing the abilities of the developed model, and representative of modern systems as a whole. An example of such systems is the Saab EWS 39 electronic warfare suite equipped on Gripens [17]. This EW suite consists of 3 channels of countermeasures that can be used simultaneously: a noise channel, a repeater channel and a digital radio-frequency memory (DRFM) channel. These channels can implement a variety of jamming techniques such as masking, saturation, deception and false target techniques. Further, this suite includes a countermeasures-dispensing system (CMDS) that controls the dispensing of chaff and flares. It is able to store a load of 80 expendable

cartridges with dimensions of one by two inches or any mixed load of different size cartridges. The CMDS is able to control the dispensing of these cartridges in terms of their delay time, the number of salvos fired, the time between salvos, number of cartridges per salvo and the time between cartridge ejections.

A further example of a modern system is the Defensive Aids Subsystem (DASS) installed on the Eurofighter Typhoon [16]. This system consists of two wingtip ECM pods, MAW, chaff and flare dispensers, and fibre-optic towed decoys.

Therefore, the EW capabilities of the platform used in the model should include multiple active jamming channels, a cartridge dispenser, MAW, and a fibre-optic towed decoy in order to representative of modern systems. Importantly, the interactions between these various elements must be accounted for.

## 2.3 OPTIMISATION METHOD SELECTION

Discrete variable optimisation problems can be divided into a number of different categories according to their characteristics [18]. The first category is one where problem functions are twice continuously differentiable and non-discrete variable values are allowed during the solution process. An example of such a problem would be the design of a structure using discrete plate thicknesses available on the commercial market. The types then progress through ones where the problem functions are non-differentiable at some points, and through to ones where non-discrete values cannot be used in the solution process. The extreme limit of these categories, are combinatorial problems that consist of purely non-differentiable, discrete problems. The problem of selecting an optimal countermeasure strategy for a platform is one such combinatorial problem, as no meaningful or useful values can be used between the discrete variable values such as the jamming techniques allocated to each channel for each time interval.

The difficulty with combinatorial problems is that there is no information available while solving the problem as to in which direction the objective function improves, thus requiring a random search. This is obviously much slower than being able to use derivatives to know exactly in which direction to disturb variables in order to attain improvement [18]. It is this fact that makes the fast and efficient optimisation of this problem difficult.

### 2.3.1 Combinatorial optimisation approaches

The most common method for solving such combinatorial problems is the use of various nature-inspired methods [18]. These methods are ones that are based on the process of generating an initial population of potential solutions before randomly altering them until a sufficiently suitable solution is found, where the alteration process is based on a natural phenomenon such as evolution. The biggest advantage of these methods is that they are relatively robust and can be used on problems where very little information is known about the problem [18].

Another option is the branch and bound method [18], which involves the process of choosing a starting solution of discrete variable values before branching off in different directions, where each variable is altered. This process is repeated along each branch until the first feasible solution in a branch is found, where thereafter the solutions would decrease in performance. The branch would then terminated, whilst the others continue. Each branch is then also terminated once it is determined that a better solution cannot be found along that branch. In this way, this method prevents the need for calculating the objective function values for weaker solutions. The issue with this technique is that it requires prior knowledge about a problem, including whether it is increasing or decreasing in a certain direction. This is an issue for the countermeasure strategy problem due to the relatively disconnected performance of different design variables.

Yet another option is the multi-agent distributed cooperative auction algorithm [5]. This is an algorithm whereby a random bidding sequence for the agents (which would be the jamming channels in this case) is initially chosen. Bidding is completed by the first bidding agent selecting the strategy with the greatest fitness. The environment and associated fitness function are then updated before the next agent's bid. This is repeated until each agent has selected a solution, with the entire process repeated with a new bidding sequence, if time permits, in order potentially develop a stronger solution. This results in the rapid generation of solutions. However, these solutions are relatively less optimised and unlikely to find the optimum.

### 2.3.2 Similar systems

Since the performance of an optimisation algorithm is dependent on the problem to which it is applied, the optimisation approaches to similar problems must be examined. Unfortunately, there

are few published systems comparable to this work due to its nature. Therefore generally similar EW problems were examined.

Wei, Ziming, and Lin [6] applied a genetic algorithm to the optimisation of the assignment of ground-to-air jamming resources to attacking adversary platforms in the context of defence. It was found that traditional allocation methods were computationally intensive and time consuming, and that genetic algorithms successfully overcame these shortfalls. This implementation included single point crossover, single gene mutation, and elitism.

Zhai and Zhuang [4] successfully used a genetic algorithm for the purpose of optimising multi-platform cooperative jamming. In this problem, multiple platforms are each allocated a threat to jam in order to protect specific targets, where the platforms originate from a number of bases. It was found that the algorithm effectively solved the problem and can be used to shorten the time required for decision making.

Liu *et al.* [5] developed a jamming resources assignment algorithm for the purposes of real-time decision support. This system was required to distribute the jamming capabilities of multiple platforms over a number of different adversary platforms by simply allocating each a platform a threat to jam. The authors acknowledged the successful use of genetic algorithms in other systems by Shanwei and Na [19], and Wang and Li [20], that assigned jammers to adversary radars. However, it was decided to use a multi-agent distributed cooperative auction algorithm due to the heavy computational requirements of the genetic algorithms in these implementations. The result was a relatively less optimised solution that could be generated very rapidly, which was sufficient for real-time decision support in a smaller solution space.

Finally, Kang *et al.* [7] investigated the problem of finding optimal EW countermeasures against threats in real time. A decision theoretic architecture was used for rapid development of solutions. However, this approach required the calculation of the utility of each possible strategy for comparison in an exhaustive-search approach. This was relatively successful in the limited solution space of the problem due to the limited jamming options available to the platform.

Therefore it is seen that in this type of problem nature-inspired techniques, and genetic algorithms in particular, are favoured. The problems where other approaches were selected were ones where

real-time functionality was preferred over more effective solutions, or where the solution space was sufficiently small. However, this work, due to its prior-to-mission-commencement application, favours a compromise that leans more towards stronger solutions. Further, a very large solution space is expected due to the need for a combination of a number of techniques for each time interval over the length of an entire mission.

### 2.3.3 Method selection

In light of the above, it is clear that a nature-inspired approach is preferable for this problem. In particular a genetic algorithm was chosen. There are a number of alternative nature-inspired techniques such as: simulated annealing, ant colony optimisation, differential evolution, and particle swarm optimisation amongst others [18]. However, beyond the precedent set by previous systems that have favoured this approach, this problem in particular lends itself to a number of intuitive modifications of the evolutionary operators of genetic algorithms. For example, an operator could be implemented that examines unsuccessful strategies and identifies time intervals in which the platform was successfully hit. The countermeasures in the time intervals leading up to this could then be replaced with the necessary last-minute countermeasures in an attempt to fix the strategy, rendering it viable. Further, due to the level of independence of countermeasure allocations for time intervals, they can be easily interchanged between strategies in a modified crossover. For example if a particular strategy performs well in the first half of a mission, but poorly in the second half, it can be combined with another strategy with the opposite characteristics in order to potentially generate a stronger strategy.

## 2.4 GENETIC ALGORITHMS

As mentioned already, as a nature-inspired method, genetic algorithms are based on generating an initial population of solutions that are then systematically altered until a satisfactory solution is found. The population size is defined as $N_p$ and a single member of the population is a complete set of variable values providing a solution to the problem. Performance or fitness of a member of the population is determined by the objective function, which in this case would be a measure of various strategy metrics such as risk to platform, financial cost, and levels of EMCON.

In particular, this approach is based on biological evolution and aims to emulate Darwin's theory of natural selection, whereby a population is systematically improved using processes that mimic genetic operations [18]. The basic method starts with a randomly generated population that is individually analysed using a fitness function (the objective function). A random subset of the population is then chosen that is skewed towards the more fit designs, and used to generate a new population of designs using crossover and mutation. As a result, the population in each iteration has a higher probability of having designs of increased fitness [18]. This is then repeated until a strop criterion is met.

In genetic algorithms, the terminology referring to the optimisation process relates back to the equivalent biological terminology. Each iteration is referred to as a generation, a chromosome is a single complete design, and a gene is a value of a particular design variable [18]. Further, it is important to note that since this method relies on random processes, it is likely to produce different results each time it is performed, both in the optimal solution found and the time to obtain it [18].

The advantage of genetic algorithms is that they do not require any prior knowledge of the optimisation problem and that they find global minima rather than local ones [18]. However, there is no guarantee that the found solution is indeed the global solution, but this is not strictly an issue as only a useful solution is required in this application, not necessarily the true optimum. Also, genetic algorithms require a large number of calculations, especially for larger population sizes, but due to the complete independence of the fitness calculations of the different solutions, large gains in performance can be obtained using parallel processing [18].

The specifics of genetic algorithms such as mutation, crossover and selection are discussed in the following section along with other optimisation choices made regarding Pareto optimisation and multi-objective methods. For this particular problem, the fitness function is to be minimised, hence any reference to improved fitness, greater fitness etcetera indicates a lower fitness function value. A strong member is one with a low fitness value and vice versa.

### 2.4.1 Design schema

In order to implement a genetic algorithm, each chromosome and gene must be represented numerically. This is most commonly done using binary encoding, but can also be achieved using real number and integer encoding, and is known as the design schema [18]. Binary encoding is

achieved by representing each design value or gene using the minimum number of bits required for the number of possible values for its particular design variable. Each binary encoded gene is then concatenated into one large binary string representing the design point or chromosome.

### 2.4.2 Seeding

Seeding is the process of generating an initial population for the genetic algorithm and is usually done randomly so as to spread the designs as equally as possible across the design space [18]. This maximises genetic diversity in the population so as to ensure that the global optimum is found. The optimisation process can be speeded up greatly by seeding the initial population with known potential solutions or approximate solutions to the problem by giving the algorithm a good starting point [18]. However, if too large a portion of the population is occupied by these intentional seeds, genetic diversity may be compromised, causing the algorithm to focus on a local minimum.

### 2.4.3 Reproduction

Reproduction is the process of selection used to create the mating pool used for crossover and other operations [18]. Selection is always biased towards the fitter members of the population so that the average fitness of the population improves over the generations. A number of methods are discussed below. There are many other techniques, but these are the most commonly used [21]. Any of these selection schemes can also be modified in a number of different ways, such as only accepting designs with a fitness value greater than a certain threshold, with the aim to speed up convergence [22]. Selection of individuals from the mating pool for genetic operations can then either be done randomly from the pool, or a specific scheme can be used such as performing crossover between the strongest and weakest designs in the pool [22].

*1) Roulette wheel selection*

This method is based on the operation of a roulette wheel, where the area of the segments on the wheel allocated to each design is proportional to their fitness value [18]. As a result, when the wheel is spun it is more likely to stop on the more fit designs. This method works well for problems when a sufficient portion of the population have similar fitness values, in this case it will only slightly favour stronger designs and maintain genetic diversity. However, in problems where a single strong design dominates the population, such as in the case of deliberate seeding of the

initial population, this method will eliminate genetic diversity too rapidly, potentially focusing on a local minimum, resulting in premature convergence [4].

*2) Tournament selection*

This method involves the random selection of a pre-determined number of competing designs, where the one with the greatest fitness is selected [18]. In this method, the tournament size can be varied to set the desired level of selectivity in the selection process, where a larger tournament size favours stronger designs and reduces genetic diversity, but increases the rate of convergence towards a solution. The other extreme is a small tournament size of, for example, two designs, where selection approaches random selection (a tournament size of one would actually be random), favouring genetic diversity. Tournament selection is also one of the most popular selection schemes due to its straight forward implementation [23].

*3) Truncation selection*

This method requires the ordering of the population according to fitness, before a chosen percentage of the weakest individuals are dropped [23]. The remaining individuals are then duplicated in order to maintain the size of the population before evolutionary operators are used. This is one of the simplest selection schemes to implement [23].

*4) Linear ranking selection*

This selection scheme is aimed at eliminating the disadvantages associated with roulette-wheel selection. The individuals of the population are first ordered according to their fitness values, before being allocated a rank. The rank $N_p$ is assigned to the strongest individual, with the others assigned decreasing ranks down until the weakest individual which is given a rank of 1. These ranks are then used to calculate their probability of selection rather than their fitness as with Roulette wheel selection [21].

*5) Technique selection*

For this particular application, the preservation of life and the reduction of mission costs are prioritised over the rate at which a solution can be found. Therefore the generation of generally stronger solutions will be preferred over rapid solution generation that converges too quickly to a poorer solution.

A comparison of the common selection schemes discussed above was performed by Blickle and Thiele [21]. Despite the fact that the specific optimisation problem itself determines the extent to which various characteristics are advantageous, it was found that, in general, roulette wheel selection is a very unsuitable selection scheme. For the same selection intensity, the truncation method replaces more weak individuals than the remaining methods, resulting in poorer genetic diversity and greater chance of premature convergence. Truncation also leads to a lower selection variance (variance in the selected individuals).

Therefore, due to the relatively good performance and ease of implementation of tournament selection, it was chosen as the selection scheme for this implementation. Importantly, this scheme allows for a user-variable balance between exploration of the solution space and exploitation of strong solutions through the modification of the tournament size. This allows for easy and intuitive modification of the performance of the genetic algorithm.

### 2.4.4 Genetic operators

The basic genetic or evolutionary operators of crossover and mutation, along with the processes of seeding the initial population, and immigration are discussed in this section. The number of these operators performed in each generation, along with their other characteristics, must be adjusted in order to fine tune the performance of the algorithm. The actual number of each is generally determined by trial and error according to user experience [18].

*1) Crossover*

The technique of crossover is used to introduce relatively large variation into a population by combining or mixing two different parent chromosomes [18]. This is usually performed by selecting one or more cut points in the binary chromosome where the parent chromosomes are split. The segments of the parent chromosomes are then recombined with each other to produce children chromosomes. A common method is to choose a single split point, creating four segments from the parent chromosomes, and then swapping the second segment on the parent chromosomes to create two children [18].

*2) Mutation*

The technique of mutation is used to introduce slight variations around strong solutions in the design space by randomly altering a gene (design value) in a chromosome [18]. Mutation maintains diversity by introducing new genes to the chromosomes of the population. This is performed by selecting a random location on the binary chromosome of a member of the population and switching a zero for a one or vice versa.

*3) Elitism*

The technique of elitism is used to preserve the population leader in each generation and as a result, prevent the algorithm from ever going backwards in the optimisation process [18]. This process simply consists of copying the fittest chromosome to the next generation unaltered and keeping it immune from mutation. However, the chromosome remains in the original pool for selection for use in crossover and other genetic operations.

*4) Immigration*

The process of immigration is sometimes used to increase genetic diversity when progress is slow by introducing completely new chromosomes into the population [18]. This is usually performed when the fitness value of the population leader hasn't improved by more than a pre-determined value for a pre-determined number of generations. The number of immigrating chromosomes and the point at which they are introduced to the population can be varied in order alter algorithm performance. If immigration occurs too early or too late then it will slow the algorithm down. Further, if too many chromosomes immigrate into the population, then it will set back the algorithm too much, but if too few immigrate, then it will have minimal impact on increasing the genetic diversity of the population.

### 2.4.5 Population schemes

The population size used in a genetic algorithm has a significant effect on its performance. A small population size requires lower computational cost per generation, but can result in poor sampling of the solution space [24]. On the other hand, a larger population size has an increased ability to differentiate between strong and weak solutions to a problem [25], thus preventing the algorithm from getting trapped in a local minimum. Typically, genetic algorithms use a single, constant population size. However, it is possible to use a population scheme that varies the population size

over the generations of the algorithm in order to improve performance. This can be implemented using any scheme such as sinusoidal [26], or saw-tooth [27] changes. For example, for the saw-tooth scheme, the population size linearly decreases over the generations. This provides a larger population in the early stages of the algorithm in order to generate better initial solutions to the algorithm through superior sampling of the solution space. Thereafter, the smaller population during the later stages provides sufficient population size for the convergent population, whilst reducing the computation time and cost of each generation.

The major disadvantage of these more-advanced techniques is that a lot of tuning is required of the amplitude and period variables in order to exploit their performance advantages [27]. As a direct result of this, along with their increased implementation complexity, a constant population size will be used for this work. As with the number of genetic operators, this value must be determined by trial and error according to user experience.

### 2.4.6 Multi-objective optimisation

Multi-objective optimisation is where there is more than one objective function to be minimised or maximised simultaneously [18]. This can be achieved through many different techniques, of which a select few are discussed below.

The weighted sum technique is the most common approach to multi-objective optimisation due to the fact that it is both simple and intuitive [18]. In this approach, the objective functions are normalised, linearly scaled, and summed in order to produce a new, single objective (or fitness) function. The scalar values, or weights, are then chosen so as to set the relative importance of the original objective functions [18]. Importantly, this approach allows a user to modify system performance with relative ease by simply adjusting the weights. Further, all original objective functions are used to calculate the fitness of a solution, meaning that all aspects of a scenario would always be considered in the optimisation process. In fact, this approach can be considered is a special case of the weighted global criterion approach, where the utopia points for the objective functions are set to zero and the exponential component, $p$, is set to 1. In this approach, the differences between the current values of the objective functions and their utopia points (ideal values) are weighted and summed, where the exponent p is often included on the difference values in order to weight the importance of being close to the utopia point [18].

On the other hand, the bounded objective function, minimax, and the constraint optimisation approaches always use a single one of the original objective functions in order to calculate the fitness of a solution. The bounded objective function and constraint optimisation approaches both use a pre-selected main objective function, whilst the remaining functions are used to constrain the problem [18] [28]. In the case of minimax, this main objective function is selected as the one with the maximum fitness value for that particular calculation [28]. For these techniques, the issue is that not all of the objective function values are considered collectively, or at all throughout the optimisation process.

Therefore, due to the above discussion, and its generally strong performance, lower computational cost, simplicity, and generally intuitive nature, the weighted sum technique was chosen for this implementation.

### 2.4.7 Pareto Optimality

A Pareto optimal point is one where there is no other point in the design domain that reduces at least one objective function without increasing another [18]. Pareto optimality is a very useful tool in multi-objective optimisation as it allows a user to choose between optimal designs that prioritise different objective functions. The set of all Pareto optimal points is known as the Pareto optimal set. There are a number of techniques that can be used to obtain Pareto solutions for genetic algorithms and these are discussed below. A few of these techniques are variations on reproduction or multi-objective optimisation techniques.

The concept of dominated and non-dominated solutions must be understood to follow some of the techniques. Essentially a non-dominated solution, is like temporary Pareto solution in that there exists no other known solution (at the time) that reduces at least one objective function, without increasing another [18]. Dominated solutions are then the opposite of this.

*1) Vector-evaluated genetic algorithms*

This technique maintains $k$ individual sub-populations, where each is optimised using one of the $k$ objective functions [18]. This creates different species within the population, with each specialising in a specific characteristic. The major disadvantage of this technique is that the solutions it

generates tend to cluster around the minima of the individual objective functions, rather than a balanced solution.

*2) Pareto-set filter*

This technique uses two population sets, one of which is the population itself, and the other is known as the filter, which contains a tentative set of non-dominated solutions [18]. Each iteration, the solutions that are not dominated by any in the filter are moved to the filter population. The filter population is then examined and any dominated solutions are removed. Once optimisation is complete, the filter population will contain a set of Pareto solutions. Additionally, the filter population can be monitored and solutions removed that are too close to another solution in order to maintain an even distribution of solutions in the design space. Also, a set number of these filter designs can be re-introduced after crossover and mutation as part of elitism.

*3) Ranking*

The technique of ranking uses the rank of a population member to determine its fitness rather than a fitness function [18]. The members are ranked by finding the non-dominated members and giving them a rank of one. These members are then removed from the population and the next group of non-dominated solutions are found and given a rank of two. This process is then repeated until the entire population has been ranked. The highest ranked members can either be moved into a filter population or maintained through elitism in order to maintain a Pareto set.

*4) Tournament selection*

This method requires a random candidate solution to be selected and compared with another set number of solutions. If the solution is non-dominated by the tournament, it is then selected [18]. As with the previously discussed method of tournament selection, the size of the tournament determines the performance of the optimisation algorithm. A large tournament size causes premature convergence, whilst a too small tournament size will find too few Pareto points.

*5) Niche techniques*

A niche in terms of Pareto optimisation is a group of similar solutions. Most multi-objective genetic algorithm methods tend to converge to a limited number of niches, known as genetic or population drift [18]. Niche techniques are techniques that are aimed at developing multiple niches

in order to encourage equal spread in a set of Pareto solutions. One such technique is fitness sharing that divides the fitness of a solution by the number of solutions within a specified vicinity, essentially sharing the fitness of a niche between the designs within it [18].

*6) Alternate parameters*

The most straight-forward method of generating a number of Pareto designs is to rerun the optimisation procedure with different parameters each time [18]. In this case this would simply be achieved by using different weights in the weighted-sum fitness function for each run of the genetic algorithm. The advantage of this technique is that it is able to generate a specific number of solutions with pre-determined characteristics and objective function prioritisations. For example, it can be used to generate three different approaches to a scenario: a cost-effective approach that uses as few expendables as possible, a conservative approach that uses as many as required to conservatively ensure the safety of the platform, as well as a stealthy approach that emphasis EMCON. Further, this technique has the advantage of reduced computational complexity, especially when only a single solution needs to be rapidly generated. Lastly, processing speed can be further increased by using the previous Pareto solution to seed the next Pareto optimisation run. It is for these reasons that this approach was chosen for generating Pareto solutions in this system.

### 2.4.8 Stop Criteria

A very important aspect of optimisation is knowing when a sufficiently good solution has been found or when no more improvement can take place. If there are no stop criteria, then the optimisation process would continue indefinitely. Once again, there a number of different stop criteria that can be used. The ones discussed here have all been selected for implementation, with the idea being that the user can choose which one is used based their particular application.

*1) Maximum iterations*

The most straight-forward stop criterion is to limit the maximum number of iterations (generations of the genetic algorithm) before the optimisation procedure is terminated and the best solution found is chosen. This technique has two main applications. The first is when there is a known time limit in which a solution must be found. In this case, the approximate time required per iteration would be previously measured and hence known, and the number of iterations chosen accordingly. This process would then result in the best possible solution obtainable in the time available. An

example of this would be when there is a limited time before mission commencement and a jamming strategy must be found to maximise the probability of survival of the platform. The second application of this stop criterion would be as a back-up to other stop criteria in order to catch the optimisation procedure if no solution exists for the chosen scenario and stop criterion.

*2) Minimum improvement*

A second stop criterion is to monitor either the fitness of the strongest member in the population or the average population fitness in each generation. If this value has not improved by a certain threshold in a set number of generations, it is assumed that the optimal solution has been found and the algorithm terminated [18]. This stop criterion can be modified to first increase the number of mutations per iteration after a number of generations of minimal improvement in order to increase genetic diversity and hence ensure that the algorithm is not trapped in a local minimum. Then, if the minimum improvement requirement is met, the number of mutations can be reset. However, if there is still minimal improvement over another specified number of generations, then the algorithm is terminated and the population leader used as the optimal solution [18].

The advantage of this technique is that it will find an as close to optimal solution as possible in as short a time as possible. Although, the time period required is unknown and may be longer than available, resulting either in delaying a mission or causing a mission to commence before the optimal jamming strategy is known. Also, the solution found may not be the actual optimal value and could be improved with further iterations, which could be allowed in situations where time permits it.

*3) Safe passage*

A final stop criterion is to monitor a specific characteristic of the population leader over the generations until it improves beyond a specified performance barrier or limit. For this specific system, this would mean monitoring the number of times the platform is successfully engaged over the course of the mission until it reaches zero, indicating the first solution found that would result in the survival of the platform. The advantage of this technique is that uses the minimum possible time to find a satisfactory jamming strategy for the platform. This is extremely advantageous in a time-critical mission that is planned and executed on very short notice, where the platform would wait until a jamming strategy is found before commencing the mission.

## 2.5 CURRENT APPROACHES

Currently, there are many threat evaluation and weapon assignment (TEWA) systems that allocate weapon systems to adversary platforms according to the threat level they pose (e.g. [1], [2]). These systems represent a similar problem to the one at hand in terms of the processes of threat evaluation and allocation of resources, but do not take into account the specific characteristics associated with actions in the electromagnetic spectrum.

To account for these differences, a few jamming allocation systems have been proposed (e.g. [3] - [7]). Most of these systems use threat levels for prioritisation of threats in conjunction with either a jamming factor, or a probability of jamming success to determine the optimal allocation of jamming resources.

For example a cooperative jamming resource allocation system has been proposed [4]. It is aimed at the optimisation of the cooperative jamming of multiple threats using multiple platforms dispatched from a number of bases in order to protect a number of specific targets. In this system, threats are first identified according to their radar characteristics and this information used to determine which protected target they are likely to engage. Each threat is then allocated a threat value according to the distance between the threat and the protected target, as well as the importance factor of this target. The effect of jamming allocation, or jamming contribution, is then calculated as the product of threat value of the threat, and the jamming effectiveness of a potential jamming resource on that threat, where the latter factor is obtained from a pre-determined lookup table. The objective function for optimisation is the weighted sum of the total jamming contribution and the total distance travelled by the jamming platforms in the strategy, where the jamming contribution must be maximised, and the distance travelled minimised. The final result of optimisation is simply the allocation of a single threat to each jamming platform.

A further example, and the most developed and similar system to the one proposed in this document, is one which is aimed at the entire process of real-time detection and classification of threats before selecting and executing countermeasures against them [3] [7]. In this system detection and classification is achieved using soft computing techniques, such as naive Bayesian classifiers, inductive decision tree algorithms, and neural networks. Threats are classified as either terminal or non-terminal according to their radar characteristics such as frequency of operation,

pulse width, pulse power, and pulse repetition frequency. Thereafter countermeasure allocation makes use of decision-theoretic architecture in order to maximise the expected utility of countermeasures. There are four available countermeasures according to the receiver which detected the threat: flares, chaff, RF jamming and IR jamming. The expected utility is calculated as the product of the probability of success of a countermeasure and its utility, where these values are calculated offline according to factors such as jamming to signal ratio and the distance between the radar and the aircraft, and stored in lookup tables.

However, these systems only allocate jamming resources as a whole to threats without determining the specific optimal jamming techniques to be used. Secondly, specific interactions inherent to the EMS are not taken into account such as the constructive and destructive interactions between different jamming techniques and their signals. This is one of the major differences to TEWA systems, in that jamming strategies will work effectively for some threats, but illuminate the platform for others depending on the frequencies and bandwidths used as well as on threat radar modes. Existing jamming allocation systems also do not take into account the future effects of current jamming actions which result from the progression of the radar modes of the threats from search to guidance, or the effect of threat counter-countermeasures such as flare rejection capabilities. Further, the inherent uncertainty involved in the threat environment is also not taken into account by these systems. Finally, and perhaps most significantly, none of these currently existing systems are aimed at, or capable of, the optimisation of the passive countermeasure cartridge load-out of a platform prior to mission commencement.

As such, the proposed system will have to account for all of the factors listed above, including the effects of different jamming techniques on each threat, interactions between jammers, radar modes and their progression, the effects of different operating frequencies, and threat uncertainties. Also, a number of user-definable parameters would allow a wide range of systems to be modelled, and a user-weighted objective function will allow for different mission strategy characteristics to be prioritised.

## 2.6 FUTURE APPROACHES

A future solution to the problem lies in the development of cognitive EW. These systems are based on the concept of autonomous adaptability through cognition. Essentially these systems are

intended to be able to identify both known and unknown targets, determine their weaknesses and to jam and suppress them with optimal, adaptive algorithms [29]. This is as opposed to current systems that rely on predetermined libraries of threats and associated countermeasures, where new threats have to be recorded and analysed post-engagement to determine a countermeasure. An example of such as system is the project known as Adaptive Radar Countermeasures (ARC) currently being developed by the United States' Defence Advanced Research Projects Agency (DARPA) [29].

However, these systems are being developed for implementation in next generation platforms with distributed sensing. This leaves a gap not only for present systems, but for military forces that, due to financial reasons, will be using current or older generation systems for many years to come.

## 2.7 CHAPTER SUMMARY

In this chapter an overview of the relevant areas of EW and optimisation were presented. It was found that a number of generic weapon systems should be modelled including: explosive and non-explosive artillery, as well as command, bream-riding, active, and semi-active missiles. Additionally, it was found that the radars associated with these threats can be broken down into an ordered sequence of stages: search, acquisition, tracking, and guidance. Thereafter, the EW capabilities of the platform were selected to be representative of modern systems and as such 5 active countermeasure techniques, and 3 passive techniques were selected for modelling along with a fibre-optic towed decoy: range gate pull off, velocity gate pull off, noise jamming, cover pulses, multiple false targets, as well as flares, distraction chaff, and dilution chaff. Lastly, it was determined that the platform should contain a missile approach warner, or a missile launch warner.

Next, examination of similar systems indicated that a genetic algorithm was the best optimisation algorithm for the task, and the specifics of the implementation of these algorithms were explored. In particular, a binary encoding design schema, combined procedural and random seeding, tournament selection, elitism, and immigration were all selected. Further, a simple weighted sum technique was selected for multi-objective optimisation, along with the alternate-parameters Pareto optimisation approach. Lastly, a number of different stop criteria were considered and all selected for implementation due to their potential use in different situations: maximum iterations, minimum improvement of the population, and safe passage of the platform.

Lastly, both the current and future approaches to this problem were discussed. It was found that the current approaches are either too simple and do not take into account a number of important factors, or are to complex and slow to be of any real use in optimisation. On the other hand, it was found that there are some potential alternative approaches that are currently under development. These include cognitive EW systems that are able to, in real-time, identify both known and unknown targets and jam them with adaptive algorithms. However, such systems are being developed for implementation in next-generation platforms, hence leaving a gap for militaries that will continue to use current or older generation systems for years to come.

# CHAPTER 3 METHOD

## 3.1 CHAPTER OVERVIEW

In this chapter the method used to solve the countermeasure and load-out optimisation problem is discussed. It begins with an overview of the threat evaluation and countermeasure allocation approach in Section 3.2. This includes all clarifications and assumptions, how scenario information is calculated, how radar and IR stage progression is handled, along with break lock and dead time. Next, the calculation of the danger value used for the purpose of threat evaluation is discussed in Section 3.3. This includes a number of factors: Platform RCS, probability of threat encounter, threat radar stages, threat accuracy, projectile time to platform and time to next radar stage, along with how these apply to IR threats. Thereafter the jamming factor used to account for the effect of countermeasure allocation is discussed in Section 3.4. This is split into separate discussions of how this factor is calculated for active channels, chaff, and flares, before finally discussing the effect of the towed decoy on this factor. Lastly, the optimisation approach is discussed in Section 3.5. This begins with an overview of the optimisation procedure as whole before discussing the variable values used, the design schema, the calculation of strategy fitness, seeding approaches, crossover, mutation, repair operators, immigration, stop criteria, and Pareto optimisation.

## 3.2 THREAT EVALUATION AND COUNTERMEASURE ALLOCATION

The approach used to solve this problem is known as threat evaluation and countermeasure allocation (TECA). This is due to the fact that it can be broken down into two major processes: a threat evaluation stage for the prioritisation of threats according to their characteristics, and a countermeasure allocation stage that takes the effect of countermeasures into account and uses this information to optimally allocate them. As such, this section consists of an initial overview to explain the overall TECA process before moving on to individual explanations of threat evaluation, countermeasure allocation, and finishing on a discussion of the optimisation procedure used. Note

that in many cases, concepts are introduced as they become relevant, and then are expanded on in detail in later sections.

### 3.2.1 Overview

Since the intention of this work is to not only optimise countermeasure allocation, but also to optimise cartridge load-out, it must be assumed that the software is executed prior to mission commencement using available intelligence information, including all potential threats and their characteristics. Further, the system is designed to operate using generic platforms and systems with a lot configurability built in. This is due to the sheer number of platform types in service, and the number of systems each individual platform can contain. This way the TECA system can be modified according to the specific platforms in a scenario, where all necessary values, and lookup tables would be populated using powerful, but slow, physics- or parameter-based simulation environments. Note that all parameters used in this work were heuristically determined and based on the general characteristics of the systems considered. They are thus not representative of specific systems due to the classified nature of such information.

The overall scenario considered is that of a single generic platform entering adversary territory. Specifically, the platform is assumed to be equipped with a self-protection system consisting of two active channels, a single passive channel, and a towed decoy through which either of the two active channels can be directed. This is due to that fact that, as discussed in the literature study, this is representative of operational systems such as the Gripen [17], and the Eurofighter Typhoon [16].

Each individual scenario is set up using mission waypoints in a three-dimensional Cartesian coordinate system, where each waypoint consists of the location and roll of the platform at a point in time measured in seconds from the commencement of the scenario. It is assumed that a direct path is traversed between these waypoints. Therefore, the speed, pitch and yaw of the platform are assumed to be constant between waypoints, and hence are determined by the difference in their coordinates. On the other hand, the roll of the platform is linearly incremented over the time intervals between waypoints so as to achieve the desired amount roll at each one with a smooth transition between. Roll is measured in degrees from the horizontal (x-y) plane and is defined such that a positive angle indicates the necessary roll for a right-hand turn. The locations of the threats are also programmed using the same Cartesian coordinate system. Further, each threat in a scenario is identified by two numbers: a threat type number, and a threat identification (ID) number. The

threat type number indicates what weapon system that threat is, and hence allocates the characteristics associated with that system to that threat. These characteristics include: weapon system type, accuracy, weapon range, radar range, projectile velocity, countermeasure resistance, and radar stage progression rate. On the other hand the threat-specific information such as its location, probability of encounter, and radius of likely encounter are allocated to a specific threat through its ID number. Further, it is assumed that the threat-type numbers are allocated in ascending frequency-band-usage order, where additionally, the threats can also be allocated into specific frequency bands. As such, the frequency band of a threat type 4 will be adjacent to those of types 3 and 5, where the amount of overlap is determined by a cross-effect factor which is discussed later. In this work, the frequency bands of operation have been limited to the L, S, C and X bands.

The scenario is then divided into a number of individual time intervals of uniform, user-defined length, where this length can be used to vary the trade-off achieved between accuracy and computational efficiency. Each interval is then handled individually, where each channel of the self-protection system is allocated a countermeasure technique, threat type for which it is optimised, and a direction in which the antenna is steered in the case of active channels. Further, the towed decoy can also be allocated to any of the active channels. Finally, the mission is optimised as a whole using a genetic algorithm according to a number of criteria. This overall process is depicted in the pseudo code in Figure 3.1, where all the individual steps are discussed in detail in later sections.

The process begins by initialising all variables. These variables include a number of user-defined lookup tables, weights, threat and platform characteristics, mission waypoints, and genetic algorithm settings that are used to setup the scenario and modify algorithm performance. Thereafter, this information is used to calculate necessary scenario information: the location and orientation of the platform at each time interval, and the associated distances and angles to each of the threats in the scenario. Since the path and manoeuvres of the platform are fixed for the scenario, and threat movements are accounted for using distributed areas of likely encounter, this calculated scenario information remains constant across all population members. Therefore, this reduces the calculations necessary within the nested for loops, and prevents the unnecessary re-calculation of values, thus reducing computational complexity.

```
1:  procedure COUNTERMEASURE OPTIMISATION
2:      INITIALISE VARIABLES
3:      CALCULATE(scenario information)
4:      GENERATE INITIAL POPULATION
5:      CALCULATE POPULATION FITNESS
6:      FIND(population leaders)
7:      for all pareto solutions do
8:          if pareto solution > 1 do
9:              SET NEW OBJECTIVE FUNCTION WEIGHTS
10:             REGENERATE POPULATION
11:             CALCULATE POPULATION FITNESS
12:             FIND(population leaders)
13:         end if
14:         for all generations do
15:             GENETIC OPERATORS
16:             CALCULATE POPULATION FITNESS
17:             POST-FITNESS GENETIC OPERATORS
18:             FIND(population leaders)
19:             if stop criterion = true do
20:                 BREAK
21:             end if
22:         end for
23:         CLEAN-UP SOLUTION
24:         SAVE SOLUTION
25:     end for
26:     return best countermeasure strategies and results
27: end procedure
```

**Figure 3.1.** Overall system pseudo code.

Following this, an initial population is generated for the genetic algorithm, and its fitness calculated. This consists of a number of both randomly generated solutions, and procedurally generated ones in order to speed up convergence. Next, a number of different Pareto solutions can be generated with different sets of user-defined weights that prioritise different strategy characteristics, where an optimised strategy is generated for each. For each of the requested strategies, the algorithm loops through a number of generations that each consist of a number of genetic operators, a calculation of the fitness of each solution in the population, and a test of the population against the selected stop criteria. If these criteria are met, then the optimisation loop is exited, the solution cleaned of trivial non-idealities, and saved. It is noted that there are a number of post-fitness genetic operators performed after the main population fitness calculation. These take advantage of the information obtained during the fitness calculation to more intelligently modify the population, and hence must be performed thereafter. The fitness of the modified members is then immediately recalculated before they are placed back into the population.

Further note that in the Pareto loop, the initial population is used to generate the first solution. Thereafter, for each new Pareto solution a portion of the population retained, and the rest is regenerated in order increase genetic diversity. Also, for each of these new solutions, the objective function weights must be changed, the fitness of the new population calculated, and the population leader found.

The calculation of the fitness of each population member is where the process of threat evaluation occurs along with where the effects of countermeasures are taken into account, as depicted in Figure 3.2. As previously stated, each individual time interval is handled separately. First, the scenario information must be updated for the time interval being examined, including the platform locations, threat distances and angles. The threats' radar stages are then updated according to the effects of the previous time interval, and their danger values calculated for the purposes of prioritisation. A danger value is a value that is representative of the level of danger a threat presents to the platform in that time interval according to the various characteristics of the threat. Thereafter, a jamming factor is calculated for each threat according to the countermeasure strategy of the population member being examined. This is a multiplicative factor that accounts for the effect of jamming on the danger value of a threat, and as such the next stage is to calculate the post-jamming danger value of the threats. This forms a major part of the countermeasure allocation process as the minimisation of this post-jamming danger value minimises the risk to the platform. Lastly, the rate of radar stage progression for that time interval is calculated according to the effects of the implemented countermeasures, and it is checked whether the lock of threats has been broken.

```
1:  procedure CALCULATE POPULATION FITNESS
2:      for all population members do
3:          for all time intervals do
4:              UPDATE(scenario information)
5:              UPDATE(threat stages for all threats)
6:              CALCULATE(danger values for all threats)
7:              CALCULATE(jamming factors for all threats)
8:              CALCULATE(post-jamming danger values for all threats)
9:              CALCULATE(radar stage progression for all threats)
10:             CHECK(break lock for all threats)
11:         end for
12:         CALCULATE(strategy fitness)
13:     end for
14: end procedure
```

**Figure 3.2.** Pseudo code for the calculation of population fitness.

Finally, after the above has been completed for each individual time interval of a scenario, the fitness of the population member and its scenario-long EW strategy can be calculated. This is achieved by summing the characteristics of the strategy over the mission: the post-jamming danger value, the number of passive cartridges used, and the number of active jamming techniques used. These are then combined in an objective function, using user-defined weights, in order to determine an overall fitness value of the strategy that must be minimised.

### 3.2.2 Clarifications and assumptions

As stated already, there are an exceedingly large number of platform types in service, each with a large number of systems that they can be equipped with. Further, due to the lack of published literature in the field, this system had to essentially be developed from scratch. This is without even considering the necessary focus on computational complexity of this work. As such, the problem must be reduced using a number of assumptions and simplifications in order keep it bounded, reasonable, and importantly, solvable. The idea is that functionalities can slowly be added to the system over time in order to improve the accuracy of the model, and improve performance. In fact, this work began as a single-time-interval active jamming allocation system known as TEJA. Passive countermeasures and towed decoys were then added along with the ability to optimise an entire mission strategy as a whole using a genetic algorithm. Thereafter, the system was modified to include the effects of the RCS of the platform and the antenna gain patterns of its ECM system, before modifications were made to the algorithm to improve performance with small time intervals. As such, a number of general simplifying assumptions made in this work are detailed below. The remaining assumptions are discussed in more relevant locations throughout the text.

Firstly, it has been assumed that each active channel of the platform's ECM system is capable of the same five major classes of jamming techniques: range gate pull off (RGPO), velocity gate pull off (VGPO), noise jamming (NJ), cover pulses (CP), and multiple false targets (MFT). It is further assumed that the passive channel can fire both flare and chaff cartridges, where the chaff can be dispensed in either a distraction (DIS) or dilution (DIL) manner. Finally, it is also assumed that either of the active channels can be directed through the platform's towed decoy, and that the required platform manoeuvres are included in these techniques.

Further, it is assumed that all flare and chaff cartridges are of an equal standard size [15] and that the only difference between the optimised strategies for each threat is simply the number of

cartridges used. As such, all differences must be accounted for in this user-defined number. Specifically, this number is defined as the number of cartridges required per time interval while the technique is implemented. Therefore, the number of each specific cartridge type required for a mission can be determined by examining both the threats being countered and the number times they are countered in the developed countermeasure strategy. It is also noted that maximum cartridge capacity of the platform is user defined according to the number of these general, equal-sized cartridges that can be loaded into the platform's ECM system.

Due to the need for computational efficiency in this system and the resultant high-level approach, the total effects of passive countermeasures are looked at as a whole, rather than detailed effects over time. It has also been assumed that all radar characteristics are captured in the user-defined parameters and lookup tables, rather than attempting to model these explicitly. Therefore, it is assumed that these overall effects and resulting lookup tables are determined and populated using more accurate, slow, highly detailed physics- or parameter-based simulators such as SADM [8] or OSSIM [30]. Further, as a result of this high-level approach, a number of effects such as atmospheric attenuation have not been considered, and the platform's radar warning receiver (RWR) and MAW are assumed to be perfect.

It is further assumed that the radar modes of RF threats can be divided into an ordered sequence of stages: search, acquisition, tracking, and guidance, where artillery threats would skip the final stage due to their inability to guide their projectiles. Thereafter, these stages can be further divided into two overall categories according to their general action and the approach to jamming them: search-type and tracking-type stages. As such, the search and acquisition stages are grouped into the first category, and the tracking and guidance stages are grouped into the second. On the other hand, due to their passive nature, IR threats remain undetected by the platform until they are fired and detected by the platform's missile approach warner (MAW) or similar system. Therefore it must be assumed that the platform is fitted with such a system, like the Eurofighter [16]. As a result of this, IR threats are assumed to progress from an undetected stage straight to the guidance stage when fired. Lastly, it is also noted that it is assumed that all threats can only fire one missile or projectile at a time, and that there is no communication amongst the threats.

Due to the difference in ranges between the weapon and radar systems of a threat, a separate one is defined for each by the user. As such, a threat is restricted to the search stage while the platform is

outside of its radar-system range, and also allocated a danger value of zero. Thereafter, once the platform enters this range, the platform will be able to detect the platform and progress to the end of the tracking stage, where it must wait until the platform enters its weapon-system range. It can then fire a projectile at the platform and enter the guidance stage if it capable of doing so. Finally, guided missiles must reach the platform before it exits the weapon-system range in order to hit it. If this is not achieved, then it is assumed that the platform will successfully outrun the missile. Note that since stages other than guidance are not considered for IR threats, these types of threats are only allocated a weapon-system range.

Finally, note that in all equations the superscript $^{k}$ is used to denote the $k^{\text{th}}$ time interval of a mission whilst subscripts $_{m}$ and $_{n}$ indicate the $m^{\text{th}}$ IR and the $n^{\text{th}}$ RF threats respectively. Further, all angles are measured in degrees, and all distances measured in kilometres unless otherwise stated.

### 3.2.3 Scenario information calculation

As stated previously, all scenario information about each threat and their relative position to the platform is calculated and stored prior to the commencement of the optimisation procedure. This is possible due to the fact that the platform's route and the threats' positions are fixed for a scenario. Further, it is preferable due to the fact that this pre-calculation approach is more computationally efficient.

Before getting into the calculations for this information, some definitions need to be made. First, the heading of the platform must be defined. A heading of zero is defined, according to standard Cartesian definitions, as travelling along the direction of the positive x axis. This can also be seen as the platform traversing to the right in all depictions of the threat area in this work, or simply as an Easterly direction. A positive heading is then chosen to direct the platform toward the positive y axis, or in a more Northerly direction towards the top of the page. Similarly, azimuth angle ($\varphi$) is defined relative to the axis of the platform such that an angle of zero indicates a direction directly to the front of the platform, with positive angles then measured counter-clockwise to the platform's left-hand side. Next, the pitch of the platform must be defined. A pitch of zero is defined as the platform travelling parallel to the x-y plane, which can also be seen as the ground plane. A positive pitch is then chosen to direct the front of the platform towards the positive z axis, which can be seen as the pitching required for a platform to ascend. Similarly, elevation angle ($\theta$) is defined

relative to the axis of the platform such that an angle of zero indicates a direction pointing directly to the horizon, with positive angles then measured upwards. Finally, a zero roll angle is defined such that an aircraft's wings would be parallel to the x-y ground plane. A positive roll angle is then chosen to rotate the platform in a clockwise direction from the perspective of the pilot, or simply seen as the banking necessary for an aircraft to turn right.

The first information that must be calculated is simply the platform location in each time interval. This is achieved by assuming that the platform moves with a constant air speed and trajectory between waypoints, and calculating the platform location accordingly. The process consists of determining the two waypoints between which the platform is busy traversing, finding the difference between each of their stipulated coordinates, and dividing these by the number of time intervals between their waypoint times. The platform's coordinates at a particular time interval can then simply be determined by incrementing each prior coordinate accordingly.

Thereafter, for each time interval, both the ground distance ($d_g$) and slant range ($d_a$) are calculated between the platform and each threat as

$$d_g = \sqrt{\Delta x^2 + \Delta y^2}, \tag{3.1}$$

and

$$d_a = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}, \tag{3.2}$$

where $\Delta x$, $\Delta y$, and $\Delta z$ are the difference between the platform and the threat's $x$, $y$, and $z$ coordinates respectively. Note that for the following calculations it is necessary to define these $\Delta$ values specifically as the coordinates of the threat minus those of the platform.

Next, it is necessary to determine the position of the threats relative to the platform's orientation. This allows for the determination of the azimuth and elevation angles at which these threats perceive the platform for the calculations that require the platform RCS or antenna gain patterns. As such, the platform's orientation must be determined in terms of roll ($\gamma$), pitch ($\beta$) and heading ($\alpha$) angles for the time interval being examined. Since it has been assumed that the platform's roll linearly changes between the values stipulated at each waypoint, its roll at a particular time interval is determined using the same technique its position. On the other hand, due to the assumption of constant trajectory between waypoints, both the pitch and the heading of the platform are also constant between waypoints. The pitch can thus be calculated as

$$\beta = \sin^{-1}\left(\frac{\Delta z_w}{d_{a,w}}\right), \tag{3.3}$$

and the heading as

$$\alpha = \begin{cases} \sin^{-1}\left(\frac{\Delta y_w}{d_{g,w}}\right), & \text{if } \Delta x_w \geq 0 \\ 180 - \sin^{-1}\left(\frac{\Delta y_w}{d_{g,w}}\right), & \text{if } \Delta x_w < 0, \end{cases} \tag{3.4}$$

where $\Delta x_w$, $\Delta y_w$, and $\Delta z_w$ are the differences between the coordinates of the two waypoints currently being traversed, and $d_{a,w}$ and $d_{g,w}$ are the corresponding slant range and ground distance. Specifically, these $\Delta$ values are defined as the coordinates of the waypoint being travelled to minus those of the waypoint being travelled from. Note that the results of these equations are undefined if the slant range or ground distance are zero. As such, the pitch is set to zero, and the heading set to that of the previous time interval in the case that the slant range is zero. However, in the case that only the ground distance is zero, the heading is set to that of the previous time interval.

Thereafter, the coordinate vectors of each threat relative to the platform's axis ($\omega'$) can be determined by rotating their original vectors ($\omega$) around the z axis using the heading angle, around the y axis using the pitch angle, and around the x axis using the roll angle. This is achieved using specifically ordered rotation matrices [31] where

$$\omega' = R_x(\gamma) \cdot R_y(-\beta) \cdot R_z(\propto) \cdot \omega, \tag{3.5}$$

where

$$\omega = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}, \tag{3.6}$$

$$R_z(\propto) = \begin{bmatrix} \cos(\propto) & \sin(\propto) & 0 \\ -\sin(\propto) & \cos(\propto) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.7}$$

$$R_y(-\beta) = \begin{bmatrix} \cos(-\beta) & 0 & -\sin(-\beta) \\ 0 & 1 & 0 \\ \sin(-\beta) & 0 & \cos(-\beta) \end{bmatrix}, \tag{3.8}$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & \sin(\gamma) \\ 0 & -\sin(\gamma) & \cos(\gamma) \end{bmatrix}. \tag{3.9}$$

Then using these new coordinates, the new relative 'ground distance' ($d_g'$) can then be calculated as in (3.1). The same can be done for the relative slant range, but it remains unchanged in value. Finally, the elevation angle can then be calculated as

$$\theta = \sin^{-1}\left(\frac{\Delta z'}{d_a}\right), \tag{3.10}$$

and the azimuth angle as

$$\varphi = \begin{cases} \sin^{-1}\left(\frac{\Delta y'}{d_g'}\right), & \text{if } \Delta x' \geq 0 \\ 180 - \sin^{-1}\left(\frac{\Delta y'}{d_g'}\right), & \text{if } \Delta x' < 0. \end{cases} \tag{3.11}$$

Note that the results of these equations are undefined if the slant range or relative ground distance are zero. As such, both the elevation and the azimuth angles are set to zero in the case that the slant range is zero. However, in the case that only the relative ground distance is zero, only the azimuth angle is set to zero degrees.

Lastly, threat distance must be separately considered for the case of threats in the guidance stage. Specifically, the distance between the platform and the missile must be calculated and used, rather than the distance to the threat itself. However, these distances are strategy-dependant, and as such must be calculated on a case-by-case basis during the optimisation procedure. First, the time interval in which a missile is fired must be recorded. Thereafter, the distance calculation can be performed by making the simplifying assumption, in each time interval, that the threat fired directly at the current location of the platform. The missile distance is then simply the previously-calculated slant range of the threat minus the distance the missile would have travelled since the time interval in which it was fired, where the projectile velocity of each threat is known. A hit then occurs when this missile distance is reduced to zero. Importantly, this approach is computationally efficient as complex trajectories do not need to be considered. Further, the generated values are conservative in that the approach ensures both the shortest possible total guidance time, and the shortest possible missile distances in each individual time interval. This in turn generates more conservative strategies, hence ensuring greater safety for the platform.

### 3.2.4 Radar stage progression

As part of the modelling process, threats are required to progress though the above-mentioned radar stages. Further, the rate of this progression must take into account the effects of implemented

countermeasures, such as platform illumination which occurs when the platform's countermeasures increase its visibility to threat radars. This includes effects such as chaff temporarily increasing the RCS of the platform, generated false targets in the vicinity of the platform increasing the signal-to-noise ratio (SNR) perceived by threat radars, and noise jamming activating track-on-jam counter-countermeasures. As such, in the case of platform illumination this radar-stage-progression rate should be increased in order to account for the greater SNR perceived by a threat's radar, and the resultant increased probability of detection and more accurate tracking. On the other hand, effective jamming should reduce this rate so as to account for the various countermeasure effects such as decreased SNR, induced angular errors, and over-loaded or confused target detection systems. As such, the radar stage progression rate must be dependent on the jamming effect ($E$) values that are used to take the effects of countermeasures into account.

At its core, radar stage progression is achieved by using user-defined search, acquisition, and tracking times that indicate the average time taken for a threat to progress to the next stage. Each threat has a time-to-next-stage counter ($X$) that is decremented by the time resolution ($Q$) in every time interval. When this counter is reduced to zero, the threat progresses to the next radar stage and the counter is set to the associated user-defined average time.

However, the above approach does not take the effect of countermeasures into account. Instead, the average stage progression rate should either be slowed down or sped up by effective countermeasures or platform illumination, where the degree of which is determined by the jamming effect ($E$) due to its relation to both the JSR and SNR. Further, the RCS of the platform as perceived by the threat should be taken into account in this rate for the same reasons, where a large perceived RCS results in a large SNR and aids detection and tracking of the platform, and vice versa. As such, the time-to-next-stage counter ($X$) for the $n^{\text{th}}$ threat in the $k^{\text{th}}$ time interval is calculated as

$$X_n^k = X_n^{k-1} - \left(B_n^k \cdot Q\right), \tag{3.12}$$

where

$$B_n^k = \begin{cases} 1 - (H_s \cdot E_n^k \cdot L_n^k), & \text{if } \left|E_n^k\right| \geq Z_s \\ 1, & \text{if } \left|E_n^k\right| < Z_s, \end{cases} \tag{3.13}$$

$$L_n^k = \begin{cases} 1 - \mathrm{RC_b}\left(\varphi_n^k, \theta_n^k\right), & \text{if } E_n^k \geq 0 \\ \mathrm{RC_b}\left(\varphi_n^k, \theta_n^k\right), & \text{if } E_n^k < 0, \end{cases} \tag{3.14}$$

In these equations $B$ is the radar-stage-progression-rate modifier. $H_s$ is the user-defined weight for setting the amount by which the radar stage progression rate can be modified for the $s^{th}$ radar stage. Importantly, when set to a value greater than one, this factor allows for a negative radar-stage-progression-rate modifier for large effective jamming effect values. This allows the platform to essentially reverse radar stage progression within a radar stage. Without this capability, the time-to-next-stage counter can only ever decrease, and as a result any threat in a search-type stage would almost be guaranteed to detect the platform unless it was sufficiently jammed the near-entire time the platform was within its radar range. Therefore, this crucially allows the platform to switch its jamming between a number of threats in order to simultaneously hold them at bay and prevent its detection. As such, it is set to 1.5 in this work for all radar stages to allow for the radar-stage-progression rate to vary between -0.5, and 2.5 times the average. Next, $L$ accounts for the effect of RCS on the radar stage progression rate. $\mathrm{RC_b}(\varphi,\theta)$ is the normalised RCS factor of the platform when encountered from at an azimuth angle of $\varphi$ and an elevation angle of $\theta$ in the $b^{th}$ frequency band, where a separate set of RCS values can be defined for each of the major frequency bands: L, S, C, and X bands. As such, $L$ essentially modifies the normalised RCS values so that a large perceived RCS reduces, and a small perceived RCS enhances effective jamming performance, and vice versa, due to its effect on the jamming-to-signal ratio (JSR). Lastly, $Z_s$ is a user-defined jamming-effect threshold below which it is assumed that the effect caused by countermeasures is negligible on the average radar-stage-progression rate. This accounts for circumstances where the effect of countermeasures is so small that it disappears below the noise floor of the threat's radar, or is counteracted by processing techniques such as Doppler processing. This value is currently set to 0.1 in this work.

Therefore in these equations the radar stage progression rate is modified from the standard time-resolution rate using the factor $B$ that is dependent on the effects of the platform's countermeasures. A positive jamming effect value, which indicates effective jamming, results in a less than one $B$ factor, that in turns slows down the rate of stage progression, and vice versa. Importantly, since the RCS is accounted for in the danger value of the threat, but not the jamming effect, it is also included in this factor. Note that the specifics about the jamming effect and the implementation of RCS are discussed in detail in their respective sections below.

Finally, the above approach is applied to all radar stages, except that of guidance. This is due to the fact that the time taken by a missile to hit a platform is determined by physical characteristics such

as the distance that it must cover and the rate at which it covers that distance, rather than an average progression rate. As such, stage progression for guidance is instead handled by considering the distance of the missile to the platform. As discussed in the previous section, this distance is calculated on a case-by-case basis, and a hit assumed to have occurred when this distance is reduced to zero. However, a time-to-next-stage value is still necessary for the purposes of danger value calculations, and this is simply calculated using the missile distance and the known projectile velocity of that threat.

### 3.2.5 IR stage progression

As stated previously, IR threats only have two stages: undetected and guidance. As such, stage progression is much simpler for these threats. Since they are undetected by the platform until they are fired, they are not jammed in the undetected phase, resulting in no need to account for different stage progression rates. Further, as with RF threats, the guidance stage rate is dependent on the physical characteristics at the time of firing, rather than the countermeasures implemented against it, resulting in no need to stray from the time-resolution progression rate. Therefore, IR threats' time-to-next-stage counters ($X$) are simply decremented by the time resolution in each time interval.

The only other difference for IR threats is to account for the fact that such systems are often man-portable shoulder-mounted systems (MANPADS) [15], rather than automated systems with consistent stage progression rates. As such, IR threats are assumed to fire upon the platform at the first opportunity that it is within range, and thereafter have to reload and take aim before firing again. Therefore, each IR threat type is allocated a reload time that is representative of the average time required for this to occur. The time-to-next-stage counter is then set to this value after each missile is fired and decremented in each time interval.

### 3.2.6 Break lock

The process of radar-stage progression outlined above works well for the high-level modelling of the entirety of the search-type stages, in that threats are able to detect the platform faster or slower depending on the countermeasures implemented by the platform. However, this does not account for the ability of certain countermeasures to break the lock of the tracking-type stages [15]. This is achieved by using both a general user-defined break-lock threshold, and a user-defined average jam

time. The threshold sets the required minimum level of radar performance reduction in order to break lock, whilst the break-lock time sets the length of time required for a countermeasure to be applied in order to be effective and break lock.

Therefore, at the end of every time interval it must be tested whether the implemented countermeasures generate a sufficient jamming factor in order to break the lock of threats in tracking-type stages. This is tested for the $n^{\text{th}}$ threat in the $k^{\text{th}}$ time interval using

$$\mathrm{RC_b}\left(\varphi_n^k, \theta_n^k\right) \cdot F_n^k < \mu_{RF}, \quad \text{if } F_n^k \leq 1, \tag{3.15}$$

where $\mu_{RF}$ is the user-defined break-lock threshold for RF threats, RC is the normalised RCS factor for the platform as perceived by the threat, and $F$ is the jamming factor. Note that the test can only be passed if the jamming factor has a value of less than one. This prevents a small RCS value from overcoming platform illumination ($F > 1$), which is not possible due to the fact that platform illumination occurs independently from the reflected signal returning from the platform. As with radar-stage progression, the effect of the platform's RCS must be accounted for here as it is accounted for in a threat's danger value, rather than in the jamming factor itself. The break-lock threshold is currently set to 0.2, thus requiring a threat's radar performance to be reduced by at least 80 percent for lock to be broken.

Thereafter, if the threshold is surpassed, the countermeasures used are examined and compared to the previous time interval. If the current break lock attempt is determined to be a new attempt due to the fact that no break lock attempt was made in the previous time interval, then the break-lock timer is reset. Further, if the countermeasures used to break lock have changed between this attempt and the last, then the break lock timer is also reset. This is because a new countermeasure, such as VGPO, would need to start its procedure again from scratch, requiring the full time to complete the break-lock process. The break-lock timer value is determined by a user-defined lookup table where a different break-lock time can be defined for each threat type, and for each countermeasure used against it. To simplify the process of understanding and analysing the results of this work, all active RF technique break-lock times are set to 4 s in this work, even though the underlying models do not require this.

Next, the break-lock counter is simply decremented by the time resolution of the scenario. Then once the counter reaches zero, the threat's lock is broken and its radar stage reset to the search stage, and the time-to-next-radar-stage counter set to the corresponding search time. It is noted that

unlike radar stage progression, the break-lock rate is fixed and hence unaffected by the jamming effect of the platform. This is due to the fact that the rate at which RGPO and VGPO work (the only active techniques effective against tracking-type stages in this work) is not dependent on the JSR. Instead, it is dependent on the techniques themselves. Specifically, these techniques work by creating a false target in the immediate vicinity of the platform before slowly moving it away in either range or velocity, and then finally breaking lock by removing the false target once it is sufficiently far away. As a result, the time required to execute this process is consistent and dependent on the characteristics of the threat's radar system, rather than the JSR.

Lastly, it is noted that the process detailed above remains the same for passive countermeasures. The only difference is that a separate break-lock threshold can be defined for IR threats, which for simplicity has also been set to 0.2 in this work. Similarly, in order to improve the readability of the results of this work, the break-lock times for all passive techniques have been set equal to the 4 s used for the active techniques, despite the underlying models not inherently requiring this. Importantly, it must be noted that in this model it has been assumed that the time to execute a particular chaff or flare pattern for a specific threat is independent of the range and angle of that threat, as the complex modelling of these chaff and flare patterns is outside the scope of this work. Further, it is acknowledged that this 4 s break-lock time for chaff and flare countermeasures is perhaps too long in many instances. However, this value has been selected, in light of the coarse time intervals used, in order to demonstrate how the model operates with multiple time intervals of passive countermeasures.

### 3.2.7 Dead time

The last phenomenon that must be accounted for in this process is the fact that it requires a finite amount of time for the platform's ECM channels to switch between countermeasure techniques and targets. During this time, the effect of the channel is negligible to non-existent. These changes include the techniques themselves, the threat types for which they are optimised, the antenna directions, and the decoy allocation. This is accounted for by implementing a dead time for a channel after a new countermeasure is selected. During this dead time, the techniques implemented by the channel have zero jamming effect, and do not use up cartridges. Further, it must be noted that this dead time does not apply to the switching of channels to no jamming technique, as the transmitter can be simply turned off, or the firing of cartridges ceased.

This is implemented using a user-defined lookup table of dead times for each ECM channel depending on whether a new technique has been selected, a new optimised threat type has been selected, the antenna orientation has changed, the decoy allocation has changed, or any of combination thereof. This allows for maximum flexibility. Further, the implementation of dead times also serves to discourage the optimisation algorithm from rapidly switching between numerous countermeasures and giving them no time to have an effect on a threat. In this work, for simplicity, a single dead time interval of 2 seconds has been chosen for all changes in countermeasures, even though a single value is not inherently required.

## 3.3 DANGER VALUE

As stated previously, a very important part of the TECA process is the ability to prioritise threats so that countermeasures can be allocated accordingly. This is achieved using a danger value (*D*) that is representative of the level of danger a threat presents to the platform and is calculated as the first major step in the TECA process. It takes into account a number of characteristics: the probability of encountering the threat (*P*), the threat's radar stage (*S*), the time before the threat will progress to the next radar stage (J), the range-adjusted accuracy of the threat (*A*), and the time that a projectile from the threat would take to reach the platform ($\omega$). Further, it also takes into account the RCS of the platform as perceived by the threat. It is calculated for the $n^{\text{th}}$ threat in the $k^{\text{th}}$ time interval using

$$D_n^k = \mathrm{RC_b}\left(\varphi_n^k, \theta_n^k\right) \cdot P_n^k\left[W_s S_n^k + W_a A_n^k + W_t\left(1 - \omega_n^k\right) + W_x\left(1 - J_n^k\right)\right], \tag{3.16}$$

where $W_s$, $W_a$, $W_t$, and $W_x$ are the user-defined weights that are used to set the relative level of priority of each threat characteristic, where the individual factors are normalised prior to being weighted.

The threat's radar stage is a strong indicator of how far along it is in its engagement procedure, and hence how close it is to firing upon and hitting the platform. Further, the more accurate a threat is, the greater the likelihood of it hitting the platform. Therefore, both of these factors increase the danger value of a threat.

On the other hand, the shorter the time before the threat progresses to the next radar stage, the further along it is in its engagement procedure, and hence the greater the danger it presents to the platform. Therefore, a smaller *J* value should increase the danger value, and as such it is included as a $1 - J$ factor. This approach is used so as to prevent negative danger values, and is feasible due

to the fact that $J$ is a normalised version of the time-to-next-stage counter ($X$) discussed previously. The projectile time to platform essentially serves the purpose of prioritising threats according to how close the threat is to the platform. Due to the differences in projectile velocities, a projectile from a threat that is further away from the platform might reach it sooner than another closer threat, and hence present a more immediate danger. As such, $\omega$ takes both the threat distance and its projectile velocity into account. As with the $J$ factor, due to its normalised nature it is used in a $1 - \omega$ manner. Importantly, both of these factors act to further differentiate the danger presented by threats in the same radar stage.

The danger value calculated using the previous factors is then scaled by the probability of the platform actually encountering that threat in the time interval being examined. It is then further scaled by the RCS of the platform as perceived by that threat in its frequency band of operation ($b$). This is due to the fact the perceived RCS directly affects the performance of the threat's radar and its associated ability to detect, track and be jammed by the platform due to its multiplicative effect on the radar range equation.

User-defined weights are used in order to allow the user to optimise system performance according to their requirements. However, weights are required in this work in order to demonstrate system performance. Therefore, since the progress of a threat through its radar stages is the greatest indication of how soon it will engage the platform, the radar-stage weight is allocated the largest weighting of 8. Thereafter, the time-to-next-stage, projectile-time-to-platform, and accuracy weightings are set to 4, 2, and 1 respectively, where these weights were chosen so that each is half the value of the previous one. This approach ensures that each weight is sufficiently lower than the previous one and first prioritises threats that are further along in their engagement procedure. Thereafter, threats that are closer to the platform are prioritised, before finally threats with greater accuracy. Note that the values of the weights themselves are not important as each is normalised by dividing it by the total of the weights.

Detailed explanations of the calculation of these values, as well as their normalisation, follow. Note that the danger value of a threat is set to zero if the platform is outside its radar range and hence unable to detect the platform. This prevents the platform from unnecessarily allocating resources to that threat.

### 3.3.1 Platform RCS

The RCS of the platform plays a major role in the engagement between itself and each threat in the mission area. This is evidenced by its role in the radar range equation, where the received power ($P_r$) is directly proportional to the platform RCS ($\sigma$) [32]:

$$P_r = \frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^3 R^4}, \tag{3.17}$$

where $P_t$ is the transmitted power, $R$ is the range of the target, $G_t$ is the transmitter gain, $G_r$ is the receiver gain, and $\lambda$ is the wavelength. As such, it has a direct effect on a threat's radar performance and hence the danger that threat presents to the platform. Further, this extends to RCS having a direct effect on the JSR of the platform's countermeasures, and hence their performance. Therefore it follows that the RCS should have a direct multiplicative effect on both the danger value of a threat, and the jamming factor used to account for the effects of countermeasures. However, since the product of these two values is used to determine the post-jamming danger value of a scenario, the inclusion of RCS in both would result in a squared effect in the optimisation process, rather than the desired effect indicated by the radar equation. Therefore the platform RCS is included solely in the calculation of danger values in (3.16) rather than in the jamming factor so as to prevent this issue, and so that threats can be correctly prioritised. However, this does mean that the platform RCS must still be explicitly included in the other places where the jamming factor is solely used in calculations. As such, the RCS factor is also included in radar stage progression, and the breaking of tracking lock as discussed previously.

RCS is implemented using a number of user-defined lookup tables of variable size, where each contains the normalised RCS factor of the platform for a specific frequency band, for each possible combination of azimuth and elevation angles. A separate RCS array can be defined for each frequency band, where the RCS of the platform at a single representative frequency is used over the entire band. Any number of bands may be used, but in this work the L, S, C, and X bands have been selected along with representative frequencies of 1 GHz, 3 GHz, 5 GHz, and 10 GHz respectively. The RCS array used for a specific threat is then dependent on its frequency band of operation as specified in a threat's characteristics. Further, a user-defined resolution is used for each of the angles in the platform's RCS arrays to allow the user to input any model into the system.

Note that the RCS factor values must be specified in decibels relative to a square meter (dBsm), and that the values must be normalised into a non-zero factor in the range of 0 to 1, relative to the largest RCS factor value across all frequency bands of operation. Importantly, this allows the RCS to be used as a scaling factor in the calculation of danger value. If linear values are used instead, the large orders of magnitude between the RCS values of different angles results in poor optimisation performance. This can be seen in Figure 3.3, which depicts the linear normalised RCS factor of an aircraft in the L band using the same flat plate approximation used and discussed later in this section. It is immediately seen that a large RCS value is perceived directly above and below the platform, with moderate values perceived directly in front of, behind, and to each side of it. At every other angle the RCS factor is non-existent in comparison. The direct result of which is that the danger value of a threat in these other angles will be virtually zero in comparison to any threats in the peak positions. In turn this will cause them to be essentially ignored in the optimisation process when threats occur in the peak positions at any point in the mission. Also, even if no threats perceive the platform from a peak position in a scenario, there is insufficient resolution to capture the differences between the RCS values at these off-peak points. This results in the RCS factor having an insignificant effect on the danger value, and in turn on the optimisation process. Further, it is important that the factor is normalised in such a way that it is non-zero because zero values would result in some threats having zero danger value. This is unrealistic because any functioning threat always presents at least some level of danger to the survival of the platform, and hence should have some bearing on the allocation of countermeasures.

As stated previously, the directions of the various threats relative to the platform's axis over the course of the mission are calculated prior to running the optimisation process due to the fact the mission route is fixed across all solutions. Therefore, these angles simply need to be checked whether they lie within the necessary ranges, and rounded to the nearest point in the lookup table according to the user-defined angular resolutions. $\mathrm{RC_b}\left(\varphi_n^k, \theta_n^k\right)$ is then simply determined by finding the corresponding lookup table value for those angles for that threat's frequency band of operation. When used as a multiplicative factor, this scales the danger value presented by that threat such that a larger RCS results in a greater danger value and vice versa.

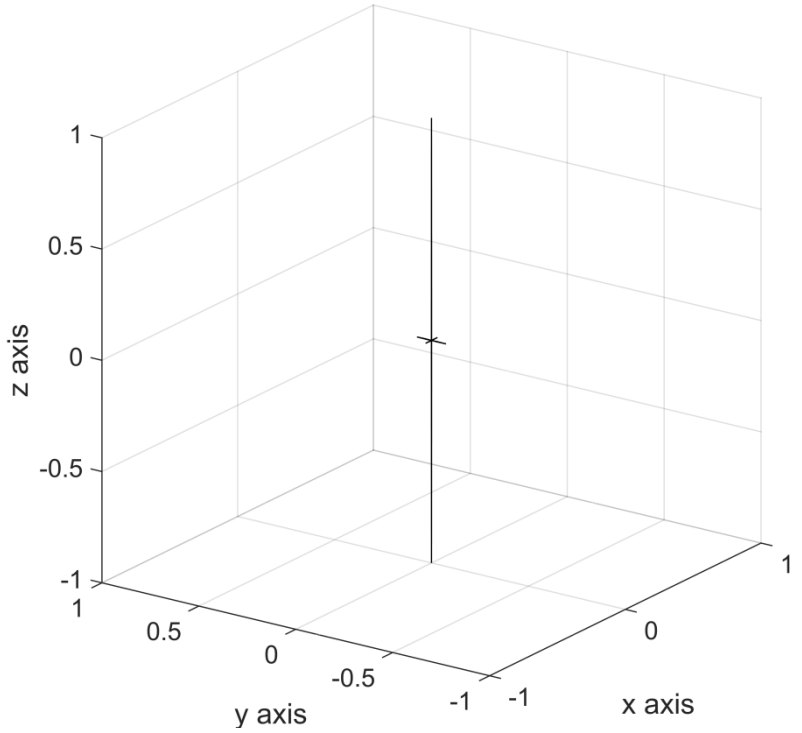



**Figure 3.3.** Linear normalised RCS factor approximation of an aircraft in the L band.

In the case of the azimuth angle, the lookup table is defined for 0° to 360°, where 360° is non-inclusive. As such, if the calculated value is outside of this range, 360° is either added to, or subtracted from the value in order to bring it within range. The value is then divided by the specified azimuth resolution and rounded off in order to determine the lookup table index. The same process is then repeated for the elevation angle, except that the range is from -90° to 90°, inclusive.

In order to generate a reasonable frequency-dependent RCS factor without relying on classified RCS measurements and models, or relying on insufficiently-detailed or non-frequency-dependent measurements and models in the literature, the flat plate approximation of an aircraft that appears in Figure 3.4 is used in this work. Note that such a flat-plate model does have precedent in the approximation of RCS data of generic aircraft [33], and that the focus of this work is on the use of the RCS data, rather than its generation, where accurate RCS data for the platform would be used instead in real-world applications. As such, all that is of importance in this work is that the model has the general RCS characteristics of an aircraft. Specifically, in this approximation the length of the aircraft is set to 15.96 m, and the wingspan is set to 10.95 m so as to emulate the dimensions of a Eurofighter Typhoon [34]. Further, the wing width is set to 5 m as this is the rounded-up width of

4.676 m required to obtain the aircraft's published wing area of 51.2 m. This was done so as to account for the surface area of the body of the aircraft. Also, the height and width of the body are both set to 3 m so as to approximate the dimensions of the aircraft. Lastly, it is noted that a resolution of one degree is used in both the azimuth and elevation angles for the RCS models in this work.

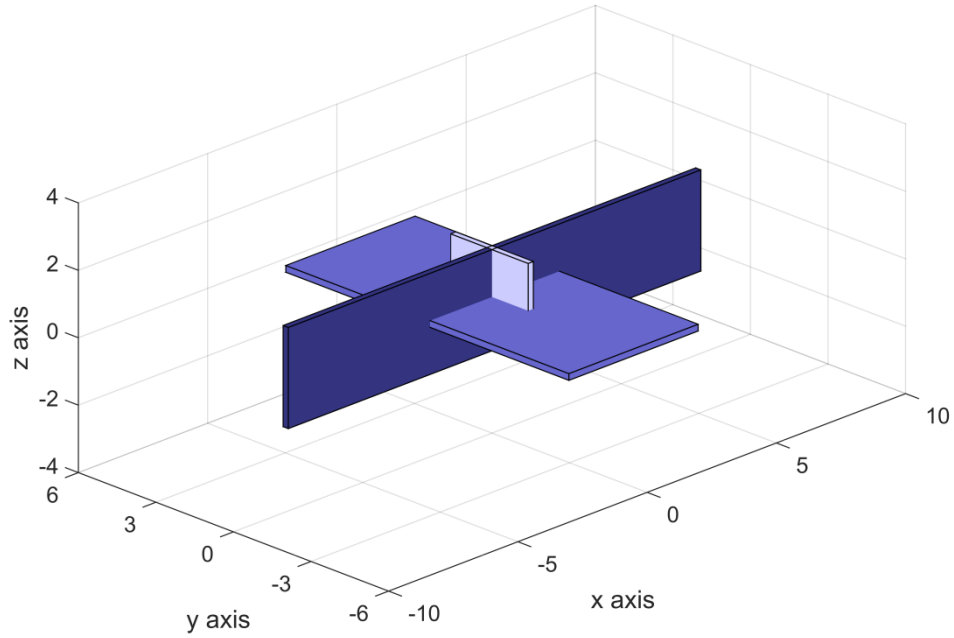



**Figure 3.4.** Rectangular flat plate approximation of an aircraft, with dimensions in meters.

The approximated RCS of each of the individual plates, of dimensions 2$a$ by 2$b$, can then be calculated for each possible combination of azimuth and elevation angles as [35]

$$\sigma(\varphi,\theta) = \frac{4\pi a^2 b^2}{\lambda^2}[\text{sinc}(ak\sin(\theta)\cos(\varphi)) \cdot \text{sinc}(bk\sin(\theta)\sin(\varphi))]^2 \cdot \cos^2(\theta), \quad (3.18)$$

where

$$k = \frac{2\pi f}{c}, \quad (3.19)$$

if the plate is assumed to be perfectly conducting and situated on the x-y plane, with the elevation angle measured from the positive z axis downward. The RCS of the plate representative of the side of the platform's body must then be rotated 90° around the x-axis using the rotation matrix $R_x$ in (3.9), and the RCS of the plate representative of the front of the aircraft must be rotated 90° around the y-axis using the rotation matrix $R_y$ in (3.8). In order for this to be achieved, the RCS must first be converted into Cartesian coordinates, and then converted back into spherical coordinates after the rotation has been performed. Finally, the total RCS of the platform is then calculated as the sum

of the RCS values for each individual plate before being converted into decibels and normalised. This is achieved by first limiting the minimum RCS value to -135 dBsm so as to maintain a reasonable resolution and distribution of the normalised factor. Thereafter the magnitude of this minimum value plus one is added to the RCS so as to move the factor into a positive non-zero range before it is normalised using the maximum value across all frequency bands. Finally, this results in a three dimensional RCS factor in the range of zero to one, that is importantly non-inclusive of zero.

Note that the minimum value of -135 dBsm used in the above normalisation process can be set by the user to any value that results in good resolution and distribution of the factor values. Specifically in this case, it has been chosen so that the median RCS value across all azimuth and elevation angles, and across all frequency bands is 0.5. Importantly, this ensures that the RCS factor values are well distributed in magnitude by ensuring that half of the view angle possibilities result in a factor occurring in the lower half of possible values, and the other half occurring in the upper half of possible values. Further, in this case this choice of median value also results in a mean value of 0.52. The culmination of the statistics means that on average the platform will be perceived to have an RCS factor of around 0.5, with threats in more advantageous positions perceiving a larger RCS factor and vice versa. A comparison of the mean and median values of the platform's RCS factor for each of the different frequency bands appears in Table 3.1, where it is seen that the average values of the lower frequency bands are greater than those of the higher bands. However, overall the factor is well distributed.

**Table 3.1** RCS factor mean and median values for each frequency band.

| Statistic | L Band | S Band | C Band | X Band | All Bands |
|---|---|---|---|---|---|
| **Mean** | 0.5808 | 0.5276 | 0.5036 | 0.4701 | 0.5205 |
| **Median** | 0.5595 | 0.5055 | 0.4806 | 0.4478 | 0.5030 |

It is acknowledged that the minimum value of -135 dBsm is low. However, only 1.317% of factor values occur in the lower third of the range. In other words this means that only 1.317% of pre-normalised RCS values occur below -90 dBsm across all frequency bands. It is further noted that these low values are a function of the flat plate approximation used, where the flat plates have a large broadside RCS, which then rapidly tails off thereafter. The bodies of real aircraft do not suffer from this issue. However, as stated previously, this model is solely used to generate a reasonable frequency-dependent RCS factor without relying on classified RCS measurements and models, or

relying on insufficiently-detailed or non-frequency-dependent measurements and models in the literature. Further, RCS modelling of a more detailed and complex nature is outside the scope of this work.

Finally, the resultant three dimensional RCS models used for this work appear in Figure 3.5. for the L, S, C, and X bands using frequencies of 1 GHz, 3 GHz, 5 GHz, and 10 GHz respectively. Thereafter, the front view of the platform RCS is compared between the different frequency bands in Figure 3.6. This is when the azimuth angle is set to zero (the front of the platform) while the elevation is varied. Figure 3.7 contains the same comparison for the side view where the azimuth angle is set to 90°. Lastly, Figure 3.8 compares the top view of the RCS for each frequency band, where the elevation angle is kept at a constant value and the azimuth angle is varied. Specifically, the elevation angle is set to a constant value of 0° according to the conventions used in this work. Note that for Figures 3.6 – 3.8, the frequency bands have been split into two pairs: the L and C bands, and the S and X bands. This has been done for maximum visibility and clarity. Further, in each case the lower frequency band has been depicted in black, and the higher in blue. Significantly, it can be seen in these figures that the flat-plate approximation does indeed generate RCS models that are representative of the general characteristics of aircraft such as the F-15, F-16, and F-35 [36] [37], in that the RCS peaks to the front, back, sides, top and bottom of the aircraft.

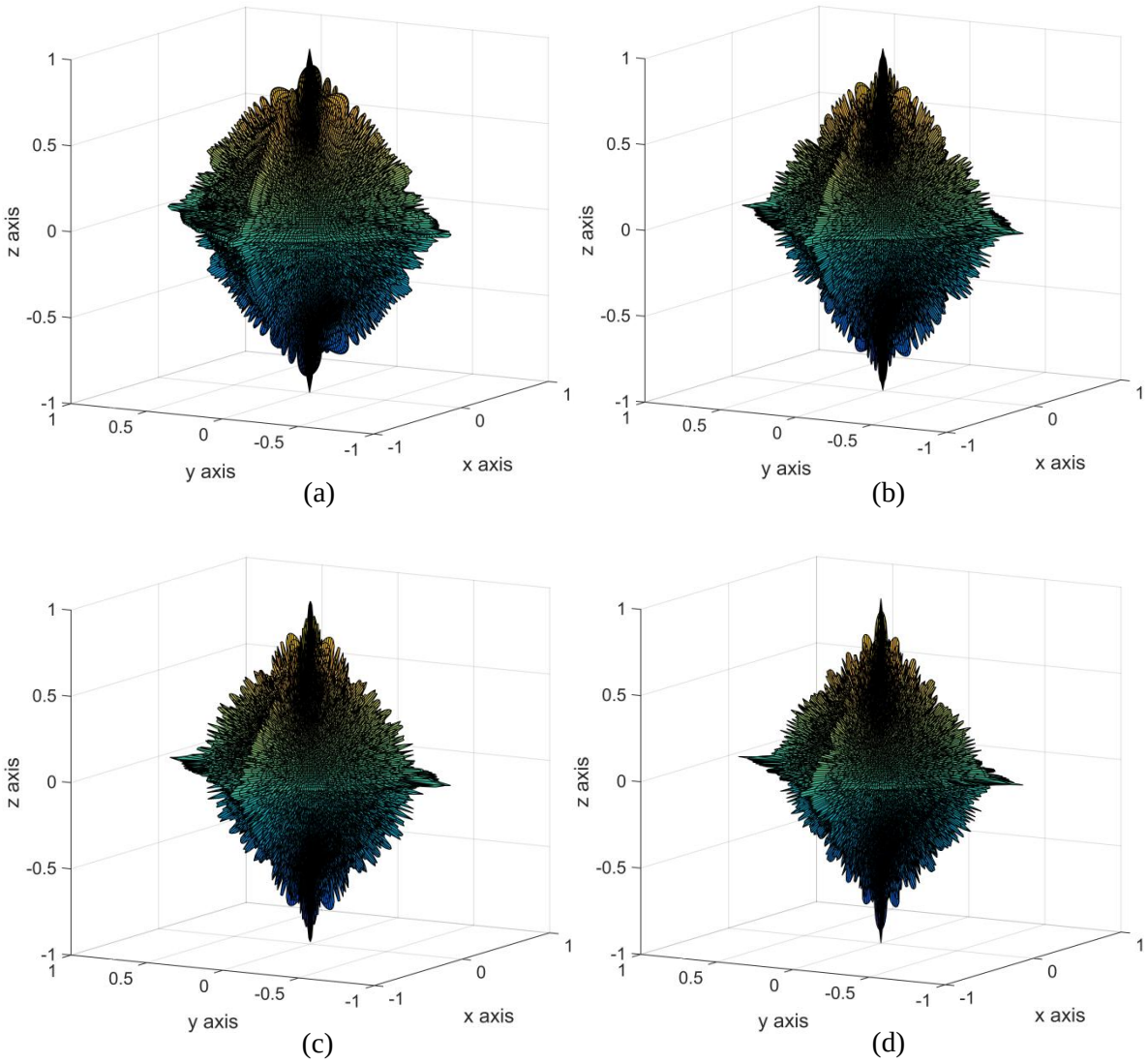
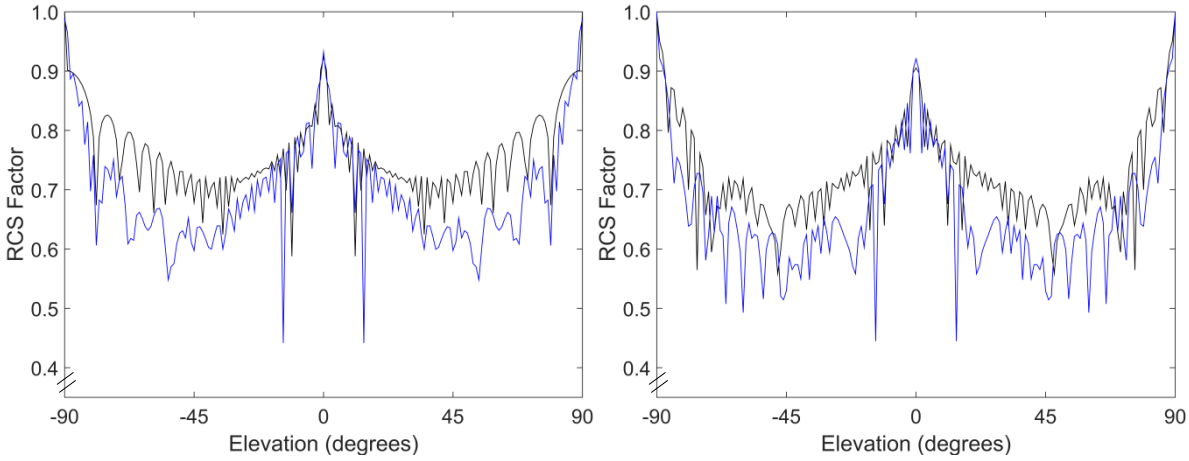
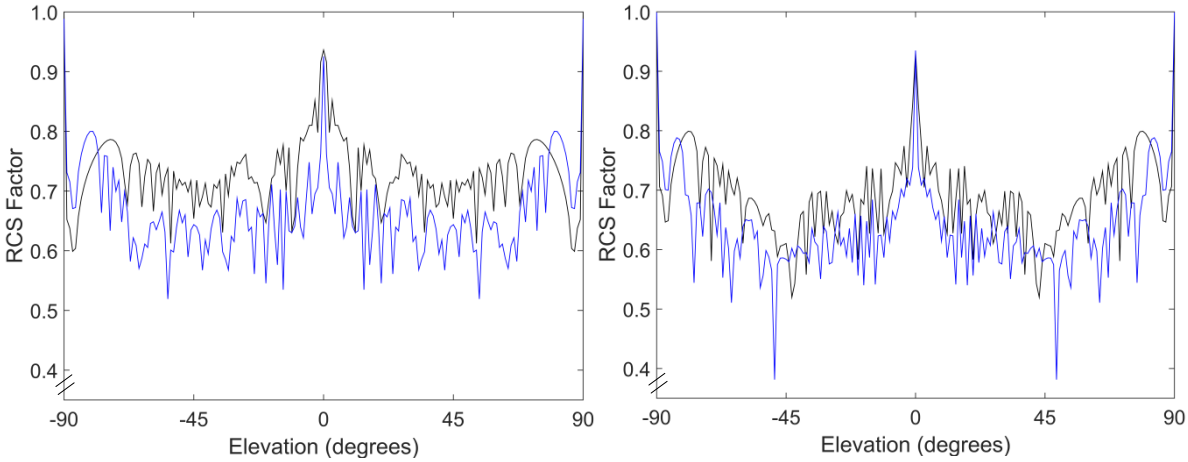



**Figure 3.5.** Three dimensional platform RCS factor for L (a), S (b), C (c), and X (d) bands.




(a) (b)

**Figure 3.6.** Platform RCS front view for L and C bands (a), and S and X bands (b), where lower frequencies are in black.




(a) (b)

**Figure 3.7.** Platform RCS side view for L and C bands (a), and S and X bands (b), where lower frequencies are in black.

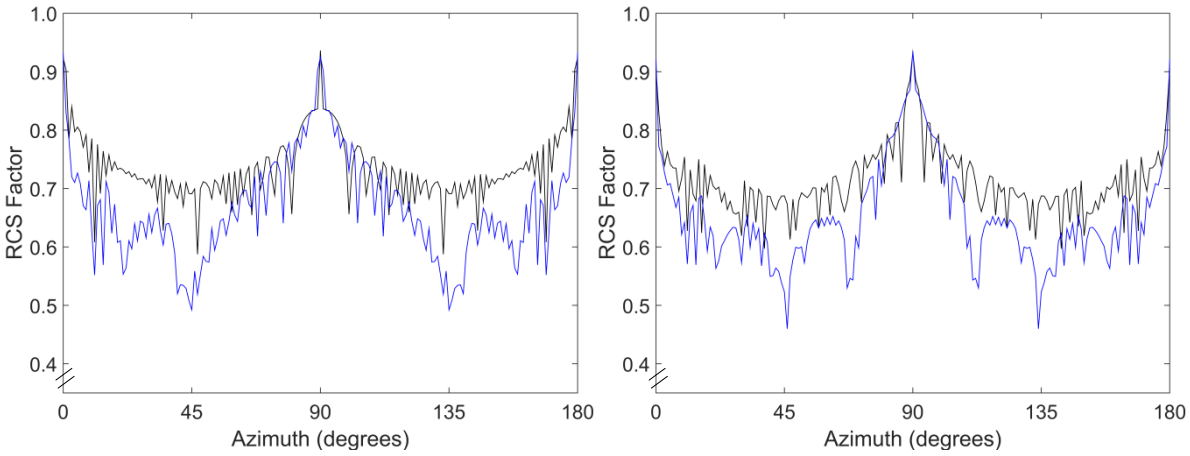



**Figure 3.8.** Platform RCS top view for L and C bands (a), and S and X bands (b), where lower frequencies are in black.

### 3.3.2 Probability of threat encounter

Due to the focus of this work on computational efficiency, it is both infeasible and beyond the scope of this work to implement complex artificial intelligence to drive the movements and reactions of the threats. As such, a simple distributed threat approach is used instead, where each threat is allocated a central point location, a radius of likely encounter, and an overall probability of occurrence ($P_{n,occ}$). The centre point indicates the expected location of the threat, while the radius indicates the possible deviation of threat location from this point. This radius accounts for a number of factors including movement of the threat (both coincidental movement and deliberate engagement of the platform) and inaccuracies of intelligence. On the other hand, the overall probability of encounter accounts for the cases that the threat leaves the area completely, it is non-functioning, under maintenance or if it is a mistakenly identified non-threat. As a result, faster moving threats will have a larger radius value and camouflaged or small threats will have a lower probability (and vice versa). The advantage of this method over just simply using point locations for threats is that it more accurately represents the nature of battlefield intelligence: a threat is known to be within a certain area, with a certain probability, rather than a guaranteed threat at a specific location.

In light of the above, the probability of the platform encountering a threat in the examined time interval ($P_n^k$) is defined as the probability that the platform will be within range of that threat. This is calculated by determining the portion of the distributed threat area that the platform is within range of, and scaling the probability of threat occurrence accordingly. This process is depicted in Figure 3.9 for a single threat in a single time interval, where $R$ is the radar range of the threat being analysed, $r$ is the user-specified distributed-threat radius and d is the ground distance between the platform location and the centre location of the threat. As such, the shaded area is the portion of the threat area that the platform is within range of, and hence poses a danger to it.

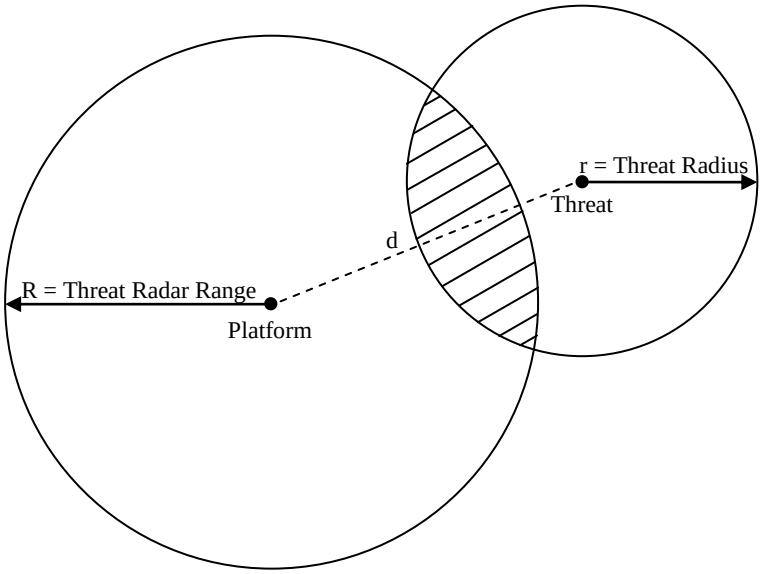



**Figure 3.9.** The portion of a distributed threat area that the platform is within range of.

The shaded area's proportion of the entire distributed threat area is calculated as

$$A_{prop} = \frac{A_{shaded}}{\pi r^2}, \tag{3.20}$$

where

$$A_{shaded} = C_1 - C_2, \tag{3.21}$$

$$C_1 = r^2 \cos^{-1}\left(\frac{d^2+r^2-R^2}{2dr}\right) + R^2 \cos^{-1}\left(\frac{d^2+R^2-r^2}{2dR}\right), \tag{3.22}$$

$$C_2 = 0.5\sqrt{(-d+r+R)(d+r-R)(d-r+R)(d+r+R)}. \tag{3.23}$$

The probability of the platform encountering the threat in the $k^{\text{th}}$ time interval can then be calculated as

$$P_n^k = P_{n,occ} \times A_{n,prop}^k, \tag{3.24}$$

where $A_{n,prop}^k$ is the proportional area of the $n^{\text{th}}$ threat in the $k^{\text{th}}$ time interval. The result of the above is a probability of encounter. The meaning of this is best explained with an example: if the user-specified overall probability of encountering a threat is 0.8 (80%) and the platform is within range of only half of the distributed threat area, then there will only be a

$$0.8 \times 0.5 = 0.4, \tag{3.25}$$

probability of encountering that threat in that time interval. Therefore, its danger value must be scaled accordingly, hence its multiplicative inclusion in (3.16).

However, (3.24) only works for the exact situation depicted in Figure 3.9, where a portion of the threat area is within range, rather than other cases where the entire area, or none of it is within range, or where the distributed area is so large that it completely envelops the range circle around the platform. These cases are handled using a number of conditional rules where

$$P_n^k = \begin{cases} P_{n,occ}, & \text{if } R \geq d + r \\ 0, & \text{if } d \geq r + R \\ P_{n,occ} \times \dfrac{\pi R^2}{\pi r^2}, & \text{if } r \geq d + R, \end{cases} \tag{3.26}$$

Finally, the centre point of the threat is used as the threat location for all calculations in this work, such as ground distances and slant ranges. This is done in order to simplify and speed up calculations by using the average point of the distributed threat instead of attempting to perform a calculation for every potential threat location in the distributed area. Since this is done for all threats, the effect is relatively equal on them all, causing minimal detrimental effect to the accuracy of the system, except for cases with extreme radius values.

### 3.3.3 Threat radar stage

In (3.16) a normalised value is required for $S_n^k$ that is representative of the radar stage of the threat and its associated levels of danger. Due to the fact that in this equation a larger number indicates an increase in danger, and the fact that the further along a threat is in its engagement procedure of search, acquisition, tracking and guidance, the greater its danger to the platform, these stages must

be allocated increasing $S_n^k$ values. As such, search, acquisition, tracking and guidance are allocated $S_n^k$ values of 0.25, 0.5, 0.75, and 1 respectively.

### 3.3.4 Threat accuracy

It is clear that a threat with a greater probability of inflicting damage to the platform poses a greater threat to its survival. Therefore, the accuracy of a threat must be included in (3.16) in the form of $A_n^k$, a probability of hit, where a greater accuracy results in a greater danger value for the threat. As such, each threat type is allocated both a user-defined average accuracy ($A_{n,sys}$) in the form of a probability of hit, weapon system type, and maximum weapon range ($R_{max}$). In this work there are five guided missiles types: IR, active, semi-active, command, and beam-riding missiles. There are also two artillery types: explosive, and non-explosive. For some threat types, such as semi-active, active and IR missiles, the average accuracy value is constant over the system's entire user-defined weapon range ($R_{max}$). This is due to the fact that their receivers and guidance systems are built into the missile itself. However, for the remaining weapon system types (artillery, as well as command and beam-riding missiles), their accuracy is not constant over their entire range. The reason for this is that these weapon systems are either fired from (without guidance) or guided by the threat itself and hence are limited by the angular accuracy of the system [15]. Therefore the accuracy of these systems is modelled to remain constant at the user-defined value for ranges up to half the maximum range before decreasing at a linear rate as depicted in Figure 3.10.

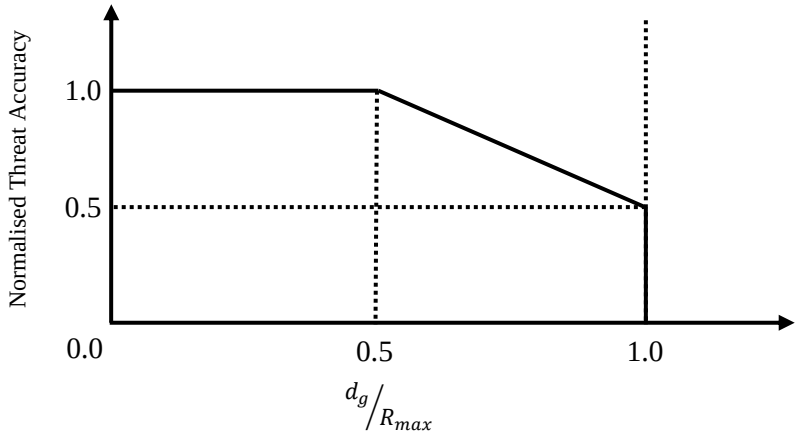



**Figure 3.10.** Normalised threat accuracy verses range for artillery, and command and beam-riding missiles.

This is because it is assumed that for distances up to half of the maximum range the weapon system's angular accuracy is greater than required, resulting in constant accuracy. After this, the accuracy decreases linearly due to the fact that the length of the arc ($L$) between two radii of a circle is equal to the product of their length ($R$) and the angle ($\theta$) in radians between them:

$$L = \theta R. \tag{3.27}$$

In this equation, $L$ can be seen as the approximate miss distance, $R$ as the range, and $\theta$ as the angular accuracy error. As such, it is seen that the miss distance can be approximated as linearly dependent on the range of the threat from the platform. Therefore, the range adjusted accuracy ($A_n^k$) of these threats is calculated as

$$A_n^k = \begin{cases} A_{n,sys}, & \text{if } R \leq 0.5R_{max} \\ A_{n,sys}\left(1.5 - \dfrac{d_g}{R_{max}}\right), & \text{if } R > 0.5R_{max}. \end{cases} \tag{3.28}$$

Importantly, in the above calculation it has been assumed that weapon ranges have been specified in ground distances rather than slant ranges, and as such the ground distance of the threat is used.

Finally, it is noted that a number of effects on accuracy, such as atmospheric effects, are not accounted for in this model. This is due to the fact that the modelling of these effects is outside the scope of this work. Also, since the user-defined accuracy is in the form of a probability of hit, there is no need to normalise this factor.

### 3.3.5 Projectile time to platform

A further way to differentiate between the levels of danger presented by each threat is by comparing how close each is to the platform. However, the actual distance itself is of little value on its own because a further away threat with a faster projectile could hit the platform before a closer threat. Therefore, the time a threat's projectile would take to reach the platform is used instead. This measure of danger is aimed at the differentiation of threats in similar stages of engagement, especially in the guidance stage, where a guided missile that is about to hit the platform should receive jamming priority. As such, projectile time to platform is included in (3.16) as the normalised factor $T_n^k$. However, it is included in the form of one minus $T_n^k$ so that a smaller time to platform results in a greater danger value.

The projectile time to platform factor is simply calculated by dividing the threat's slant range from the platform by the user-defined velocity of its projectile. This value is then normalised using the

maximum possible projectile time to platform which is calculated using the largest possible slant range and the slowest projectile velocity of all the threats. Specifically, the largest possible slant range that can be encountered in a mission is calculated using a combination of the longest radar range, and the maximum height achieved by the platform. It is also important to note that in the case of guided missiles, the distance to platform must be calculated using the missile's current position, rather than that of the threat itself.

### 3.3.6 Time to next radar stage

The next factor to be taken into account is the normalised time to next radar stage ($J_n^k$). This is simply the normalised time-to-next-stage counter ($X_n^k$) of each threat that is calculated using (3.12), (3.13), and (3.14). It is normalised by dividing each counter value by the largest user-defined radar stage time of all the threats whether it is a search, acquisition, or tracking time. As with the projectile time to platform, this factor is also included in (3.16) as one minus $J_n^k$ so that a smaller time-to-next-stage counter results in a greater danger value for that threat. This is because this factor, along with that of the threat radar stage factor ($S_n^k$), allocates danger according to how far along a threat is in its engagement procedure, where the further along it is, the closer it is to firing upon the platform, and the greater danger it presents. Specifically, this factor helps differentiate between threats that are in the same radar stage by looking at how close they are to progressing to the next one.

### 3.3.7 IR threats

For the most part, IR threats are handled in the same way as RF threats. The only major difference is that IR threats in the undetected stage are allocated zero danger value due to the fact that their presence is not known at that point, and that the implemented countermeasures are unable to counter them until they enter the guidance stage anyway. Further, it is noted that RCS does not affect the performance of IR threats.

## 3.4 JAMMING FACTOR

As stated previously, the jamming factor (*F*) forms part of the countermeasure allocation process of TECA. It is a multiplicative factor that accounts for the effect of countermeasures on the danger

value of a threat. As such, it forms an important part of the optimisation procedure where the resultant post-jamming danger values ($V$) are used to determine an overall risk level of a particular strategy for comparison purposes. The jamming factor is defined such that a value of 0.8 indicates a 20% reduction in the radar performance of a threat, and it is equal to one minus the jamming effect ($E$). For RF threats this post-jamming danger value is calculated on a threat-by-threat basis for each time interval as

$$V_n^k = F_n^k \cdot D_n^k = \left(1 - E_{n,final}^k\right) \cdot D_n^k, \tag{3.29}$$

where

$$E_{n,final}^k = E_{max} \cdot (E_{n,adj}^k + E_{n,3}^k), \tag{3.30}$$

$$E_{n,adj}^k = \left(E_{n,1}^k + E_{n,2}^k\right) \cdot \left[1 + \left(\frac{d_a}{R_j}\right)^2\right]. \tag{3.31}$$

$E_{n,1}^k$, $E_{n,2}^k$, and $E_{n,3}^k$ are the individual jamming effects of active channel 1, active channel 2, and the passive channel on the $n^{\text{th}}$ threat in the $k^{\text{th}}$ time interval respectively. Next, $d_a$ is the slant range of the platform from the threat, and $R_j$ is the user-defined maximum jamming range of the platform. It is currently set to 20 km as this value aligns with the ranges contained in the scenario used to generate results in this work.

A positive jamming effect value indicates that the jamming is reducing the danger posed by that threat, and results in a jamming factor of less than one. This in turn results in a reduced post-jamming danger value, indicating lower risk to the platform. On the other hand, a negative jamming effect value indicates platform illumination, and results in a greater than one jamming factor. This in turn results in a greater post-jamming danger value, indicating a greater risk to the platform.

As such, in the above equations the individual jamming effects of each countermeasure channel are combined in order to determine the total jamming effect of the platform against a particular threat in a particular time interval. This is achieved by first adjusting the jamming effects, and hence effectiveness, of the active channels ($E_{n,1}^k$ and $E_{n,2}^k$) for the effect of path loss over distance in (3.31). As with RCS, this is implemented as a multiplicative factor, where the distance is normalised relative to the maximum range of the jammer, due to the nature of range's inclusion in the radar range equation in (3.17). However, it has also been assumed that the platform makes use

of a saturated jamming approach with constant output power due to the fact that this results in the greatest efficiency for the ECM system's amplifier [38]. As such, a squared distance effect is used that has the net result of increasing the absolute jamming effects as a function of range. This is because the skin return power decreases with $d^4$, whilst the jammer power only decreases with $d^2$, thus causing the JSR to increase with $d^2$. Lastly, it must be noted that in order to prevent this factor from overpowering the jamming effect, a value of one is added to this squared, normalised distance so that it instead results in a percentage enhancement to the jamming effect. This is because otherwise half of the possible distances would result in a squared, normalised distance of less than 0.25 that would in turn render most jamming useless in the model.

Thereafter the jamming effect of the passive channel is combined with that of the range-adjusted active channel effects, and adjusted using the user-defined maximum-jamming-effect factor $E_{max}$ in (3.30). This factor sets the maximum effect that jamming can have. In this work it is set to 0.9, which prevents a countermeasure from reducing a threat to below ten percent effectiveness. More importantly, it prevents a threat's post-jamming danger value from being reduced to zero. This is due to fact that any functioning threat always presents at least some non-zero level of danger to the platform, and as such no countermeasures should be able to reduce a threat's post-jamming danger value to zero.

It is noted that in the above equations $E_{n,3}^{k}$ is set to zero in cases where there is no chaff used. Also, if the jamming effect exceeds a value of one prior to being scaled by the maximum effect, it simply set equal to one. This is due to the fact that a countermeasure technique cannot be more than 100% effective. Further, it is noted that in the above equations the jamming effect of passive countermeasures is not adjusted for the distance between the platform and the threat. This is due to the fact that the JSR is independent of distance for these techniques. In the case of chaff this is because both the chaff and platform radar returns are dependent on the signal power that reaches the platform, and thereafter travel the same distance and undergo the same $d^4$ level of path loss. Similarly, the signal emitted by flares and the IR signature of the platform both propagate the same distances to a passive seeker and hence suffer the same $d^2$ path loss.

The individual jamming effects are discussed in detail in the following sections since the calculation of these values is different for active channels and chaff. Further, IR threats and flares

are handled completely separately to RF threats and hence are discussed separately in Section 3.4.3.

### 3.4.1 Active channels

The jamming effects of each individual channel need to take into account numerous factors including the countermeasure techniques used, the radar stage of the threat being examined, interactions with other countermeasures, antenna gain patterns, and relative frequency domain usage.

As such, the jamming effect for each active channel is calculated on a threat-by-threat basis for each time interval using a number of multiplicative factors in the form of lookup tables: stage effectiveness (SE), cross effect (CE), and chaff interference (CI). Further, a countermeasure resistance (CR) factor is also included along with a separately calculated antenna gain pattern (AG) factor, and technique interaction (*I*) factor. The jamming effect of active channels 1 and 2 can be calculated for the $n^{\text{th}}$ threat in the $k^{\text{th}}$ time interval as

$$E_{n,1}^{k} = \mathrm{AG}_{1,b}\left(\varphi_n^k, \theta_n^k, \varphi_{L1}^k\right) \cdot \mathrm{SE}\left(S_n^k, J_1^k\right) \cdot I_{1,2}^k \cdot \mathrm{CE}\left(O_1^k, Y_n\right) \cdot \mathrm{CI}\left(J_1^k, J_3^k\right) \cdot \rho_{n,1}^k, \tag{3.32}$$

$$E_{n,2}^{k} = \mathrm{AG}_{2,b}\left(\varphi_n^k, \theta_n^k, \varphi_{L2}^k\right) \cdot \mathrm{SE}\left(S_n^k, J_2^k\right) \cdot I_{2,1}^k \cdot \mathrm{CE}\left(O_2^k, Y_n\right) \cdot \mathrm{CI}\left(J_2^k, J_3^k\right) \cdot \rho_{n,2}^k, \tag{3.33}$$

where

$$\rho_{n,1}^{k} = \begin{cases} 1 - \mathrm{CR}(Y_n), & \text{if } \mathrm{SE}\left(S_n^k, J_1^k\right) \geq 0 \\ 1 + \mathrm{CR}(Y_n), & \text{if } \mathrm{SE}(S_n^k, J_1^k) < 0, \end{cases} \tag{3.34}$$

$$\rho_{n,2}^{k} = \begin{cases} 1 - \mathrm{CR}(Y_n), & \text{if } \mathrm{SE}\left(S_n^k, J_2^k\right) \geq 0 \\ 1 + \mathrm{CR}(Y_n), & \text{if } \mathrm{SE}(S_n^k, J_2^k) < 0. \end{cases} \tag{3.35}$$

In these equations $J_1^k$, $J_2^k$, and $J_3^k$ are the three countermeasure techniques implemented in the time interval for the first active channel, second active channel, and the passive channel respectively. $O_1^k$ and $O_2^k$ are the threat types for which the countermeasure techniques are optimised, $S_n^k$ is the radar stage of the threat, and $Y_n$ is the threat type. $\mathrm{AG}_{1,b}$ and $\mathrm{AG}_{2,b}$ are the antenna gain pattern factors for the first and second channels in the frequency band of operation (*b*) of the examined threat. $\varphi_n^k$ and $\theta_n^k$ are the previously-calculated azimuth and elevation angles of the examined threat relative to the rolled, pitched, and yawed platform. $\varphi_{L1}^k$ and $\varphi_{L2}^k$ are the steering angles of the first and second

channel antennas respectively. Lastly, $\rho_{n,1}^{k}$ and $\rho_{n,2}^{k}$ are the countermeasure resistance factors that have been modified according to the sign of the stage effectiveness of the countermeasure techniques of the first and second channels, respectively. These factors are all discussed in the following sections.

*1) Antenna gain pattern*

This factor accounts for the effects of the gain patterns of the platform's antennas, and hence the relative jamming signal level perceived by a threat. Due to the multiplicative effect of antenna gain in the radar range equation in (3.17), this factor is used in a multiplicative fashion on the jamming effect for each threat so as to either increase or decrease its value according to the signal level directed towards that threat. This factor is important because a threat that the platform might otherwise be illuminating itself to could actually be positioned in a low-gain region of the antenna gain pattern, resulting in little to no illumination, and vice versa. Further, the use of this factor allows for the modelling of antenna steering for each of the individual jamming channels, where the optimisation of which can drastically improve the performance of countermeasure strategies by allowing the platform to steer its antennas in directions that offer the best compromise between effective jamming of some threats and platform illumination to others. For example, a jamming channel's antenna can be steered such that it aims between two similar threats thus suppressing both simultaneously, or it can be purposely steered such that a threat the platform would otherwise be illuminated to sits in a low-gain region of the antenna pattern.

In order to allow for antenna steering, it is assumed in this work that an antenna array is used for each active channel on the platform. Further, in order to simplify the optimisation of the problem, it is assumed that the platform is only capable of steering its antenna in the horizontal plane. Therefore, steering is only optimised in the azimuth axis for each active jamming channel to a user-defined resolution of 6°. However, vertical steering can be modelled using a similar approach. Note that this resolution was specifically chosen so as to make better use of the binary encoding of the chromosome in the genetic algorithm, but it can be set to any reasonable value.

Due to the above assumption of an antenna array, electronic steering is used in this work. With this approach and the resultant changing antenna gain pattern shape, it is infeasible to use a lookup table as it would require a separate one for each possible steering angle. Therefore, the calculated mathematical factor approach detailed below is used instead. However, mechanical steering can be

simply implemented with a single gain pattern lookup table that can be rotated. This would also allow for the implementation of any possible antenna gain pattern by the user. Lastly, it has been assumed that a single average antenna gain pattern for each frequency band is sufficient for the modelling purposes in this work. However, any number of patterns and frequency bands can be implemented using the same approach.

Due to the fact that grating lobes occur for element spacing that is greater than a half wavelength [32], the array must be designed for operation at the highest frequency band considered in this system due to its shortest wavelength. As such, using the highest operational frequency band (X-band) as the design frequency (10 GHz), the element spacing is calculated as:

$$d = \frac{\lambda}{2} = \frac{c}{2f} = 0.015 \text{ m}. \tag{3.36}$$

Further, an assumption of a 10 element horizontal array is made.

A two dimensional antenna gain pattern can then be calculated using the normalised power pattern of a uniformly-excited antenna array situated along the y-axis as [32]

$$\mathrm{G}(\psi) = \left| \frac{\sin\left(\frac{N_a \psi}{2}\right)}{N_a \sin\left(\frac{\psi}{2}\right)} \right|^2, \tag{3.37}$$

where

$$\psi = \frac{2\pi f d}{c} \sin(\varphi), \tag{3.38}$$

where $N_a$ is the number of antenna elements, $d$ is the spacing between these elements, $c$ is the speed of light, $f$ is the frequency of the signal being transmitted, and $\psi$ is the wavenumber. Then, assuming antenna elements with a squared cosine elevation pattern, the three dimensional pattern can be approximated as:

$$A_{3d}(\varphi, \theta) = \cos^2(\theta) \cdot \mathrm{G}(\psi). \tag{3.39}$$

This takes into account this work's chosen convention that elevation is measured from the x-y plane, and as such the choice of a squared cosine pattern results in the broadside being situated in the horizontal plane.

Next, assuming that the antenna arrays are mounted in wing-mounted pods, the effect of the platform's body on the antenna gain pattern must be taken into account. Specifically, if it is assumed that active channel one is mounted on the left wing, then it would have reduced gain in

directions directly to the right of the aircraft, and vice versa. This is approximated by using a further squared cosine factor on all azimuth angles that are on the opposite side of the platform to the jamming channel such that

$$\mathrm{A}_{\mathrm{body}}(\varphi,\theta) = \cos^2(\varphi)\cdot\cos^2(\theta)\cdot \mathrm{G}(\psi), \tag{3.40}$$

for those angles. The effect of this is to gradually reduce the antenna gain on the opposite side of the platform's body, before culminating in a value of zero on the exact opposite side of the platform.

Further, as an extension to the assumption of wing mounted pods, it is assumed that the gain pattern of the front-mounted antenna array is only directed in front of the pod, with only a small leakage pattern emitted behind. As such, a second array would be installed at the back of the pod to direct jamming behind the platform, with only a small leakage pattern emitted towards the front of the platform. This is achieved in two parts. First, a 45° taper off of the antenna gain pattern is assumed and implemented using a further squared cosine factor such that

$$\mathrm{A}_{\mathrm{taper}}(\varphi,\theta) = \cos^2(2\varphi)\cdot\cos^2(\theta)\cdot \mathrm{G}(\psi), \tag{3.41}$$

for the corresponding angles. Then secondly by setting the gain pattern factor to the minimum antenna gain pattern value ($\tau$) for the remaining azimuth angles in the range of 90° to 270° for the forward facing array, where it is assumed that this array is used for all antenna steer directions that are in front of the platform, and vice versa. Specifically, this value corresponds to the logarithmic minimum antenna gain pattern value used to normalise the factor into a range of zero to one, as discussed later.

Lastly, the effects of electronic steering must be considered. The antenna gain pattern of the first channel's steered array ($\mathrm{G}'$) at a specific set of azimuth and elevation angles $\left(\varphi_n^k, \theta_n^k\right)$ in the $k^{\text{th}}$ time interval can be calculated as [32]

$$\mathrm{G}'\left(\psi_{n,b}^k\right) = \mathrm{G}\left(\psi_{n,b}^k - \psi_{L1,b}^k\right), \tag{3.42}$$

where $\psi_{n,b}^k$ is the wavenumber of the view direction, and $\psi_{L1,b}^k$ is the wavenumber of the look direction of the array. These are calculated as

$$\psi_{n,b}^k = \frac{2\pi f d}{c}\sin\left(\varphi_n^k\right), \tag{3.43}$$

$$\psi_{L1,b}^k = \frac{2\pi f d}{c}\sin\left(\varphi_{L1}^k\right), \tag{3.44}$$

where $\varphi_{L1}^{k}$ is the direction in which the array is steered in the time interval being examined, and $f$ is equal to the frequency value of the chosen frequency band. Similarly, the look-direction wavenumber for the second channel can be calculated as

$$\psi_{L2,b}^{k} = \frac{2\pi f d}{c}\sin\left(\varphi_{L2}^{k}\right). \tag{3.45}$$

Combining all of the above equations, the antenna gain pattern factor perceived by the $n^{\text{th}}$ threat in the $k^{\text{th}}$ time interval can then be calculated as

$$\mathrm{AG}_{1,b}\left(\varphi_n^k, \theta_n^k, \varphi_{L1}^k\right) = \begin{cases} \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L1,b}^k\right), & \text{if} \quad 0° \le \varphi_n^k \le 90° \\ \cos^2\left(2\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L1,b}^k\right), & \text{if} \quad 90° < \varphi_n^k \le 135° \\ \tau, & \text{if} \quad 135° < \varphi_n^k < 270° \\ \cos^2\left(\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L1,b}^k\right), & \text{if} \quad 270° \le \varphi_n^k < 360°, \end{cases} \tag{3.46}$$

for the first channel on the left-hand side, and

$$\mathrm{AG}_{2,b}(\varphi_n^k, \theta_n^k, \varphi_{L2}^k) = \begin{cases} \cos^2\left(\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L2,b}^k\right), & \text{if} \quad 0° < \varphi_n^k \le 90° \\ \tau, & \text{if} \quad 90° < \varphi_n^k < 225° \\ \cos^2\left(2\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L2,b}^k\right), & \text{if} \quad 225° \le \varphi_n^k < 270° \\ \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L2,b}^k\right), & \text{if} \quad 270° \le \varphi_n^k \le 360°, \end{cases} \tag{3.47}$$

for the second channel on the right-hand side, if the forward-facing array is used ($-90° \le \varphi_L^k \le 90°$). In the case of the backward-facing array ($180° < \varphi_L^k < 270°$), the limits on the azimuth angle are simply changed such that

$$\mathrm{AG}_{1,b}\left(\varphi_n^k, \theta_n^k, \varphi_{L1}^k\right) = \begin{cases} \tau, & \text{if} -90° < \varphi_n^k < 45° \\ \cos^2\left(2\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L1,b}^k\right), & \text{if} \quad 45° \le \varphi_n^k < 90° \\ \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L1,b}^k\right), & \text{if} \quad 90° \le \varphi_n^k \le 180° \\ \cos^2\left(\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L1,b}^k\right), & \text{if} \quad 180° < \varphi_n^k \le 270°, \end{cases} \tag{3.48}$$

for the first channel, and

$$\mathrm{AG}_{2,b}(\varphi_n^k, \theta_n^k, \varphi_{L2}^k) = \begin{cases} \tau, & \text{if} -45° < \varphi_n^k < 90° \\ \cos^2\left(\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L2,b}^k\right), & \text{if} \quad 90° \le \varphi_n^k < 180° \\ \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L2,b}^k\right), & \text{if} \quad 180° \le \varphi_n^k \le 270° \\ \cos^2\left(2\varphi_n^k\right) \cdot \cos^2\left(\theta_n^k\right) \cdot \mathrm{G}\left(\psi_{n,b}^k - \psi_{L2,b}^k\right), & \text{if} \quad 270° < \varphi_n^k \le 315°, \end{cases} \tag{3.49}$$

for the second. Then, as with the RCS of the platform, the normalised antenna gain pattern factor is calculated by converting the pattern to decibels and normalising it. This is achieved by first limiting the lowest value to a variable minimum of -43 dBsm so as to maintain reasonable resolution and distribution of the normalised factor. Thereafter, the magnitude of this minimum value plus one is

added to the gain pattern so as to move the factor into a positive, non-zero range before it is normalised using the maximum value across all frequency bands.

Note that as with the RCS factor, the minimum value used in the above normalisation process can be set by the user to any value that results in good resolution and distribution of the factor values. Again, in this work this value has been chosen so as to obtain a median value of 0.5 for the factor across all frequency bands. However, in this case the factor values examined by the statistic are only those of the region of interest, which ignores the leakage pattern behind the antenna array. As such, setting the steer angle to 0° on the left-hand channel, this area of interest is defined as all elevation angles in the azimuth angle range of -90° to 135°. The reason for this is that the very-low value leakage pattern forms a large portion of the pattern, and as such if it is used in the determination of this statistic, it tends to dominate the result. Importantly, this ensures that the antenna gain pattern factor values in the region of interest are well distributed in magnitude by ensuring that half of the factor values are in the lower half of possible values, and the other half are in the upper half of values. Further, the resultant mean value of 0.4877 also means that on average the antenna gain factor perceived by a threat in the region of interest will be around 0.5, with threats in the main beam of the antenna array perceiving a larger antenna gain, and vice versa. A comparison of the resultant mean and median values of the antenna gain pattern factors for each of the different frequency bands appears in Table 3.2, where it is seen that as with RCS, the average values of the lower frequency bands are greater than those of the higher bands. However, the overall factor values remain reasonably well-distributed.

**Table 3.2** Antenna gain factor mean and median values for each frequency band.

| Statistic | L Band | S Band | C Band | X Band | All Bands |
|---|---|---|---|---|---|
| **Mean** | 0.7298 | 0.5092 | 0.4138 | 0.2980 | 0.4877 |
| **Median** | 0.8049 | 0.5318 | 0.4289 | 0.2856 | 0.5013 |

The culmination of all of the above results in the non-steered antenna gain factor patterns for each frequency band shown as a function of azimuth angle, at an elevation angle of 0°, in Figures 3.11-3.14. Specifically, these gain patterns are for the first active channel that is mounted on the left-hand side of the platform, hence the platform body effect is visible on the right-hand side of the pattern. Further, it is also seen that since the forward-facing array is being used, only the leakage pattern is visible behind the platform. Lastly, the steered antenna gain factor patterns for the first

channel in the X band are shown in Figures 3.15 and 3.16 for a positive and a negative 45° look angle respectively in order to show the effects of antenna steering.

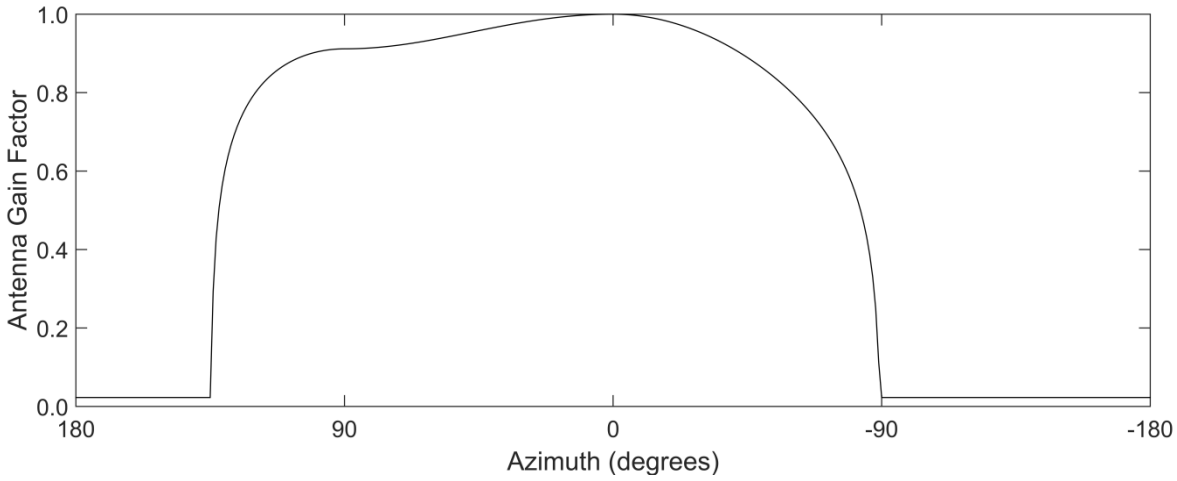



**Figure 3.11.** Antenna gain factor for the platform in the L band (1 GHz) when the left channel is aimed directly forward at an azimuth angle of 0°.

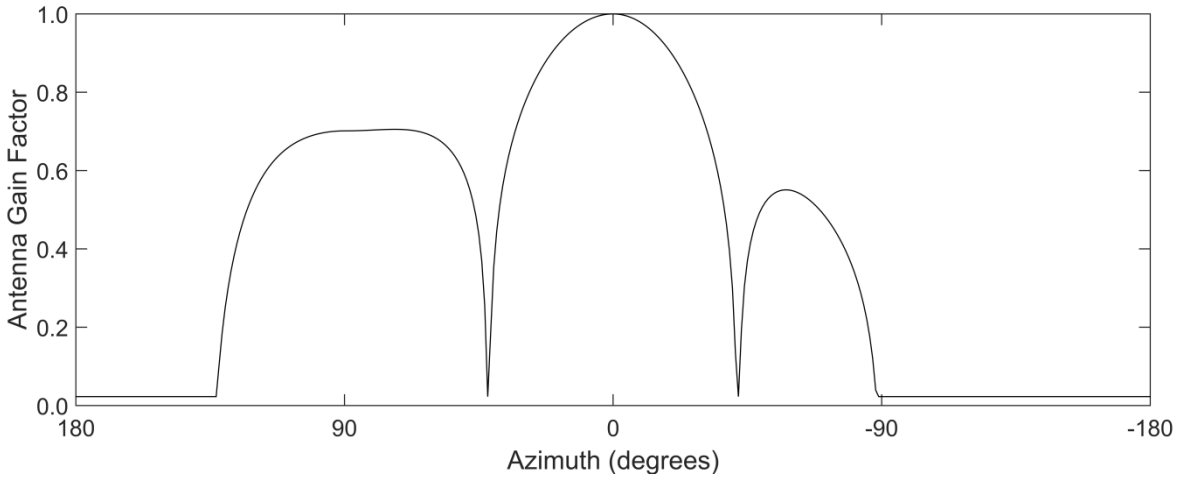



**Figure 3.12.** Antenna gain factor for the platform in the S band (3 GHz) when the left channel is aimed directly forward at an azimuth angle of 0°.

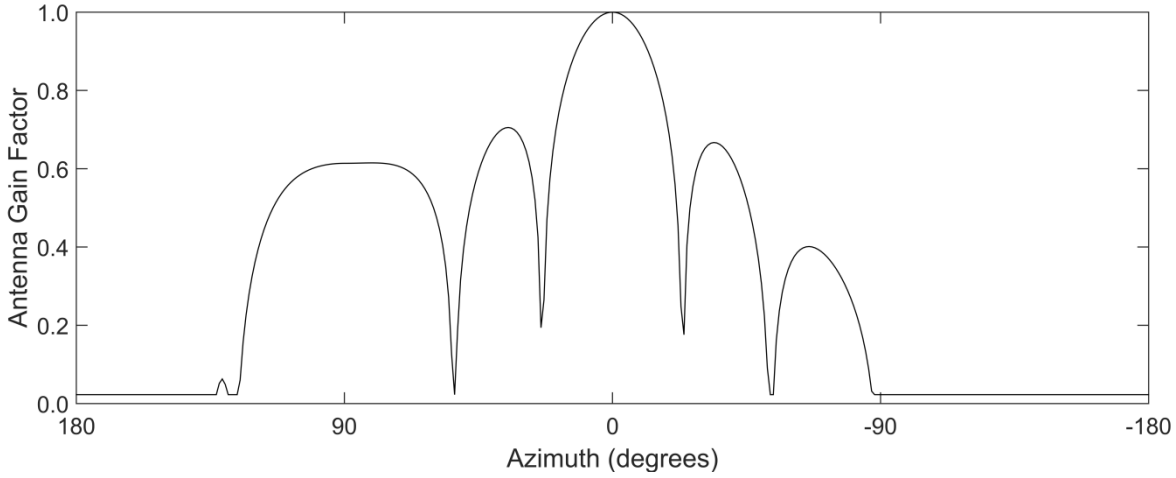



**Figure 3.13.** Antenna gain factor for the platform in the C band (5 GHz) when the left channel is aimed directly forward at an azimuth angle of 0°.

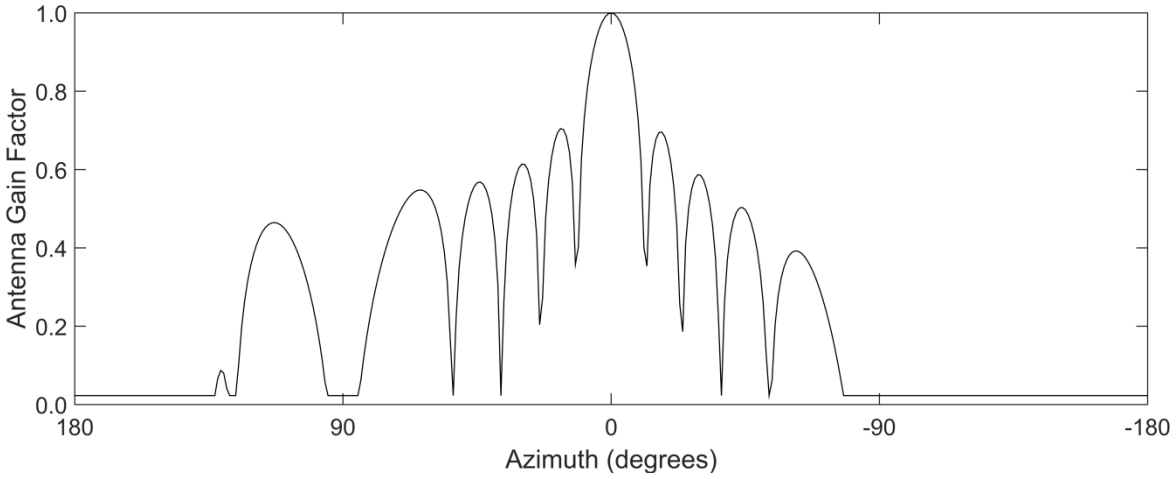



**Figure 3.14.** Antenna gain factor for the platform in the X band (10 GHz) when the left channel is aimed directly forward at an azimuth angle of 0°.

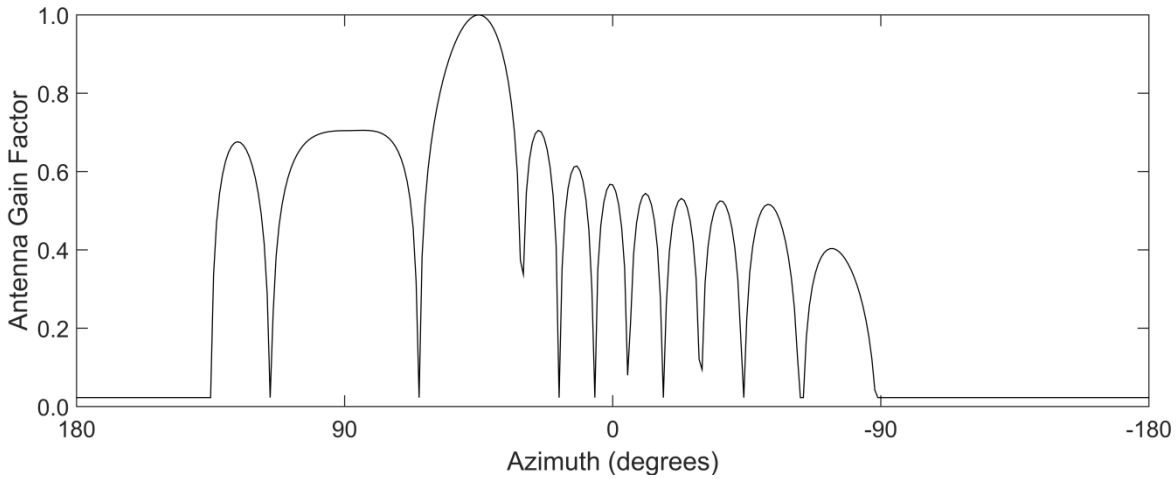



**Figure 3.15.** Antenna gain factor for the platform in the X band (10 GHz) when the left channel is aimed at an azimuth angle of 45°.

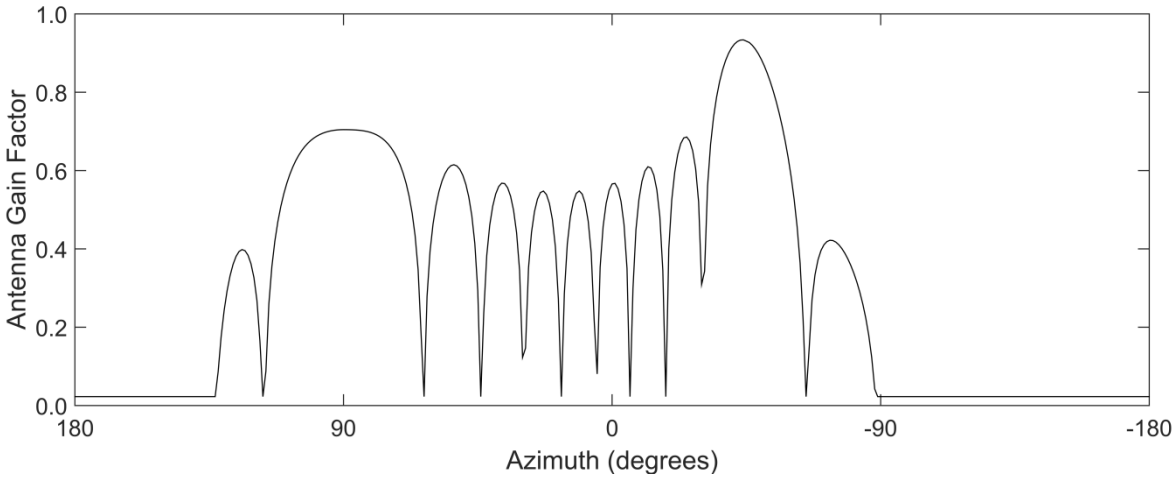



**Figure 3.16.** Antenna gain factor for the platform in the X band (10 GHz) when the left channel is aimed at an azimuth angle of -45° or 315°.

Note that it has been assumed that the antenna of the towed decoy is omnidirectional, so if an active channel is directed through it, the antenna gain pattern factor is simply set to 1. In the case of chaff illumination by active channels, the effect of the antenna gain pattern is accounted for by assuming that the chaff is situated directly behind the platform. Further, in the case of interaction between active channels, the effect of antenna gain patterns is taken into account along with the relative frequencies of operation to determine the relative signal level of the interfering signal

perceived by a threat, and hence the level of interference. These are discussed in greater detail in their relevant sections.

*2) Stage effectiveness*

This factor accounts for the effectiveness of an implemented countermeasure technique against a threat's radar stage. The stage effectiveness factor values used in this work appear as in Table 3.3 below, where a positive number indicates effectiveness against a radar stage, whilst a negative value indicates that the technique results in platform illumination that either aids detection or tracking of the platform for that radar stage.

**Table 3.3** Stage effectiveness factors (SE).

| Radar Stage | RGPO | VGPO | CP | NJ | MFT |
|---|---|---|---|---|---|
| **Search** | -1.0 | -1.0 | 1.0 | 0.8 | 1.0 |
| **Acquisition** | -1.0 | -1.0 | 1.0 | 0.9 | 1.0 |
| **Tracking** | 1.0 | 1.0 | -1.0 | -1.0 | 0.0 |
| **Guidance** | 1.0 | 1.2 | -1.0 | -1.0 | 0.0 |

RGPO and VGPO are techniques based on similar principles, and are specifically designed to effectively jam tracking-type radar stages. On the other hand, the large false targets generated by these techniques in the direct vicinity of the platform actually illuminate the platform to search-type radars. Further, VGPO has the advantage of being able to draw guided missiles towards stationary clutter when it is used in a decreasing velocity sense, and hence has a slightly increased effectiveness against these threats [15]. As such, these techniques are allocated strong negative factors against search-type stages, and strong positive factors against tracking-types stages, with a slightly larger factor for VGPO against guidance stages.

A cover pulse (CP) is a technique specifically aimed at fooling a search radar's CFAR detector, and hence is highly effective against search-type radars. However, the large cover pulse signal that is emitted from the platform can illuminate the platform to tracking-type stages, resulting in a strong negative factor for these threats.

Noise jamming (NJ) is effective at raising the noise-floor of the threat's radar, making the platform more difficult to detect for search-type radar stages. However, the large signal strength emitted can trigger track-on-jam counter-countermeasures, hence illuminating the platform to tracking-type

stages. Further, for the sake of demonstrating the capability, in this work it has been assumed that the noise jamming has been optimised for greater performance in jamming the acquisition phase.

Finally, multiple false targets (MFT) are effective at over-loading a search-type radar, making it difficult for it to discern the platform from the false targets, whilst this has no effect on a tracking-type radar already in the process of tracking the platform.

3*) Technique interaction*

This factor is used to account for the fact that some combinations of techniques can interact in a positive way, enhancing each other's performance, whilst others can have no effect, or even a detrimental one. Note that interactions between the channels themselves are accounted for in the combining of their jamming effects in (3.31). Instead, this factor specifically focuses on the interactions of the techniques themselves where, for example, an interfering RGPO technique can assist the adversary radar in singling out the platform when it is surrounded by multiple false targets, hence reducing the effectiveness of that MFT technique. It is calculated as the effect of channel 2 on channel 1 using

$$I_{1,2}^{k} = 1 + \left[\mathrm{IN}(J_1^k, J_2^k) \cdot \mathrm{CE}(O_2^k, Y_n) \cdot \mathrm{AG}_{2,\mathrm{b}}(\varphi_n^k, \theta_n^k, \varphi_{L2}^k)\right], \tag{3.50}$$

and similarly as the effect of channel 1 on channel 2 using

$$I_{2,1}^{k} = 1 + \left[\mathrm{IN}(J_2^k, J_1^k) \cdot \mathrm{CE}(O_1^k, Y_n) \cdot \mathrm{AG}_{1,\mathrm{b}}(\varphi_n^k, \theta_n^k, \varphi_{L1}^k)\right]. \tag{3.51}$$

In these equations, IN is the interaction factor that appears in Table 3.4, CE is the cross effect factor that is discussed in the next section, and AG is the antenna gain pattern factor discussed above. As such, the interference factor is dependent on the techniques being used, the relative frequencies bands they are being used in, and the relative signal level of the interfering technique as perceived by the examined threat due to the steered antenna array pattern. Therefore, if the interfering technique is implemented in a different part of the frequency spectrum to what is used by the threat, or the interfering antenna is steered in such a way that almost no signal reaches the threat being examined, then no interference will occur, and vice versa. Further, note that in Table 3.4 a value of zero indicates no interaction effect, a positive value indicates an enhancing effect, and a negative value indicates a detrimental interaction between techniques. Lastly, in this table the rows represent the technique being examined, and the columns are the interfering technique.

**Table 3.4** Jamming interaction factors (IN).

| Technique | RGPO | VGPO | CP | NJ | MFT |
|---|---|---|---|---|---|
| **RGPO** | 0.0 | 0.2 | -0.3 | -0.2 | 0.0 |
| **VGPO** | 0.2 | 0.0 | -0.3 | -0.2 | 0.0 |
| **CP** | -0.2 | -0.2 | 0.0 | 0.2 | 0.2 |
| **NJ** | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 |
| **MFT** | -0.3 | -0.3 | 0.3 | 0.0 | 0.0 |

Beginning with RGPO as the technique being examined, the jamming interaction factors can be explained as follows. First, VGPO is set to have an enhancing effect due to the similarities between these two techniques, where the likelihood of a tracking-type radar following one of the two techniques is greater than that of the radar being led away by just one. Next, an interfering cover pulse is set to have a negative effect. This is because the cover pulse's large false target in the vicinity of the platform can mask that of the RGPO, hence reducing the likelihood of it being detected and followed. Further, depending on its characteristics, the large cover pulse may require greater separation between the platform and the RGPO false target before break-lock can occur. Similarly, noise jamming is set to negatively interfere with RGPO as it also reduces the likelihood that the tracking-type radar will detect and follow the RGPO false target. Lastly, MFT is set to have zero interference with RGPO. This is because creating false targets around the platform once it is already being tracked will have no effect on the performance of RGPO. Note that due to the similarity of the two techniques, the IN factors for VGPO have been set equal to those of RGPO for the same reasons.

Next, considering a cover pulse as the technique being examined, the IN factors can be explained as follows. Both RGPO and VGPO have been set to have a negative interference effect. This is because when a cover pulse is being used to jam a search-type radar, a RGPO or VGPO technique will generate a false target in the immediate vicinity of the platform, which in turn will disturb the carefully shaped cover pulse that hides the platform, thus reducing its effectiveness. Thereafter, noise jamming is set to have a positive interference effect with a cover pulse because they utilise similar principles, with both raising the detection threshold of a CFAR detector. Lastly, MFT is set to have an enhancing effect due to the fact that the multiple false targets will distract the adversary radar from the platform hidden under the cover pulse.

In the case of noise jamming being the examined technique, all techniques are set to have no interference, except a cover pulse. This is due to the fact that, as stated previously, noise jamming

and cover pulses utilise similar principles, with both raising the detection threshold of a CFAR detector. On the other hand, the remaining techniques do nothing to interact with noise jamming's action of raising the noise floor. They instead operate independently from this technique.

Lastly, when considering a MFT technique, both RGPO and VGPO are set to have a negative interference effect. This is because the moving false targets in the immediate vicinity of the platform associated with these two techniques will single out which is the real platform. Next, a cover pulse is set to have a positive effect due to the fact that this technique will help hide the platform from adversary radar while attention is focused on the multiple false targets. Lastly, noise jamming is set to have no interference with MFT because the noise will not increase the likelihood of the adversary radar being distracted by the false targets.

*4) Cross effect*

This factor accounts for the relative frequency-band use of the different threat systems and hence the associated levels of interaction between them where, as stated previously, the threat-type numbers are allocated in ascending frequency-band-usage order. It is implemented using a lookup table that can be modified by the user to account for any combination of frequency band usage and bandwidth, and would, as with the other lookup tables in this work, be populated using low-level physics- or parameter-based simulators. Threat types can be set to have a relatively wider bandwidth by causing countermeasures optimised for them to have a larger effect on a greater number of neighbouring threat types or vice versa. The lookup table used for the ten threat types in this work appears in Table 3.5, where the rows represent the threat type for which the jamming technique is optimised (*O*), and the columns represent the threat type of the threat currently being examined (*Y*).

**Table 3.5** Cross effect factors (CE).

| | **T1** | **T2** | **T3** | **T4** | **T5** | **T6** | **T7** | **T8** | **T9** | **T10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **T1** | 1.0 | 0.4 | 0.2 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **T2** | 0.4 | 1.0 | 0.4 | 0.4 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **T3** | 0.2 | 0.4 | 1.0 | 1.0 | 0.4 | 0.2 | 0.2 | 0.0 | 0.0 | 0.0 |
| **T4** | 0.2 | 0.4 | 1.0 | 1.0 | 0.4 | 0.2 | 0.2 | 0.0 | 0.0 | 0.0 |
| **T5** | 0.0 | 0.2 | 0.4 | 0.4 | 1.0 | 0.4 | 0.4 | 0.0 | 0.0 | 0.0 |
| **T6** | 0.0 | 0.0 | 0.2 | 0.2 | 0.4 | 1.0 | 0.6 | 0.2 | 0.0 | 0.0 |
| **T7** | 0.0 | 0.0 | 0.2 | 0.2 | 0.4 | 0.6 | 1.0 | 0.6 | 0.4 | 0.2 |
| **T8** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 0.6 | 1.0 | 0.4 | 0.2 |
| **T9** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.4 | 0.4 | 1.0 | 0.4 |
| **T10** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 0.2 | 0.4 | 1.0 |

In this work it has been assumed that almost all the threat types use an equally spaced, similar bandwidth such that there is only a slight over-lap. However, for capability demonstration purposes there are three exceptions: threat types 3, 4 and 7. Threat types 3 and 4 are set to operate in the same frequency band. On the other hand, threat type 7 is set to operate in a wider band than the other threats by increasing both the number of threat types its optimised countermeasures affect, and the number of optimised countermeasures that affect it. For the general threats, their bandwidth over-lap has been chosen such that if an adjacent threat type is jammed, the jamming only has 40% effectiveness against the threat being examined and 20% effectiveness on a threat-type one type further away. On the other hand in the case of threat type 7, if an adjacent threat type is jammed, the jamming is 60% effective against it. Thereafter jamming one threat type further away results in 40% effectiveness, and then yet another threat type further away results in 20% effectiveness. Note that in Table 3.5 threat types 3 and 4 may appear at a glance to have a wider band of operation than the majority of threat types, but since their frequency bands are the same as each other, their optimised countermeasures both only affect 5 other frequency bands. This is also true of the remaining threat types in that they may appear to have different bandwidths, but it must be kept in mind that the T3 and T4 rows and columns are essentially treated as one.

The cross effect and associated Table 3.5 are implemented as above for all RF jamming techniques, except for noise jamming. This is due to the fact that the implemented system has 5 different noise jamming bandwidths available to it: narrow (N), medium-narrow (MN), medium (M), medium-wide (MW) and wide (W) bandwidth. This accounts for the limited maximum power output of jamming systems, where noise jamming can be implemented as a powerful narrowband technique with a strong effect on one threat, or it can be implemented as a weaker wideband technique with less of an effect over a greater number of threats.

As a result, a separate lookup table must be used for the cross effect for noise jamming, that allows for the implementation of each of these different bandwidths (B/W). Further, a separate table must be defined for each individual threat type in order to account for their bandwidth, and resultant level of overlap with their neighbouring threat types. This is achieved with Table 3.6 for all threat types except type 7, and Table 3.7 for that remaining type. This is due to the fact that, as discussed above, types 1-6 and 8-10 all use an equally spaced, similar bandwidth, whilst threat type 7 is set to operate in a wider bandwidth. However, different tables can be defined for each threat type if needed. In these tables the rows contain the different bandwidths of noise jamming, whilst the

columns contain the threat types relative to the threat type being considered (*Y*). The centre column (C) represents the actual threat type being considered, whilst the increasing H columns to the right represent the threat types in higher frequency bands, and the L columns the ones in lower frequency bands. The cross-effect value used in each instance is determined by the bandwidth of noise jamming used in the time interval, along with the difference between the threat type for which the jamming is optimised, and the threat type being considered. This process is best explained with an example. If the cross effect value for an examined threat type 6 is needed for a time interval in which a medium-narrow noise jamming technique optimised for threat type 4 is used, the difference is calculated as -2. Then the cross effect value can be read from the L2 column and MN row of Table 3.6 as 0.32. However, if the examined threat was of type 7, then the difference would be calculated as -3, and the cross effect value read from the L3 column and MN row of Table 3.7 as 0.32 as well, due to its greater bandwidth. Lastly and importantly, if the threat type being examined was of type 2, then the difference would only be calculated as 1. This is due to the fact that threat types 3 and 4 occupy the same frequency band, and hence type 2 is adjacent to type 4. As such, the cross-effect value would be read from the H1 column and the MN row in table 3.4 as 0.8.

**Table 3.6** Noise jamming cross effect for examined threat types (*Y*) 1-6 and 8-10.

| B/W | L7 | L6 | L5 | L4 | L3 | L2 | L1 | C | H1 | H2 | H3 | H4 | H5 | H6 | H7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **N** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.40 | 1.00 | 0.40 | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **MN** | 0.00 | 0.00 | 0.00 | 0.00 | 0.16 | 0.32 | 0.80 | 0.80 | 0.80 | 0.32 | 0.16 | 0.00 | 0.00 | 0.00 | 0.00 |
| **M** | 0.00 | 0.00 | 0.00 | 0.12 | 0.24 | 0.60 | 0.60 | 0.60 | 0.60 | 0.60 | 0.24 | 0.12 | 0.00 | 0.00 | 0.00 |
| **MW** | 0.00 | 0.00 | 0.08 | 0.16 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.16 | 0.08 | 0.00 | 0.00 |
| **W** | 0.00 | 0.04 | 0.08 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.08 | 0.04 | 0.00 |

**Table 3.7** Noise jamming cross effect for examined threat type (*Y*) 7.

| B/W | L7 | L6 | L5 | L4 | L3 | L2 | L1 | C | H1 | H2 | H3 | H4 | H5 | H6 | H7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **N** | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.40 | 0.60 | 1.00 | 0.60 | 0.40 | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 |
| **MN** | 0.00 | 0.00 | 0.00 | 0.16 | 0.32 | 0.48 | 0.80 | 0.80 | 0.80 | 0.48 | 0.32 | 0.16 | 0.00 | 0.00 | 0.00 |
| **M** | 0.00 | 0.00 | 0.12 | 0.24 | 0.36 | 0.60 | 0.60 | 0.60 | 0.60 | 0.60 | 0.36 | 0.24 | 0.12 | 0.00 | 0.00 |
| **MW** | 0.00 | 0.08 | 0.16 | 0.24 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.24 | 0.16 | 0.08 | 0.00 |
| **W** | 0.04 | 0.08 | 0.12 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.12 | 0.08 | 0.04 |

Essentially, the noise jamming cross effect values have been chosen so that the narrow bandwidth has 100% effectiveness on one threat type, the medium-narrow bandwidth has 80% effectiveness spread over 3 adjacent threat types, the medium bandwidth has 60% effectiveness spread over 5 adjacent threat types, the medium-wide has 40% effectiveness over 7 adjacent threat types, and the wide bandwidth has 20% effectiveness spread over 9 adjacent threat types. Next, the overlapping

region values were chosen so as to emulate those in Table 3.5 as can be seen by the fact that the narrow bandwidth cross effect values in Tables 3.6 and 3.7 emulate those of the general threats, and those of threat type 7 respectively. Thereafter, these overlapping region values were scaled for each of the different bandwidths according to their respective 80%, 60%, 40%, and 20% maximum effect. Again it must be noted that these arrays can be modified by a user to implement any combination of noise jamming schemes and bandwidth distributions.

*5) Chaff interference*

The chaff interference factor accounts for the interference effects of different chaff techniques on the various active jamming techniques available to the platform. As with the technique interaction factor (*I*), this specifically accounts for the interactions between the techniques themselves, where the interaction between channels has been accounted for in the combining of jamming effect values in (3.30). Further, since the chaff cartridges are assumed to contain a number of dipole lengths, the interference of these countermeasures is independent of frequency. Also, since the directionality of fired passive countermeasures is outside the scope of this work, their interference is assumed to be omnidirectional. As such, these factors could be ignored and the chaff interference factor simply implemented as a single lookup table with a zero-effect value of one so as to be equivalent to (3.50) and (3.51).

The factor values for various combinations of techniques appear in Table 3.8, where 1.0 indicates zero effect and a greater value indicates enhancement. Due to the similarity of distraction chaff's action to that of multiple false targets, its values were chosen so as to emulate that technique's jamming interaction factors (IN) in Table 3.4 for the same reasons as discussed there. On the other hand, the values chosen for dilution chaff were chosen so as to emulate the jamming interaction factors of RGPO due to their similarity. Note that in this table, the active techniques in the rows are the examined technique, whilst the passive techniques in the columns are the interfering ones. As such, the values were taken from Table 3.4 accordingly. As a result, distraction chaff is enhanced by cover pulses. On the other hand, dilution chaff is enhanced by RGPO and VGPO, whilst it is negatively interfered with by cover pulses and MFT. Lastly, note that this factor is set to 1.0 when no chaff is used in the time interval being examined so as cause zero effect to the jamming effect.

**Table 3.8** Chaff interference factors (CI).

| Technique | Distraction | Dilution |
|---|---|---|
| **RGPO** | 1.0 | 1.2 |
| **VGPO** | 1.0 | 1.2 |
| **CP** | 1.2 | 0.8 |
| **NJ** | 1.0 | 1.0 |
| **MFT** | 1.0 | 0.7 |

*6) Countermeasure resistance*

Lastly, this factor (CR) takes into account the resistance of the examined threat to the effects of countermeasures due to its counter-countermeasure capabilities. It is set for each individual threat type and is defined such that a factor of 0.2 indicates a 20% reduction in jamming effectiveness. Further, it is modified according to the sign of the relevant stage effectiveness factor in order to generate $\rho_{n,1}^{k}$ and $\rho_{n,2}^{k}$ in (3.34) and (3.35). This is because improved counter-countermeasures reduce the impact of effective jamming, but not that of illumination. Instead, the negative effect of platform illumination is enhanced by counter-countermeasures such as track-on-jam capabilities, and hence must be modified accordingly. Therefore in these equations, CR is subtracted from one in cases of effective jamming, and added to one in cases of illumination in order to either increase or decrease the jamming effect by the appropriate percentage.

Any values in the range of 0 to 1 can be used for this factor, but factors that are too large in value can render the platform defenceless against that threat type, and even prevent the breaking of its lock. As such, in this work factors in the range of 0.0 to 0.3 are used, where the latter indicates a more advanced and modern threat type, whereas the former indicates an older threat type with little to no counter-countermeasure capabilities. Since the values used are directly dependent on the threats encountered in a specific scenario, the specific values used in this work are detailed in the results section along with the other threat characteristics.

### 3.4.2 Chaff

The jamming effect of chaff is calculated in a similar manner to that of the active channels in that it is calculated as product of factors that account for: the effectiveness of the chosen chaff technique on the examined threat's radar stage (CS), chaff illumination by active jamming techniques ($U_1$ and $U_2$), and the cross effect of seducing a threat with a technique optimised for another threat (בּ). This is calculated as

$$E_{n,3}^{k} = \mathrm{CS}\left(S_n^k, J_3^k\right) \cdot \beth_n^k \cdot U_1^k \cdot U_2^k, \tag{3.52}$$

where

$$\beth_n^k = \begin{cases} 1, & \text{if } \mathrm{CS}\left(S_n^k, J_3^k\right) \geq 0 \text{ and } O_3^k = Y_n \\ 1 - \mathrm{CR}(Y_n), & \text{if } \mathrm{CS}\left(S_n^k, J_3^k\right) \geq 0 \text{ and } O_3^k \neq Y_n \\ 1 + \mathrm{CR}(Y_n), & \text{if } \mathrm{CS}\left(S_n^k, J_3^k\right) < 0, \end{cases} \tag{3.53}$$

$$U_1^k = 1 + \left[U_{max} \cdot \mathrm{RR}(J_1^k, J_3^k) \cdot \mathrm{CE}(O_1^k, Y_n) \cdot \mathrm{AG}_{1,b}(\varepsilon, \mu, \varphi_{L1}^k) \cdot \Omega_{n,1}^k\right], \tag{3.54}$$

$$U_2^k = 1 + \left[U_{max} \cdot \mathrm{RR}(J_2^k, J_3^k) \cdot \mathrm{CE}(O_2^k, Y_n) \cdot \mathrm{AG}_{2,b}(\varepsilon, \mu, \varphi_{L2}^k) \cdot \Omega_{n,2}^k\right], \tag{3.55}$$

$$\Omega_{n,1}^k = \begin{cases} 1 - \mathrm{CR}(Y_n), & \text{if } \mathrm{RR}\left(J_1^k, J_3^k\right) \geq 0 \\ 1 + \mathrm{CR}(Y_n), & \text{if } \mathrm{RR}(J_1^k, J_3^k) < 0, \end{cases} \tag{3.56}$$

$$\Omega_{n,2}^k = \begin{cases} 1 - \mathrm{CR}(Y_n), & \text{if } \mathrm{RR}\left(J_2^k, J_3^k\right) \geq 0 \\ 1 + \mathrm{CR}(Y_n), & \text{if } \mathrm{RR}(J_2^k, J_3^k) < 0. \end{cases} \tag{3.57}$$

As before $J_1^k$, $J_2^k$, and $J_3^k$ are the three countermeasure techniques implemented in the time interval for the first active channel, second active channel, and the passive channel respectively. $O_1^k$, $O_2^k$, and $O_3^k$ are the threat types for which the countermeasure techniques are optimised. $S_n^k$ is the radar stage of the examined threat, and $Y_n$ its threat type. $U_{max}$ is the maximum-chaff-illumination-effect factor, RR is the chaff-illumination technique factor, and CE is the cross effect factor. $\mathrm{AG}_{1,b}$ and $\mathrm{AG}_{2,b}$ are the antenna gain pattern factors of the first and second active channels in the frequency band of operation of the threat being examined, $b$. Next, $\varepsilon$ and $\mu$ are the azimuth and elevation angles of the chaff cloud relative to the platform, and $\varphi_{L1}^k$ and $\varphi_{L2}^k$ are the steering angles of the first and second channel antennas respectively. Lastly, $\Omega_{n,1}^k$ and $\Omega_{n,2}^k$ are the sign-adjusted countermeasure resistance (CR) factors.

CS is a factor that accounts for the effectiveness of the chosen chaff technique against the radar stage of the threat currently being examined. It is implemented using the user-defined lookup table that appears in Table 3.9, where positive values indicate effectiveness against a particular radar stage, and negative values indicate illumination of the platform. As stated previously, distraction chaff works in a very similar way to multiple false targets in that the false targets generated are used to distract and overwhelm search-type radar stages. As such, this technique is also set to have

a positive effect against these radar stages, and no effect against tracking-type stages. However, distraction chaff's effectiveness has been set lower for search stages so as to discourage the use and wasting of chaff cartridges in lower-danger situations. On the other hand, dilution chaff is set to illuminate the platform to search-type stages and to be effective against tracking-type stages. This is due to the similarity of action between this technique and that of RGPO in that they both attempt to draw a tracking radar away from the platform, hence illuminating the platform to search-type radars by creating a false target in the immediate vicinity of the platform.

**Table 3.9** Chaff stage effectiveness factors (CS).

| Radar Stage | Distraction | Dilution |
|---|---|---|
| **Search** | 0.8 | -1.0 |
| **Acquisition** | 1.0 | -1.0 |
| **Tracking** | 0.0 | 1.0 |
| **Guidance** | 0.0 | 1.0 |

The chaff cross-effect factor (ב) is used to account for the effectiveness of a chaff strategy on a threat type for which it is not optimised. It is assumed that the chaff cartridges consist of a number of different dipole lengths, hence it is also assumed that each cartridge is equally effective against all threat radar frequencies. Therefore, any differences in effectiveness are solely due to the size and timing of salvos. As such, in the case of effective threat seduction, the value of this factor must be 1.0 for the threat type for which the chaff strategy is optimised. Thereafter, its value is dependent on the counter-countermeasures of the threat being examined, and hence its chaff rejection capabilities. This can be explained using two extreme examples: an advanced guided missile with high countermeasure resistance, and a basic guided missile with no countermeasure resistance. In the case of non-optimal chaff countermeasures being dispensed, the basic missile would be unable to distinguish between the chaff and the platform, and hence still be successfully seduced away from the platform. On the other hand, the more advanced missile would have an increased capability to distinguish between the chaff and the platform, especially since the chaff dispensing pattern is not optimised for that threat and its associated range and velocity bins. This in turn then results in a reduced seduction effect. Therefore in such cases where the threat type does not match the type for which the chaff is optimised, ב is dependent on that threat's countermeasure resistance (CR) factor due the fact that this factor is used to account for the threats counter-countermeasure capabilities.

Further, it is noted that in the case of chaff illumination of the platform caused by a negative stage effectiveness factor, the ב factor is calculated in (3.53) in such a way that a larger countermeasure resistance enhances this illuminating effect. Importantly, in contrast to effective threat seduction, this effect is independent of the threat type for which the countermeasure was optimised. This is because the chaff dispersion pattern is optimised for a specific effect against a particular radar stage of a threat type. As such, illuminating chaff patterns were not optimised in their illuminating capacity, and hence do not have an increased effect on the threat type for which they were optimised (*O*). Instead, their effect is solely dependent on the counter-countermeasures, or countermeasure resistance (CR), of the threat being examined (*Y*). This can be further explained by means of an example: in the case of determining the effect of dilution chaff against a search-type radar stage, the greater the ability of a threat to discern the platform from the chaff, the greater the illuminating effect, and hence greater the rate at which that threat can detect the platform, and vice versa.

$U_1$ and $U_2$ are factors that account for the effect of the chaff being illuminated by the first and second active jamming channels, and the interference that results. These factors are defined such that detrimental inference results in a value of less than one, whilst positive interference results in a value greater than one. In both cases, these factors are scaled by the user-defined maximum effect of chaff illumination, $U_{max}$, which has been set to 2. Considering an instance where all factors are aligned, this results in an absolute maximum change in chaff performance of 60%, where most of the time in non-ideal circumstances this change will be less. RR is a factor that accounts for the interaction between different techniques in the illumination of chaff by active jamming. This value is obtained from the lookup table that appears in Table 3.10, where negative values indicate detrimental interference, whilst positive values indicate enhancement. These values were chosen to so as to mirror the effects and interactions contained in Table 3.4 for the same reasons as discussed there. CE is the same cross effect factor that appears in Table 3.5, and accounts for the frequency-band usage of threat radar systems. Further, $\Omega_{n,1}^{k}$ and $\Omega_{n,2}^{k}$ are the sign-adjusted countermeasure resistance (CR) factors that have been adjusted for the first and second channels, respectively. These account for counter-countermeasure capabilities of the threat type being examined, and are the same as the $\rho$ factors used for the active techniques, where they have been sign-adjusted so that effects of illumination are enhanced, and those of effective threat seduction are reduced accordingly. This is for the same reasons as discussed above. Lastly, it is noted that this approach

in general allows for the implementation of specific illuminated chaff techniques through the use of large enhancement factors, and aiming the active channel antennas directly at the chaff.

**Table 3.10** Chaff illumination technique factors (RR).

| Technique | Distraction | Dilution |
|---|---|---|
| **RGPO** | -0.3 | 0.2 |
| **VGPO** | -0.3 | 0.2 |
| **CP** | 0.3 | -0.3 |
| **NJ** | 0.0 | -0.2 |
| **MFT** | 0.0 | 0.0 |

Lastly, within the chaff illumination factors, AG is the normalised antenna gain pattern factor of the active channels, and is the same factor as discussed in Section 3.4.1.1. It accounts for the relative signal level illuminating the chaff from each active channel due to their frequency-band specific antenna gain patterns, along with the effect of the steer directions of the antennas in the time interval being examined. Therefore in this case, because the factor is used to determine the level of chaff illumination, the position of the chaff cloud relative to the platform must be considered for these factors rather than the position of the threat itself. As such, for the purposes of this work it is assumed that the chaff is situated directly behind the platform with an azimuth angle of 180°, and an elevation angle of 0°. As a result, $\varepsilon$ and $\mu$ are set to 180° and 0°, respectively.

### 3.4.3 Flares

The jamming factors of flare countermeasures are relatively simply calculated in comparison to those of RF countermeasures. This is because in this work flares do not experience interference as they are the only implemented IR countermeasure technique, and further there is no need to take into account stage effectiveness as IR threats are not detected or countered until they are in the guidance stage. Also, the assumption that a cocktail of different flare types is used means that the effects of these countermeasures are not dependent on frequency bands either. Instead the effect is considered to be dependent on the guidance system of the specific threat and its associated level of flare rejection capabilities or countermeasure resistance.

Therefore, the jamming effect ($E$) of a flare technique on the $m^{th}$ IR threat can be calculated similarly to the jamming effect of chaff without the effects of interference and stage effectiveness as:

$$E_{m,IR}^{k} = \begin{cases} 1, & \text{if } O_3^k = Y_m \\ 1 - \mathrm{CR}(Y_m), & \text{if } O_3^k \neq Y_m, \end{cases} \tag{3.58}$$

where $O_3^k$ is the threat type for which the flare strategy is optimised, $Y_m$ is the threat type of the threat being examined, and CR is the countermeasure resistance of that threat. Essentially, similarly to chaff countermeasures, the jamming effect of a flare technique is set to the maximum possible for a threat type for which it is optimised. This is because it has been assumed that the passive countermeasure strategies for each threat type have individually been optimised as a whole in other low-level simulators and include all necessary manoeuvres, and hence are assumed to be fully effective at seducing that threat type. Then for threat types for which the flare technique is not optimised, effectiveness is calculated according to the countermeasure resistance (CR) factor of the threat type being examined. This is for the same reasons as discussed for chaff countermeasures, where threats with worse countermeasure resistance and hence less flare rejection capabilities, are less likely to differentiate between the flares and the platform. On the other hand, more advanced threats with greater flare rejection capabilities require more tailored strategies in order to have the same effect. The net result of this is that threats with minimal to no flare rejection capabilities can be successfully seduced using non-optimal techniques, whereas more advanced threats with such capabilities will be affected less. As with the RF CR factors, the specific factor values for each threat type will be defined along with the other threat characteristics in the results section where the example scenario is detailed. Lastly, it is noted that unlike chaff countermeasures, an exception does not need to be made in this factor for instances of illumination. This is because, as stated above, the passive nature of these threats means that they are only dealt with in their guidance stage with techniques specifically designed for this stage. This in turn means no illumination occurs for these threats and hence does not need to be considered.

Thereafter, similarly to RF, these effectiveness values are then multiplied by a user-defined maximum flare effect ($E_{max,IR}$) so as to prevent countermeasures from reducing a threat's post-jamming danger value to zero. This is important due to the fact that any functioning threat within range must always present at least some non-zero danger to the platform. As such, in this work this factor is currently set to 0.9, and hence IR guidance systems cannot be reduced below 10% effectiveness. Finally, the multiplicative jamming factor ($F_{m,IR}^{k}$) is then set equal to one minus this value and used to adjust the danger value ($D_{m,IR}^{k}$) of the threat accordingly, resulting in a post-jamming danger value of

$$V_{m,IR}^{k} = F_{m,IR}^{k} \cdot D_{m,IR}^{k} = \left[1 - \left(E_{max,IR} \cdot E_{m,IR}^{k}\right)\right] \cdot D_{m,IR}^{k}. \tag{3.59}$$

It is noted that the angle of approach of the IR missiles in relation to one another, and to the platform and its flares, plays a major role in determining the interaction between flare countermeasure strategies and threat missiles. However, the modelling of these complex interactions in a computationally efficient manner is outside the scope of this work. As a result, the simplified model outlined above is used here.

### 3.4.4 Decoys

A single fibre-optic towed decoy was implemented by treating it as an optional multiplicative modifier that can be applied to the channel effect ($E$) of either the first active jamming channel or the second, but not both. This means that the decoy acts as another antenna from which the central ECM system can broadcast the signal from one of the jamming channels.

The user-defined modifier is dependent on both the radar stage of the threat being examined and the technique being directed through the decoy, where the values for each combination appear in Table 3.11. Essentially, the use of the decoy enhances the jamming effect of RGPO and VGPO against tracking-type stages by 20%, whereas it reduces the jamming effect in all other cases by 20%. In the former case, this is due to the fact that the separation between the decoy and platform enhances the effectiveness of techniques that target the angular accuracy of threats. Whereas in the case of effective jamming of search-type stages (CP, NJ, or MFT), the use of the decoy essentially spreads the platform out, making it a larger target to detect and thus reducing countermeasure effectiveness. Finally, in the case of platform illumination, the use of the towed decoy causes the location of the decoy to be illuminated rather than the actual platform, hence reducing the illumination effect.

**Table 3.11** Decoy multiplication factors.

| Technique | Search | Acquisition | Tracking | Guidance |
|---|---|---|---|---|
| **RGPO** | 0.8 | 0.8 | 1.2 | 1.2 |
| **VGPO** | 0.8 | 0.8 | 1.2 | 1.2 |
| **CP** | 0.8 | 0.8 | 0.8 | 0.8 |
| **NJ** | 0.8 | 0.8 | 0.8 | 0.8 |
| **MFT** | 0.8 | 0.8 | 0.8 | 0.8 |

Lastly, it is assumed for simplicity that the antenna gain pattern of the decoy is omnidirectional, and that due to the separation between it and the platform, the effects of the body of the platform are negligible. Further, the antenna gain pattern factor of this omnidirectional antenna should not be

equal to that of the main-lobe gain of the directional antennas onboard the platform. As such, the AG factor for an active jamming channel that is directed through the decoy is always set to a user-defined value of 0.9 for all threats. In this work the value of 0.9, has been chosen so as to be approximately equal to the AG factor of the 90° view angle of the onboard antenna in the L band due to its almost omnidirectionality. As a result of the values used in the normalisation of the antenna gain pattern factor, this value is representative of a 4.5 dB drop in gain verses the main beam of the onboard antennas.

## 3.5 OPTIMISATION

The optimisation of this problem must be performed over the extent of the entire mission so that the future effects of each countermeasure selection are properly taken into account. This allows for better strategies to be found, especially in situations where a stealthier low-emission approach to a mission would be more effective in the long run, rather than reduced danger in the short term. Also, this approach is important for the allocation of expendable countermeasures as these must be reserved for when they are most needed in the mission. Without overall optimisation, the platform would tend to use up all its available cartridges too soon and be left defenceless in the later parts of missions. Further, a number of other situations can be properly taken into account with this approach such as the platform leaving a threat's weapon range before it is able to fire, where it would be inefficient to waste countermeasure resources on it.

Ideally, an optimal countermeasure strategy should be determined using an exhaustive search through all possible strategies for a mission, but the need for overall mission optimisation makes this infeasible. In this work, there are 10 different active jamming techniques ($N_{AT}$), 60 possible antenna directions ($N_{DIR}$), 4 passive techniques ($N_{PT}$), 2 active channels ($N_{AC}$), and 1 passive channel ($N_{PC}$). Additionally each of the two active channels can be emitted through a towed decoy, or it can remain unused. Working with a 120 s example scenario with 2 s time intervals, 10 RF threat types ($N_{RFT}$) and 4 IR threat types ($N_{IRT}$), results in 61 discrete time intervals ($N_I$). The total number of possible countermeasure strategies for the mission ($N_{TOT}$) can be calculated as

$$N_{DC} = N_{AC} + 1, \tag{3.60}$$

$$N_{ACC} = (N_{AT} \cdot N_{RFT} \cdot N_{DIR})^{N_{AC}} \times N_{DC}, \tag{3.61}$$

$$N_{PCC} = [N_{PT} \cdot (N_{RFT} + N_{IRT})]^{N_{pc}}, \tag{3.62}$$

$$N_{CI} = N_{ACC} \cdot N_{PCC}, \tag{3.63}$$

$$N_{TOT} = (N_{CI})^{N_I} = 4.7678 \times 10^{596}, \tag{3.64}$$

where $N_{DC}$ is the number of possible combinations for the towed decoy (channel 1, 2 or off), $N_{ACC}$ is the number of active channel combinations, $N_{PCC}$ is the number of passive channel combinations, and $N_{CI}$ is the number of combinations per individual time interval. Note that the passive channel can be directed to either RF threats or IR threats, hence the inclusion of both in the calculation of $N_{PCC}$.

This clearly shows that the number of possible combinations is simply too large for an exhaustive search to be feasible, and hence why a genetic algorithm is necessary to solve the problem in a reasonable amount of time. Further, this also demonstrates the need for specialised genetic algorithm operators and procedural seeding in order speed up convergence to a good solution.

### 3.5.1 Overall optimisation procedure

Optimisation is achieved through the use of the genetic algorithm that is depicted in the form of pseudo code in Figure 3.17. It is noted that this pseudo code is the same as that which appears in Figure 3.1, except that the genetic operators have been expanded upon. This has been done both for the ease of the reader, and more importantly in order to contextualise these operators within the overall system.

The process begins with the initialisation of variables according to the user's specifications. These include the population size, the tournament size, the maximum number of generations, the number of mutations and crossovers, immigration settings, stop-criterion settings, and seeding settings. Next, an initial population is generated according to these settings. This includes a number of randomly generated solutions, as well as a number of procedurally generated solutions in order increase the rate of convergence in such a large problem space. Thereafter the fitness of each individual member of this initial population is calculated and the population leaders found.

The algorithm then enters the Pareto loop that is executed according to the number of desired Pareto solutions in order to generate a Pareto set. In its first iteration, the population is not regenerated as it has just been initialised, hence the generation loop is immediately entered instead. In subsequent iterations, a portion of the population is regenerated in order to increase genetic

diversity. The population leader of the previous Pareto design is also kept in the population in order to increase the rate of convergence. The fitness of this new population is then calculated and the population leaders found. Thereafter the objective function weights are updated and the generation loop commenced.

```
1: procedure COUNTERMEASURE OPTIMISATION
2:       INITIALISE VARIABLES
3:       CALCULATE(scenario information)
4:       GENERATE INITIAL POPULATION
5:       CALCULATE POPULATION FITNESS
6:       FIND(population leaders)
7:       for all pareto solutions do
8:             if pareto solution > 1 do
9:                   SET OBJECTIVE FUNCTION WEIGHTS
10:                  REGENERATE POPULATION
11:                  CALCULATE POPULATION FITNESS
12:                  FIND(population leaders)
13:            end if
14:            for all generations do
15:                  OPERATOR(elitism)
16:                  OPERATOR(direct member copy)
17:                  OPERATOR(leader crossover)
18:                  OPERATOR(random crossover)
19:                  OPERATOR(specialised crossover)
20:                  OPERATOR(leader repair)
21:                  OPERATOR(immigration)
22:                  OPERATOR(clean-up operator)
23:                  OPERATOR(random mutation)
24:                  CALCULATE POPULATION FITNESS
25:                  OPERATOR(survival operator)
26:                  FIND(population leaders)
27:                  if stop criterion = true do
28:                        BREAK
29:                  end if
30:            end for
31:            CLEAN-UP SOLUTION
32:            SAVE SOLUTION
33:      end for
34:      return best countermeasure strategies and results
35: end procedure
```

**Figure 3.17.** Overall system pseudo code with expanded optimisation procedure.

The generation loop contains the actual optimisation process and all the associated genetic operators. The process begins by creating a new generation of the population by copying across the leader of the old population, and subsequently a number of tournament-selected unchanged members. Thereafter a number of different crossovers are performed on the old population in order to generate further new population members, including in particular crossovers between the old population leader and the next fittest member. Lastly, the remaining population members are then separately generated using two different repair operations on the unchanged leader of the old population. The now completed population then undergoes checks to determine if immigration is necessary in order to increase genetic diversity, before a number of randomly-selected population members are then cleaned-up using a repair operator. Specifically, this clean-up focuses on removing unnecessary dead times, and decoy allocations from the strategies. Finally, random mutation is performed on a number of randomly selected members of the population, and the fitness of the entire population calculated. Importantly, another repair operation is performed thereafter as this technique makes use of information generated during the fitness calculation process. Specifically, this second operator is aimed at modifying the randomly selected unsuccessful strategies such that threats that successfully hit the platform are jammed prior to that hit. Finally, the population leaders are found and the stop criterion checked. If the stop criterion is met, or the maximum number of generations reached, the generation loop is terminated, the best solution cleaned-up, and then saved for later use.

Note that all of the above concepts are covered in detail in the following subsections. Further, it is noted that in the above cases all selections or reproductions are performed using tournament selection. Also, since a repair operator is performed after the fitness calculation, fitness must be immediately recalculated for any chromosomes modified by this process.

### 3.5.2 Variables

All the variables in the implemented genetic algorithm can be set by the user in order to optimise algorithm performance according to their specific application. There are no rules regarding the selection of most of these values. Instead, they are based on rules of thumb and observation of the performance of the algorithm. A summary of the important variables appears in Table 3.12 for convenience and easy reference. The values chosen for most of the variables and the reasoning behind their selection are discussed in their respective sections.

**Table 3.12** Genetic algorithm variables.

| Variable | Value |
|---|---|
| Tournament Size | 3 |
| Population size ($N_p$) | 100 |
| Maximum Iterations | 150 |
| Number of Crossovers ($N_c$) | 73 |
| Number of Mutations ($N_m$) | 33 |
| Immigration Population Percentage | 40 |
| Number of Seed Solutions in Initial Population | 20 |
| Number of direct copies | 5 |
| Total Number of Specialised Crossovers | 12 |
| Number of Clean-Up Operators | 17 |
| Number of Survival Operators | 17 |
| Percentage of Population Regenerated for Pareto | 90 |

Two of the most important variables in the performance of a genetic algorithm are the population size and the maximum number of generations. Both of these have a direct effect on the time taken to solve the problem, as well as the strength of the final solution. A larger population allows for a larger initial spread of samples across the problem space, which results in better genetic diversity, and thus a greater likelihood that the global minimum will be found. However, a large population requires more fitness calculations, crossovers, and mutations per generation, and hence slows down the rate at which a solution can be found. On the other hand, a large maximum number of iterations increases the likelihood of finding the global optimum, but increases the time taken for the algorithm to run. Further, the rate of solution improvement drastically tapers off over the iterations resulting in diminishing returns for larger numbers of generations. Therefore it is up to the user to decide the relative importance of speed versus the quality of the solution.

Taking into consideration the performance of the algorithm in this work, the population size has been set to 100 and the maximum number of generations set to 150 so as to achieve a general combination of solution strength and speed.

### 3.5.3 Design schema

In this work it is assumed that the platform has two active jamming channels, one passive jamming channel and a towed decoy. Each channel is allocated a jamming technique, a threat type for which it is optimised, and in the case of the active channels, an antenna direction. As such, a countermeasure strategy is composed of selections for each of the above for each individual time

interval over the course of a mission. For optimisation purposes this is encoded into a binary chromosome as the concatenated string of the strategies for each time interval as in Figure 3.18. Each time interval is then split into active channel one technique, threat type, and antenna direction, active channel two technique, threat type, and antenna direction, passive channel (channel 3) technique and threat type, and the decoy allocation as in Figure 3.19.

| Interval 1 | Interval 2 | Interval 3 | ... | $n^{th}$ Interval |
|---|---|---|---|---|

**Figure 3.18.** The design schema for a complete chromosome in terms of time intervals.

| Tech 1 | Threat 1 | Dir 1 | Tech 2 | Threat 2 | Dir 2 | Tech 3 | Threat 3 | Decoy |
|---|---|---|---|---|---|---|---|---|

**Figure 3.19.** The design schema for each individual time interval, where tech is the countermeasure technique allocated to a channel, and dir is the antenna direction.

### 3.5.4 Strategy fitness

The fitness of a countermeasure strategy must take into account a number of different factors that each forms a major part of strategy selection such as the risk to the platform, the financial cost of the mission, and the desired level of EMCON. Further, this must be achieved using a number of user-defined weights to allow for the optimisation of the algorithm's performance for a particular application. As such, the fitness ($\varsigma$) of a solution is calculated as

$$\varsigma = W_V V_{mission} + W_P P_H + W_C N_C + W_J N_J, \tag{3.65}$$

where $V_{mission}$ is the total post-jamming danger value, $P_H$ is the representative probability of the platform being hit, $N_C$ is the number of cartridges used, $N_J$ is the number of countermeasure techniques used, and $W_V$, $W_P$, $W_C$, and $W_J$ are their respective weights. Typically, a fitness function is maximised, but due to the nature of the problem where all the individual objectives must be minimised, fitness is minimised.

The first factor, the total post jamming danger value, is a value representative of the level of danger or risk presented to the platform over the course of a mission. It is the weighted sum of the post-jamming danger values of each of the $N$ RF and the $M$ IR threats over all $K$ time intervals of the mission. That is

$$V_{mission} = \sum_{k=1}^{K} V_{total}^{k} = \sum_{k=1}^{K}\left[\sum_{n=1}^{N}\left(V_{n,RF}^{k}\right) + \sum_{m=1}^{M}\left(V_{m,IR}^{k}\right)\right], \tag{3.66}$$

where $V_{n,RF}^{k}$ and $V_{m,IR}^{k}$ are the post-jamming danger values of the $n^{\text{th}}$ RF and the $m^{\text{th}}$ IR threats, respectively. Minimising this objective function thus minimises the danger to platform, and hence reduces the risk it experiences. This factor is normalised by dividing it by both the total number of threats (IR and RF) and the total number of time intervals so as to keep its magnitude approximately similar in magnitude regardless of the length or size of the scenario.

The second factor, the representative probability of the platform being hit, is a value representative of the probability that the platform was hit by a threat's projectile in the mission, for that strategy. It is defined as the sum of the probability of hit of each individual successful attempt to hit the platform, and hence it is only representative of the probability of hit rather than a true probability. A successful attempt to hit the platform occurs when a guided missile reaches the end of its guidance stage, or when an artillery system reaches the end of its tracking stage. The probability of hit for the $n^{\text{th}}$ threat ($P_{h,n}$) in the event of a successful attempt to hit is calculated as a combination of the probability that the threat was encountered ($P_n$, calculated in (3.24) and (3.26)), and its range adjusted accuracy ($A_n$, calculated in (3.28)):

$$P_{h,n} = P_n \times A_n, \tag{3.67}$$

where both values are calculated for the time interval at the time of firing. By its very definition, the minimisation of this factor minimises the likelihood of the platform being hit, and hence minimises the risk to the platform. There is no straight-forward way of normalising this factor as there is no way to determine a reasonable maximum possible number of successful hits in a scenario. However, this is actually advantageous as the value of this factor should be reduced to zero before the other factors are minimised. This is due to the fact the main objective of this system is the survival of the platform, and hence a large $P_H$ factor simply assists in the prioritisation of this.

The third factor, the number of cartridges used, is simply the number of cartridges used over the course of the entire mission by the strategy being examined. It is normalised as a fraction of the total cartridge capacity of the platform. Due to the fact that a fixed mission route is assumed for this work, the minimisation of this value is the only way to reduce the cost of the mission.

The final factor, the number of countermeasure techniques used, is simply the total number of times a countermeasure technique other than 'none' is allocated to any of the channels. As such, it is a measure of the level of force used by a strategy, and more importantly, it is a measure of the

level of RF and IR emissions generated by the platform over the course of the mission. Minimising this factor thus implements EMCON by preventing unnecessary countermeasure use, and hence minimises platform illumination. This is especially important in preventing the platform from illuminating itself to unknown threats. The value is normalised by dividing it by the product of the total number of time intervals and the total number of jamming channels, as that is the maximum possible number of countermeasure techniques that can be used, and hence the maximum possible amount of EM emissions.

Therefore, each of the four weights sets the relative importance of each major decision-making factor in determining an optimal countermeasure strategy. Although their values are user-defined, typically the largest weight would be applied to $P_H$, then $V_{mission}$, and then either $N_C$ or $N_J$. This prioritises finding a solution to the problem where the platform can complete the mission unscathed. Thereafter, lower-risk strategies are prioritised, before lastly prioritising either cost effective solutions, or EMCON. As such, using the previously-used double weighting scheme, the standard weights for this work are set as 8, 4, 2 and 1 for $W_P$, $W_V$, $W_C$, and $W_J$ respectively, thus favouring cost over EMCON. This scheme ensures that each factor is weighted sufficiently more than the next-smallest one. Again, as with the danger-value weights, these weights are normalised using their total within the algorithm so as to ensure that the fitness values always occur in a similar range.

Further, it is necessary to note that the average values of the above factors play an important role in the normalisation process and allocation of weights. These must be assessed to ensure that the values of the factors typically occur with the same relative magnitude, and hence are fairly weighted in the optimisation process. This is especially evident in the total post-jamming danger value. Considering the way that danger values are calculated in (3.16), it is seen that an average danger value can be approximated as 0.2125. This is because the weighted sum in brackets will have an approximate average value of 0.5 due to the normalisation of those weights. Thereafter this value is multiplied by the normalised RCS factor, and the probability of encounter. The RCS factor has an average value of 0,5 by design, whereas the average of the probability factor can be approximated as 0.85 if the probability values contained in the example scenario are used, and the effects of distributed threats are ignored. Lastly, this value is then multiplied by a jamming factor with a further average value of 0.5 in order to attain the post-jamming danger value, finally resulting in an average value of 0.10625. In comparison, the average values for the number of

cartridges used, and the number of countermeasure techniques used can simply be approximated as 0.5. As such, in order to attain a fair comparison, the post-jamming danger weight must include a further multiplier of 4.7. Therefore this multiplier is included in the algorithm, resulting in the rounded modified weight of 19 for $W_V$ in the standard set of weights.

Lastly, disallowed combinations are handled in the algorithm by simply allocating a fitness value of 999 to the offending time interval. The result of which is to drastically reduce the associated strategy's fitness and in so doing prevent their reproduction. Examples of disallowed combinations include flares being directed towards RF threats (and vice versa), firing cartridges when there are insufficient remaining, and out of range variable values.

### 3.5.5 Seeding

Seeding is the process of introducing good solutions into the initial population in order to increase the rate of convergence. Obviously, the fitter this initial population is, the faster the rate of convergence that can be achieved. However, if there is insufficient genetic diversity in the population, premature convergence can occur, causing the algorithm to get trapped in a local minimum rather than find a global minimum. Therefore, a good compromise must be found between the two extremes. This is achieved by generating the initial population using a combination of randomly generated solutions for diversity, and procedurally developed solutions to increase the rate of convergence.

For this work, a random population generation method was implemented along with two different procedural seeding methods. The number of members generated by each can be set by the user in order to optimise the performance of the algorithm. Currently, a population size of 100 is used, with 10 members generated by each procedural technique, and the remaining 80 generated randomly. In the case of the procedural techniques, only a single seed solution is generated, so the remaining members are generated through the random mutation of that single seed. Specifically, 5 mutations are performed on the procedurally generated strategy in order to generate each new solution. This is done so as to ensure that the procedurally generated solutions form a reasonable percentage of the initial population to increase the rate of convergence.

*1) Random population generation*

This is simply achieved by randomly selecting a technique, threat type, and antenna direction, where applicable, for each of the three countermeasure channels, as well as a channel allocation for the decoy. This is then repeated for each of the time intervals of the scenario in order to generate a complete chromosome. Some basic checks were included to prevent the allocation of disallowed combinations such as the directing of flares towards RF threats etcetera.

*2) First procedural seeding technique*

Seeding requires the development of a basic reasonable solution to the problem in minimal time. This is achieved through the procedural generation of solutions. The first approach used in this work is essentially to jam the two most dangerous threat types in each time interval using active techniques, with cartridges being reserved for IR threats and time intervals in which the active countermeasures alone are insufficient.

This is achieved by beginning with IR threats. In each time interval in which there is an IR threat in the guidance stage, flares are allocated to that threat type. In circumstances in which there are multiple IR threat types in the guidance stage, the flares are allocated to the threat type with the greatest countermeasure resistance. This is because that threat type will have the greatest flare rejection capabilities and hence will require a more optimised flare strategy in order for its lock to be broken. Further, non-optimised flare strategies are still effective against threat types with lower countermeasure resistance.

Next, the RF threats are handled on an iteration-by-iteration basis by allocating the most effective countermeasure techniques to the two most dangerous threat types in each time interval, ignoring external factors. Importantly, this allocation is based on the two most dangerous threat types, rather than the two most dangerous threats. This is done so as to prevent interference issues. For example, if the two most dangerous threats are of the same type in two different stages, with one in tracking and another in acquisition, then the allocated countermeasures would interfere negatively, reducing their effectiveness. Further, this allocation process includes directing each channel's antenna directly at the threat it has been allocated, but only if it is an attempt at jamming a new threat. Thereafter, if that threat type remains one of the most dangerous in the following time intervals, thus requiring the same countermeasure allocation, then the antenna direction is kept at that original value. Importantly, this prevents inducing antenna-direction-change dead time that would

otherwise render the countermeasures ineffective. Also, it is necessary to note that channel allocation needs to take into account the position of the threat relative to the platform as it would be both inefficient and less effective to, for example, attempt to jam a threat on the right-hand side of the platform using its left-hand side channel. As such, the most dangerous threat and its associated threat type are allocated to the appropriate jamming channel on the same side of the platform. The second most dangerous threat and its associated threat type must then be allocated to the remaining channel, regardless of whether it is on the correct side of the platform or not. Lastly, the decoy is then allocated to the most dangerous threat and its allocated channel if that threat is in a tracking-type stage so as to increase the likelihood of breaking its lock. For the lookup table settings used in this work, threats in search and acquisition stages are allocated the cover pulse technique, threats in tracking are allocated RGPO, and threats in guidance are allocated VGPO.

Finally, the remaining cartridges are allocated to chaff countermeasures in time intervals in which the third most dangerous RF threat type is sufficiently close in danger value to the first two, and is in a tracking-type stage. Importantly, this must only be done for time intervals in which flares have not already been allocated to the passive channel. This is achieved by comparing the difference in danger value between the second and third largest threat types to the difference between the largest and the second largest. If the difference between the second two threat types is smaller than the difference between the first two, then it is deemed that the third largest threat is sufficiently threatening to require countermeasures. Since the threat is in a tracking-type stage, dilution chaff is used. However, if there are insufficient cartridges remaining, then the passive jamming channel is allocated no technique instead.

*3) Second procedural seeding technique*

This second procedural method is the same as the first, except how it handles threats in the search and acquisition stages. So when the most or second most dangerous threat type is in one of these stages, a wide-band noise jamming technique is used instead of the previously-discussed most-effective technique against that single threat. In order to maximise effectiveness, the bandwidth and centre frequency ($O$) used for the noise jamming technique are determined by an exhaustive search that maximises that channel's jamming effect in the time interval in which the technique is commenced. As with the previous approach, regardless of the stages of the threats, the jamming is allocated to the channel on the correct side of the platform for the most dangerous threat type, and its antenna steered in the direction of that threat, with the second most dangerous allocated to the

remaining channel. Thereafter, the antenna direction of a channel is kept constant until the threat allocated to it changes. Lastly, in order to counter the rapid technique changes caused by the miniature optimisation scheme, the technique allocated to a channel is also kept constant until a new threat is allocated to it. The reason for this overall approach is that threats in these search-type stages present less danger to the platform, hence it is potentially more beneficial to the platform to attempt to jam a number of threats simultaneously rather than focus solely on one threat.

### 3.5.6 Crossover

Two methods of crossover were implemented in the optimisation algorithm: a standard random crossover for diversity, and a specialised crossover that takes the characteristics of the problem into account for a greater rate of convergence.

*1) Random crossover*

This is implemented by first randomly selecting two tournaments of three chromosomes, where the fittest member of each is chosen for crossover. The process of crossover itself is then performed by randomly selecting a point anywhere along the entire length of the chromosome, where both chromosomes are split. The first section of the first chromosome is then combined with the second section of the second chromosome to generate a single new chromosome, where the generation of a single chromosome from each pair is in order to increase diversity. The number of random crossovers can be set by the user in order to optimise the performance of the algorithm. It is currently set to generate 73 members of the total population of 100.

*2) Specialised crossover*

The approach of this crossover type is based on the fact that the countermeasure selections for individual time intervals are relatively independent of one another for similar strategies. However, drastically different strategies handle the threats differently, resulting in them being in different radar stages at different points throughout the mission, thus requiring different strategies. As such, this specialised crossover is divided into four approaches. The first compares the post-jamming danger values of a tournament of three chromosomes on a time-interval-by-time-interval basis, and selects the fittest countermeasure strategy for each individual interval of the child chromosome. The other approaches use the same concept, except that the chromosomes are broken down into halves, quarters, and sixths, and compared for these larger groups of time intervals, with the best

countermeasure strategy for each section chosen for the child chromosome. This process is depicted in Figure 3.20 for a single subsection comparison, where this process is repeated for each individual subsection, and the subsection size varies from individual time intervals all the way to half chromosome lengths.

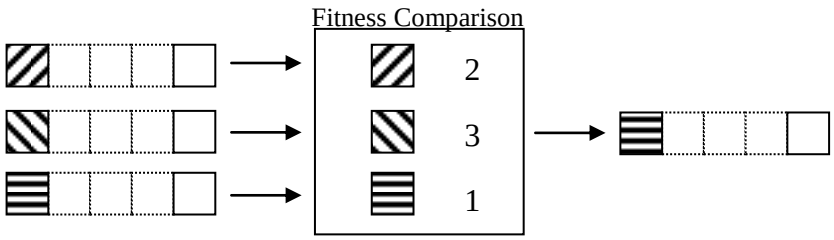



**Figure 3.20.** The process of the specialised crossover technique.

The effect or advantage of each can be best described by comparing the two extremes of this process. For similar parent chromosomes, the threat environment would be very similar over the course of the mission, thus allowing for direct comparison on an interval-by-interval basis. This works particularly well for chromosomes that are mostly the same throughout the mission, but have different strategies for a few time intervals in which they have poor performance. This specialised crossover would then essentially take the fittest member from the pool and replace each of its poor performing time intervals with the best strategy from the other chromosomes, thus allowing for the improvement of its overall performance. On the other hand, the comparison of half chromosomes is for dissimilar chromosomes where the different strategies result in different threat environments over the course of the mission. In this case, the three chromosomes are compared to see which handled each half of the mission the best, and their strategy used for the child chromosome accordingly. This is essentially aimed at improving an otherwise good strategy by replacing an entire section of poor performance. This approach works because the larger segments negate the effect of differing threat environments. The remaining types are then simply filling the gaps between the two extremes, catering for different levels of similarity.

Therefore, the half chromosome approach theoretically speeds up convergence in earlier iterations when dissimilarity is more prevalent, with each smaller-segment-size approach speeding up convergence in later and later iterations. The iteration-by-iteration approach should then help convergence to the optimum in the late stages of the optimisation process when strategies are more similar. Each type of this custom crossover is used to create 3 new chromosomes in each iteration, resulting in a total of 12.

*3) Population leader crossover*

In order to increase the rate of convergence of the algorithm, both the random and the specialised crossover are specifically performed on the population leader and the next fittest, different chromosome. This has the potential to increase the rate of convergence because this process combines the characteristics of the two fittest designs in the population. In this work, a single specialised crossover of each type is performed on the two chromosomes, along with 3 random crossovers.

### 3.5.7 Mutation

Mutation introduces new genetic material to the population by randomly altering chromosomes. In this work it is achieved by selecting a random chromosome other than the population leader (which is preserved through elitism), then selecting and flipping a random bit in that chromosome. In this work, the number of mutations per generation is set to 33.

### 3.5.8 Repair operators

Two different repair operators were implemented that non-randomly repair chromosomes by searching for errors and attempting to fix them in order to aid convergence. The first is a clean-up operator that searches for and removes non-ideal strategy allocations, and the second is a survival operator that identifies and attempts to fix unsuccessful strategies.

*1) Clean-up operator*

The goal of this operator is to clean-up randomly selected members of the population so as to improve optimisation performance. This is achieved by indentifying and fixing unnecessary dead time intervals, and intervals in which the decoy has been allocated a channel with no jamming technique. The first is achieved by searching for patches of dead time that are longer than the maximum user-defined amount. The unnecessary dead time intervals are then simply replaced with no technique for that channel. Further, due to the fact that switching a channel off does not incur dead time, there should never be dead time intervals occurring before intervals in which no technique is allocated. Thus, these instances are also searched for and corrected by replacing them with no technique allocation. Lastly, since it does not make sense for the decoy to be allocated to a channel in a time interval in which no technique is allocated, these intervals are fixed by setting the

decoy allocation to none. In this work, the number of these repairs performed per generation has been set to 17. Further, it is noted that this repair operator is also used as a clean-up procedure on the final solution after optimisation has been performed so as to ensure that the saved solution does not contain the above issues and is as close to optimal as possible.

*2) Survival operator*

The primary goal of this optimisation procedure as a whole is to rapidly find a solution where the platform escapes from the mission unscathed. As such, an operator is needed that identifies the times in a countermeasure strategy when the platform is hit by a threat's projectile, and attempts to prevent this. The basic concept is that if the platform is successfully hit in a time interval, then the lock of that particular threat should be broken in the previous time interval while it is in its guidance stage, or in the case of artillery, its tracking stage.

In order to achieve this it is necessary to keep track of the time intervals in which the platform is hit, along with which threats fired the projectiles. This information is determined as part of the fitness calculation, and as such it must be stored for the entire population for analysis. However, this does require that this operator be performed after the fitness of the population has been calculated for a generation, where the fitness of a modified member must be immediately recalculated after modification.

If the platform is hit by a single RF threat in a particular time interval then the most effective countermeasure technique (ignoring external factors) must be allocated to it in the previous time intervals. As with the procedural seeding techniques, the threat is allocated to the appropriate channel that is on the same side of the platform as it is. Further, the antenna direction is chosen in such a way that it is aimed directly towards the threat in the time interval in which break lock is intended to occur. In other words: the time interval prior to when the projectile otherwise would have successfully hit the platform. Again, as with the procedural seeding techniques, this antenna direction is kept constant throughout the time during which the threat is to be jammed so as to prevent antenna-direction-change dead time. Then if there is a second RF threat that simultaneously hits the platform in that same time interval, then the remaining active channel is allocated the most effective technique for that threat type for the previous time intervals, with its antenna direction allocated in the same way. Thereafter, if there is a third RF threat that hits the platform in that time interval, then the passive channel is set to fire the most effective chaff technique for that threat.

Lastly, it is noted that for the settings used in this work, the most effective active countermeasure technique for artillery is RGPO, for missiles it is VGPO, and in terms of chaff it is dilution chaff for both.

Importantly, the number of time intervals for which a channel must be set to the most effective countermeasure technique must be determined on a case-by-case basis. Obviously, the countermeasure must be implemented for its user-defined average jam time in order to break lock, but it also must be allocated to the channel for enough time before that in order to allow for the dead time required to change countermeasures. The dead time for that channel is dependent on whether it is the technique, threat type, antenna direction, or any combination thereof that changes from the previous countermeasures. As such, for efficiency it must be determined which is the shortest possible change that can be implemented with the pre-existing countermeasure allocations in the strategy. This can be achieved by simply beginning with the shortest dead time option, say technique change only, and checking to see if its required dead time fits with the previous countermeasure allocations. If it does, then that is the shortest possible dead time, and the countermeasure strategy for that number of time intervals is changed accordingly. If it is not, the next shortest option, say antenna direction change only, is checked and so on.

IR threats are handled in exactly the same way, except that when there are multiple threats that hit the platform simultaneously the passive jamming channel is set to fire flares optimised for the threat type with the greatest countermeasure resistance. This is for the exact same reasons as discussed in the procedural seeding techniques: there is no second passive channel, and this approach will have the greatest effect on the greatest number of these IR threats. In the unlikely event of there being three RF threats that hit the platform and one or more IR threats that hit the platform simultaneously, then the IR threats are given priority. This is due to the fact that there is no other channel that can be used to counter the IR threats and because IR threats can only be countered during their guidance stage in this work.

The user-defined number of these operators performed per generation is set to 17 in this work.

*3) Population leader operators*

As with crossover, each of the repair operators is used on the population leader in order to increase the rate of convergence. Due to the non-random nature of these techniques, they are only

performed once each and the two resultant chromosomes are saved in different locations so as not to corrupt the population leader which is protected through elitism.

### 3.5.9 Immigration

Immigration is a technique used to increase the genetic diversity of a population once the algorithm has been left to run for a relatively large number of generations and has converged towards a single solution. Importantly, it can help the algorithm escape a local minimum. The technique keeps track of the levels of genetic diversity by counting the number of different chromosomes in the population in each generation, as well as the number of individuals with each chromosome. If the number of members with a single design increases over a user-defined threshold, then all but one of these members are regenerated with new random designs. In this work, this threshold is set to 40 percent of the population.

### 3.5.10 Stop criteria

There are three different stop criteria in this system for the user to choose between, where each has its own advantages and application. However, the maximum limit on iterations does remain in effect regardless of what stop criterion has been selected. This is because it acts as a back-up to the other stop criteria in the event that they cannot be met.

*1) Maximum iterations*

The first stop criterion simply limits the number of iterations or generations that the genetic algorithm is run for. When this limit is reached, the algorithm is terminated and the population leader's chromosome returned as the solution regardless of its performance. This approach is suited to applications where the optimisation can be run for a known, limited amount of time, where the iteration limit would then be set according to the known run-time of the algorithm. This would then result in the best possible countermeasure strategy for that limited time period. For example, this approach would be used if it is known at what time a mission must commence and hence what time the platform must be loaded with its optimal cartridge load-out. The iteration limit would then be set so that the solution is ready in time for loading to commence, resulting in the best possible solution in the available time.

*2) Safe passage*

The second criterion is the representative probability of the platform being hit over the course of the mission. This method runs the algorithm until the population leader's strategy results in the platform escaping the mission unscathed as this is the minimum requirement for a strategy to be considered successful, or reasonable. As such, this method results in the fastest possible generation of a solution because it runs the algorithm until the first reasonable solution is generated before terminating. It is suited to applications where speed is favoured over strategy performance. For example, it would be used in a situation where the platform must be scrambled in order to engage a target. The algorithm can then be run for the shortest possible amount of time, the platform loaded, and the mission commenced.

*3) Minimal improvement*

The final criterion is a compromise between the two previous extremes in terms of performance. This method works using the fact that the performance improvement between the generated strategies reduces over the time that the algorithm is run, where the diminishing returns eventually are no longer worth the time required to generate them. As such, this method runs the algorithm until a user-defined minimum improvement in the fitness of the population leader has not been met for a specified number of consecutive iterations. However, the algorithm first increases the number of mutations per generation in an attempt to increase the rate of improvement. If the minimum improvement is met after this, the number of mutations is reset, and the algorithm is allowed to continue. If it is not, the algorithm terminates after a number of iterations. The number of iterations of insufficient improvement before the number of mutations is increased, the proportional increase in the mutations, and the total number of iterations allowed for the improvement to increase before the algorithm is terminated are set to 5, 2, and 10 in this work, respectively. The required minimum improvement in the fitness value is set to 0.0001.

This approach would be used in circumstances that are time sensitive, but where more optimal strategy performance is desired. For example, it would be used in circumstances where the platform must be sent out on a mission as soon as possible, but it does not have to be immediately. The algorithm would then be run until the improvement in the strategies' performance was no longer worth the time it took to generate.

### 3.5.11 Pareto optimisation

Pareto optimisation is implemented by running the genetic algorithm multiple times, each with a different set of fitness function weights in order to create individual Pareto solutions that prioritise different characteristics. Once the chosen stop criterion has been met for each run of the algorithm, the population leader is saved as the Pareto solution for that set of weights. Next, the previous population leader is protected through elitism, and a set percentage of the population randomly regenerated. The fitness of the new population is then calculated and the population leaders found. Finally, the weights of the fitness function are then updated to those of the next desired solution and the genetic algorithm started. This entire process is repeated until all the desired Pareto designs have been found.

The reason why the population regeneration is handled in this way is because this reuses the fit designs found during the previous optimisation process to give the genetic algorithm a strong initial population in order to increase the rate of convergence. A percentage of the population is also regenerated in order to maintain genetic diversity, especially in cases where a large portion of the previous population may have converged to a single solution. This percentage of regenerated designs can be varied by the user to alter performance. It is set to 90% in this work.

For demonstration purposes in this work, three combinations of weights are used to generate three different Pareto solutions. These appear in Table 3.13, where the representative-probability-of-hit weight ($W_P$) is largest for all three sets, because the main priority is always to ensure the survival of the platform before other considerations can be taken into account. The first set of weights, which is used as the standard set of weights, is for a conservative solution that focuses on the safety of the platform by favouring lowering danger levels over cost and EMCON. The second set favours minimising cost over safety and EMCON, and the final set of weights is then for a solution that favours EMCON over cost and safety, where in both cases the second strongest weighting, excluding $W_P$, has been assigned to the post-jamming danger value. This is due to the fact that platform risk should always be an important parameter for optimisation. Note that, as previously, the value of each weight has been set equal to half that of the previous weight so as to ensure that the effect of each is sufficiently large as compared to each other. Further, as discussed previously, in the algorithm the weights for the post-jamming danger value will include a multiplier of 4.7, and that all weights are normalised using the set's total.

**Table 3.13** Pareto solution fitness function weights.

| Weight | Conservative | Cost | EMCON |
|---|---|---|---|
| **Probability of hit ($W_P$)** | 8 | 8 | 8 |
| **Post-jamming danger ($W_V$)** | 4 | 2 | 2 |
| **Financial cost ($W_C$)** | 2 | 4 | 1 |
| **EMCON ($W_J$)** | 1 | 1 | 4 |

## 3.6 CHAPTER SUMMARY

In this chapter the implemented approach to the problem of EW countermeasure and load-out optimisation was discussed, including modelling, threat evaluation, countermeasure allocation, and optimisation. This began with an overview of the entire system using pseudo code, and a number of clarifications and assumptions. Thereafter the underlying model was discussed including how scenario information is calculated, as well as how radar stage progression, IR stage progression, break lock, and dead time are handled. Next, the prioritisation of threats using a danger value was discussed along with the associated sub-factors: platform RCS, probability of threat encounter, threat radar stage, threat accuracy, projectile time to platform, time to next radar stage, as well as how these are applied to IR threats. Then the process of determining the effects of countermeasure allocation was considered in the form of how to calculate the jamming factor for both active and passive channels. Specifically, the effect of active channels was broken down into a number of multiplicative factors that were also discussed: antenna gain pattern, stage effectiveness, technique interaction, cross effect, chaff interference, and countermeasure resistance. Similarly, the effect of chaff was broken down into the following factors: stage effectiveness, chaff illumination, and cross effect. Lastly, the effect of the towed decoy on the jamming factor of each channel was discussed.

Lastly, the optimisation process itself was covered. This began with a motivation for the need for advanced optimisation techniques, along with a discussion of the overall optimisation procedure in the context of the pseudo code of the entire system. Thereafter, the specifics of the genetic algorithm were discussed. This included: the variable values used, the design schema, fitness calculation, random and procedural seeding, random and specialised crossover, mutation, clean-up and survival repair operators, immigration, stop criteria, and Pareto optimisation.

# CHAPTER 4 RESULTS

## 4.1 CHAPTER OVERVIEW

In this chapter the results achieved for the example scenarios are presented. This begins with a discussion of the example scenarios themselves in Section 4.2 including the scenario layout, the mission waypoints and the threat characteristics. It ends with an analysis to confirm that the selected threat characteristics are reasonable within the context of the scenario. This is followed by an in-depth analysis of the strategies developed for both the full and the half scenarios in Section 4.3. Lastly, an analysis of the performance of the genetic algorithm is presented in Section 4.4. This begins with a comparison of the three implemented stop criteria. Thereafter the performance of the Pareto optimisation is demonstrated with a comparison of three different Pareto designs. Next, an overall analysis of the genetic algorithm is presented, including the population leader fitness and the population diversity over time. The section concludes with analysis of the implemented specialised operators in order to determine their effect on optimisation performance.

## 4.2 EXAMPLE SCENARIO

Performance of the algorithm is best illustrated through the detailed analysis of an example scenario, and associated half scenario, where the developed strategy is validated through the demonstration that the proposed actions are reasonable within the context of the mission. Importantly, this type of explanation capability is essential in building the confidence necessary for a system to be entrusted with human lives. The scenario used in this work is one of an airborne platform entering adversary territory in order to engage a target that is in turn defended by a number of different ground-based threats. The scenario is initiated and terminated close to the target, and makes use of a moderate time interval of 2 s, so as to keep the resulting data reasonably presentable. However, it is this section of the mission, with a large number of threats in a small area, that is the most critical and shows the ability of the system to operate under pressure. This

scenario is the basis for all results that follow unless otherwise stated, where the simplified half scenario just uses a subset of the threats presented here.

### 4.2.1 Layout

The layout of the scenario is depicted in Figures 4.1 and 4.2, where the first depicts the radar ranges of the threats, and the second shows their weapon-system ranges. Note that in this work IR threats do not have a radar range, and hence do not have a range depicted in Figure 4.1.

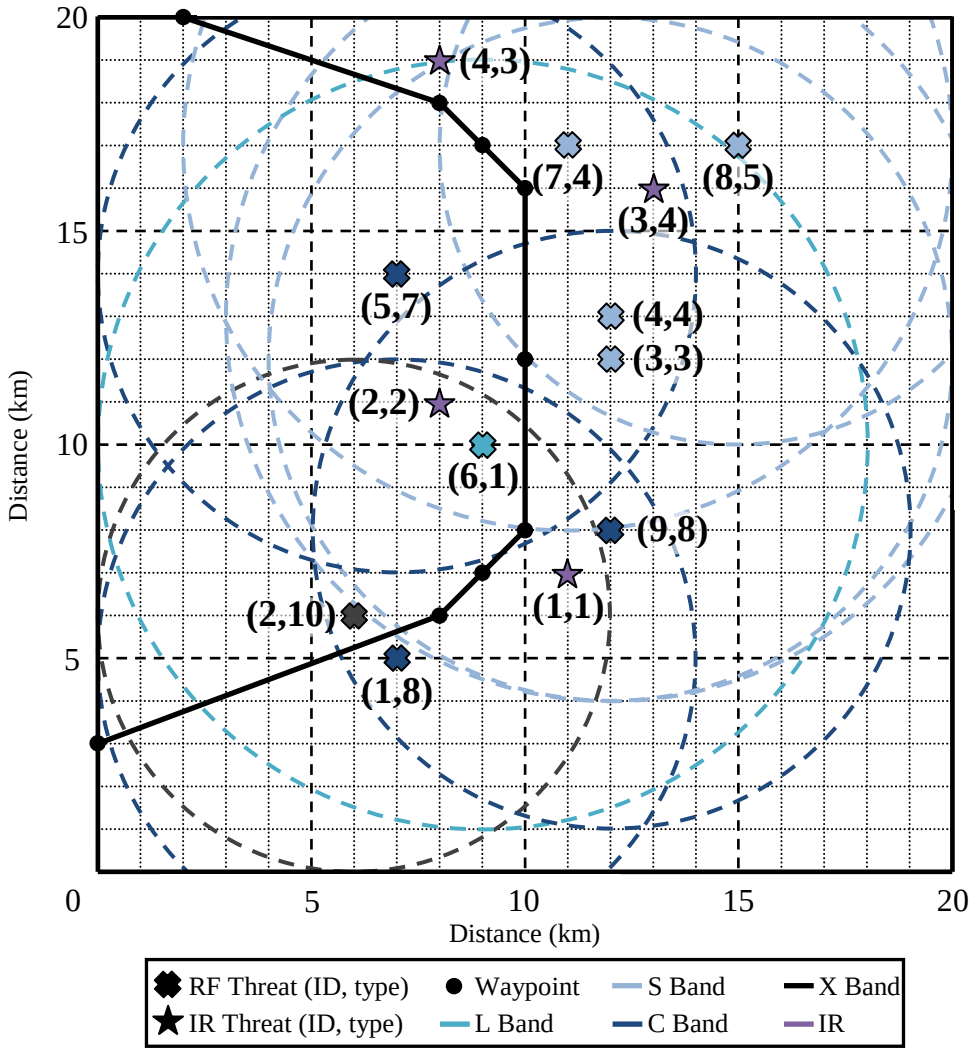



**Figure 4.1.** Scenario layout and radar ranges.

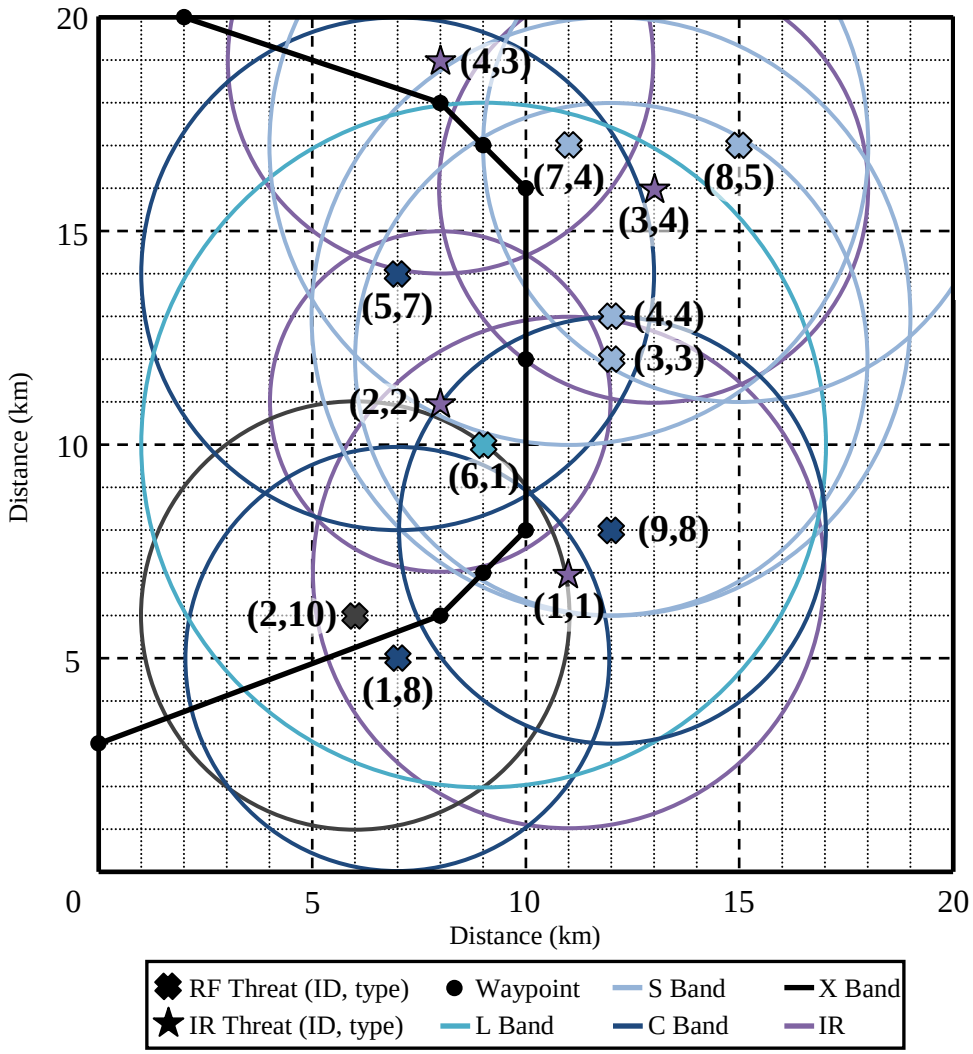



**Figure 4.2.** Scenario layout and weapon system ranges.

In these figures, each RF threat is depicted using a cross, whilst IR threats are depicted using a star. Further, each is labelled with two numbers: the first indicates that threat's ID number, and the second indicates its threat type. The frequency band of each threat is colour coded such that turquoise, light blue, dark blue, and black indicate the L, S, C, and X bands respectively, whilst IR is indicated by purple. This not only conveys more information, but also helps the reader to more easily determine where potential illumination issues and interactions may occur. Lastly, the mission waypoints of the platform are depicted as solid black circles, and the path traversed between them as a thick black line, where the platform moves from the bottom of the arena towards the top.

### 4.2.2 Waypoints

The example scenario contains 9 waypoints, where each consists of a set of coordinates, a time from the commencement of the mission at which the waypoint is reached, and a platform roll angle. The individual waypoints are detailed in Table 4.1, where it is noted that the remaining characteristics of pitch, heading and airspeed are determined by the waypoint information, rather than directly programmed by the user. It is seen that platform enters the mission area at an altitude of 14 km (45932 ft), descends to an altitude of 8 km (26247 ft) in order to engage the target at the 5$^{th}$ waypoint, before finally ascending to its original altitude in order to exit the area. This results in a downward pitch of 35° in the beginning of the mission, and an upwards pitch of 43.5° at the end. Further, waypoints 3 and 7 are allocated a roll of -45°. According to the conventions chosen in this work, this means that the platform performs a left-hand turn by beginning a counter-clockwise roll (from the perspective of the pilot) at the previous waypoints (2 and 6), reaches a peak roll of 45° at those waypoints, before then rolling back to a horizontal position at the following waypoints (4 and 8). The platform remains level throughout the rest of the mission. Finally, it is noted that the waypoint times have been chosen so to keep an approximate airspeed of 1000 km/h (539.96 knots) throughout the course of the mission, except for the periods during which the aircraft is turning where it slows down.

**Table 4.1** Scenario waypoints.

| Waypoint | Coordinates (km) | | | Time (s) | Roll (°) | Pitch (°) | Heading (°) | Airspeed (km/h) | Airspeed (knots) |
|---|---|---|---|---|---|---|---|---|---|
| | X | Y | Z | | | | | | |
| 1 | 0 | 3 | 14 | 0 | 0 | n/a | n/a | n/a | n/a |
| 2 | 8 | 6 | 8 | 36 | 0 | -35.0 | 20.6 | 1044 | 564 |
| 3 | 9 | 7 | 8 | 42 | -45 | 0.0 | 45.0 | 848 | 458 |
| 4 | 10 | 8 | 8 | 48 | 0 | 0.0 | 45.0 | 848 | 458 |
| 5 | 10 | 12 | 8 | 62 | 0 | 0.0 | 90.0 | 1029 | 556 |
| 6 | 10 | 16 | 8 | 76 | 0 | 0.0 | 90.0 | 1029 | 556 |
| 7 | 9 | 17 | 8 | 82 | -45 | 0.0 | 135.0 | 848 | 458 |
| 8 | 8 | 18 | 8 | 88 | 0 | 43.5 | 135.0 | 848 | 458 |
| 9 | 2 | 20 | 14 | 120 | 0 | 0.0 | 161.6 | 981 | 530 |

### 4.2.3 Threats

Due to the classified nature of detailed platform characteristics, generic threats have been used instead, where the scenario has been specially designed to include a number of situations that both challenge the algorithm, and highlight how it handles such issues. As such, the characteristics of the threats have been chosen accordingly. This means that threat characteristics were chosen so as to create a good example in the context of the size of the mission area, rather than attempting to accurately represent real systems. Specifically, threat ranges have been chosen so as to create a dynamic threat environment where the combination of threats presenting danger to the platform at any given point in time changes over the course of the mission. Further, it was ensured that the mission begins and ends out of range of all threats so that the generated strategy completely encapsulates the entire mission and does not take advantage of the mission ending early.

Specifically, this scenario has been designed such that threat types 3 and 4 occur near to one another, as is the case with threat IDs 3 and 4. This is because these two threat types have been chosen to operate in the same frequency range, and as such techniques directed at one will have a strong effect on the other, potentially resulting in large amounts of interference and platform illumination. Also, two threats close in frequency-band usage, in the form if threat IDs 1 and 2 (types 8 and 10), have been placed immediately either side of the platform's path. This is because, in terms of the platform's RCS and antenna gain patterns, this results in the two threats appearing close to one another at long distances as the platform approaches. Thereafter, the two threats move apart as the platform passes between them, before then finally moving back together again. Due to their similarity in frequency of operation, this means that the techniques directed at these two threats will at first interfere a lot, then reduce to almost no interference, before returning to large interference values again. Further, threat ID 8 was placed specifically so that the platform only passes through the edge of its range. Importantly, it was also chosen to be an artillery type. This means that this threat will only have to reach the end of the tracking stage before the platform leaves its weapon-system range in order to obtain a hit. The average time taken to progress from search through to the end of the tracking stage is 24 s, whilst the platform is only within range for 20 s. This means the platform can ignore the threat in normal circumstances. However, there are frequency-adjacent threat types in the form of threat IDs 3, 4, and 7 in the immediate vicinity whose allocated countermeasures could illuminate the platform. This in turn requires the developed

strategy to be careful so as not to illuminate the platform too much and allow threat ID 8 to fire and hit the platform, whilst still sufficiently suppressing threat IDs 3, 4, and 7.

Overall, the remaining threats have been placed so as to create a more sparse threat environment in the beginning of the mission, and a more saturated one around the target at the $5^{th}$ waypoint and towards the end. This also includes the placement of a threat type 7 that uses a larger bandwidth as it will have increased interactions with the other threats. IR threats have also been spread throughout the mission. Lastly, it is noted that in general a greater number of RF threats have been included in the scenario due to the fact that these types of threats have been explored in greater depth in this work, and hence present the greatest complexity.

The characteristics of the RF threats appear in Table 4.2: ID number, threat type (Y), weapon-system category (W), weapon-system subcategory, accuracy (acc.), weapon-system range (weap. range), radar range, projectile velocity, probability of occurrence (prob.), radius of likely encounter (rad.), countermeasure resistance (res.), cartridge requirement (cart.), and frequency band of operation (B). Note that in the weapon-system category column (W), an artillery system is indicated using an ‘A’, whilst a guided missile is indicated using an ‘M’. The characteristics of each threat type have been chosen so as to create a combination of more threatening types, and less threatening ones, whilst the probabilities and distribution radii have been chosen at random. The average search, acquisition, and tracking times have been universally set to 10, 6 and 8 s respectively in order to make the results easier to follow as this makes any deviations in these times more obvious. Also note that the cartridge requirement for a threat is the number of cartridges required per time interval for a chaff technique directed at that threat type, and that the total cartridge capacity of the platform has been set to 110. Lastly, the frequency bands of operation have been allocated according to the convention in this work that the threat type numbers must be allocated in increasing frequency-usage order. As such, threat types 1 and 2 have been allocated to the L band, types 3, 4, and 5 have been allocated to the S band, types 6, 7, and 8 have been allocated to the C band, and types 9 and 10 have been allocated to the X band.

It is noted that two pairs of identical threat types have been included in the scenario: threat IDs 4 and 7 (type 4), and IDs 1 and 9 (type 8). As such, these pairs have been allocated the exact same threat characteristics, except for their probability and radii of occurrence as these are independent of the threat type. This more accurately emulates the real world where multiple threats in a mission

would be of the same type. Further, it is noted that threat types 2, 6, and 9 do not appear in the scenario despite the fact that the highest threat type is a type 10. This demonstrates how the threat type numbers can be allocated to more accurately model the frequency-band usage of the threats, considering the known cross-effect lookup table values. For example, in this scenario threat type 10 is not adjacent to threat type 8 in the frequency spectrum, and hence should experience less interference. As such it was allocated the threat type number 10 instead of 9. Further, not all potential threat types in the threat library will be encountered in every mission.

**Table 4.2** RF Threat characteristics.

| ID | Y | W | Weapon Subcategory | Acc. | Weap. Range (km) | Radar Range (km) | Projectile Velocity (km/h) | Prob. | Rad. (km) | Res. | Cart. | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 8 | A | Non-explosive | 0.75 | 5 | 7 | 1400 | 0.95 | 1 | 0.2 | 1 | C |
| 2 | 10 | M | Beam riding | 0.70 | 5 | 6 | 1700 | 0.80 | 2 | 0.3 | 2 | X |
| 3 | 3 | M | Command | 0.80 | 6 | 8 | 1900 | 0.90 | 1 | 0.0 | 1 | S |
| 4 | 4 | M | Active | 0.95 | 7 | 9 | 2200 | 0.85 | 2 | 0.2 | 3 | S |
| 5 | 7 | M | Semi-Active | 0.90 | 6 | 7 | 1800 | 0.80 | 3 | 0.1 | 2 | C |
| 6 | 1 | M | Command | 0.85 | 8 | 9 | 2000 | 0.95 | 2 | 0.2 | 3 | L |
| 7 | 4 | M | Active | 0.95 | 7 | 9 | 2200 | 0.80 | 1 | 0.2 | 3 | S |
| 8 | 5 | A | Explosive | 0.85 | 6 | 7 | 1400 | 0.85 | 3 | 0.3 | 2 | S |
| 9 | 8 | A | Non-explosive | 0.75 | 5 | 7 | 1400 | 0.90 | 2 | 0.2 | 1 | C |

The characteristics of the IR threats appear in Table 4.3, where it is seen that the threat characteristics have also been chosen so as to create a combination of more threatening types, and less threatening ones. Threat probabilities of occurrence and radii of likely encounter have been chosen at random. Further, it is noted that similarly to RF threats, the reload times of IR threats have been universally set to 20 s to allow for easier scenario analysis. Lastly, it is noted that the threat type numbers have simply been allocated in the range of one to four. This is because of the assumption made in this work that a cocktail of different flare types is used, which in turn means that the frequency bands used by each threat have no effect.

**Table 4.3** IR threat characteristics.

| ID | Y | Acc. | Range (km) | Projectile Velocity (km/h) | Rad. (km) | Prob. | Res. | Cart. |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.95 | 6 | 2200 | 1 | 0.90 | 0.3 | 2 |
| 2 | 2 | 0.80 | 4 | 1700 | 2 | 0.85 | 0.1 | 2 |
| 3 | 4 | 0.85 | 5 | 1900 | 1 | 0.80 | 0.0 | 1 |
| 4 | 3 | 0.90 | 5 | 1600 | 3 | 0.95 | 0.2 | 3 |

### 4.2.4 Parameter confirmation

Lastly, to check the values chosen for the threat ranges and radar-stage progression rates, these values must be compared. This ensures that each threat has sufficient time to hit the platform in the absence of any countermeasures, because otherwise those threats would not actually present any danger to the platform. This is achieved by looking at the hit times for each threat, which appear in Table 4.4. For RF threats, the hit time is the time from when the platform enters that threat's radar range until the time that the platform exits its weapon-system range. This is the time in which the threat must progress until the end of the guidance stage (or tracking in the case of artillery systems) and for a hit to occur before the platform escapes unscathed. As such, this time should be longer than the average time taken for the entire progression to occur. The time required to progress to the end of the tracking stage is simply the total of the average search, acquisition, and tracking times, and is equal to 24 s. The guidance time of a threat is variable and depends on the projectile velocity of the threat, its position relative to the platform's path, and the time when the missile is fired. Due to the fact that this process is simply used to check the values chosen, an approximation is sufficient. If the platform passes directly over the slowest projectile speed threat (1400 km/h or 389 m/s) at an altitude of 8 km (26247 ft) (the altitude of the aircraft throughout the most dangerous part of the mission), the guidance time of the missile will be 20.57 s. Rounding this value up to a conservative 24 s results in a total average progression time of 48 s in the absence of countermeasures. As such, it is seen that each threat type meets this minimum, except for threat ID 8 and 9. However, both these threats are artillery-type systems and hence only require 24 s to hit the platform due to the exclusion of the guidance stage for these threats. Further, for threat ID 8 this is as expected due to the design of the scenario that intentionally placed this threat in that location. In the case of IR threats, the hit time is simply the time in which the platform is within the weapon system range. Further, IR threats do not undergo radar stage progression and are able to fire immediately once the platform enters their weapon-system range. As such, the IR hit times show that threat IDs 2 and 3 will potentially be able to fire twice, and threat IDs 1 and 4 will potentially be able to fire 3 times due to their universal reload time of 20 s.

**Table 4.4** RF and IR threat hit times.

| ID | Time (s) | |
|---|---|---|
| | **RF** | **IR** |
| 1 | 51 | 41 |
| 2 | 48 | 28 |
| 3 | 50 | 25 |
| 4 | 72 | 44 |
| 5 | 63 | n/a |
| 6 | 77 | n/a |
| 7 | 59 | n/a |
| 8 | 20 | n/a |
| 9 | 38 | n/a |

## 4.3 STRATEGY ANALYSIS

The analysis of the generated results begins with a detailed breakdown of the countermeasure strategies developed for the previously discussed example scenario, along with the associated half scenario. As stated previously, this ability to explain the developed countermeasure strategies in the context of a mission is a very important part of the validation process of the developed model. Note that the half scenario is exactly the same as the first except that it contains only half of the threats: RF threat IDs 2, 4, 5, and 9, along with IR IDs 1 and 3. Importantly, this scenario shows the performance of the system in a less saturated environment. Finally, note that all results contained in this section, and in this work in general were obtained by running the developed program in MATLAB R2018b on a server with two 6-core Intel Xeon E5-2630 processors equipped with 32 GB of RAM.

### 4.3.1 Full scenario

Running the algorithm a total of 30 times for the full scenario, using the standard set of conservative weights, and using the maximum iterations stop criterion set to 150 iterations, results in the performance statistics contained in Table 4.5 for the four objective functions, the total fitness, the number of iterations required to find a successful solution to the scenario, and the total run time. Note that due to the stochastic nature of the genetic algorithm, it does not always generate a successful solution to the scenario in the allotted time. As such, the number of times the algorithm failed to generate a successful solution is contained in the right-most column. Further, these failed solutions do not have a known number of iterations required to solve them, and also

have a disproportionately poor fitness value, which affects the presented statistics. In particular, the maximum values, the associated range value, and the overall average are affected, and hence are repeated for the successful solutions only. On the other hand, the minimum is completely unaffected, whilst the median is relatively unaffected due to its robustness to outliers. As such, these have not been repeated. Lastly, the fittest solution was selected for in-depth analysis, and its individual performance is included in this table for comparison purposes. Note that this fittest solution was selected for analysis due to its relative lack of stochastic-optimisation artefacts. However, any of the successful solutions could be used.

**Table 4.5** Performance statistics of the full scenario.

| Stat. | Prob. of Hit | Danger Score | Cost Score | EMCON Score | Total Fitness | Iterations to Solve | Run Time (s) | No. Failed |
|---|---|---|---|---|---|---|---|---|
| **Ave.** | 0.1150 | 0.0735 | 0.1518 | 0.3069 | 0.0976 | n/a | 793.82 | |
| **Med.** | 0.0000 | 0.0736 | 0.1455 | 0.3142 | 0.0670 | 6.00 | 794.71 | |
| **Min.** | 0.0000 | 0.0697 | 0.1455 | 0.2240 | 0.0652 | 2.00 | 771.12 | |
| **Max.** | 0.7125 | 0.0771 | 0.1818 | 0.3607 | 0.2563 | n/a | 821.32 | |
| **Ran.** | 0.7125 | 0.0074 | 0.0363 | 0.1367 | 0.1911 | n/a | 50.20 | |
| **Successful Only:** | | | | | | | | 5 |
| **Ave.** | 0.0000 | 0.0733 | 0.1531 | 0.3124 | 0.0671 | 9.32 | 794.02 | |
| **Max.** | 0.0000 | 0.0771 | 0.1818 | 0.3607 | 0.0695 | 102.00 | 821.32 | |
| **Ran.** | 0.0000 | 0.0074 | 0.0363 | 0.1148 | 0.0043 | 100.00 | 50.20 | |
| **Selected Solution:** | | | | | | | | |
| **Sel.** | 0.0000 | 0.0707 | 0.1455 | 0.3224 | 0.0652 | 5 | 817.63 | |

It is seen in the above table that whilst the selected solution has the best overall fitness, it actually has a worse than average EMCON score, and a median cost score and probability of hit. However, it has a very strong danger score, which means that this strategy in particular is a conservative one that favours using more countermeasures in order to expose the platform to less danger. It also took slightly fewer intervals to solve than the median, and its run time is also slightly more than average. Further, it is noted that its cost score of 0.1455 indicates that a total of 16 cartridges out of the platform's total capacity of 110 have been allocated. Most importantly, this strategy results in the platform being able to complete the mission unharmed.

Next, the developed countermeasure strategy must be analysed in the context of the mission. In order to do so, the strategy and its interaction with the radar stage progression of the threats must be displayed in an easy-to-read format. This is especially important in decision-support systems and training applications, where the information must be displayed in such a way as to instil confidence in a user, and allow them to be able to gleam as much information as possible at a

glance. Further, the information display must be as intuitive as possible so as to reduce the time required to train the user to use the software. As such, there will be strong focus on the display methods used in this work.

The first of these display methods shows the countermeasure strategy itself by depicting the countermeasure allocations for each of the platform's channels on a time-interval-by-time-interval basis, along with the towed decoy allocation. Each countermeasure allocation consists of the allocated technique, the threat type for which it is optimised, and the antenna direction where applicable. Further, the omnidirectional towed decoy can be allocated to either of the two active channels, where that channel's antenna direction is set to a 'D' accordingly. Importantly, each countermeasure allocation is colour-coded such that countermeasures directed at search-type stages are coloured using cool colours such as blue and green, whilst countermeasures directed at tracking-type stages are coloured using warm colours such as red and orange. This helps the reader differentiate between the actions of the techniques, and allows them to better see the overall approach of the mission. It also helps differentiate between the relative urgency or importance of the countermeasure allocations, where a warm-coloured technique would be necessary to break the lock of a threat that is far along its engagement procedure, and hence close to hitting the platform. On the other hand, a cool-coloured technique would be performing more of a delaying action on a threat that is not far along in its engagement procedure. However, it is noted that both categories of techniques can be essential to a countermeasure strategy, where the removal of either can result in the failure of that strategy. Specifically, in this work NJ has been coloured green, CP has been coloured blue, whilst RGPO has been coloured orange, and VGPO has been coloured red. Separately, flares have been coloured purple to indicate their independence from the RF countermeasures. Finally, dead time intervals during which a channel is changing its allocated countermeasure are greyed out so as to indicate that channel's inactivity. The countermeasure-allocation table for the selected strategy appears in Table 4.6.

The second of these display methods shows the effect of the developed countermeasure strategy on the stage progression of the threats. Specifically, the stage of each threat is shown on a time-interval-by-time-interval basis. Similarly to the countermeasure allocation table, search-type stages are coloured using cool colours, whilst tracking-type stages are coloured using warm colours. This helps the reader to immediately identify the increasing level of danger presented by a threat as it progresses through its radar stages, and gets closer to firing upon and hitting the platform. Further,

this helps the reader to visually see the how saturated the threat environment is at any point in time, as well as which threats need to be prioritised. Specifically, in this work the acquisition stage (A) has been coloured blue, the tracking stage (T) has been coloured orange, and the guidance stage (G) has been coloured red. The guidance stage of IR threats has also been coloured red. Lastly, it is noted that both the search (S) and undetected (U) stages have been left uncoloured due to their wide proliferation. This makes the table less crowded, and easier to read. The threat stage table for the selected strategy appears in Table 4.7 below.

**Table 4.6** Countermeasure allocation table for the selected strategy.

| Time (s) | Active Channel 1 | | | Active Channel 2 | | | Passive Channel | | Decoy |
|---|---|---|---|---|---|---|---|---|---|
| | Tech | Threat | Antenna(°) | Tech | Threat | Antenna(°) | Tech | Threat | |
| 0 | CP | 10 | 0 | CP | 8 | 0 | | | |
| 2 | None | | | | | | | | |
| 4 | Change | | | | | | | | |
| 6 | CP | 10 | 0 | | | | | | |
| 8 | | | | | | | | | |
| 10 | | | | | | | | | |
| 12 | | | | | | | None | | |
| 14 | | | | | | | | | |
| 16 | None | | | | | | | | |
| 18 | | | | | | | | | |
| 20 | Change | | | None | | | | | |
| 22 | CP | 1 | 18 | Change | | | | | |
| 24 | | | | CP | 8 | 354 | | | None |
| 26 | | | | | | | Change | | |
| 28 | None | | | | | | Flare | 1 | |
| 30 | Change | | | | | | | | |
| 32 | CP | 1 | 24 | | | | | | |
| 34 | | | | None | | | | | |
| 36 | None | | | | | | | | |
| 38 | | | | | | | | | |
| 40 | Change | | | Change | | | None | | |
| 42 | VGPO | 1 | 0 | RGPO | 8 | 132 | | | |
| 44 | | | | | | | | | |
| 46 | | | | None | | | | | |
| 48 | | | | Change | | | | | |
| 50 | | | | RGPO | 8 | D | Change | | 2 |
| 52 | | | | | | | Flare | 1 | |
| 54 | | | | | | | | | |
| 56 | | | | None | | | | | None |
| 58 | None | | | | | | None | | |
| 60 | | | | Change | | | | | |
| 62 | | | | RGPO | 3 | D | Change | | 2 |
| 64 | | | | | | | Flare | 4 | |
| 66 | | | | Change | | | | | None |
| 68 | | | | RGPO | 7 | D | None | | 2 |
| 70 | | | | | | | | | |
| 72 | Change | | | | | | Change | | |
| 74 | RGPO | 1 | 156 | | | | Flare | 3 | |
| 76 | | | | | | | | | |
| 78 | | | | | | | | | None |
| 80 | None | | | | | | | | |
| 82 | | | | | | | | | |
| 84 | | | | | | | | | |
| 86 | Change | | | | | | | | |
| 88 | VGPO | 4 | D | | | | | | 1 |
| 90 | | | | | | | | | |
| 92 | | | | | | | | | |
| 94 | | | | | | | | | |
| 96 | | | | None | | | | | |
| 98 | | | | | | | | | |
| 100 | | | | | | | | | |
| 102 | | | | | | | | | |
| 104 | | | | | | | | | |
| 106 | None | | | | | | | | None |
| 108 | | | | | | | | | |
| 110 | | | | | | | | | |
| 112 | | | | | | | | | |
| 114 | | | | | | | | | |
| 116 | | | | | | | | | |
| 118 | | | | | | | | | |
| 120 | | | | | | | | | |

**Table 4.7** Threat stages for the selected strategy.

| Time (s) | RF Threat ID | | | | | | | | | IR Threat ID | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 |
| 0 | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| 10 | | S | | | | S | | | | | | | |
| 12 | S | | | | | | | | | U | | | |
| 14 | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | |
| 18 | | | | | | | | | S | | | | |
| 20 | | | S | S | | | | | | | U | | |
| 22 | | A | | | | A | | | | | | | |
| 24 | | | | | | | | | | | | | |
| 26 | | | | | S | | | | | G | | | |
| 28 | A | T | | | | | S | | | | | | |
| 30 | | | | | | | | | | | | U | |
| 32 | | | | | | | | | | | | | |
| 34 | | | | | | T | | S | | | | | U |
| 36 | T | G | | | | | | | | | | | |
| 38 | | | | | | | | | | | | | |
| 40 | | | | | | | | | A | U | | | |
| 42 | | | | | | G | | | | | | | |
| 44 | | | A | A | | | | | | | G | | |
| 46 | | | | | | | | | T | | | | |
| 48 | S | | | | | | | | | G | | | |
| 50 | | | T | T | | S | | | | | | | |
| 52 | | T | | | | | | | | | | | |
| 54 | A | | | | A | A | | | | | | | |
| 56 | | | | | | | | | | | | | |
| 58 | | | G | G | | | | | S | | | | |
| 60 | | | | | | | A | | | | | | |
| 62 | | | | | T | T | | | | | | G | |
| 64 | | | | | | | | | A | | | | |
| 66 | | | | | | | T | | | | | | |
| 68 | | | | | | | | | T | | | | |
| 70 | | | S | S | | | | | | | | | |
| 72 | | | | | | G | | A | | | | | G |
| 74 | | | | | | | | | | | | | |
| 76 | | | A | A | S | | G | | | | | U | |
| 78 | | | | | | | | T | | | | | |
| 80 | | | | | | | | | | | | | |
| 82 | | | T | T | A | S | | | | | | | |
| 84 | | | | | | | | | | | | | |
| 86 | | | | | | | | | | | | G | U |
| 88 | | S | | | T | A | | | | U | U | | |
| 90 | S | | | | | | | | | | | | |
| 92 | | | | | | | | | | | | | |
| 94 | | | | | | T | | | | | | | |
| 96 | | | | S | | | S | | S | | | | |
| 98 | | | | | G | | | | | | | | G |
| 100 | | | | | | | | | | | | | |
| 102 | | | | A | | | A | | | | | | |
| 104 | | | | | | | | S | | | | U | |
| 106 | | | S | | | | | | | | | | |
| 108 | | | | | | S | | | | | | | |
| 110 | | | | | T | | T | | | | | | |
| 112 | | | | S | | | | | | | | | |
| 114 | | | | | | | | | | | | | |
| 116 | | | | | | | | | | | | | |
| 118 | | | | | S | | S | | | | | | U |
| 120 | | | | | | | | | | | | | |

It is seen that the overall approach of the strategy is to initially delay detection of the platform using CP techniques while the threat environment is relatively unsaturated. Thereafter, once the platform is within range of a majority of the threats, it is no longer able to remain undetected, and hence it must resort to well-timed RGPO and VGPO techniques in combination with the towed decoy in order to break radar lock. Finally, towards the end of the mission the platform reaches a point where it will be able to outrun all the threats, and all countermeasures are dropped. On the other hand, IR threats are mostly just countered immediately after they have fired upon the platform, unless it is known that the platform will be able to outrun the missiles. Obviously, the strategy of relying on the platform's ability to leave the weapon range of a threat before its missile hits the platform (in other words to outrun it) is extremely risky, but it is clearly a more efficient use of countermeasure resources. Further, it can be beneficial in circumstances where the necessary jamming would illuminate the platform to either known, or unknown threats, and hence place the platform in greater danger. The risk of this approach can be reduced by using conservative estimates of threat ranges, as well as projectile and platform velocities in the model. Importantly, this highlights one of the decision-support and training applications of this system, where not immediately obvious strategies can be brought to the attention of a human user.

Further, it is noted that there is a distinct lack MFT and chaff countermeasures in this strategy, and all other strategies shown in this work. Both can be attributed to the optimisation procedure, where there is a distinct bias towards using NJ and CP against search-type stages, and RGPO and VGPO against tracking-type stages in both the procedural seeding techniques and the survival operator. This is because, in general, these techniques are set to use the most effective countermeasure technique against a threat in each instance according to the stage effectiveness table, where CP is used against both search and acquisition stages, RGPO is used against the tracking stage, and VGPO is used against the guidance stage. It is noted that for the search and acquisition stages, both CP and MFT have the same stage effectiveness, but only one can be selected, and the algorithm chooses the first one in the table in the event of a tie, which is CP in this work. This also applies to the tracking stage, where RGPO and VGPO have the same effectiveness, but RGPO appears first in the table. Also, NJ is specifically used against search-type stages in the second procedural seeding technique due to its variable bandwidth. On the other hand, in these techniques chaff allocation is avoided in search stages, and only used in tracking stages as a last resort when there are multiple different threat types simultaneously in tracking-type stages. This bias is then hard to overcome due to both the dead time, and the break-lock time, which require countermeasures to be implemented

for a number of simultaneous time intervals before having an effect, or breaking the lock of a tracking-type stage. This means that the random processes in the algorithm need to generate MFT or chaff countermeasures in groups before they can have a positive impact on the fitness of a strategy, and reproduce. Further, the single instances of these countermeasures will often be removed by the cleaning operator before they manage to create such a group. This is obviously not ideal when trying to find the true optimal strategy, rather than just a local minimum. However, the positive impact these specialised operators have on both the fitness of the generated solutions, and the rate at which they are produced, far out-weighs this negative impact. Note that the effect of these specialised operators on the optimisation performance of the algorithm is covered in detail in Section 4.4.4.

Further, chaff use is also discouraged because cartridge use increases the cost objective function, whereas active techniques do not. As such, the use of active techniques is preferable to the algorithm. However, this in particular is not an issue in this work because IR threats can only be countered using cartridges, so it is actually preferable to save the platform's cartridge capacity for any unexpected IR threats.

Next, the countermeasure strategy and its interaction with threats can be analysed in greater depth by using a graphical display that depicts the entire threat environment on a time-interval-by-time-interval basis. To save space, only the time intervals during which significant RF jamming occurs will be examined, where the analysis will be performed on an allocation-by-allocation basis. The first set of these images appears in Figure 4.3 for the time intervals beginning 24, 26, 28 and 30 s into the mission. Note that each threat is depicted using a stage-dependent icon, whose size is scaled according to that threat's danger value, where a larger size indicates a greater danger and vice versa. This allows the reader to immediately determine how far along each threat is in their engagement procedure, and prioritise them. For RF threats, the search stage is depicted by a plus ('+'), acquisition is depicted using a cross ('×'), tracking is depicted using a star ('*'), and finally guidance is depicted using a square ('■'). On the other hand, for IR threats the undetected stage is depicted using a triangle ('Δ'), and guidance is depicted using a diamond ('◇'). Further, the colour of each icon is dependent on the jamming effect (E) of the countermeasures implemented in that time interval, where a positive jamming effect is indicated using green, and an illuminating effect is coloured red. Lastly, red triangles are used to indicate the locations of missiles that are in the guidance stage for all threat types. Also, the platform's two active jamming channels are depicted

using a pink arrow for the left-hand channel, and orange for the right-hand channel, where the arrow points in the platform-roll-corrected antenna direction. Note that these arrows are not displayed during time intervals in which no technique is allocated to that channel, including dead time intervals. Further, an appropriately coloured circle around the platform is used instead of a channel's arrow in instances where the towed decoy is allocated to that channel in order to indicate its omnidirectional nature. On the other hand, IR countermeasures are not depicted in this manner due to their lack of directionality in this work. Finally, the RF threats types that are the target of the examined time interval's countermeasures are further highlighted by an appropriately coloured pentagon, where this pentagon is left uncoloured when no decoy is used for that channel, or coloured if it is. On the other hand, IR threats that are the target of flares are highlighted by a coloured blue circle. Therefore, this display approach allows the reader to immediately see the threat environment and the effect of the implemented countermeasures in any given time interval.

Firstly, it is noted that the time intervals in the beginning 24 s of the mission during which the strategy calls for various CP techniques have not been shown in the interests of saving space. This is because in those time intervals, only threat IDs 1 and 2 are affected, where they are simply held in their search stage. Thereafter, it is seen that Figure 4.3 shows a combination of CP directed at threats type 1 and 8, along with flares directed at IR type 1 whose lock is broken at the end of the 30 s time interval. Further, it is also seen that the associated antenna directions are well chosen to aim towards the intended targets. Initially, the CP allocated to channel 2 has no effect on threat ID 9, due to the platform being out of range, but once the platform gets close enough in the 26 s time interval, it begins to successfully suppress threat ID 9 in addition to threat ID 1. Also in that time interval, IR threat ID 1 enters its guidance stage, causing the platform to begin firing flares in the next two time intervals in order to break its lock. Importantly, it is noted that this initial stage of CP techniques forms an important part of the strategy whereby the platform suppresses the first couple threats in order to reduce the difficulty of the threat environment in the middle of the mission. In particular, it is seen in Table 4.7 that threat ID 2 fires upon the platform and enters the guidance stage 36 s into the mission, but its lock is never broken. This is because the platform is able to outrun the missile. Therefore, this means that the CP techniques used in this strategy were purposely chosen so as to suppress the threat just enough that it could never hit the platform. Thereafter, it can basically be ignored by the platform, thus allowing the platform to focus its limited jamming resources on other threats.

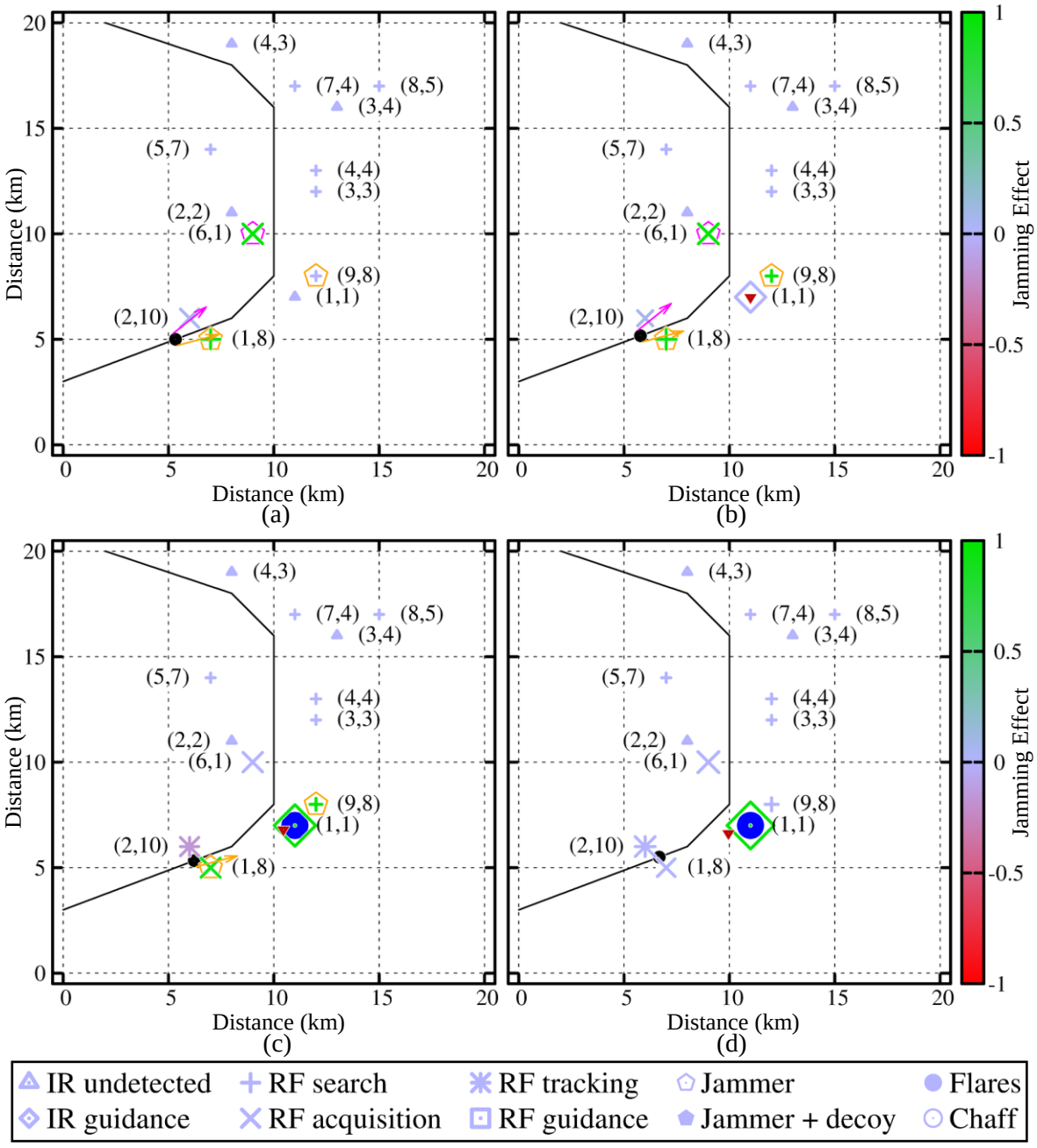



**Figure 4.3.** Scenario images for the time intervals starting 24 (a), 26 (b), 28 (c), and 30 s (d) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Next, the combined VGPO and RGPO techniques beginning in the 42 s time interval are shown in Figure 4.4. Here it is seen that VGPO is being used to break the lock of threat ID 6, which has entered the guidance stage, and RGPO is being used to break the tracking lock of threat ID 1.

Threat ID 2 is in its guidance stage as well, but as previously discussed, the platform can outrun the missile, hence it can be ignored. Further, IR threat ID 2, also enters its guidance stage in the 44 s time interval. However, it can be ignored for now because IR threat ID 1 will soon enter its guidance stage, and both can be seduced away from the platform simultaneously using flares optimised for IR threat ID 1 in the 52 s time interval, due to IR threat ID 2's relatively low countermeasure resistance. As before, this clearly is a risky approach, but it results in a more efficient use of the platform's limited resources. Further, it is seen that the VGPO allocated to the first channel is slightly illuminating the platform to threat IDs 3 and 4 due to them both being in the search stage and because they are relatively close in frequency to threat ID 1. In comparison, the RGPO allocated to channel 2 isn't illuminating the platform to threat ID 9, even though it is of the same threat type, because the channel's antenna is directed backwards. Lastly, it is noted that the antenna angles of these two techniques are not perfect, but they are in the general area. This is simply an effect of the optimisation process.

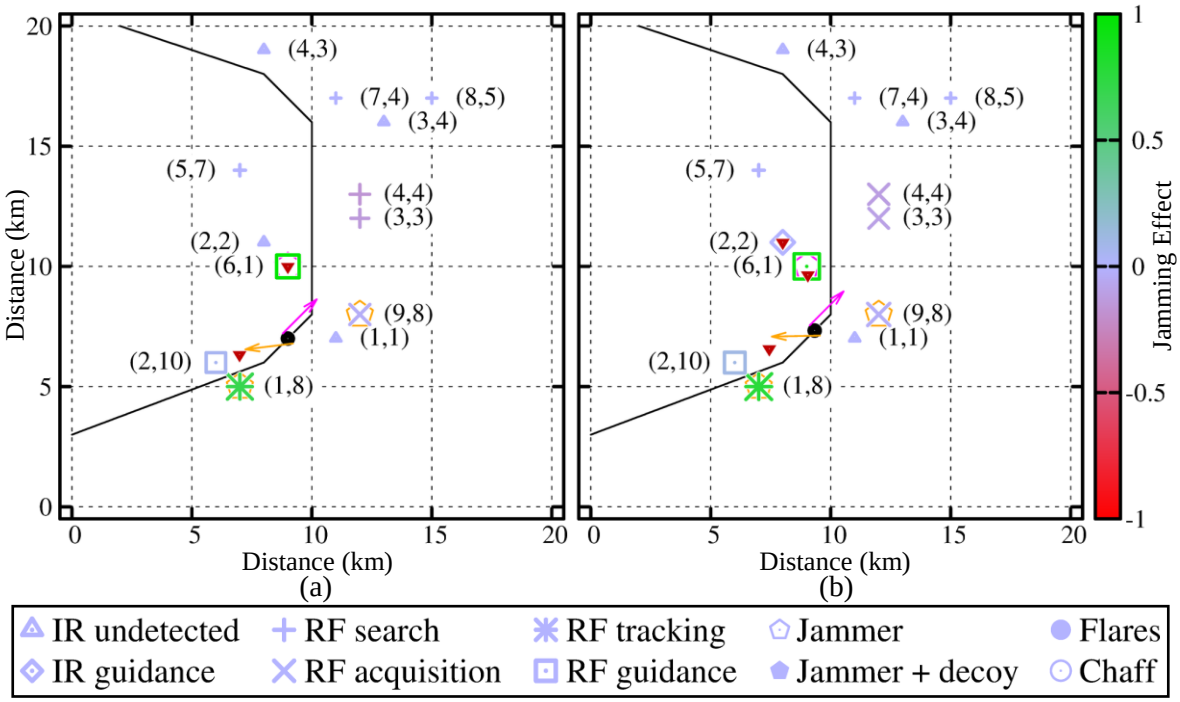



**Figure 4.4.** Scenario images for the time intervals starting 42 (a), and 44 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

The next set of images in Figure 4.5 show the use of RGPO optimised for threat type 8 directed through the towed decoy in the 50 and 52 s time intervals, even though RGPO optimised for the

same threat type was allocated to the same channel just 4 s earlier. However, threat ID 9 was only in the acquisition stage at the time, and threat ID 1, an artillery type, was close to the end of its tracking stage and hence needed its lock broken imminently. Further, as a consequence of this the use of the omnidirectional towed decoy in this time interval results in the platform illuminating itself to the newly searching threat ID 1, along with threat ID 5. However, the use of the decoy is necessary in order to create enough separation from the platform, and resultant angular error in order to generate a jamming effect sufficient enough to break the lock of threat ID 9. Further, the platform will leave the range of threat ID 1, before it is able to fire, hence meaning that its illumination is relatively inconsequential. It is also noted that the RGPO technique also has a weak positive jamming effect on threat ID 2, although this has little effect due to the platform outrunning its missile, as seen in this Figure. Lastly, it is also seen that IR threat ID 1 fires upon the platform in the 52 s time interval, causing the platform to use flares in the next time interval that simultaneously seduces it and IR threat ID 2, as discussed previously.

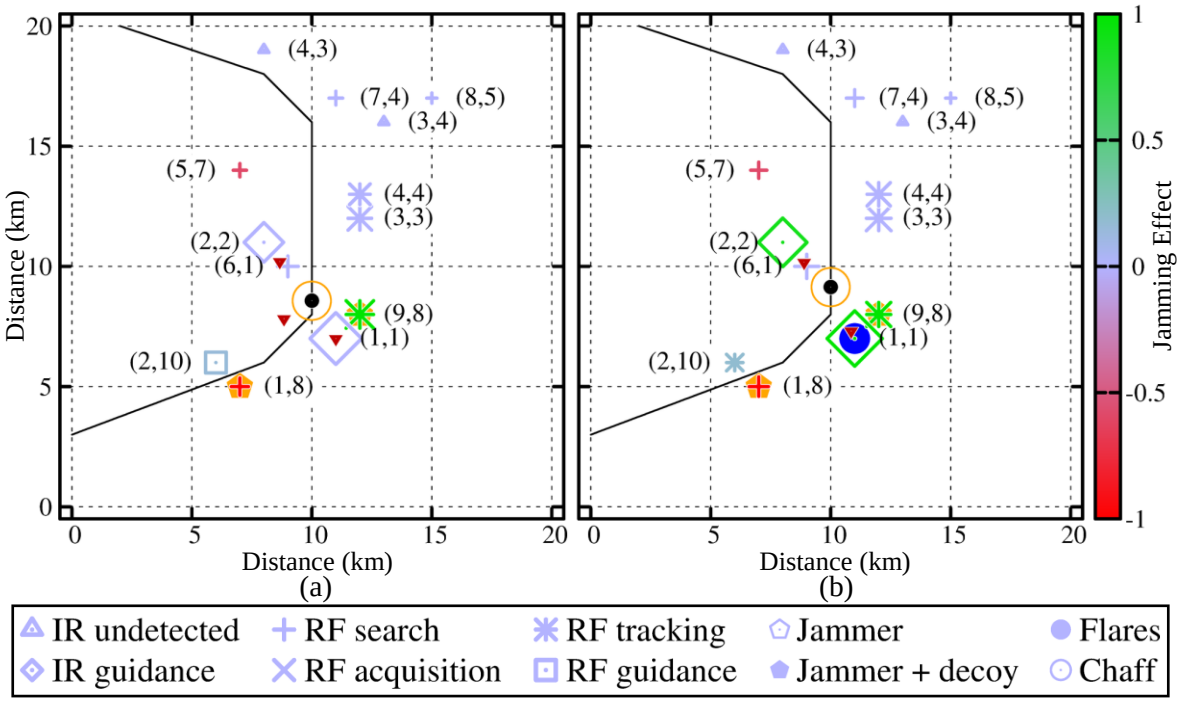



**Figure 4.5.** Scenario images for the time intervals starting 50 (a), and 52 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Figure 4.6 then depicts the time intervals starting 62 and 64 s into the mission, where the strategy calls for RGPO optimised for threat type 3 and directed though the towed decoy. This results in the

break of guidance-lock of both threat IDs 3 and 4, due to their use of the same frequency band, where the use of the towed decoy is necessary in order to break lock due to the fact that the platform is passing very close to these threats, and is unable to change its antenna direction fast enough. In particular, it is seen that in the first time interval, a forward-facing antenna direction would be required, whereas in the second time interval threat ID 3 would require a reverse-facing antenna direction. The omnidirectionality of the towed decoy also results in the minor suppression of threat IDs 5 and 6. This can be attributed to the RGPO being relatively close to threat ID 6's frequency band, and threat ID 7's wider than usual frequency band. Further, this countermeasure allocation results in the illumination of the platform to threat IDs 7 and 8, where the former operates in the same frequency band, and the later in an adjacent frequency band. Note that the platform only enters the range of threat ID 8, and hence illuminates itself to it, in the 64 s time interval. However, the illumination of this threat is not an issue as it is never able to fire upon the platform. On the other hand, IR threat ID 3 fires upon the platform in the 62 s time interval, causing the platform to fire flares in the following time interval.

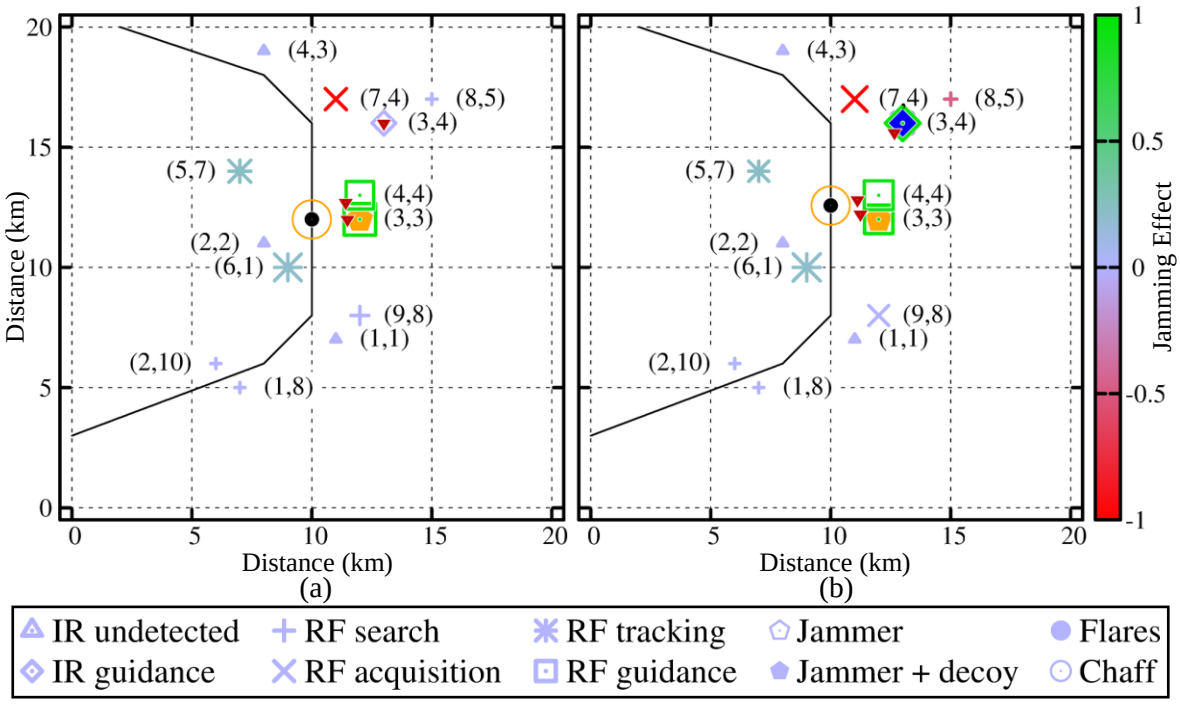



**Figure 4.6.** Scenario images for the time intervals starting 62 (a), and 64 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Thereafter, Figure 4.7 depicts the 68 and 70 s time intervals during which the strategy calls for a RGPO technique optimised for threat type 7 directed through the towed decoy in order to break the lock of threat ID 5. It is noted that the decoy is also necessary in this case in order to break lock. Again, this due to the fact that the platform is in the process of passing the threat as it is attempting to jam it. However, the unintended consequences of this decoy use are not entirely detrimental or severe. The worst being the fact that it strongly illuminates the platform to threat ID 9 while it is in the acquisition stage due to the fact that its threat type is frequency adjacent to the extra-wide bandwidth threat type 7. The RGPO then actually has a relatively strong suppressing effect when that threat progresses to the tracking stage. However, the platform then exits the range of that threat in the very next time interval, resulting in the illumination not having any real consequence. Further, this countermeasure allocation has a light suppressing effect on threat ID 7, a moderate illuminating effect on ID 8, and a weak illuminating effect of IDs 3 and 4. This is good considering the danger presented by threat ID 7, and the relatively lower danger presented by IDs 3, 4 and 5. In fact, overall it is seen that both threat IDs 6 and 7 are nearing the end of their engagement procedure and present a large danger to the platform. This is confirmed in the next figure where both of these threats have entered the guidance stage, and threat ID 6 is being jammed.

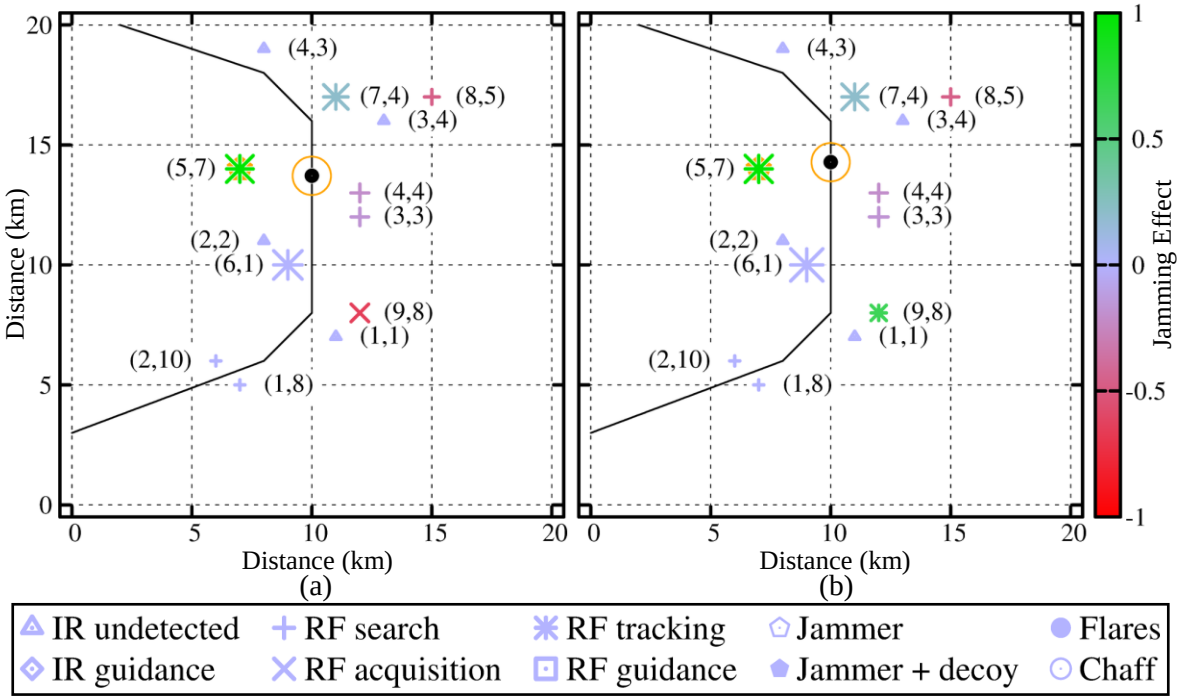



**Figure 4.7.** Scenario images for the time intervals starting 68 (a), and 70 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Figure 4.8 shows the 74 and 76 s time intervals, which are just after threat ID 6 has entered the guidance stage and needs to be countered using RGPO allocated to the left-hand jamming channel. Note that the antenna direction has been specifically chosen so as to account for the platform turning in the middle of this countermeasure, where threat ID 6 sits directly between the two different antenna directions. It does result in some minor illumination of threat IDs 3 and 4, especially in the 76 s time interval when the antenna is pointed towards them. However, this is a small price to pay in order to break the lock of threat ID 6. Lastly it is also noted that IR threat ID 4 also entered guidance in the previous time interval, and hence is seduced away using flares in the two time intervals depicted here.

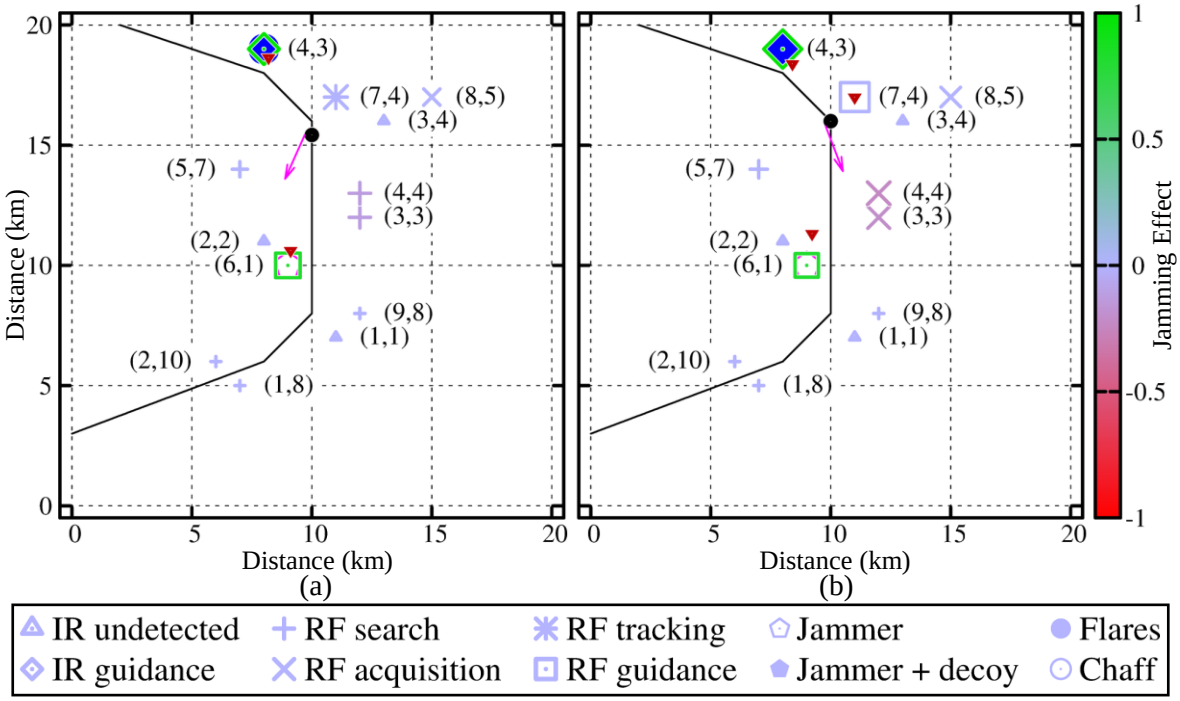



**Figure 4.8.** Scenario images for the time intervals starting 74 (a), and 76 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Finally, Figure 4.9 depicts the final countermeasure allocation of the strategy in the 88 and 90 s time intervals. Specifically, the allocated countermeasure is VGPO optimised for threat type 4, and emitted through the towed decoy in order to break the guidance lock of threat ID 7, and the tracking locks of threats IDs 3 and 4. In this case, the decoy use is almost purely beneficial as it firstly allows for a strong positive effect on all three targets that are at different angles to the platform, as well as a moderate suppressing effect on threat ID 5, which is in the tracking stage.

The only drawback is the slight illumination of the platform to threat ID 6, but the platform leaves its range shortly thereafter, before it is able to fire. Importantly, it must be noted that the strategy specifically calls for this very late jamming of threat ID 7. This is because the platform exited the range of threat ID 8 in the previous time interval. This artillery-type threat was on the verge of reaching the end of its tracking stage, and any more illumination of the platform would have allowed it to fire upon the platform. Further, this approach has the added benefit of not giving threat ID 7 enough time to fire upon the platform again before the end of the mission. Leaving the jamming of threat ID 7 this late is obviously very risky, but theoretically necessary for the survival of the platform. As stated previously, the risk associated with this can be reduced by ensuring the use of conservative weapon ranges, projectile velocities, and platform velocity. Lastly, it is noted that hereafter both threat ID 5 and IR threat ID 4 fire missiles at the platform, but the platform escapes their range before they hit. This also applies to IR threat ID 3 which fired in the previous time interval that began 86 s into the mission, where the platform exited the range of that threat in the very next time interval.

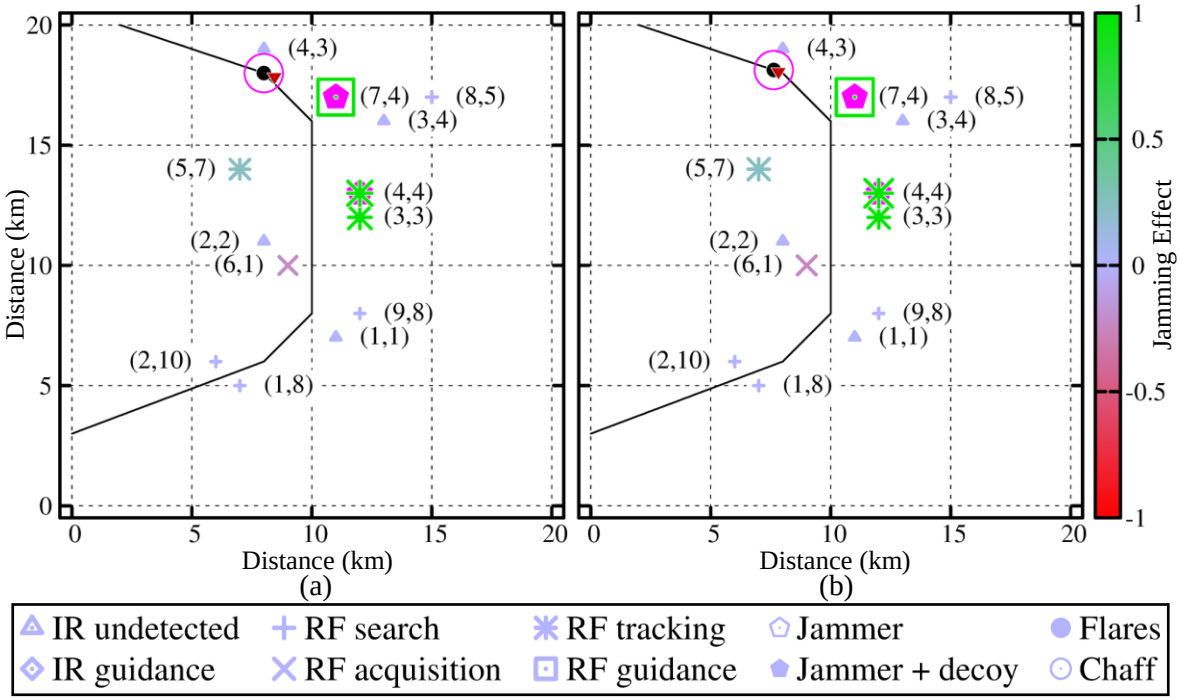



**Figure 4.9.** Scenario images for the time intervals starting 88 (a), and 90 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Lastly, the scenario was purposely designed with a number of situations that test the algorithm's approach to countermeasure allocation. The first of which is the inclusion of the two similar threats right next to each other, which in this case are threat IDs 3, and 4 that actually use the same frequency band. Looking at Table 4.7, it would appear that these two threats simply progress simultaneously, which in turn allows them to simply be jammed simultaneously. However, when all countermeasures are removed, these two threats do not progress simultaneously. They begin only 2 s out of synchronisation, but this worsens over the course of the mission to a peak of 6 s, resulting in threat ID 4 being able to fire upon the platform a whole extra time. As such, it is seen that the countermeasures in the first 44 s align the two threats' radar stage progressions, and then thereafter they are jammed simultaneously. This allows them to almost be treated as a single threat, which in turn allows them to continue to be simultaneously jammed, which is the most effective approach.

The second of these situations was the inclusion of threat ID 8. This threat is an artillery type that is placed so that the platform is only within its range for 20 s, when the threat requires 24 s to fire upon the platform. Further, it was placed so that it is surrounded by frequency-adjacent threats, thus making it easy to accidentally illuminate the platform to this threat. This in turn required careful planning of the jamming of these frequency-adjacent threats so as to minimise this. It was seen in the analysis so far that the platform was indeed illuminated to this threat while jamming others. However, it was unable to fire upon the platform. This essentially saw a trade-off where the platform had to make the risky choice of jamming threat ID 7 in the very last possible second in order to jam threat IDs 3, 4, and 5 along the way. This was clearly very risky, but necessary in order to generate an efficient countermeasure strategy.

The last of these situations was the inclusion of threat IDs 1 and 2, which are two threats operating in similar frequency bands that were placed directly either side of the platform's path. The idea is that to the platform, these two threats will appear next to one another at long distances as the platform approaches. Then, as the platform passes between them, the two threats separate drastically from its perspective, before coming together again as it moves further away. It is seen that the strategy counters them simultaneously in the beginning of the mission using CP techniques directed directly towards the front of the platform. Importantly, this approach means that the two channels work together as the positive cross effect from each channel combines with the main effect of the other channel due so as to increase the jamming effect. Thereafter, as the platform

approaches, the strategy begins to follow the increasing angle of these threats by setting the antenna direction of the second channel to 354°. However, once the platform gets too close to these threats and actually passes between them, the jamming channels cannot change direction fast enough, and the platform chooses to stop jamming these two threats. This is also seen in Figure 4.3. Thereafter, the strategy turns its focus to other threats. This is in part because the other threats are more pressing, but also because threat ID 2 has been sufficiently suppressed such that it no longer presents a danger to the platform and can be essentially ignored.

### 4.3.2 Half scenario

The very saturated threat environment of the full scenario mostly requires a strategy that jams threats in tracking-type stages using RGPO and VGPO, rather than making use of the various other countermeasures available to the platform. As such, for comparison purposes the results for a less-saturated scenario that contains half of the full scenario's threats are presented here. This scenario contains RF threat IDs 2, 4, 5, and 9, along with IR IDs 1 and 3 from the first scenario, where their characteristics are completely unchanged. However, note that their ID numbers have been changed to reflect the number of threats in the scenario, but keep the same numerical order. As such, they will be identified here as RF threat IDs 1, 2, 3 and 4, as well as IR IDs 1 and 2, respectively.

The results of running the algorithm 30 times for this scenario are presented in Table 4.8 below using a maximum iterations stop criterion of 150 iterations. Importantly, note that neither of the procedural seeding techniques were able to generate a successful solution to the full scenario, hence requiring the algorithm to run for a median 6 generations before such a solution was found. In comparison, the second procedural seeding technique does generate a successful solution to the half scenario, albeit a relatively weak one. As such, the number of failed solutions has been excluded from the table along with the number of iterations required to generate a successful solution and the successful only statistics. Further, the successful solutions mean that the probability of the platform being hit was zero for all solutions, and as such this has also been excluded.

**Table 4.8** Performance statistics of the half scenario.

| Statistic | Danger Score | Cost Score | EMCON Score | Total Fitness | Run Time (s) |
|---|---|---|---|---|---|
| **Average** | 0.0487 | 0.0909 | 0.3357 | 0.0481 | 761.63 |
| **Median** | 0.0480 | 0.0909 | 0.3443 | 0.0479 | 762.72 |
| **Minimum** | 0.0452 | 0.0909 | 0.2459 | 0.0470 | 734.51 |
| **Maximum** | 0.0547 | 0.0909 | 0.3880 | 0.0498 | 787.79 |
| **Range** | 0.0095 | 0.0000 | 0.1421 | 0.0028 | 53.28 |
| **Selected Solution:** | | | | | |
| **Selected** | 0.0456 | 0.0909 | 0.3607 | 0.0470 | 762.51 |

Again, the fittest solution was selected for demonstration purposes due to its relative lack of stochastic optimisation artefacts, but any of the generated solutions could be used. It is seen that this selected strategy also has an above average EMCON score, but below average danger score. This in turn means that the strategy is a conservative one that uses more countermeasures in order to expose the platform to less danger over the course of the mission. It was also generated in an average amount of time, which is less than that of the full scenario. This is because the danger values and jamming factors need to be calculated for fewer threats in each fitness calculation. Lastly, the cost score of 0.0909 for this strategy indicates that it requires 10 cartridges out of the platform's total capacity of 110 cartridges. Most importantly, this strategy results in the platform escaping the mission unharmed.

The strategy itself appears in Table 4.9, and the resultant threat stages in Table 4.10 below that. Here it is immediately seen that in comparison to the full scenario, this strategy makes extensive use of various bandwidths of NJ in order to delay detection of the platform by search-type radars. Although, there are still no MFT or chaff techniques due to the reasons already discussed. Overall, the approach is to use CP techniques in the beginning of the mission in order to delay detection of the platform, where it is noted that the very first CP technique is used while out of range of all threats and is simply an artefact of the stochastic optimisation process. However, this is insufficient to completely prevent detection, and threat ID 1 enters the tracking stage, requiring its lock to be broken by a VGPO technique. Here the platform switches to various NJ techniques to both suppress the search and acquisition stage threats, and counteract the illumination caused by the VGPO and RGPO techniques required to break lock of the other threats as they progress. Then towards the end of the mission, the strategy calls for a combination of RGPO and VGPO in order to break the tracking and guidance lock of threat IDs 2 and 3, before then using one last NJ technique to protect the platform as it escapes. Note that this approach only allows for a single instance of RF

guidance, and in general breaks the lock of tracking-stage threats immediately. This results in the threats spending very little time in tracking-type stages, hence exposing the platform to less danger. Finally, it is noted that IR threats are simply countered as soon as they enter the guidance stage using the appropriate flare technique, except for the missile fired by IR threat ID 2 86 s into the mission. This is because the platform is able to outrun that missile.

Separately, it is noted that this strategy calls for a break in RF countermeasures in a relatively long period from 64 to 80 s into the mission, which is what allows threat ID 3 to enter the guidance stage. The reason for this is that the platform has already left the range of threat ID 1, and will leave the range of threat ID 4 during this period. As such, fewer threats need to be juggled by the platform. Further, this allows for the simultaneous jamming of threat IDs 2 and 3 immediately thereafter, which is very beneficial in terms of EMCON as it only requires the platform to emit RF signals for 4 s out of a total period of 28 s.

**Table 4.9** Countermeasure allocation table for the selected strategy.

| Time | Active Channel 1 | | | Active Channel 2 | | | Passive Channel | | Decoy |
|---|---|---|---|---|---|---|---|---|---|
| | Tech | Threat | Antenna(°) | Tech | Threat | Antenna(°) | Tech | Threat | |
| 0 | CP | 1 | D | | | | | | 1 |
| 2 | None | | | | | | | | |
| 4 | | | | | | | | | |
| 6 | Change | | | | | | | | |
| 8 | CP | 10 | 0 | | | | | | |
| 10 | | | | None | | | | | |
| 12 | | | | | | | None | | |
| 14 | | | | | | | | | |
| 16 | | | | | | | | | |
| 18 | | | | | | | | | None |
| 20 | | | | | | | | | |
| 22 | | | | | | | | | |
| 24 | | | | Change | | | | | |
| 26 | | | | CP | 8 | 0 | Change | | |
| 28 | | | | | | | Flare | 1 | |
| 30 | None | | | None | | | | | |
| 32 | | | | | | | | | |
| 34 | Change | | | Change | | | | | |
| 36 | VGPO | 10 | D | NJ (M) | 6 | 6 | | | 1 |
| 38 | | | | | | | | | |
| 40 | Change | | | Change | | | None | | |
| 42 | NJ (M) | 6 | 42 | NJ (MN) | 9 | 42 | | | |
| 44 | | | | | | | | | None |
| 46 | | | | | | | | | |
| 48 | Change | | | Change | | | | | |
| 50 | NJ (N) | 4 | 336 | RGPO | 8 | D | Change | | 2 |
| 52 | | | | | | | Flare | 1 | |
| 54 | Change | | | Change | | | | | None |
| 56 | NJ (MN) | 8 | 36 | RGPO | 4 | D | | | 2 |
| 58 | | | | | | | None | | |
| 60 | | | | | | | | | |
| 62 | | | | | | | Change | | |
| 64 | | | | | | | Flare | 4 | |
| 66 | | | | | | | | | |
| 68 | | | | None | | | | | None |
| 70 | None | | | | | | | | |
| 72 | | | | | | | | | |
| 74 | | | | | | | | | |
| 76 | | | | | | | | | |
| 78 | Change | | | Change | | | | | |
| 80 | VGPO | 7 | D | RGPO | 4 | 144 | | | 1 |
| 82 | | | | | | | | | |
| 84 | | | | | | | | | |
| 86 | None | | | | | | | | |
| 88 | | | | | | | | | |
| 90 | Change | | | | | | | | |
| 92 | NJ (MN) | 5 | 144 | | | | | | |
| 94 | | | | | | | None | | |
| 96 | | | | | | | | | |
| 98 | | | | | | | | | |
| 100 | | | | | | | | | |
| 102 | | | | None | | | | | None |
| 104 | | | | | | | | | |
| 106 | | | | | | | | | |
| 108 | | | | | | | | | |
| 110 | | | | | | | | | |
| 112 | None | | | | | | | | |
| 114 | | | | | | | | | |
| 116 | | | | | | | | | |
| 118 | | | | | | | | | |
| 120 | | | | | | | | | |

**Table 4.10** Threat stages for the selected strategy.

| Time (s) | RF Threat ID | | | | IR Threat ID | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 |
| 0 | | | | | | |
| 2 | | | | | | |
| 4 | | | | | | |
| 6 | | | | | | |
| 8 | | | | | | |
| 10 | | | | | | |
| 12 | S | | | | U | |
| 14 | | | | | | |
| 16 | | | | | | |
| 18 | | | | S | | |
| 20 | | | | | | |
| 22 | | S | | | | |
| 24 | | | | | | |
| 26 | A | | S | | G | |
| 28 | | | | | | |
| 30 | | | | | | U |
| 32 | | | | | | |
| 34 | | | | | | |
| 36 | T | | | | | |
| 38 | | | | | | |
| 40 | | | | A | U | |
| 42 | | | | | | |
| 44 | S | | | | | |
| 46 | | A | | | | |
| 48 | | | | | | |
| 50 | A | | | T | G | |
| 52 | | | | | | |
| 54 | | | | | | |
| 56 | T | T | A | | | |
| 58 | | | | S | | |
| 60 | | | | | | |
| 62 | | | | | | G |
| 64 | | S | | A | | |
| 66 | | | T | | | |
| 68 | | | | T | | |
| 70 | | A | | | | |
| 72 | | | | | | |
| 74 | | | G | | | |
| 76 | | T | | | | U |
| 78 | | | | | | |
| 80 | | | | | | |
| 82 | | | | | | |
| 84 | | | | | | |
| 86 | | | | | | G |
| 88 | S | | S | | U | |
| 90 | | S | | | | |
| 92 | | | | | | |
| 94 | | | | | | |
| 96 | | | A | S | | |
| 98 | | A | | | | |
| 100 | | | | | | |
| 102 | | | | | | |
| 104 | | | | | | U |
| 106 | | | T | | | |
| 108 | | | | | | |
| 110 | | | | | | |
| 112 | | S | | | | |
| 114 | | | | | | |
| 116 | | | | | | |
| 118 | | | S | | | |
| 120 | | | | | | |

Next, the approach of the strategy can be analysed in detail by looking at the scenario images. Again, it is very inefficient to show these images for every single time interval, so this analysis will be performed on a RF countermeasure-by-countermeasure basis. The first time intervals to be examined are the time intervals beginning 26 and 28 s into the mission, and their associated images appear in Figure 4.10 below. As before, this is simply because the first CP technique that begins 8 s into the mission just suppresses threat ID 1, with no further effects. In these time intervals, the platform uses CP directed towards threat types 8 and 10. It is seen that this has the effect of suppressing both threat IDs 1 and 4, whilst the platform is out of range of the remaining threats. Note that the antenna direction for channel 1 appears poor, but this is simply because this jamming technique began a number of time intervals before. Lastly, it is also seen that IR threat ID 1 fires upon the platform 26 s into the mission, and the platform responds with the appropriate flare technique in the very next time interval.

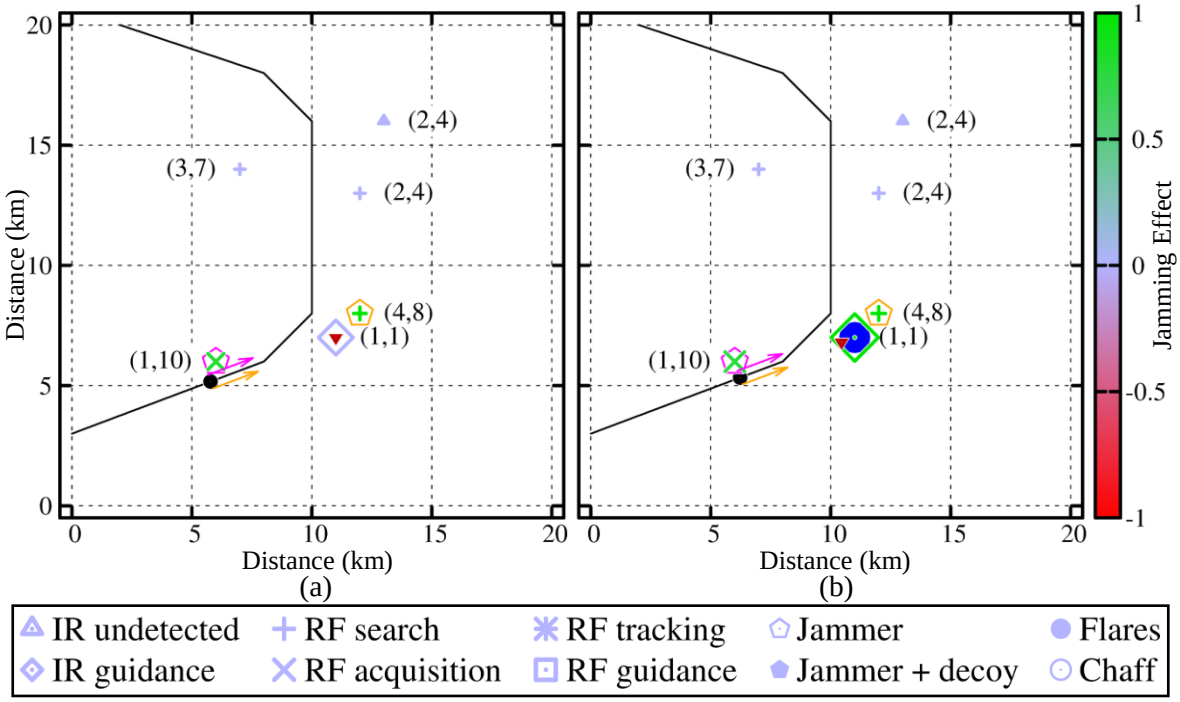



**Figure 4.10.** Scenario images for the time intervals starting 26 (a), and 28 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Next, the time intervals beginning 36 and 38 s into the mission are examined. Their associated images appear in Figure 4.11. In these intervals, the platform is using a combination of VGPO optimised for threat type 10 directed through the decoy, and medium-band NJ optimised for threat

type 6. Firstly, the decoy is needed to generate a jamming factor sufficient enough to break the tracking-lock of threat ID 1. This in turn illuminates the platform to threat ID 4, but this is counteracted by the NJ which has been specifically chosen to be directed directly between threat IDs 2 and 4, both in terms of antenna direction and frequency. In particular, medium-band noise jamming has been chosen because it has a 0.6 jamming effect on threat types 4, 5, 6, 7 and 8. As such, the platform is only slightly illuminated to threat ID 4 in the first interval, with that threat actually slightly suppressed in the second interval. Threat ID 2 is moderately suppressed in both these two intervals, whilst the remaining threat types are out of range. Lastly, it is noted that the forward-facing direction of the antenna means that there is no NJ signal emitted in the direction of threat ID 1, hence ensuring that there is no interference.

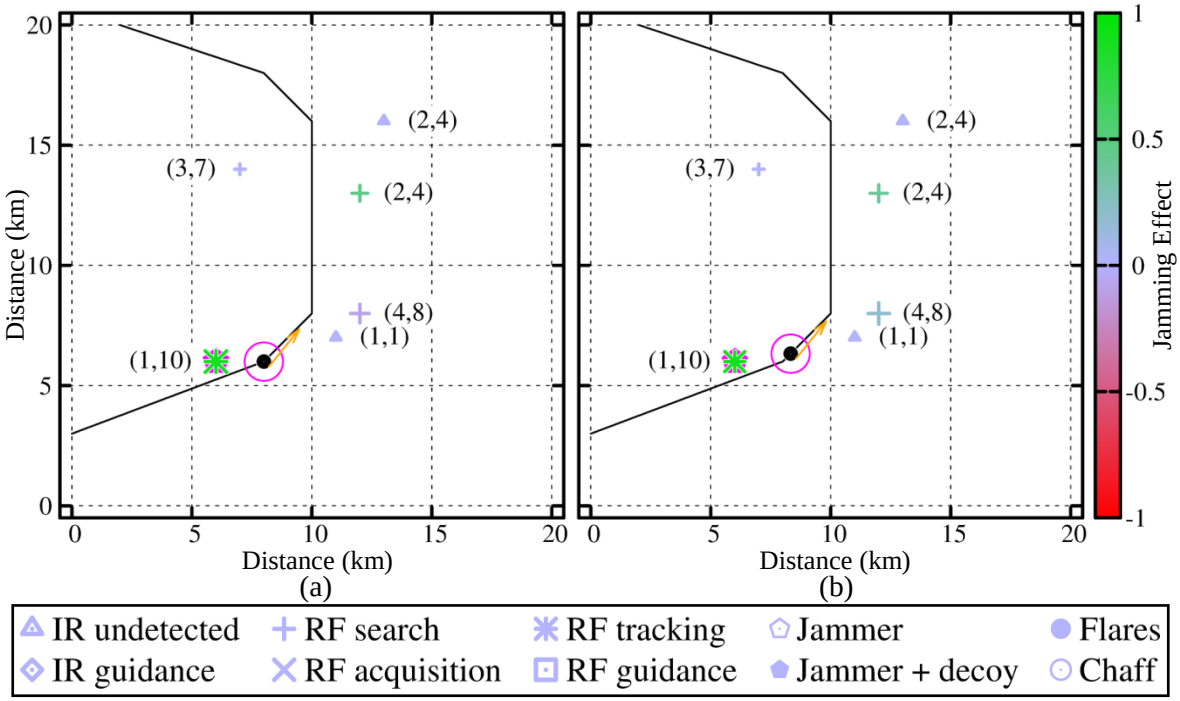



**Figure 4.11.** Scenario images for the time intervals starting 36 (a), and 38 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

In the next countermeasure allocation, the strategy calls for medium and medium-narrow band NJ optimised for threat types 6 and 9 respectively, with both channel antennas set to the same direction. Note that this allocation applies for three time intervals, but only the latter two have been shown in Figure 4.12. This is because the image for the first time interval does not add any more information to what is seen here. Firstly, it is noted that the chosen antenna directions allow for the

jamming of threat IDs 2, 3, and 4, but the platform is only within range of threat ID 3 in the final time interval. This results in the moderate to strong suppression of threat IDs 2 and 4 in the 42, 44 and 46 s time intervals, with ID 3 being suppressed a similar amount in the 46 s time interval. In order to determine the frequency distribution of the effect of the combination of these two techniques, their cross effects are combined in Table 4.11, where the centre frequency of each technique is highlighted. Note that in this table, the frequency bands of the threats have been accounted for where threat IDs 3 and 4 use the same frequency band, and threat ID 7 uses an extra-wide frequency band. The latter of which results in threat ID 3 receiving 0.48 cross effect from the medium-narrow NJ technique. However, since that threat is in the main band of the medium-band technique, there is no change in cross effect for that countermeasure allocation. This means that threat ID 2 receives a combined cross effect of 0.6, ID 3 receives 1.08, and ID 4 receives 1.4. It is also seen in this table that the medium-band technique has a well-selected centre frequency that results in all three threats receiving the maximum possible cross effect, without any energy being wasted. On the other hand, the medium-narrow band technique is optimised for a threat type that is higher than any of the threats being countered, resulting in a lot of signal energy being wasted. However, when the antenna direction is considered, it is seen that threat ID 4 is a worse position, whilst threat ID 2 is in a better position. This balances out their cross effect values, and results in similar jamming effects.

**Table 4.11** Noise jamming cross effect combination.

| **Technique** | **RF Threat type** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **NJ (M)** | 0.12 | 0.24 | 0.60 | 0.60 | 0.60 | 0.60 | 0.60 | 0.60 | 0.24 | 0.12 |
| **NJ (MN)** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.16 | 0.48 | 0.80 | 0.80 | 0.80 |
| **Total** | 0.12 | 0.24 | 0.60 | 0.60 | 0.60 | 0.76 | 1.08 | 1.40 | 1.04 | 0.92 |

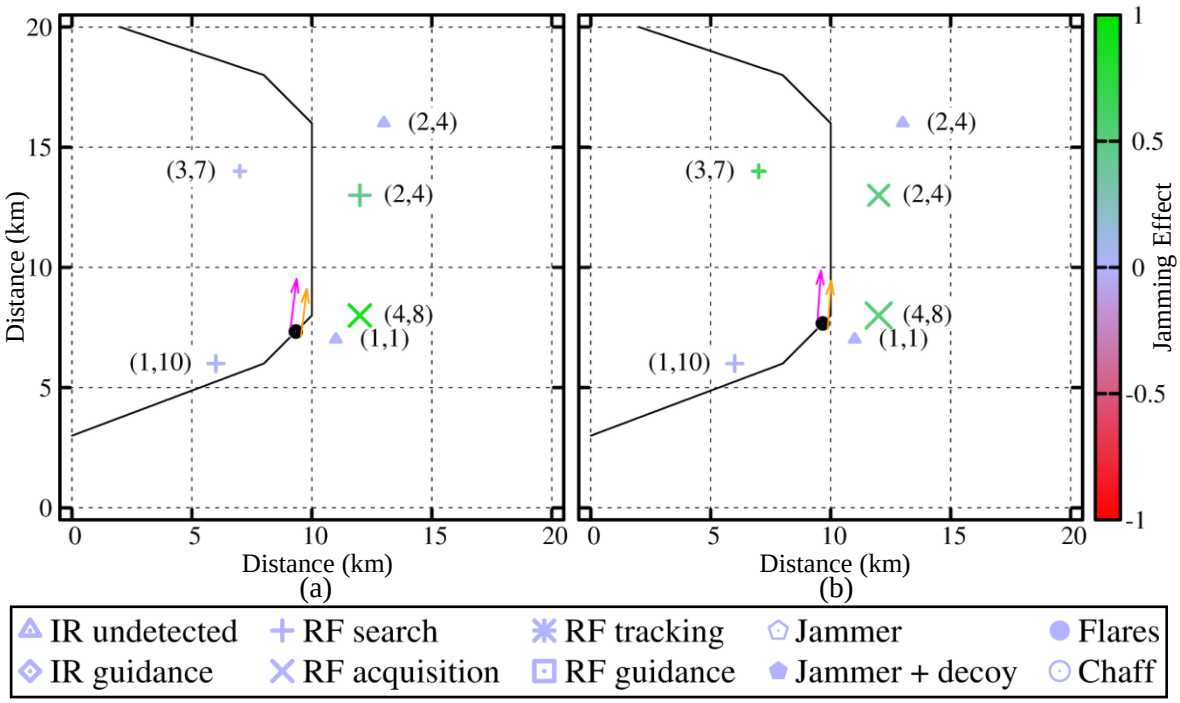



**Figure 4.12.** Scenario images for the time intervals starting 44 (a), and 46 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Next, the 50 and 52 s time intervals are considered in Figure 4.13, where the strategy calls for a combination of narrow-band NJ optimised for threat type 4, and RGPO optimised for threat type 8 directed through the towed decoy. Again, the decoy is necessary to generate enough of a jamming factor to break the tracking lock of threat ID 4 due to the platform jamming it as it passes close by. The associated omnidirectionality in turn slightly illuminates the platform to threat ID 1, and results in moderate to strong illumination to threat ID 3, where this greater effect on ID 3 is due to its wider bandwidth and the fact that it is frequency adjacent to threat ID 4. On the other hand, the NJ is directed solely at suppressing threat ID 2 both through its frequency band allocation, and its associated antenna direction. Lastly, it is also noted that again IR threat ID 1 enters guidance in the 50 s time interval, and the strategy responds with flares in the very next time interval.

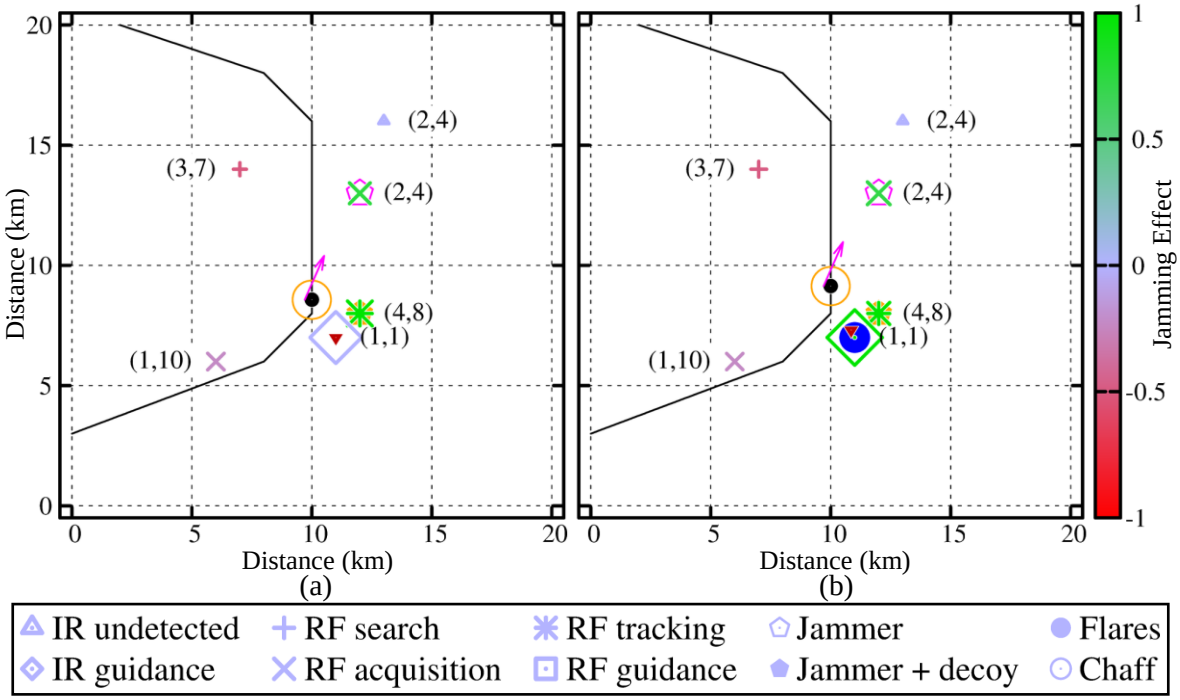



**Figure 4.13.** Scenario images for the time intervals starting 50 (a), and 52 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Figure 4.14 then shows the 56 and 58 s time intervals during which medium-narrow band NJ optimised for threat type 8 has been allocated along with RGPO optimised for threat type 4. Note that the NJ technique continues for a further 4 s, but its effect remains unchanged and as such the associated images have been excluded. In this case, the towed decoy is not actually necessary in order to break the tracking lock of threat ID 2. It has simply been included to increase the jamming effect on this threat. However, it does result in the slight illumination of the platform to threat ID 3 due to its extra-wide bandwidth, but this is countered by the use of NJ. Further, the only alternative would be to use a forward-facing antenna direction which would still illuminate the platform to this threat anyway. It is noted that the NJ centre frequency has been unnecessarily allocated to threat type 8, where threat ID 3 is a type 7, and the only threat of type 8 in the mission is behind the platform, and hence unaffected due to the forward-facing antenna direction. This is as a direct result of the genetic algorithm. The current allocation of medium-narrow band NJ is sufficiently far from threat type 4 in frequency that it does not interfere with the RGPO. However, if the centre frequency of the NJ was changed to threat type 7, then it would begin to interfere with the RGPO.

As such, the NJ would also need to be changed to the narrow-band version of the technique in order to prevent this. However, if only the bandwidth was changed in the original technique, then the cross effect on threat ID 3 would drop from 0.8 to 0.6. Therefore, for improvement to occur, the entire NJ technique of 5 time intervals (including the dead time) must undergo two changes simultaneously, which is clearly unlikely. Although, the current allocation is still favourable, and results in a moderate to strong suppressing effect on threat ID 3 and successfully overcomes the illuminating effect of the RGPO.

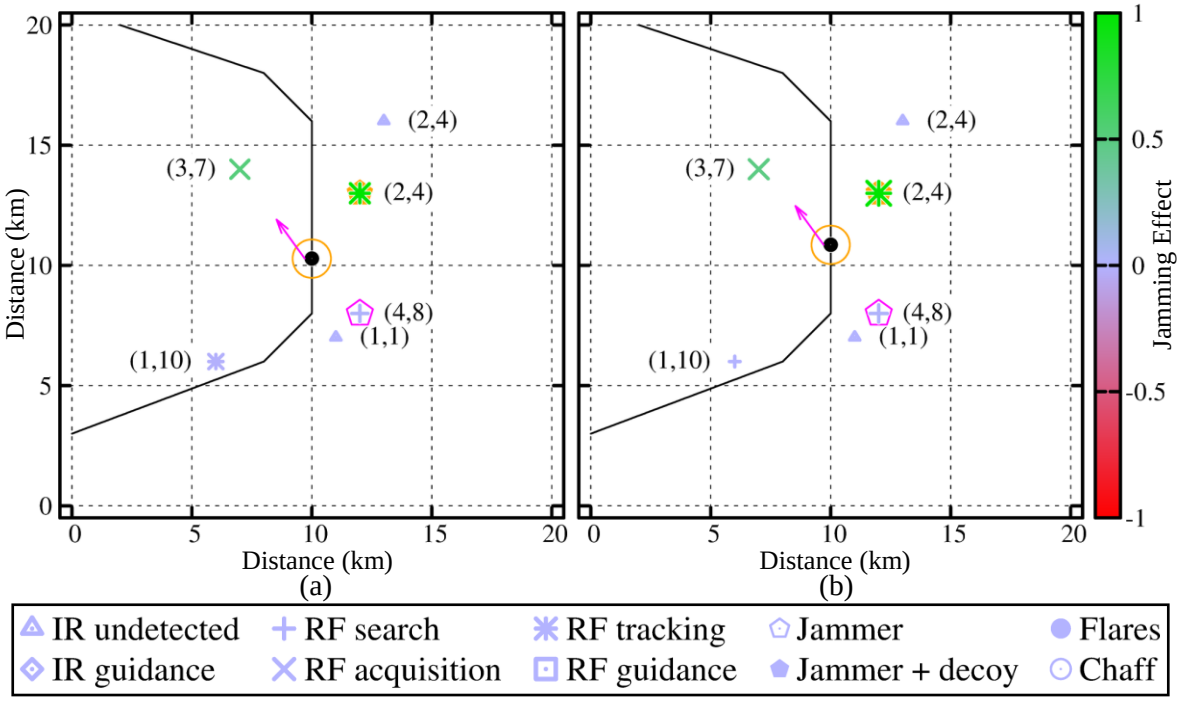



**Figure 4.14.** Scenario images for the time intervals starting 56 (a), and 58 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Next, Figure 4.15 depicts the time intervals starting 80 and 82 s into the mission, where the strategy calls for a combination of VGPO optimised for threat type 7 directed through the towed decoy, and RGPO optimised for threat type 4 in order to break the guidance and tracking locks of threat IDs 3, and 2. As with the previous figure, the towed decoy allocation is not actually necessary in order to break the lock of threat ID 3 in this case. In fact, it has specifically been included in order to maximise the positive interference with the RGPO used against threat ID 2, due to the extra wide band width of threat type 7. Further, it is used to enhance the jamming effect of the VGPO technique due to the large danger presented by threat ID 3 which is in its guidance stage. Lastly,

VGPO in particular has been used due to its enhanced effectiveness against guidance stages. On the other hand, a similar approach has been used for the second jamming channel, where its antenna direction has been chosen to be closer to threat ID 3 rather than being pointed directly at threat ID 2. This allows for an increased positive interference. Finally, it is noted that using the right-hand channel to jam a threat on the left-hand side of the platform is not ideal. However, this is simply a function of the algorithm, where it allocates the most dangerous threat to its appropriate channel, which is threat ID 3 and the left-hand channel in this case.

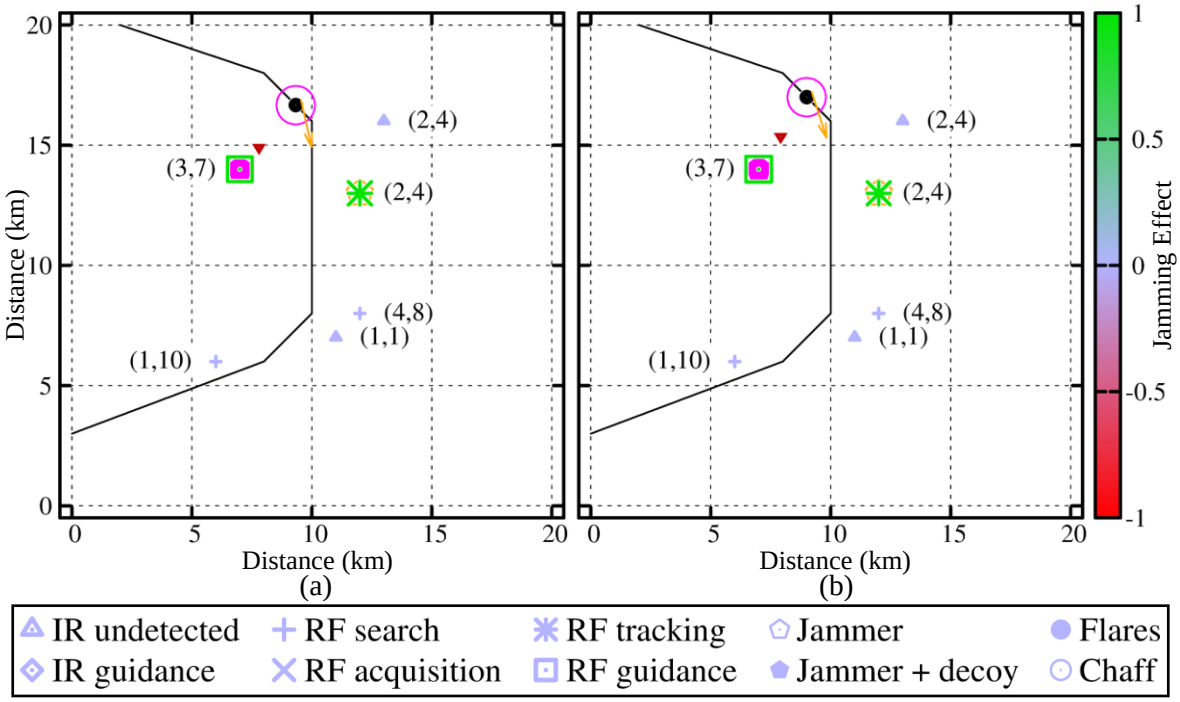



**Figure 4.15.** Scenario images for the time intervals starting 80 (a), and 82 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

Lastly, Figure 4.16 depicts the final countermeasure allocation for the mission in the 92 and 94 s time intervals. Here the strategy calls for a medium-narrow NJ technique optimised for a threat type 5, and an antenna direction of 144° for a total of 12 s, beginning in the 92 s interval. As such, this countermeasure continues for another 4 time intervals after what is depicted here, but their associated images do not convey any new information, and have been excluded as a result. Here it is seen that this technique has been specifically chosen and the antenna direction aimed such that both threat IDs 2 and 3 are suppressed as the platform escapes. In particular, the bandwidth and centre frequency of the NJ have been chosen so as to have a decent effect on both threats, with a

cross effect value of 0.8 for ID 2, and 0.48 for threat ID 3 due to its increased bandwidth. This is seen in Figure 4.16, where there is a moderate to strong suppression of both of these threats, where a stronger suppression occurs for threat ID 2. On the other hand, the platform can simply ignore the other threats as it is out of their range.

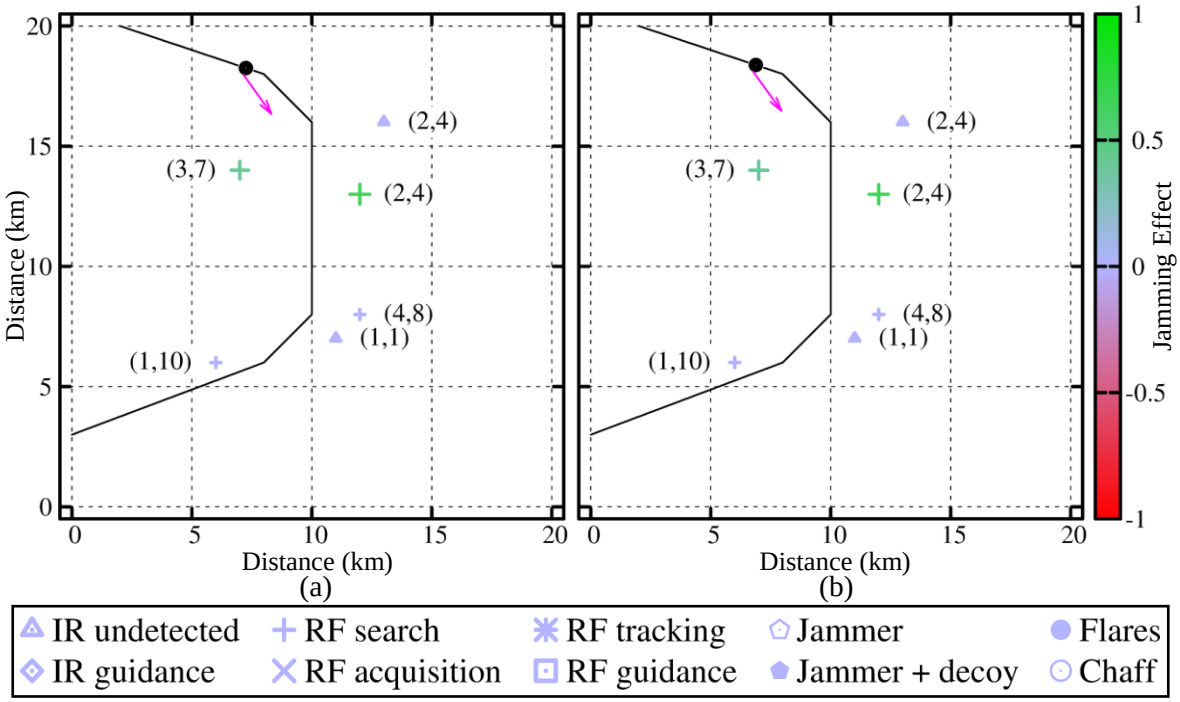



**Figure 4.16.** Scenario images for the time intervals starting 92 (a), and 94 s (b) into the mission, where the axes show the position of the threats in km, and the colour legend shows the jamming effect (E).

## 4.4 OPTIMISATION RESULTS

In this section, the performance of the genetic algorithm itself will be analysed. This consists of a comparison the performance of the three different stop criteria, and separately the performance of the three Pareto solutions. Thereafter, the characteristics of the population over the course of optimisation procedure will be analysed, before finally the effects of the individual specialised operators are investigated.

### 4.4.1 Stop criterion comparison

In this section, the performances of the different stop criteria are compared using the full scenario described above. Due to the fact that the genetic algorithm is based on stochastic processes, the

performances of the stop criteria need to be compared over a number of runs of the algorithm. Therefore, the performance statistics of 30 runs of each individual stop criterion are presented in Table 4.12. As with the previous section, this stochastic nature means that the algorithm is not guaranteed to generate a feasible solution to the problem in every run. As such, the number of times the algorithm failed to generate a successful solution appears in the right-most column, and a separate set of statistics for only the successful solutions has been included to counteract the disproportional effect of the unsuccessful ones. Importantly, this allows for a more direct comparison between the criteria because they all use the same maximum-iteration limit of 150, and as such any differences in the number of unsuccessful solutions would purely be as a result of chance.

**Table 4.12** Stop criterion performance statistics.

| Stop Criterion | Statistic | Total Fitness | Total Iterations | Run Time (s) | No. Failed |
|---|---|---|---|---|---|
| **Maximum Iterations** | **Average** | 0.0976 | 150.00 | 793.82 | 5 |
| | **Median** | 0.0670 | 150.00 | 794.71 | |
| | **Minimum** | 0.0652 | 150.00 | 771.12 | |
| | **Maximum** | 0.2563 | 150.00 | 821.32 | |
| | **Range** | 0.1911 | 0.00 | 50.20 | |
| | **Successful Only:** | | | | |
| | **Average** | 0.0671 | 150.00 | 794.02 | |
| | **Maximum** | 0.0695 | 150.00 | 821.32 | |
| | **Range** | 0.0043 | 0.00 | 50.20 | |
| | | | | | |
| **Safe Passage** | **Average** | 0.1081 | 26.17 | 159.73 | 4 |
| | **Median** | 0.0868 | 6.00 | 56.26 | |
| | **Minimum** | 0.0771 | 2.00 | 35.83 | |
| | **Maximum** | 0.2562 | 150.00 | 802.86 | |
| | **Range** | 0.1791 | 148.00 | 767.03 | |
| | **Successful Only:** | | | | |
| | **Average** | 0.0861 | 7.12 | 63.12 | |
| | **Maximum** | 0.0997 | 45.00 | 260.59 | |
| | **Range** | 0.0226 | 43.00 | 224.76 | |
| | | | | | |
| **Minimal Improvement** | **Average** | 0.0867 | 74.50 | 419.33 | 3 |
| | **Median** | 0.0682 | 63.00 | 359.07 | |
| | **Minimum** | 0.0664 | 47.00 | 273.47 | |
| | **Maximum** | 0.2566 | 150.00 | 821.07 | |
| | **Range** | 0.1902 | 103.00 | 547.60 | |
| | **Successful Only:** | | | | |
| | **Average** | 0.0682 | 66.11 | 376.40 | |
| | **Maximum** | 0.0726 | 140.00 | 752.30 | |
| | **Range** | 0.0062 | 93.00 | 478.83 | |
| | | | | | |

Note that for this section the standard set of fitness function weights has been used. Further, for the minimum improvement criterion, the required minimum improvement in fitness value of the population leader is set to 0.0001, and the number of generations before doubled mutation is set to 5. Thereafter, a further 5 generations are allowed for the minimum improvement to be attained.

Overall, it can be seen that the intended performance has been achieved, where the safe-passage stop criterion results in the shortest median run time of just 56.26 s, and the maximum-iterations criterion results in the longest median run time of 794.71 s. It can further be seen that, as expected, the fitness of a stop criterion's solutions is proportional to the run time it allows, where the rate of fitness improvement tapers off over time. As such, the fittest solutions are generated using the maximum-iterations stop criterion resulting in a median fitness of 0.0670, whilst the weakest are generated by the safe-passage stop criterion with a median fitness of 0.0868. Importantly, the minimal improvement criterion falls in between the other two in terms of both fitness and run time. Specifically, this technique achieves a median fitness value of 0.0682 which is quite close to that of the maximum-iterations technique, whilst attaining a median run time of just 359.07 s, which is under half of that technique. This means that the minimal-improvement approach achieves its intended goal of finding a good compromise between fitness and run time by avoiding the period of diminishing returns in later iterations. The performance of each individual stop criterion is analysed more in-depth below.

*1) Maximum iterations*

As discussed previously, this stop criterion runs the algorithm for a relatively fixed time period by running it for a fixed number of iterations or generations of the genetic algorithm. It is suited to applications where a known amount of time is available for strategy optimisation, especially cases where a relatively long period of time is available and a strong solution is desired. This can be seen in the results, where the performance of the solutions is both consistent and strong, achieving the overall fittest solution (0.0652), the best median fitness (0.0670), and the smallest successful-only fitness range (0.0043) of all the stop criteria. However, this approach does require a long, but relatively consistent, median run time of 794.71 s (13.25 minutes) with a small range of just 50.2 s.

*2) Safe passage*

Running the developed TECA program using a safe-passage stop criterion runs the algorithm until the first successful solution is found that results in the platform leaving the mission unscathed. It is

best suited to applications where speed is favoured over strategy performance. This can be seen in the results that this approach is indeed the most rapid with a solution generated in a median of just 6 iterations and 56.26 s, where the shortest run time of 35.83 s was achieved in only 2 iterations. However, it also resulted in the poorest median fitness of 0.0868, which is 29.55% worse than that of the maximum-iterations approach, and the overall weakest successful solution with a fitness value of 0.0997. This is 37.33% worse than the weakest successful design of the other stop criteria. Further, it is also noted that in comparison, the slowest successful solution in the data set required 45 iterations and 260.59 s to generate. As such, it is important to note that both the optimisation time required for this technique and the fitness of the developed strategies vary widely, resulting in successful-only ranges of 224.76 s and 0.0226, respectively. This run-time range may be less than that of the minimal-improvement stop criterion in magnitude, but relative to their respective medians, this run time range is substantially larger at 4 times the median, versus only 1.33 times. Further, the successful-only fitness range is also more than 3.5 times that of the next most variable approach.

*3) Minimal improvement*

This approach stops the optimisation process when the fitness of the population leader is no longer improving sufficiently, so as to avoid the diminishing returns of optimising beyond a certain point. It results in the most efficient compromise between the two previous stop criteria, achieving a median fitness of 0.0682 in a median 63 iterations and resultant median time of just 359.07 s (5.98 minutes). This means that this criterion achieves a median fitness that is only 1.79% worse than that of the maximum-iterations criterion in just 45.18% of the time. However, due to the stochastic nature of the algorithm, the point of termination does vary quite widely from 47 to 140 iterations, which in turn results in a run time that varies from 273.47 to 752.30 s (4.56 to 12.54 minutes). This is clearly more than that of the maximum-iterations criterion, but as stated previously, when compared to their respective median run times, this run time range is less than that of the safe-passage approach. Lastly, it is noted that the fitness of the generated solutions doesn't vary as widely with a success-only range of just 0.0062, which is slightly worse than that of the maximum-iteration technique's 0.0043, but still better than 0.0226 of the safe-passage approach. Therefore, this criterion is indeed suited to applications where both time and strategy fitness are of importance, where it finds the best compromise between the two.

### 4.4.2 Pareto optimisation

The performance of the three different Pareto designs is compared using the half scenario. This is the exact same scenario as described previously, except that it contains only half of the threats: RF threat IDs 2, 4, 5, and 9, along with IR IDs 1 and 3. Specifically, the half scenario was used due to the fact that the less saturated threat environment allows for more variety in countermeasure strategies, thus resulting in larger differences between the Pareto solutions.

*1) Statistical performance analysis*

A summary of the overall performance appears in Table 4.13, whilst the individual Pareto performance appears in Table 4.14 for 30 runs of the algorithm. Note that all three Pareto designs are generated in a single run of the algorithm, where the conservative design is found first, followed by the cost-focused design, then finally the EMCON-focused design is found, where the weights used for each are detailed in Table 3.13 in the previous section. Further, it is noted that the minimum-improvement stop criterion is used for this section, along with a maximum-iteration cap of 150. Specifically, this stop criterion is used in order to take advantage of the reuse of the population leader in the generation of subsequent Pareto designs, resulting in the generation of a number of useful alternative options for the user in a reasonable amount of time. Lastly, note that the second seed method results in a successful solution to this half scenario, which in turn means that no runs of the algorithm failed to generate a successful strategy. This in turn means that the representative probability of engagement objective function score has been excluded from Table 4.14 as it is simply zero for all criteria.

**Table 4.13** Overall Pareto performance statistics.

| Statistic | Total Fitness | Total Iterations | Time |
|---|---|---|---|
| **Average** | 0.1557 | 157.40 | 821.02 |
| **Median** | 0.1560 | 160.50 | 831.51 |
| **Minimum** | 0.1493 | 119.00 | 622.65 |
| **Maximum** | 0.1631 | 203.00 | 1024.21 |
| **Range** | 0.0138 | 84.00 | 401.56 |

It is seen in these results that the 3 Pareto solutions are generated in a median total of 160.5 iterations and a resultant median time of just 831.51 s (13.86 minutes), which is only 68.79 s slower than the 762.72 s (12.71 minutes) required to generate a single 150 maximum-iteration stop criterion solution for the same half scenario (see Section 4.3.2). This can be attributed to the reuse

of the most-recently-generated Pareto design as a seed for the next, because this process produces a fitter seed, which in turn results in a shorter optimisation time before the minimum improvement criterion is met. This can be seen by looking at the median iterations required to generate each Pareto design, which decreases from 66 for the conservative design that does not benefit from reuse, down to 35 and 56.50 for the cost-focused and EMCON-focused designs that do. Importantly, this means that this approach can be used to generate a number of different approaches to a scenario in a reasonable amount of time. This is vital in decision-support applications, where such a capability would instil greater confidence in a human operator as they can combine their experience with the algorithm to select the most feasible solution.

**Table 4.14** Pareto solution performance statistics.

| Pareto Design | Statistic | Danger Score | Cost Score | EMCON Score | Total Fitness | Total Iterations |
|---|---|---|---|---|---|---|
| **Conservative** | **Average** | 0.0473 | 0.0930 | 0.3681 | 0.0484 | 64.03 |
| | **Median** | 0.0467 | 0.0909 | 0.3689 | 0.0484 | 66.00 |
| | **Minimum** | 0.0436 | 0.0909 | 0.2623 | 0.0476 | 35.00 |
| | **Maximum** | 0.0537 | 0.1000 | 0.4317 | 0.0494 | 89.00 |
| | **Range** | 0.0101 | 0.0091 | 0.1694 | 0.0018 | 54.00 |
| **Cost-Focused** | **Average** | 0.0556 | 0.0842 | 0.2583 | 0.0498 | 38.93 |
| | **Median** | 0.0534 | 0.0909 | 0.2569 | 0.0503 | 35.00 |
| | **Minimum** | 0.0498 | 0.0545 | 0.1803 | 0.0458 | 13.00 |
| | **Maximum** | 0.0689 | 0.1000 | 0.3224 | 0.0524 | 82.00 |
| | **Range** | 0.0191 | 0.0455 | 0.1421 | 0.0066 | 69.00 |
| **EMCON-Focused** | **Average** | 0.0630 | 0.0833 | 0.1537 | 0.0575 | 54.43 |
| | **Median** | 0.0618 | 0.0909 | 0.1503 | 0.0571 | 56.50 |
| | **Minimum** | 0.0555 | 0.0545 | 0.1148 | 0.0534 | 16.00 |
| | **Maximum** | 0.0755 | 0.1000 | 0.2131 | 0.0656 | 82.00 |
| | **Range** | 0.0200 | 0.0455 | 0.0983 | 0.0122 | 66.00 |

Further, it is seen that the characteristics of the Pareto designs follow their weights, hence allowing them to fulfil their intended purposes. The conservative Pareto design results in the best median danger-value score of 0.0467. Further, it also results in the worst average cost and median EMCON scores of 0.0930 and 0.3689. This means that this approach tends to rather use more countermeasures in order to keep the platform safer, resulting in the lowest risk to the platform. On the other hand, the EMCON-focused designs have the worst median danger score of 0.0618, but the best median EMCON score of 0.1503 along with the best average cost score of 0.0833. In fact, the EMCON score is 59.26% less than that of the conservative approach, whilst only increasing the danger score by 32.33%. This means that this approach drastically reduces the amount of

countermeasures used over the course of the mission, and hence the levels of EM emissions. Essentially this requires the platform to take a greater risk with the known threats in this scenario, but in exchange reduces the risk presented by any unknown threats in the mission.

On the other hand, the reason for this EMCON-focused design having the lowest average cost score can be attributed to the fact that the conservative approach uses a minimal number of passive countermeasures to begin with, thus giving the cost-focused design little room to improve. Thereafter, this cost-focused design gets passed onto the EMCON-focused design to be improved upon further where the EMCON objective function indirectly, but roughly, takes the number of cartridges used into account by counting the number of passive countermeasures used. However, for the most part, this means that the EMCON-focused design tends to use the same passive countermeasure allocations as the cost-focused design. As a further result of these factors, the cost-focused design tends to fall in between the two extremes of the conservative and the EMCON-focused design in terms of the trade-off between danger to the platform and EMCON levels. It achieves a 30.36% improvement in EMCON score in exchange for an increase in danger score of 14.35% in comparison to the conservative approach.

*2) Singular Pareto set analysis*

For the direct comparison and analysis of the generated Pareto designs, the performance statistics of a single run of the algorithm are presented in Table 4.15. Further, the countermeasure strategies developed during this run are presented in Tables 4.16 – 4.18, for the conservative, cost-focused, and EMCON-focused designs respectively. Lastly, the resultant threat stages are presented in Table 4.19, where the stages of each threat are directly compared between the different Pareto designs. Note that D indicates the danger-focused conservative design, whilst the C indicates the cost-focused design, and E indicates the EMCON-focused design.

**Table 4.15** Performance statistics of the example Pareto set.

| Statistic | Danger Score | Cost Score | EMCON Score | Fitness | Iterations | Total Iterations | Total Run Time (s) |
|---|---|---|---|---|---|---|---|
| **Conservative** | 0.0463 | 0.0909 | 0.3661 | 0.0476 | 81 | 197 | 1014.93 |
| **Cost** | 0.0645 | 0.0545 | 0.2240 | 0.0465 | 67 | | |
| **EMCON** | 0.0755 | 0.0545 | 0.1202 | 0.0552 | 49 | | |

**Table 4.16** Conservative countermeasure strategy.

| Time | Active Channel 1 | | | Active Channel 2 | | | Passive Channel | | Decoy |
|---|---|---|---|---|---|---|---|---|---|
| | Tech | Threat | Antenna(°) | Tech | Threat | Antenna(°) | Tech | Threat | |
| 0 | NJ (N) | 1 | 0 | NJ (W) | 10 | 0 | | | |
| 2 | | | | | | | | | |
| 4 | | | | None | | | | | |
| 6 | None | | | | | | | | |
| 8 | | | | | | | | | |
| 10 | | | | Change | | | | | |
| 12 | | | | CP | 10 | 0 | None | | |
| 14 | Change | | | | | | | | |
| 16 | CP | 10 | 0 | | | | | | None |
| 18 | | | | None | | | | | |
| 20 | | | | | | | | | |
| 22 | | | | | | | | | |
| 24 | | | | Change | | | | | |
| 26 | | | | CP | 8 | 0 | Change | | |
| 28 | | | | | | | Flare | 1 | |
| 30 | None | | | None | | | | | |
| 32 | | | | | | | | | |
| 34 | Change | | | Change | | | | | |
| 36 | VGPO | 10 | D | NJ (M) | 6 | 6 | | | 1 |
| 38 | | | | | | | | | |
| 40 | Change | | | Change | | | None | | |
| 42 | NJ (M) | 6 | 42 | NJ (MN) | 9 | 42 | | | |
| 44 | | | | | | | | | None |
| 46 | | | | | | | | | |
| 48 | Change | | | Change | | | | | |
| 50 | NJ (N) | 4 | 336 | RGPO | 8 | D | Change | | 2 |
| 52 | | | | | | | Flare | 1 | |
| 54 | | | | | | | | | |
| 56 | | | | None | | | | | |
| 58 | | | | | | | None | | None |
| 60 | None | | | | | | | | |
| 62 | | | | Change | | | Change | | |
| 64 | | | | VGPO | 4 | D | Flare | 4 | 2 |
| 66 | | | | | | | | | |
| 68 | Change | | | None | | | | | None |
| 70 | RGPO | 7 | D | | | | | | 1 |
| 72 | | | | Change | | | | | |
| 74 | None | | | CP | 4 | 228 | | | |
| 76 | Change | | | | | | | | |
| 78 | CP | 4 | 168 | | | | | | |
| 80 | | | | | | | | | None |
| 82 | None | | | | | | | | |
| 84 | Change | | | None | | | | | |
| 86 | CP | 4 | 138 | | | | | | |
| 88 | Change | | | | | | | | |
| 90 | RGPO | 4 | D | | | | | | 1 |
| 92 | | | | | | | | | |
| 94 | Change | | | Change | | | None | | None |
| 96 | RGPO | 7 | D | CP | 4 | 162 | | | 1 |
| 98 | | | | | | | | | |
| 100 | | | | | | | | | |
| 102 | | | | | | | | | |
| 104 | | | | | | | | | |
| 106 | | | | | | | | | |
| 108 | | | | | | | | | |
| 110 | None | | | None | | | | | None |
| 112 | | | | | | | | | |
| 114 | | | | | | | | | |
| 116 | | | | | | | | | |
| 118 | | | | | | | | | |
| 120 | | | | | | | | | |

**Table 4.17** Cost-focused countermeasure strategy.

| Time | Active Channel 1 | | | Active Channel 2 | | | Passive Channel | | Decoy |
|---|---|---|---|---|---|---|---|---|---|
| | Tech | Threat | Antenna(°) | Tech | Threat | Antenna(°) | Tech | Threat | |
| 0 | CP | 2 | D | | | | | | 1 |
| 2 | | | | | | | | | |
| 4 | | | | | | | | | |
| 6 | None | | | | | | | | |
| 8 | | | | | | | | | |
| 10 | | | | None | | | | | |
| 12 | | | | | | | None | | |
| 14 | Change | | | | | | | | |
| 16 | CP | 10 | 0 | | | | | | |
| 18 | | | | | | | | | None |
| 20 | | | | | | | | | |
| 22 | | | | | | | | | |
| 24 | | | | Change | | | | | |
| 26 | | | | CP | 8 | 0 | Change | | |
| 28 | | | | | | | Flare | 1 | |
| 30 | None | | | None | | | | | |
| 32 | | | | | | | | | |
| 34 | Change | | | Change | | | | | |
| 36 | VGPO | 10 | D | NJ (M) | 6 | 6 | | | 1 |
| 38 | | | | | | | | | |
| 40 | Change | | | | | | | | |
| 42 | NJ (M) | 6 | 42 | None | | | | | |
| 44 | | | | | | | | | None |
| 46 | | | | | | | None | | |
| 48 | Change | | | Change | | | | | |
| 50 | NJ (N) | 4 | 336 | RGPO | 8 | D | | | 2 |
| 52 | | | | | | | | | |
| 54 | | | | | | | | | |
| 56 | | | | None | | | | | |
| 58 | | | | | | | | | None |
| 60 | None | | | | | | | | |
| 62 | | | | Change | | | Change | | |
| 64 | | | | VGPO | 4 | D | Flare | 4 | 2 |
| 66 | | | | | | | | | |
| 68 | Change | | | | | | | | None |
| 70 | RGPO | 7 | D | | | | | | 1 |
| 72 | | | | | | | | | |
| 74 | | | | | | | | | |
| 76 | | | | | | | | | |
| 78 | | | | | | | | | |
| 80 | None | | | | | | | | None |
| 82 | | | | | | | | | |
| 84 | | | | | | | | | |
| 86 | | | | | | | | | |
| 88 | Change | | | | | | | | |
| 90 | RGPO | 4 | D | | | | | | 1 |
| 92 | | | | | | | | | |
| 94 | | | | None | | | None | | |
| 96 | | | | | | | | | |
| 98 | | | | | | | | | |
| 100 | | | | | | | | | |
| 102 | | | | | | | | | |
| 104 | | | | | | | | | |
| 106 | None | | | | | | | | None |
| 108 | | | | | | | | | |
| 110 | | | | | | | | | |
| 112 | | | | | | | | | |
| 114 | | | | | | | | | |
| 116 | | | | | | | | | |
| 118 | | | | | | | | | |
| 120 | | | | | | | | | |

**Table 4.18** EMCON-focused countermeasure strategy.

| Time | Active Channel 1 | | | Active Channel 2 | | | Passive Channel | | Decoy |
|---|---|---|---|---|---|---|---|---|---|
| | Tech | Threat | Antenna(°) | Tech | Threat | Antenna(°) | Tech | Threat | |
| 0 | CP | 2 | D | | | | | | 1 |
| 2 | | | | | | | | | |
| 4 | | | | | | | | | |
| 6 | | | | | | | | | |
| 8 | None | | | | | | | | |
| 10 | | | | None | | | | | |
| 12 | | | | | | | None | | |
| 14 | | | | | | | | | |
| 16 | Change | | | | | | | | |
| 18 | CP | 10 | 0 | | | | | | |
| 20 | | | | | | | | | |
| 22 | | | | | | | | | |
| 24 | | | | Change | | | | | None |
| 26 | | | | CP | 8 | 0 | Change | | |
| 28 | | | | | | | Flare | 1 | |
| 30 | | | | | | | | | |
| 32 | | | | | | | | | |
| 34 | | | | | | | | | |
| 36 | | | | None | | | | | |
| 38 | | | | | | | | | |
| 40 | | | | | | | | | |
| 42 | | | | | | | | | |
| 44 | None | | | | | | | | |
| 46 | | | | Change | | | None | | |
| 48 | | | | RGPO | 8 | D | | | 2 |
| 50 | | | | | | | | | |
| 52 | | | | | | | | | |
| 54 | | | | | | | | | |
| 56 | | | | None | | | | | None |
| 58 | | | | | | | | | |
| 60 | | | | | | | | | |
| 62 | | | | Change | | | Change | | |
| 64 | | | | VGPO | 4 | D | Flare | 4 | 2 |
| 66 | | | | | | | | | |
| 68 | Change | | | | | | | | None |
| 70 | RGPO | 7 | D | | | | | | 1 |
| 72 | | | | | | | | | |
| 74 | | | | | | | | | |
| 76 | | | | | | | | | |
| 78 | | | | | | | | | |
| 80 | | | | | | | | | |
| 82 | | | | | | | | | |
| 84 | | | | | | | | | |
| 86 | | | | | | | | | |
| 88 | | | | | | | | | |
| 90 | | | | | | | | | |
| 92 | | | | | | | | | |
| 94 | | | | None | | | None | | |
| 96 | None | | | | | | | | None |
| 98 | | | | | | | | | |
| 100 | | | | | | | | | |
| 102 | | | | | | | | | |
| 104 | | | | | | | | | |
| 106 | | | | | | | | | |
| 108 | | | | | | | | | |
| 110 | | | | | | | | | |
| 112 | | | | | | | | | |
| 114 | | | | | | | | | |
| 116 | | | | | | | | | |
| 118 | | | | | | | | | |
| 120 | | | | | | | | | |

**Table 4.19** Pareto threat stages.

| Time (s) | RF Threat ID | | | | | | | | | | | | IR Threat ID | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | 2 | | | 3 | | | 4 | | | 1 | | | 2 | | |
| | D | C | E | D | C | E | D | C | E | D | C | E | D | C | E | D | C | E |
| 0 | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | |
| 6 | | S | S | | | | | | | | | | | | | | | |
| 8 | S | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | U | U | U | | | |
| 14 | | A | A | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | |
| 18 | | | A | | | | | | | S | S | S | | | | | | |
| 20 | A | A | | | S | S | | | | | | | | | | | | |
| 22 | | | | S | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | |
| 26 | A | | T | | | | S | S | S | | | | G | G | G | | | |
| 28 | | T | | | | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | | | | U | U | U |
| 32 | | | G | | | | | | | | | | | | | | | |
| 34 | T | | | | | | | | | | | | | | | | | |
| 36 | | G | | | | | | | | | | | | | | | | |
| 38 | | | | | | | | | | | | A | | | | | | |
| 40 | | | G | | | | | | | A | A | | U | U | U | | | |
| 42 | | | | | | A | | | | | | T | | | | | | |
| 44 | S | S | | | A | | | | | A | | | | | | | | |
| 46 | | | | A | | | | | | | T | | | | | | | |
| 48 | | | | | A | T | | | | | | | | | | | | |
| 50 | A | A | | | | | | | | T | T | | G | G | G | | | |
| 52 | | | T | A | | | | | | | | | | | | | | |
| 54 | | | | | | | | | | | | | | | | | | |
| 56 | T | T | | T | T | G | A | A | A | | | S | | | | | | |
| 58 | | | | | | | | | | S | S | | | G | G | | | |
| 60 | | | | T | T | | | | | | | | | | | | | |
| 62 | | | | | | G | | | | | | A | | | | G | G | G |
| 64 | | | | G | G | | T | T | T | A | A | | | | | | | |
| 66 | | | | | | | | | | | | T | | | | | | |
| 68 | | | | | | | | | | T | T | | | | | | | |
| 70 | | | | | | | | | | | | | | | | | | |
| 72 | | | | S | S | S | | | | | | | | | | | | |
| 74 | | | | | | | | | | | | | | | | | | |
| 76 | | | | | | | | | | | | | | | | U | U | U |
| 78 | | | | | A | A | S | S | S | | | | | | | | | |
| 80 | | | | A | | | | | | | | | | | | | | |
| 82 | | | | | | | | | | | | | | | | | | |
| 84 | | | | | T | | A | A | A | | | | | | | | | |
| 86 | | | | | | | | | | | | | | | | G | G | G |
| 88 | S | S | S | T | | | | | | | | | U | | | | | |
| 90 | | | | | | T | | | | | | | | | | | | |
| 92 | | | | | | | T | T | T | | | | | U | U | | | |
| 94 | | | | | | | | | | | | | | | | | | |
| 96 | | | | | | | | | | S | S | S | | | | | | |
| 98 | | | | | | | | | | | | | | | | | | |
| 100 | | | | | | | | | | | | | | | | | | |
| 102 | | | | | | | | | | | | | | | | | | |
| 104 | | | | | | | S | G | G | | | | | | | U | U | U |
| 106 | | | | S | S | | | | | | | | | | | | | |
| 108 | | | | | | | | | | | | | | | | | | |
| 110 | | | | | | | | | | | | | | | | | | |
| 112 | | | | | | S | A | T | T | | | | | | | | | |
| 114 | | | | | | | | | | | | | | | | | | |
| 116 | | | | | | | | | | | | | | | | | | |
| 118 | | | | | | | S | S | S | | | | | | | | | |
| 120 | | | | | | | | | | | | | | | | | | |

Analysis of the overall performance statistics of the different Pareto designs indicates that they again meet expectations in that the conservative design has the best danger score, the EMCON-focused design has the best EMCON score, and the cost-focused design falls between the two with the tied best cost score. Further, the conservative design uses the most cartridges, resulting in a higher cost score. This can be seen in the countermeasure strategies themselves. Overall, the strategies follow the same general approach to the mission, where the different designs successively use fewer and fewer countermeasures. This is further reinforced by the EMCON scores of the different designs which indicate that the conservative, cost-focused and EMCON focused designs have 36.61%, 22.40%, and 12.02% of possible countermeasures allocated. That makes sense given the approach used whereby the population leader is passed on between the different designs. Importantly, this not only allows for the faster generation of solutions, but also for the more direct comparison of the similar strategies, which is favourable in the context of decision-support systems.

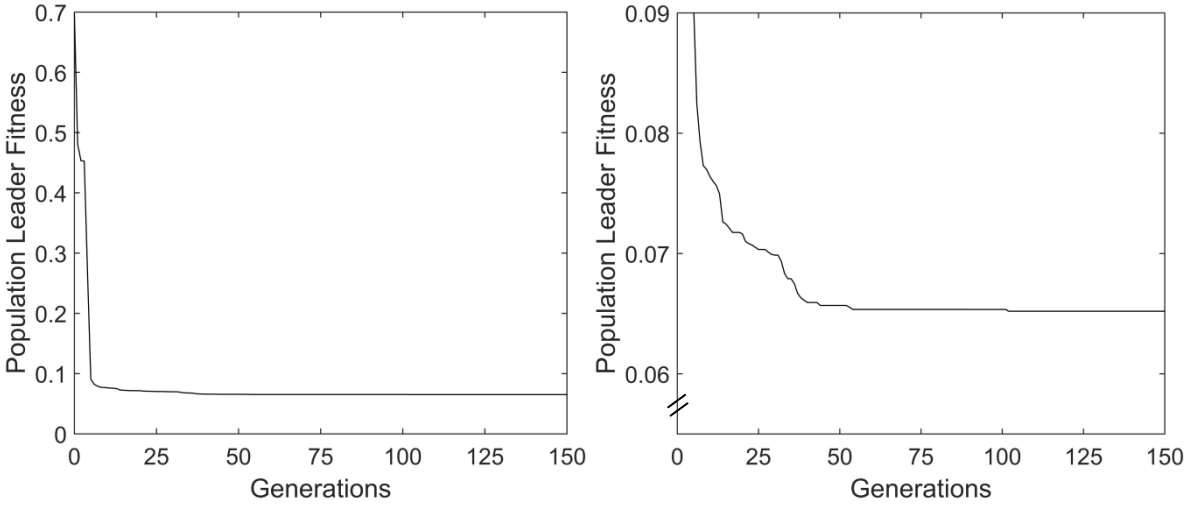
In this example the conservative approach uses a lot of countermeasure techniques directed toward search-type stages: specifically NJ and CP. Radar lock is broken six times by RGPO and VGPO starting in the 36, 50, 64, 70, 90, and 96 s time intervals. IR threats are also countered a total of three times using flares starting in the 28, 52, and 64 s time intervals. Specifically, it is seen that this strategy essentially begins with CP in order delay detection of the platform. Thereafter, the radar lock of threat type 10 (ID 2) is broken using VGPO in the time interval beginning 38 s into the mission. Around this time, the platform also switches to NJ in order to delay its detection by the remaining threats. However, it must then break the lock of threat ID 9 (type 8) shortly thereafter using RGPO, before then breaking the lock of threat ID 4 (type 4) using VGPO and subsequently threat ID 5 (type 7) using RGPO. The strategy then calls for switching back to CP in order to prevent redetection by threat ID 4 (type 4). However, it must once more break the lock of threat IDs 4 and 5 (types 4 and 7) in order to prevent them from firing at the platform. Interestingly, the strategy also calls for a simultaneous CP technique directed at threat ID 4 (type 4) while jamming threat ID 5 (type 7) in order to counteract its illuminating effect caused by threat type 7’s increased bandwidth. On the other hand, when the strategy handles IR threats, it simply breaks their lock as soon as possible using flares optimised for the relevant threat type. Importantly, this strategy results in only one threat firing upon the platform and entering guidance.

In comparison, the cost-focused design maintains a similar strategy overall, but removes two lock-breaking techniques, and also reduces the use of CP and NJ in general. The first lock-breaking technique that is removed is the RGPO right at the end of the mission that begins at 96 s. The direct effect of this is to allow threat ID 5 (type 7) to fire upon the platform and enter the guidance stage, where the strategy relies on the fact that the platform is fast enough to exit the range of that threat before the missile can hit. The second removed lock-breaking technique is the middle flare technique directed at IR threat ID 1 (type 1) that begins 52 s into the mission. This too has the same effect, where the strategy relies on the platform being able to escape the range of the threat before it gets hit. On the other hand, the reduction of NJ and CP occurs throughout the mission, with a little being removed in the beginning and middle sections, and all being removed from the latter part of the mission. The general effect of this is that it allows the threats to progress further through their radar stages, hence placing the platform in greater danger. This can clearly be seen by looking at the threat stages table, where all the threats detect the platform sooner, progress sooner, and progress further along their engagement procedure. In particular the reduction in the use of CP in the beginning of the mission allows threat ID 2 (type 10) to fire upon the platform and enter the guidance stage before its lock is broken by the VGPO technique beginning 36 s into the mission.

Finally, it is seen that the EMCON-focused design also retains the same basic approach to the mission, except that it further reduces the number of NJ and CP techniques to almost nothing. More importantly, it also removes two further RGPO and VGPO techniques from the strategy. The first is the VGPO directed toward threat ID 2 (type 10) 36 s into the mission, which then requires the platform to outrun a further missile. The second is the RGPO directed towards threat ID 4 (type 4) that starts 90 s into the mission, which means that this strategy stops all countermeasures 74 s into the 120 s mission. This too leaves the platform at great risk, relying on the platform to outrun all threats at the end of the mission, adding threat ID 4 to the previously-discussed danger presented by threat ID 5. However, it is clear that that neither of these techniques are strictly necessary for the survival of the platform, and as such it does make sense to exclude them when minimising EM emissions. Further, this approach also has the clear advantage of not illuminating the platform to any unknown threats, especially ones towards the end of the mission, or even outside the mission area. Lastly, the effect of reducing the CP and NJ techniques even further is seen in the threat stages table, where it is seen that the threats detect the platform even sooner than before, before continuing to progress to later stages sooner.

### 4.4.3 Genetic algorithm analysis

Analysis of the genetic algorithm itself can be performed by examining the fitness of the population leader on a generation-by-generation basis over the course of the mission. This is shown in Figure 4.17 for a 150 maximum-iteration stop criterion optimisation of the full example scenario, where (a) depicts the entire fitness range. On the other hand, (b) focuses on the fitness range of the chromosomes after a successful solution was found in the fifth generation so that the effects of optimisation can be seen over the rest of the process. Here it can be seen that the population leader of the initial seed solutions had a very poor fitness value, which then rapidly improved to the point that a usable solution was found in just 5 intervals. Next, regular improvements occurred up to around the 55$^{th}$ generation, with very little improvement achieved thereafter. Importantly, this shows the effect of diminishing returns of running the genetic algorithm for extended periods of time as it converges toward a single solution, and hence shows the advantages of using the minimal-improvement stop criterion.




(a) (b)

**Figure 4.17.** Population leader fitness over the course of a single run of the algorithm, where (a) shows the entire fitness range and (b) focuses on the values after a successful solution was found.

Next, the genetic diversity of the population is shown in Figure 4.18 as the number of different chromosomes in each generation over the course of the same single 150 maximum-iteration stop criterion optimisation of the full example scenario. It is seen that the algorithm begins with a large amount of genetic diversity after the seeding process, but this reduces rapidly thereafter thus

requiring replenishment by the immigration operator. This can easily be seen as the spikes in genetic diversity over the course of the optimisation proceedure, where immigration occured 40 times over the 150 generations. This can be seen as allowing the algorithm to search over a larger sample space, without having to calculate the fitness of an unnecessarily large population every single generation.

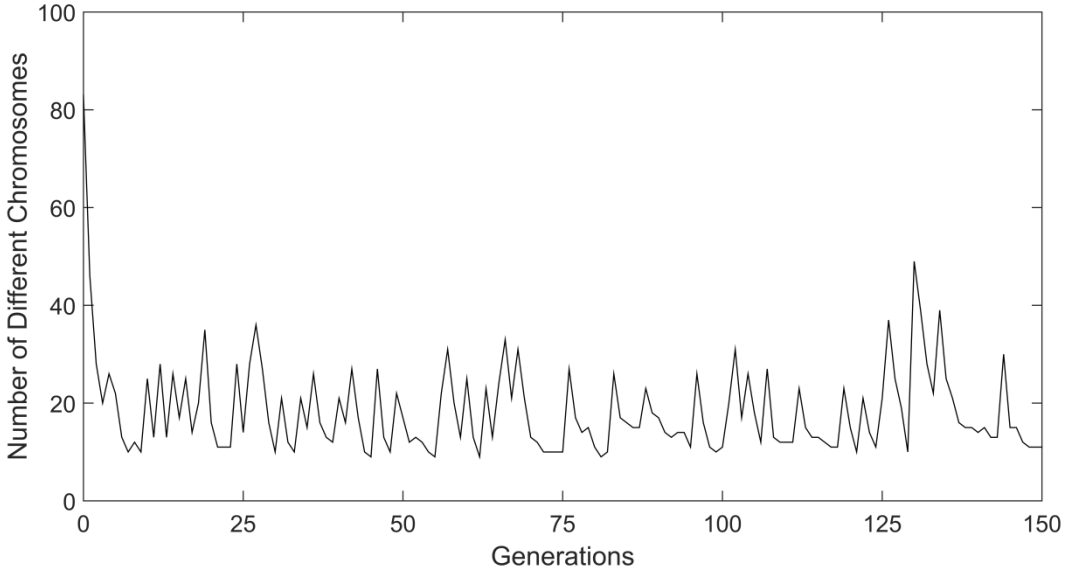



**Figure 4.18. N**umber of different chromosomes in each generation over a single run of the algorithm.

### 4.4.4 Specialised operator performance

The effect of each of the implemented specialised operators on the performance of the genetic algorithm can be seen in Table 4.20. This table contains the performance statistics of the algorithm when each of the specialised operators is removed. For comparison purposes, the performance statistics of the entire algorithm have also been included. Note that all statistics have been generated using the full scenario, along with a maximum-iteration stop criterion of 150 generations, and calculated over 30 runs of the algorithm. Again, as with the stop-criterion performance statistics, the number of times the system failed to generate a successful solution has been included in the right-most column, and the results have been divided into an overall results section, and one that only considers the successful solutions. This prevents the unsuccessful solutions from overshadowing the underlying performance of the algorithm. Further, it is noted that a lot of the overall performance statistics for the iterations to solve column are reported as “n/a”. This is due to the fact that runs that failed to find a successful solution obviously do not have a known number of

**Table 4.20** Algorithm performance statistics with various specialised operators removed.

| Removed Operator | Stat. | Prob. of hit | Danger Score | Cost Score | EMCON Score | Total Fitness | Iterations to Solve | Run Time (s) | No. Failed |
|---|---|---|---|---|---|---|---|---|---|
| None | **Ave.** | 0.1150 | 0.0735 | 0.1518 | 0.3069 | 0.0976 | n/a | 793.82 | 5 |
| | **Med.** | 0.0000 | 0.0736 | 0.1455 | 0.3142 | 0.0670 | 6.00 | 794.71 | |
| | **Min.** | 0.0000 | 0.0697 | 0.1455 | 0.2240 | 0.0652 | 2.00 | 771.12 | |
| | **Max.** | 0.7125 | 0.0771 | 0.1818 | 0.3607 | 0.2563 | n/a | 821.32 | |
| | **Ran.** | 0.7125 | 0.0074 | 0.0363 | 0.1367 | 0.1911 | n/a | 50.20 | |
| | **Successful Only:** | | | | | | | | |
| | **Ave.** | 0.0000 | 0.0733 | 0.1531 | 0.3124 | 0.0671 | 9.32 | 794.02 | |
| | **Max.** | 0.0000 | 0.0771 | 0.1818 | 0.3607 | 0.0695 | 102.00 | 821.32 | |
| | **Ran.** | 0.0000 | 0.0074 | 0.0363 | 0.1148 | 0.0043 | 100.00 | 50.20 | |
| Specialised Crossover | **Ave.** | 0.1188 | 0.0754 | 0.1537 | 0.2985 | 0.0996 | n/a | 790.43 | 5 |
| | **Med.** | 0.0000 | 0.0749 | 0.1455 | 0.3060 | 0.0691 | 7.50 | 789.69 | |
| | **Min.** | 0.0000 | 0.0709 | 0.1455 | 0.2022 | 0.0653 | 2.00 | 756.01 | |
| | **Max.** | 0.7125 | 0.0828 | 0.1818 | 0.3497 | 0.2563 | n/a | 811.93 | |
| | **Ran.** | 0.7125 | 0.0119 | 0.0363 | 0.1475 | 0.1910 | n/a | 55.92 | |
| | **Successful Only:** | | | | | | | | |
| | **Ave.** | 0.0000 | 0.0762 | 0.1553 | 0.2911 | 0.0683 | 13.64 | 791.34 | |
| | **Max.** | 0.0000 | 0.0828 | 0.1818 | 0.3497 | 0.0720 | 120.00 | 811.93 | |
| | **Ran.** | 0.0000 | 0.0114 | 0.0363 | 0.1475 | 0.0067 | 118.00 | 40.24 | |
| Clean-up Operator | **Ave.** | 0.0475 | 0.0713 | 0.1567 | 0.7335 | 0.0927 | n/a | 811.70 | 2 |
| | **Med.** | 0.0000 | 0.0707 | 0.1500 | 0.7322 | 0.0799 | 7.00 | 813.23 | |
| | **Min.** | 0.0000 | 0.0686 | 0.1455 | 0.7104 | 0.0773 | 1.00 | 789.61 | |
| | **Max.** | 0.7125 | 0.0753 | 0.2000 | 0.7814 | 0.2696 | n/a | 839.52 | |
| | **Ran.** | 0.7125 | 0.0067 | 0.0545 | 0.0710 | 0.1923 | n/a | 49.91 | |
| | **Successful Only:** | | | | | | | | |
| | **Ave.** | 0.0000 | 0.0714 | 0.1572 | 0.7311 | 0.0801 | 21.18 | 812.33 | |
| | **Max.** | 0.0000 | 0.0753 | 0.2000 | 0.7650 | 0.0839 | 88.00 | 839.52 | |
| | **Ran.** | 0.0000 | 0.0063 | 0.0545 | 0.0546 | 0.0066 | 87.00 | 49.91 | |
| Survival Operator | **Ave.** | 0.6872 | 0.0722 | 0.1464 | 0.3293 | 0.2497 | n/a | 739.67 | 28 |
| | **Med.** | 0.7125 | 0.0718 | 0.1455 | 0.3333 | 0.2563 | n/a | 742.85 | |
| | **Min.** | 0.0000 | 0.0690 | 0.1455 | 0.2787 | 0.0669 | 5.00 | 717.32 | |
| | **Max.** | 0.8075 | 0.0760 | 0.1727 | 0.3716 | 0.2833 | n/a | 767.01 | |
| | **Ran.** | 0.8075 | 0.0070 | 0.0272 | 0.0929 | 0.2164 | n/a | 49.69 | |
| | **Successful Only:** | | | | | | | | |
| | **Ave.** | 0.0000 | 0.0713 | 0.1591 | 0.3689 | 0.0680 | 6.00 | 731.15 | |
| | **Max.** | 0.0000 | 0.0714 | 0.1727 | 0.3716 | 0.0691 | 7.00 | 740.38 | |
| | **Ran.** | 0.0000 | 0.0003 | 0.0272 | 0.0055 | 0.0022 | 2.00 | 18.46 | |
| Seed One | **Ave.** | 0.0494 | 0.0771 | 0.1636 | 0.2419 | 0.0810 | n/a | 799.20 | 2 |
| | **Med.** | 0.0000 | 0.0771 | 0.1636 | 0.2459 | 0.0676 | 7.00 | 798.45 | |
| | **Min.** | 0.0000 | 0.0746 | 0.1455 | 0.1694 | 0.0655 | 3.00 | 777.76 | |
| | **Max.** | 0.8075 | 0.0810 | 0.1909 | 0.2896 | 0.2842 | n/a | 819.12 | |
| | **Ran.** | 0.8075 | 0.0064 | 0.0454 | 0.1202 | 0.2187 | n/a | 41.36 | |
| | **Successful Only:** | | | | | | | | |
| | **Ave.** | 0.0000 | 0.0771 | 0.1643 | 0.2406 | 0.0678 | 9.68 | 799.68 | |
| | **Max.** | 0.0000 | 0.0810 | 0.1909 | 0.2896 | 0.0699 | 54.00 | 819.12 | |
| | **Ran.** | 0.0000 | 0.0064 | 0.0454 | 0.1202 | 0.0044 | 51.00 | 41.36 | |
| Seed Two | **Ave.** | 0.6148 | 0.0795 | 0.1646 | 0.2605 | 0.2340 | n/a | 802.58 | 16 |
| | **Med.** | 0.7163 | 0.0782 | 0.1500 | 0.2732 | 0.2590 | n/a | 799.87 | |
| | **Min.** | 0.0000 | 0.0719 | 0.1455 | 0.1639 | 0.0671 | 7.00 | 775.82 | |
| | **Max.** | 2.1075 | 0.0887 | 0.2636 | 0.4153 | 0.6296 | n/a | 838.75 | |
| | **Ran.** | 2.1075 | 0.0168 | 0.1181 | 0.2514 | 0.5625 | n/a | 62.93 | |
| | **Successful Only:** | | | | | | | | |
| | **Ave.** | 0.0000 | 0.0814 | 0.1669 | 0.2373 | 0.0706 | 47.64 | 803.91 | |
| | **Max.** | 0.0000 | 0.0868 | 0.2455 | 0.3005 | 0.0779 | 121.00 | 825.68 | |
| | **Ran.** | 0.0000 | 0.0127 | 0.1000 | 0.1366 | 0.0108 | 114.00 | 49.85 | |

iterations required to find a solution to the scenario. Finally, it is noted that in most cases, the number of times a specialised operator is performed is simply set to zero. This includes not performing the operator at any point outside the generational loop. However, where a directly comparable technique exists within the algorithm, its number of executions was adjusted accordingly to allow for a fair test. In particular, this applies to the two procedural seeding techniques where, for example, the number of seed solutions generated by the second seeding technique were set to 20 when the first seeding technique was removed so as to keep the percentage of non-random solutions in the initial population at 20%.

*1) Specialised crossover*

This specialised operator takes advantage of the varying levels of independence between the countermeasures of different time intervals by breaking a number of selected chromosomes up into smaller pieces and using tournament selection to compare the strategies for each subsection of the mission in order to generate a child chromosome. Specifically, the chromosomes are broken up into halves, quarters, sixths, and individual time intervals in order to improve performance over the entire optimisation process, where combining larger, more-different segments theoretically improves performance in the beginning. Thereafter, combining the more homogenous small segments should theoretically improve performance in the later parts of the optimisation process. In all, this should improve both the rate at which the algorithm finds a successful solution, and the median fitness of the solution generated over the entire course of optimisation. This is seen in the results, where removing this specialised operator worsens median fitness from 0.0670 down to 0.0691, and increases the median number of iterations required to find a solution from 6 to 7.5, which equates to a 3.13% and a 25.00% change respectively. The 3.13% change may not appear to be significant, but the weakest successful solution generated by the full algorithm has a fitness of 0.0695, hence indicating that the median performance of this handicapped algorithm is not far off the worst performance of the full. Importantly, removing this operator has a very minimal effect on the run time of the algorithm, dropping the median run time down from 794.71 s to 789.69 s, which equates to a 0.63% change. Therefore, this clearly indicates that this operator is not very computationally intensive and hence definitely worth the trade-off.

*2) Clean-up operator*

This specialised operator aims to improve optimisation performance by cleaning-up randomly selected chromosomes by identifying and removing unnecessary dead time intervals, and intervals

in which the towed decoy is allocated to channels with no jamming technique. Theoretically, this allows for chromosomes to be compared more accurately because their fitness will depend directly on their underlying strategy, rather than artefacts of the stochastic optimisation process. As such, this operator should improve the overall performance of the genetic algorithm. This is seen in the results, where the removal of this operator drastically worsens the median fitness of the generated results from 0.0670 to 0.0799, which equates to a 19.25% change. Further, the median number of iterations required to find a successful solution to the scenario increases by 16.66% from 6 to 7, whilst the successful-only average number of iterations to solve increases by 127.25% from 9.32 to 21.18. Interestingly, the median probability of hit, danger, and cost objective function scores are relatively unaffected. However, the median EMCON score weakened dramatically by 133.04% when the clean-up operator was removed, which equates to a score increase from 0.3142 to 0.7322. This makes sense as the dead time intervals are considered in the calculation of the EMCON score. As such, these strategies contain many unnecessary technique changes, causing the platform's channels to be in a constant state of change, which in turn results in poor strategies. Also, it is noted that the removal of this operator has a minimal effect on the run time of the algorithm, with it actually slightly slowing down the algorithm from a median run time of 794.71 s to 813.23 s, which in turn indicates that this operator is computationally efficient. Finally, it is noted that the removal of this operator actually decreases the number of failed runs of the algorithm from 5 down to 2. However, the increased fitness of the generated solutions, and this operator's computational efficiency makes this trade-off worthwhile.

*3) Survival operator*

The goal of this specialised operator is to rapidly find successful solutions to the scenario by examining randomly selected chromosomes and identifying the times in which the platform is hit by a threat, and then breaking the lock of that threat in the preceding time intervals. As such, the removal of this specialised operator should result in reduced algorithm performance. In particular, the handicapped algorithm should have a reduced ability to generate successful solutions to the scenario. This is abundantly clear in the results where the removal of this operator reduces the algorithm's success rate from 83.33% down to just 6.67%, with the handicapped algorithm only generating 2 successful strategies out of 30 runs of the algorithm. Further, the average fitness of the two successful solutions is only 0.0680, which is worse than the 0.0671 success-only average fitness of the full algorithm. However, the median run time of the handicapped algorithm is 742.85 s, which is 6.53% faster than median run time of the standard algorithm. This indicates that

this operator is very computationally expensive, but its exceedingly large impact on the performance of the algorithm more than makes up for this.

*4) Seed method one*

This specialised operator procedurally generates a reasonable solution to the scenario to seed the initial population so as to increase the overall rate of convergence of the algorithm. Overall, this method essentially allocates the most effective countermeasure technique to the two most dangerous threats in every time interval. As such, this operator should increase algorithm performance both in terms of the overall fitness of the generated solutions and the number of iterations required to find a successful solution. This is seen in the results, although to a relatively small extent. The removal of this operator reduces the median fitness of the generated solutions from 0.0670 to 0.0676, which is only a 0.90% reduction. Further, the median number of iterations required to find a successful solution increases from 6 to 7, which equates to 16.67% increase. However, the removal of this operator has almost no impact on the run time of the algorithm, with its removal actually increasing the median run time by 0.47%. Therefore, due to the lack of downsides, the small improvement in performance justifies the use of this specialised operator in the algorithm.

*5) Seed method two*

The goal of this specialised operator is also to improve overall algorithm performance by seeding the initial population with a procedurally generated reasonable solution to the scenario. This seed method is the same as the previous one except that when the most or second most dangerous threat is in a search-type stage it allocates a wide-band NJ technique, where the centre frequency and bandwidth are determined by a small exhaustive search. In comparison to the first seed method, the benefits of this operator are clearly seen in the results, where the removal of this operator reduces the algorithm's success rate from 83.33% to just 46.67%, with the handicapped algorithm generating just 14 successful solutions to the scenario. Further, the average fitness of these successful solutions is only 0.0706 versus the 0.0671 success-only average fitness of the full algorithm. In fact, this average fitness of the handicapped solution is worse than the weakest successful solution generated by the full algorithm. Again, as with the previous seed method, this operator is computationally inexpensive, as its removal has almost no effect on the run time of the algorithm, with it actually causing a 0.65% increase in the median run time. Therefore, the inclusion of this operator is essential in the algorithm

*6) Pareto reuse*

In the case of Pareto optimisation, the specialised operator is the method of reseeding the population for each run of the algorithm, whereby the population leader of each run is protected through elitism and a set percentage of the remaining population randomly regenerated. This essentially seeds the next run of the algorithm with very fit solutions by reusing the previously optimised population. It has the advantage being able to generate solutions more rapidly due to the algorithm's better starting point. It also tends to generate more similar solutions, which is beneficial for decision-support systems as it allows for more direct comparison of the Pareto solutions by human users. However, it does mean that the generated Pareto designs tend to gravitate towards the same local minimum, rather than the global minimum for their particular set of weights.

As previously, the half scenario provides a better example for demonstrating the differences between the Pareto designs due to its less saturated threat environment. As such, the overall algorithm performance for 30 runs of the algorithm is presented in Table 4.21. Note that the full algorithm's performance (reuse) is repeated from Section 4.4.2 alongside the handicapped algorithm's performance (random) for the ease of the reader. Further it is noted that the handicapped algorithm generates a single initial population that is simply reused for each individual Pareto design. Also, for these results, the minimal improvement stop criterion has been used with an iteration limit of 150. This is for the same reasons as discussed in Section 4.4.2, where a user would require a number of Pareto designs in a short amount of time, and because the reuse operator was specifically included to improve the Pareto performance using this stop criterion. Lastly, since these results are generated using the half scenario, there is a 100% success rate. As such, the associated probability of hit and number of failed run columns have been excluded from the results.

**Table 4.21** Overall Pareto performance statistics comparison for different reseed approaches.

| Reseed | Statistic | Total Fitness | Total Iterations | Run Time (s) |
|---|---|---|---|---|
| **Reuse** | **Average** | 0.1557 | 157.40 | 821.02 |
| | **Median** | 0.1560 | 160.50 | 831.51 |
| | **Minimum** | 0.1493 | 119.00 | 622.65 |
| | **Maximum** | 0.1631 | 203.00 | 1024.21 |
| | **Range** | 0.0138 | 84.00 | 401.56 |
| **Random** | **Average** | 0.1557 | 222.97 | 1121.69 |
| | **Median** | 0.1546 | 229.00 | 1156.63 |
| | **Minimum** | 0.1493 | 151.00 | 778.52 |
| | **Maximum** | 0.1667 | 294.00 | 1454.48 |
| | **Range** | 0.0174 | 143.00 | 675.96 |

It can be seen that, as expected, the alternative Pareto approach generates generally fitter solutions than the approach implemented in the algorithm. It achieves a total median fitness across the three Pareto designs of 0.1546 versus the 0.1560 of the implemented algorithm. However, both designs actually achieve the exact same average value. Further, it is seen that the alternative approach requires a median 229 iterations to generate its 3 Pareto designs in comparison the just 160.5 of the implemented design. This in turn translates into median run times of 1156.63 s and 831.51 s respectively, which means that the alternative approach is 39.10% slower in order to attain a fitness improvement of just 0.90%. Therefore, this trade-off favours the selection of the implemented approach.

## 4.5 CHAPTER SUMMARY

In this chapter the results of the implemented system were presented using two example scenarios: a complex scenario, and a comparatively simple one. This began with a breakdown of the example scenarios including their layout, waypoints, and threat characteristics. Thereafter, the results themselves were presented. This began with an in-depth analysis of the best countermeasure strategies developed for each scenario. It was found that these strategies indeed made sense within the context of their respective missions, and highlighted some interesting approaches to countermeasure allocation. Also, some issues with the system were highlighted such as the optimisation algorithm's bias towards NJ, CP, RGPO and VGPO techniques.

Next, the performance of the genetic algorithm itself was analysed. This began with a comparison of the performance statistics of the various stop criteria. It was found that the stop criteria performed as desired, with the maximum-iterations technique generating consistently strong solutions, the safe-passage criterion rapidly generating successful solutions, and the minimal-improvement criterion generating a strong compromise between the two. Thereafter, a comparison of the performance statistics of three different Pareto solutions found that each achieved its intended purpose, where conservative weights generated strategies with the lowest danger to platform, and the EMCON-focused weights generated strategies with the lowest EM emissions. Further, the cost-focused weights generated strategies with performance statistics between the two due to population reuse and the fact that the conservative strategy used minimal cartridges to begin with. Additionally, the overall genetic algorithm performance was discussed with regard to the fitness of the population leader over time, along with the population diversity due to the effects of

immigration. Lastly, the effects of the specialised operators were considered by presenting and discussing the performance of the genetic algorithm with each individually removed. It was found that each improved the performance of the algorithm.

# CHAPTER 5 DISCUSSION

## 5.1 CHAPTER OVERVIEW

In this chapter a discussion of the results is presented. It begins with a discussion of the performance of the generated strategies themselves, and their shortcomings, in Section 5.2. A number of potential solutions to these shortcomings are also presented such as simulator-populated lookup tables, and improved modelling. Thereafter, the performance of the optimisation procedure is discussed in Section 5.3. Again, the shortcomings are highlighted and a number of potential solutions are presented, including the use of more seeding techniques, island optimisation, and antenna angle sub-optimisation, amongst others. Lastly, both the current and future potential applications of this work are explored in Section 5.4. This includes uses in training, as well as service applications where this system could be used either in a decision-support role, or directly for the optimisation of cartridge load-out and countermeasure strategy. The section concludes with a discussion of potential sub-applications of this work where it could be used in conjunction with other systems as an initial coarse optimisation stage, or for other applications such as route and manoeuvres optimisation.

## 5.2 DISCUSSION OF RESULTS

It was seen in the previous section that the strategies developed by this system do indeed make sense in the context of the scenario, where the choices can be explained on an allocation-by-allocation basis. Further, the results of this approach, along with the model itself, have been validated through the successful publication and peer review of this work in respected scientific publications. Importantly, the computational complexity of the model has been shown to be sufficiently low so as to allow for the rapid generation of these strategies in an average time of just 13 minutes. In comparison, contact was made with the team leader of the SEWES electronic warfare simulation environment at the Council for Scientific and Industrial Research (CSIR) [39].

It was stated that SEWES, which is a parameter-level simulation environment, is capable of running simulations at 10 to 20 times faster than real time on a large, powerful computer cluster. However, this performance is only achieved for scenarios that are considerably simpler than the full one in this work. Therefore, making the optimistic assumption that SEWES could run this scenario at the upper bound of 20 times faster than real time, means that the 15 000 simulations required for the maximum-iteration stop criterion, would require over 25 hours, and a full physics-based simulator would take even longer. Similarly, investigating chaff firing patterns using SADM "on one machine can take many days to simulate" [8], and that is only to consider seduction chaff operating in isolation. Therefore, it can be concluded that this work does indeed achieve its original research goal of developing a system capable of determining a reasonable countermeasure strategy and load-out in a reasonable amount of time.

There are a number of instances where non-optimal countermeasure allocations have been made, but this convergence to local minima is to be expected considering the exceedingly large problem space in which the optimisation algorithm operates, especially when a focus has been placed on the rapid generation of solutions. Importantly, these non-optimal allocations can be easily identified in the type of in-depth analysis used in this work, where changes can be made to a developed strategy and its fitness recalculated in order to determine the suitability of these changes. This highlights the application of this work as a decision-support tool that can be used to provide a framework countermeasure strategy to a human user that highlights more efficient approaches to the mission that the human user otherwise might not be aware of. The user can then modify and improve this framework strategy by identifying and fixing such non-optimal allocations, along with making other modifications according to their experience, before testing the validity of their changes by recalculating the fitness of the new strategy.

However, it is noted that this model does make a number of simplifying assumptions both in order to reduce computational complexity, and in order to keep the work within a reasonable scope. This creates the opportunity for a number of potential improvements that can be made in the future. Some possibilities are discussed below.

### 5.2.1 Lookup-table population

The first and foremost improvement that can be made to this work would be the development of a specific set of simulations that can be run in a low-level physics- or parameter-based simulator

such as SEWES in order to generate accurate lookup tables and parameter values. This would allow a user to simply run this set of simulations with the specific platform and threats it will encounter, and all the necessary values will be generated as accurately as possible, in turn resulting in the most accurate countermeasure optimisation possible.

### 5.2.2 Improved modelling

Due to the lack of published research into this field in the literature, this work has been developed from scratch, with additional functionalities added, expanded and developed over time. As such, there is a lot of scope to continue this process and improve the accuracy of the model. The first example of this would be the development of the modelling of passive countermeasures as a whole. Currently, the system relies heavily on other low-level simulators to determine an overall passive-countermeasure effect, rather than attempting to model these effects in a detailed way over time. For instance, in the current system angular information is ignored in the modelling of these countermeasures, even though this has a very large impact on their effectiveness, especially in cases where the jamming effect must be determined for non-primary targets. Further, this impact is time-based as the platform and threats move, and the passive countermeasures bloom. Therefore, the accurate modelling of these phenomena is a clear avenue for improvement of the system's performance.

Also, there is still a lot of room for development in the modelling of active countermeasures. For instance, the current system allocates an overall countermeasure resistance to each threat type according to its counter-countermeasure capabilities. However, this can be expanded upon so that each threat type has a separate countermeasure resistance for each individual countermeasure technique according to the specific counter-countermeasures it has at its disposal. For example, if a specific threat type has a track-on-jam capability, that threat type can be set to have a strong countermeasure resistance against noise jamming.

Further, a number of the assumptions made in this work can also be replaced with more detailed modelling. A very important example of such a potential improvement would be to include threat communication in the model. This would not only improve the model overall, but also drastically change the generated approach to a scenario, where the platform would have to prioritise the prevention of its detection, especially in the early stages of the mission. Also, the platform would be discouraged from leaving threats in the tracking stage any longer than absolutely necessary.

Another such example would be to include a number of friendly platforms to the optimisation process, so that the countermeasure strategy for the entire group may be optimised as a whole. Importantly, this would require all interactions and interference between the friendly platforms to be modelled, as well require changes to the models used for the threats themselves. In turn, the addition of friendly platforms would also allow for the inclusion of alternative jamming approaches such as escort jamming, stand-in jamming, and stand-off jamming, over and above the current approach of self-protection jamming. In addition, the system's often-implemented approach of short-range jamming is potentially dangerous. This could be overcome with a weapon-specific keep-out range, within which it would be too late to jam that threat, subsequently requiring the system to break its lock earlier in the mission.

Lastly, in this work the effects of the environment have been ignored for simplicity. As such, this is yet another avenue that could be explored, where in particular the effects of terrain and weather can be included so that the platform is able to take advantage of these in the development of its countermeasure strategy and overall approach to a mission.

## 5.3 OPTIMISATION PERFORMANCE

It was seen in the results section that the genetic algorithm performed very well overall in that it was able to consistently generate not only successful, but good solutions to difficult, saturated scenarios in a relatively short amount of time. Further, the genetic algorithm has the capability to tailor its performance according to the application and needs at hand, where it can rapidly find an acceptable solution, a highly optimised solution, or something in between. Also, it has the ability to generate a number of different Pareto solutions that focus on different strategy characteristics, which in turn rapidly provides a human user different approaches to the mission. This not only improves their confidence in the selected countermeasure strategy, but also allows them to apply their experience in choosing and possibly combining strategies.

However, a number of issues were noted that were the direct consequence of the optimisation algorithm and its genetic operators. In particular, a number of issues were the result of the implemented specialised operators, such as the system's bias towards RGPO, VGPO, CP and NJ techniques. These are far from ideal, but it was also shown that these operators were necessary in order to improve the performance of the algorithm, both in terms of the strength of the generated

solutions and the rate in which they were generated. As such, the solution to this issue and others would be to modify the genetic algorithm itself, and add a number of different specialised operators that attempt to balance this. A number of these potential improvements are discussed below.

### 5.3.1 More seeding techniques

One of the main reasons for the bias of the genetic algorithm towards specific countermeasures is due to the use of these countermeasures in the procedural seeding techniques, which is then further compounded by the effects of dead time. This can be overcome by creating procedural seeding techniques that introduce other countermeasure techniques such as MFT into the population in comparatively fit solutions, thus allowing them to be more prevalent, and in turn increases the algorithm's ability to converge to the optimal solution. This effect can actually be observed using the second procedural seeding technique on the half scenario. This seeding technique specifically introduces various bandwidths of NJ into the population, and if it is removed from the algorithm, the generated results tend to use CP against search-type threats. This in turn results in reduced solution fitness for this scenario.

### 5.3.2 Island optimisation

Due to the nature of the system, where for example an entire countermeasure allocation will often need to be replaced with a new one before an improvement in fitness is seen, the genetic algorithm tends to get trapped in local minima. In particular, this is seen in the times where the algorithm is unable to generate a successful solution to the scenario. Island optimisation can potentially overcome this issue and also generate generally fitter solutions by maintaining a number of independent solutions that will each tend toward a different local minimum. These populations can then be mixed after a number of generations in order to combine the characteristics of the different populations.

### 5.3.3 Dead-time switching

This is a more specialised approach to the above problem that focuses on one of the biggest culprits: dead time. Dead time counteracts the ability of the mutation operator to introduce new genetic material into the population. This is because when a mutation occurs, it changes a single characteristic of a single time interval of a countermeasure allocation out of the numerous time

intervals in which it is executed. This constitutes a change in countermeasures and tends to cause that altered time interval to become a dead time interval, which for example can stop a tracking-focused countermeasure from breaking lock, thus actually reducing the fitness of that solution. That solution will then be less likely to reproduce, even if that mutation was moving the strategy towards becoming a fitter one. As such, the proposed solution would be to temporarily switch off the effects of dead time during the optimisation process. This would only be done for a number of generations at a time, and would include the cleaning operator as it can potentially exaggerate this effect. This would then allow mutations to occur feely and potentially find improvements to the population. Thereafter, the effects of dead time, and the cleaning operator can be switched back on for a couple generations in order to clean up the population, and allow for more direct comparisons of fitness. This can then be cycled throughout the entire optimisation process, before ending on a cleaning period in order to ensure clean solutions.

### 5.3.4 Allocation mutation

This specialised operator is also a potential way of overcoming the effects of dead time described above. Similarly to standard mutation, this operator will introduce new genetic material to the population by randomly changing the characteristics of the countermeasure allocations. However, instead of changing the characteristics of a single time interval, this operator would change the characteristics of an entire countermeasure allocation that has been randomly selected. This approach would help avoid causing unnecessary dead time and reduce the reliance on chance that a number of time intervals in a row will receive the same mutation.

### 5.3.5 Antenna angle sub-optimisation

This operator could potentially be used to improve the antenna angles of countermeasure strategies, by implementing a separate numerical sub-optimisation procedure for optimising the antenna directions of countermeasures. This would improve performance due to the large number of potential antenna directions, and prevent poorly selected antenna directions from reducing the effectiveness of otherwise well selected countermeasure allocations.

## 5.4 POTENTIAL APPLICATIONS

This work has been developed with a number of applications in mind, where some can be realised with the work in its current form, whilst others can only be realistically achieved with continued development. These are discussed below.

### 5.4.1 Training applications

The most promising application of this work is in the training of military personnel. Due to limited interaction with the EMS as a whole, these personnel often lack the intuitive understanding required to effectively operate systems that function in the EMS. Even experienced EW operators and decision makers are not fully aware of all the effects and implications of their actions through no fault of their own, but simply due to the sheer complexity of the environment. There is no way for them to know for certain which threats in a situation should receive priority for limited jamming resources, or whether it would be more effective to jam a single more dangerous threat, or to jam a number of less dangerous ones simultaneously. That is before even considering the future effects of current actions, or overall strategic goals such as balancing the competing objectives of mission cost and platform safety. However, this is not limited to the operation of EW systems alone, as this lack of understanding can have dire consequences for all military personnel at all levels. Note that this section is based on the work published in the author's conference paper [10].

This must be overcome through better training that aims to build a more intuitive understanding of EW and the EMS as a whole. This can be achieved through placing a greater emphasis on more visual and interactive approaches rather than purely theoretical or mathematical ones. This is where a system such as this one comes in. The problem needs to be approached at a number of levels, where the first is to build an innate understanding of EW interactions for personnel at all levels, by demonstrating the effects of countermeasures in a visual and interactive way. This requires that all necessary information about the EMS should be gleaned at a glance: the danger presented by threats, the effects of countermeasures, illumination of the platform, and the progress of threats through their engagement procedure. Further, training of EW operators and decision makers should help them to identify better approaches to countermeasure allocation than, for example, simply jamming the most imminent threat at a point in time. Lastly, the quality of training can be greatly

improved by the quantitative evaluation of trainees' strategies against benchmarks, so as to indicate areas of possible improvement.

Before this work, this type of EW study could only be achieved with complex low-level simulations that are simply too slow for EW strategy optimisation purposes. Further, in these systems the user is left to sift through the data and propose better solutions without any assistance from the tool. Also, the previously developed high-level simulation approaches could not be used for these purposes as those systems simply allocate countermeasure resources, rather specific techniques. More importantly, these systems do not take into account a number of vital interactions that are inherent to the EMS, hence also producing poor solutions for this application.

As such, this work clearly provides the opportunity to develop and improve EW and EMS training methods at all levels. Three different applications of this work will be discussed below. Most importantly, such training is already possible with this work in its current form. It is noted that for this section, a trainee's strategy is needed for comparison purposes. Due to the first procedural seeding technique's simple and intuitive approach of simply jamming the most and second-most dangerous threats using pre-determined techniques, it will be used as a representative strategy for a trainee human strategist. Significantly, it is further noted that this approach was unable to find a successful strategy for the full scenario, hence indicating the importance of improving approaches to countermeasure allocation.

*1) Visual and interactive learning*

The first training application of this work is to use it to create a visual and interactive learning experience for all military personnel in order to improve their general understanding of EW and the EMS as a whole. In order to do this, all essential information must be displayed on a time-interval-by-time-interval basis in such a way that the user can glean all of it at a glance. This is achieved using the scenario images in Figures 4.3 – 4.16 that were used to do an in-depth analysis of both the full and half scenario strategies in Section 4.3.1 and 4.3.2. These images depict all threats, the level of danger they present, their stage of engagement, how they are affected by the current countermeasures, as well as the countermeasures themselves and their associated antenna directions or decoy use in a very easy-to-digest manner. Further, these images are generated for each individual time interval, and can be used to create a picture of the entire countermeasure strategy and its effects. This allows for the very in-depth analysis of EW strategies as seen in those sections,

which in turn results in many important insights into a particular mission, countermeasure strategy generation, and even the EMS and EW environment as a whole. This is absolutely essential in both building an innate understanding of this environment, and improving trainees' ability to develop EW countermeasure strategies. Further, the low computational cost of TECA allows for the possibility of expanding this to interactive training approaches where the user can, on a time-interval-by-time-interval basis, select a countermeasure and see its effect immediately, not only on the current time interval, but also on the mission as a whole. This allows trainees to see and interact with, and hence learn about the unique characteristics of the EMS, such as platform illumination, technique interaction and the only temporary nature of countermeasures.

*2) Superior strategy as an evaluation tool*

The next training application is aimed more specifically towards the training of EW operators and decision makers, where the computer-developed strategies can be used as benchmarks for rapid and quantitative evaluation of trainees' strategies. More specifically, the individual objective function values and resultant total fitness used to measure the performance of the population members in the genetic algorithm, can be used to quantify the performance of a trainee's strategy. The performance of the trainee's strategy in each metric can then be compared to what is theoretically superior, highlighting their areas of weakness that need to be improved upon. Moreover, these metrics can also be viewed on a time-interval-by-time-interval basis, allowing the trainee to see specifically which parts of the mission they need to focus on.

Using the full scenario as an example, the first procedural seed method as a representative trainee, and the previously-discussed 0.0652 fitness 150 maximum-iteration stop-criterion strategy as the benchmark, results in the metrics comparison in Table 5.1. Here it is seen that the trainee's solution performed very poorly in almost all areas, and results in the platform being hit 4 times, with a total representative probability of hit of 2.9150, which equates to an average probability of hit of 0.7288 for each individual instance. Interestingly, the trainee's solution actually results in a lower danger value over the course of the entire mission. However, it can also be seen that this 3.39% improvement comes at the expense of a 159.34% increase in EM emissions, and 124.95% increase in cartridge use, which is clearly a poor trade-off. Therefore, it can be seen that the approach used by this trainee is far too conservative, and does not place enough emphasis on the other important strategy characteristics. Moreover, the poor trade-off between danger and the other metrics

indicates that the countermeasure use of this approach is very inefficient, which in turn results in poor fitness.

**Table 5.1** Strategy metrics evaluation.

| Metric | Computer Developed | Seed ('Trainee') | Percentage Difference |
|---|---|---|---|
| **Probability of Hit** | 0.0000 | 2.9150 | n/a |
| **Danger Score** | 0.0707 | 0.0683 | 3.39 |
| **Cost Score** | 0.1455 | 0.3273 | 124.95 |
| **EMCON Score** | 0.3224 | 0.8361 | 159.34 |
| **Total Fitness** | 0.0652 | 0.8703 | 1234.82 |
| **Cartridges Used** | 16 | 36 | 125.00 |
| **No. of Times Hit** | 0 | 4 | n/a |

Next, the performance of the trainee's strategy can be analysed in greater depth by looking at these metrics over the course of the mission. As such, the representative probability of hit, danger score, cost score, EMCON score, and total fitness over time are depicted in Figures 5.1 – 5.5, where the TECA generated strategy is depicted in black and the trainee strategy is depicted in blue. Note that for all of these metrics, a lower value is preferred. Beginning with Figure 5.1, this figure shows the cumulative representative probability of hit increasing over the course of the mission. Here it is immediately seen that all four potential hits occurred during the saturated middle section of the mission. More specifically, the platform was potentially hit in the time intervals beginning 46, 58, 78, and 88 s into the mission, indicating where the platform needs to be saved by last-minute countermeasures. Thereafter, Figure 5.2 shows the normalised post-jamming danger value presented to the platform in each individual time interval. Immediately it is seen that the mission can be divided into three segments: the beginning, middle, and end. In the beginning part of the mission, the two strategies actually perform quite similarly, but they begin to diverge as the platform approaches the more saturated middle part of the mission, where the juggling of threats is more difficult. In this middle section, it is seen that the TECA-generated strategy actually exposes the platform to a very large initial spike in danger, but then this has the trade-off of exposing the platform to reduced levels of danger later in the section. Thereafter, the TECA strategy actually exposes the platform to a greater level of danger in the end section of mission than the trainee strategy. However, this is intentional as this is the period of the mission where that strategy has ceased using countermeasures as the platform is capable of escaping the mission unharmed. In fact, it is in this section of the mission where the trainee strategy gains its 3.39% reduction in danger, and the majority of its poor cost and EMCON scores. This is confirmed by looking at the other figures, where the cumulative cost score of the trainee strategy continues to climb steeply in Figure

5.3, whilst the score of the TECA strategy remains constant. Also, in this section of the mission in Figure 5.4, the cumulative EMCON score of the trainee strategy stagnates at this point, whilst the score for the TECA strategy drops linearly over time. The culmination of these observations is seen the final figure that depicts the total cumulative fitness of the strategies over time, where the two strategies are very similar in the beginning section of the mission, but then diverge towards the middle section, especially once the platform starts getting hit by threats.

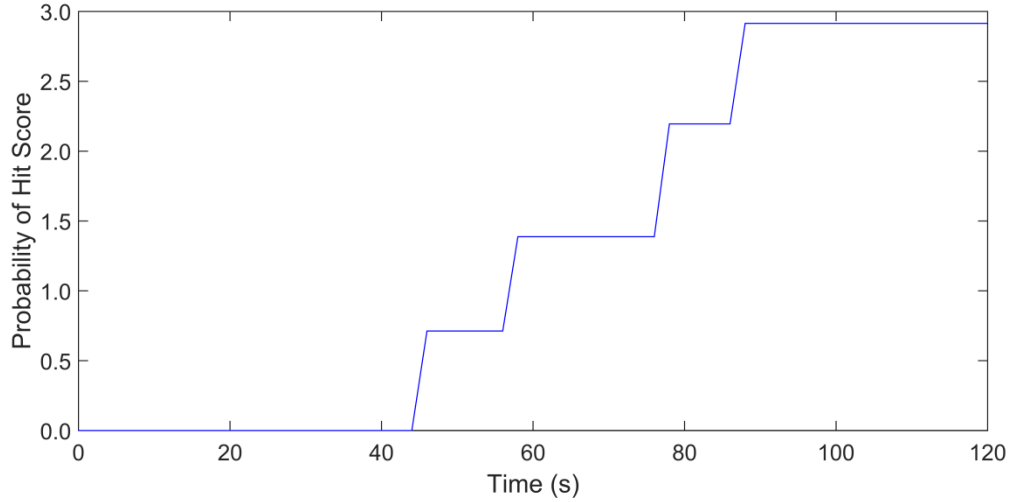



**Figure 5.1.** Cumulative **r**epresentative probability of hit versus time.

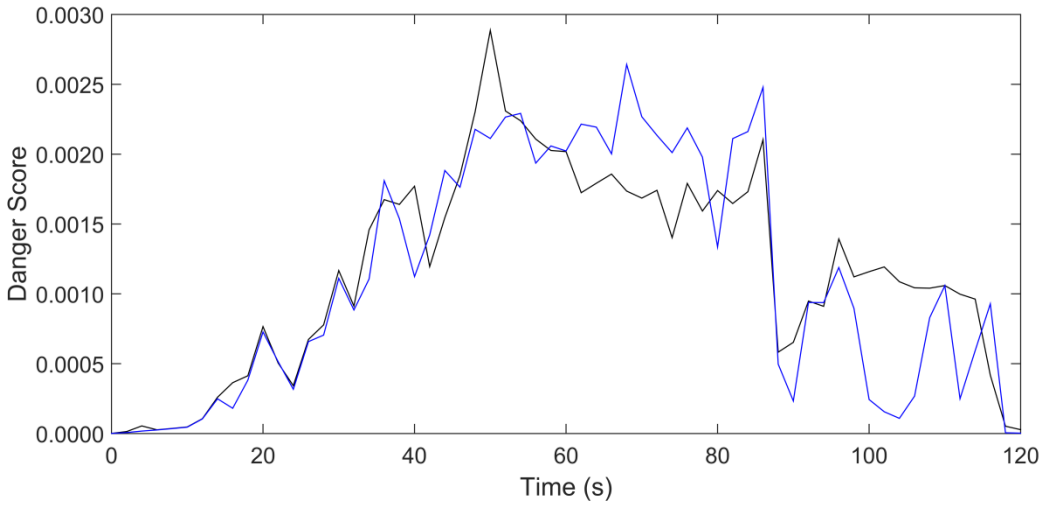



**Figure 5.2.** Danger score versus time on a time-interval by time-interval basis.

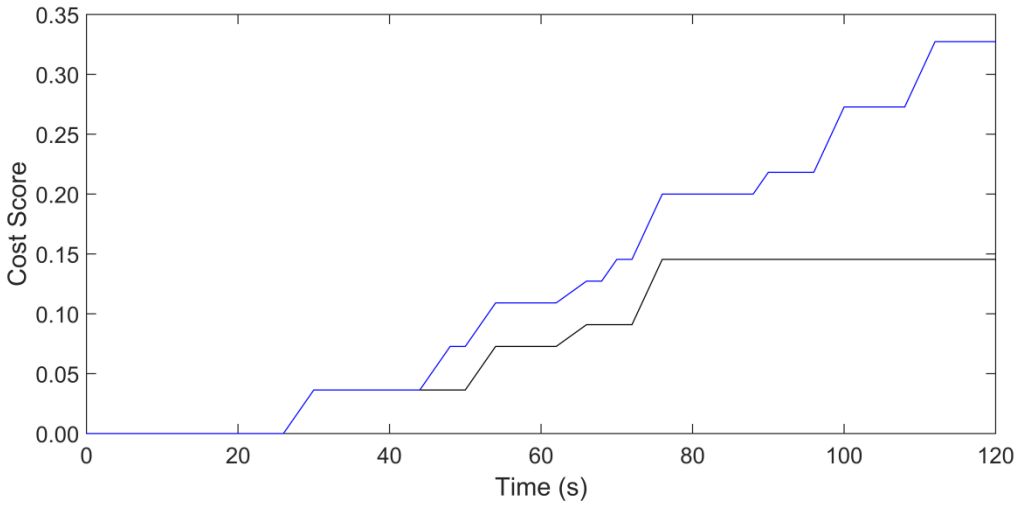



**Figure 5.3.** Cumulative cost score versus time.

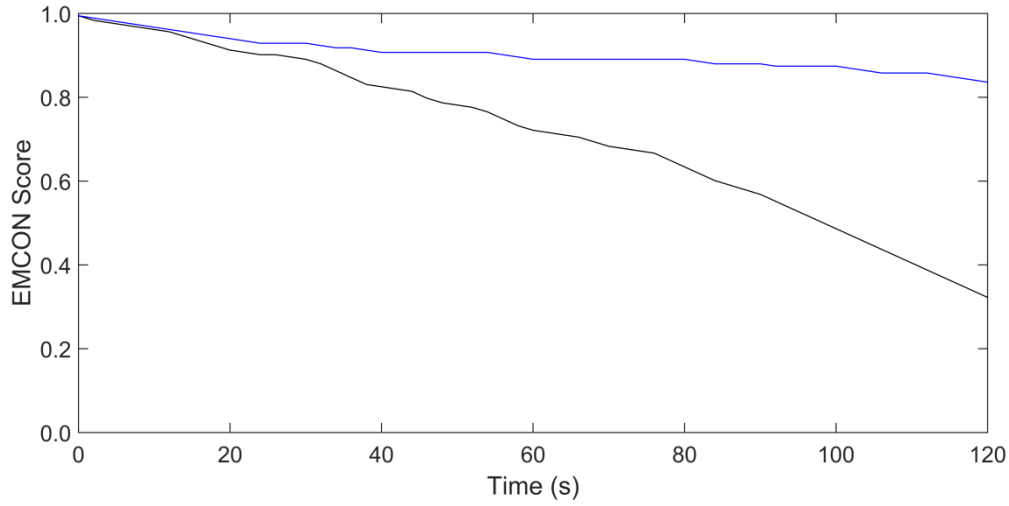



**Figure 5.4.** Cumulative EMCON score versus time.

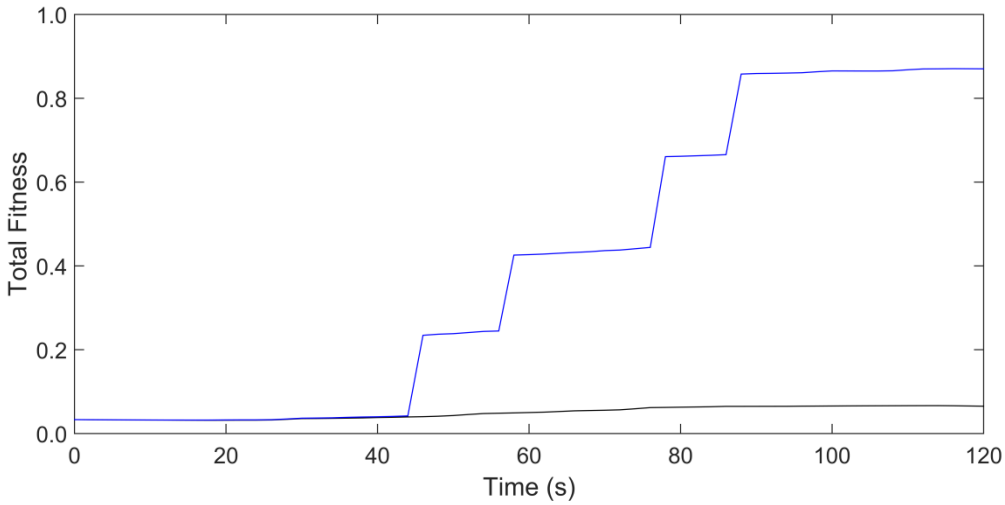



**Figure 5.5.** Total cumulative fitness versus time.

As such, it can be concluded that the trainee performs well in low-danger situations with relatively few threats, but struggles once the threat environment becomes more saturated. Further, they need to learn to determine the point at which the platform can safely cease using countermeasures and rely on its ability to outrun the remaining threats. In particular, it can be seen that this trainee strategy can be quite substantially improved by simply including some last-minute RGPO or VGPO in the middle section of the mission, and simply ceasing the use of countermeasures in the last quarter of the mission. For instance, it can be seen by looking at Figure 5.3 that if all passive countermeasures were ceased in the last quarter of the mission, the cost score of the trainee strategy would not be too far off that of the TECA system. Further, in Figure 5.4 it can be seen that if this was done for all countermeasures, the resulting linear drop in EMCON score would also result in a score that is a lot closer to that of the TECA strategy.

*3) Superior strategy as a training tool*

The final application of this work also applies to EW operators and decision makers in training, where this system can be used to demonstrate superior approaches to certain scenarios that otherwise would not be immediately obvious. Further, continuous training will help trainees to develop an eye for these opportunities, so that they can be exploited.

For example, previous analysis of the 0.0652 fitness maximum-iteration stop-criterion TECA strategy has shown that one of the most effective strategies is to determine the point at which all

countermeasures can be ceased, and the platform left to escape based on its ability to outrun any missiles that may be fired. At first, this approach appears to be very risky as it leaves the platform defenceless against guided missiles, but it actually makes sense in a mission environment where threats do not communicate as it drastically reduces the platform's EM emissions. In turn, this reduces the likelihood of the platform illuminating itself to any unknown threats in the area. As an another example, the analysis of this strategy also highlighted times in which it was deemed more efficient to ignore a threat in the middle of the mission, and allow it to enter the guidance stage, knowing that the platform will be able to outrun the missile. For example, threat ID 2 was suppressed with just the right amount of the CP technique such that that threat could be ignored thereafter. This not only freed up the platform's limited countermeasure resources, but also prevented any potential illumination of the platform. Again, letting a threat enter the guidance stage is risky, but this approach is more likely to ensure the safety of the platform and pilot in this scenario. Perhaps more importantly, making the pilot aware that such outcomes are anticipated, but necessary, makes it less likely that the pilot will panic and make dangerous errors when a launch occurs.

Another example is the situations where this strategy has shown that it is more efficient to actually delay jamming of certain threats until others are in more favourable positions or stages. In particular, it was seen that the strategy only called for VGPO to break the guidance lock of threat ID 7 once the platform had left the range of threat ID 8, even though ID7's missile was on the verge of hitting the platform. This was because up until that point, threat ID 8 was right at the end of its tracking stage and on the verge of firing upon the platform, and any illumination would have allowed it to do so. So even though this approach was very risky, it was indeed necessary for the survival of the platform, and in the greater scheme of things resulted in a superior countermeasure strategy. As a final example, it was seen that when IR threat ID 2 entered the guidance stage 44 s into the mission, its jamming was delayed by 6 s in order to allow IR threat ID 1 to also enter the guidance stage. This in turn allowed for both of these threats to be jammed simultaneously due to the relatively low countermeasure resistance of IR threat type 2. Again this is risky, but clearly uses fewer flare cartridges, and hence is a more efficient use of the platform's resources.

### 5.4.2 Service applications

The second category of applications involves the direct use of the system in the field, either directly, or as an aide. Unlike the training applications that do not require human lives to be put in

the care of this system, these applications would require a more developed version of this system in order to be feasible.

*1) Decision support*

The primary service application of this work would be the hybrid application of decision support, whereby the system is used to rapidly develop a number of Pareto strategy options for the EW decision maker. The user can then use these solutions in conjunction with their experience and knowledge to develop a superior strategy to what they would be able to on their own, in a shorter amount of time. Further, they can use the system to check the effects of any countermeasure change, both in terms of overall fitness, and time-iteration level effects such as platform illumination.

*2) Cartridge load-out*

The next most-feasible application of this work would be to use it to generate superior countermeasure load-outs. In this case, the system would be used to generate an entire countermeasure strategy for a mission, which in turn would only be used as a context to the development of a passive-countermeasure strategy, and hence a cartridge load-out for the platform. This can be performed just prior to mission commencement, and the platform loaded accordingly.

*3) Strategy optimisation*

This application would rely almost entirely on the developed countermeasure strategies to directly control the countermeasures of the platform, based on the assumption that the developed strategies are near optimal and superior to those developed by the average human operator. Theoretically once this system has undergone sufficient development, this approach would be the most reliable and consistent way of protecting a platform and the human pilot. However, because it entrusts human life directly to the system, an extensive amount of development and validation would be required before this approach can be considered.

*4) Real-time operation*

The current system is only designed to be run prior to mission commencement. However, the very high computational efficiency attained indicates that this system could be modified to operate in real time, where the threat environment is updated continually according to what is actually

encountered, both in terms of threats and their radar stages. In this case, the system could be run prior to mission commencement to both develop a superior cartridge load-out, and to generate a seed countermeasure strategy that can be modified and updated over the course of mission as necessary. Further, the system could then use the independence of the fitness of each population member to take advantage of parallel processing, or even use variable size time intervals, where the next period of the mission is optimised in fine detail, whilst further ahead in the mission is optimised very coarsely. This could even be extended so that the system only optimises a certain amount of time ahead of the platform so as to keep the problem space reasonable. Theoretically, this approach would result in the most robust countermeasure strategies that in turn result in the greatest probability of survival of the platform, but it would certainly require significant development.

### 5.4.3 Sub-applications

This final category of applications explores the potential of integrating this system and its capabilities into another system, or adding additional capabilities to it based on what it can do currently.

*1) Coarse optimisation*

The first possibility would be to integrate this system into complex low-level physics- or parameter-based simulators such as SEWES or SADM. This can be done as simply as including this system as an additional functionality, or more importantly, it can be used a first high-level coarse countermeasure optimisation. Thereafter, the low-level simulator can use this as a seed solution for a more detailed and accurate optimisation in order to generate an even more accurate solution. This would generate a more accurate solution than the current system could, whilst generating a solution faster than such a low-level simulator ever could on its own. Importantly, such an application could be possible with even the current form of the system.

*2) Route and manoeuvres optimisation*

The inclusion of directional information such as platform RCS and antenna gain patterns allows this system to be used in the optimisation of mission routes, and countermeasure manoeuvres. This can either be achieved by integrating this system into an existing system, or adding the capabilities to it. Importantly, this would allow for the effect of countermeasures to be considered dynamically

in the development of optimal mission routes. Further, since both of these are inherently tied together, the optimisation of one can improve the other, and vice versa, theoretically resulting in even greater platform survival probability.

## 5.5 CHAPTER SUMMARY

In this chapter a discussion of the implications and applications of the results was presented. It was found that the generated strategies make sense in the context of the mission, and importantly are developed in a significantly shorter time than what could be achieved using current low-level simulators. However it was also found that a number of non-optimal countermeasure allocations were made in the analysed strategies, and that a number of simplifying assumptions were made in order to reduce computational complexity and keep the work within a reasonable scope. As such, a number of potential future improvements were presented to overcome these issues. These included the population of lookup tables using low-level simulators, and improved modelling through the replacement of simplifying assumptions.

Further, it was also found that the genetic algorithm performed well overall and was able to generate good solutions to very difficult problems in a short space of time, whilst also being able to use a number of different stop criteria and generate useful Pareto solutions. However, a number of issues were noted such as the bias of the algorithm towards specific techniques, and the effects of dead time on overall optimisation performance. As such, a number of potential future improvements were discussed such as including more seeding techniques to generate more genetic diversity and overcome the first issue, as well as island optimisation, dead-time switching, and allocation mutation in order to overcome the latter issue. Also the potential for the sub-optimisation of antenna angles was discussed.

Lastly, the potential applications of this work were considered. This included a number of specific ways in which this work can be used for training applications, as well as how it can be used in decision support, cartridge load-out optimisation, automated strategy optimisation, and real-time strategy optimisation. Further, it was discussed how this work could be incorporated into existing low-level simulators. Specifically, it could be used as a first-stage coarse optimisation of countermeasure strategy in these simulators, or be used for more detailed route and manoeuvre optimisation.

# CHAPTER 6 CONCLUSION

A countermeasure strategy and passive cartridge load-out optimisation system was proposed and implemented. Overall, it uses a process of threat evaluation and countermeasure allocation, whereby on a time-interval-by-time-interval basis threats are prioritised according to the danger they present to the platform, before countermeasures are allocated to them so as to minimise this danger. It is run prior to mission commencement based on the intelligence gathered about the threats in the mission area, and generates a full countermeasure strategy and associated cartridge load-out for the platform. This system overcomes the many limitations of previously proposed high-level countermeasure resource allocation systems by both developing more detailed countermeasure strategies, and considering a number of issues and interactions that are inherent in the EMS that those systems did not. Further, this was achieved in a computationally efficient manner that generates useful results in a reasonable amount of time, hence also overcoming the shortcomings that prevented the use of existing low-level physics- or parameter-based simulators for this application.

The first improvement over current countermeasure resource allocation systems is that the varying effects of specific countermeasure techniques on different threats and radar modes were considered. Importantly, the interactions between these different countermeasure techniques when used by multiple ECM channels, were also modelled. This further required the modelling of the progression of threats through various radar modes, thereby allowing for the future effects of current countermeasures to be considered. Further, the effects of frequency and bandwidth were also considered along with the effects of the platform's RCS, antenna directions and antenna gain patterns. Arguably, one of the most significant improvements is the ability of the system to model the effects of passive countermeasures, and thereby develop an optimised load-out of expendable cartridges for a platform that maximises its probability of survival. Consequently, the inherent interactions between active jammers and these passive countermeasures were also considered. Furthermore, the uncertainty of the threat environment that arises due to inaccuracies in

intelligence gathering was yet another factor that was accounted for in order to more accurately represent real-world systems. Lastly, optimisation was achieved using a genetic algorithm, in conjunction with a number of specialised operators and seeding techniques, that optimises using multiple objective functions that allow for the user-defined prioritisation of different strategy characteristics such as danger to platform, mission cost, and levels of EMCON.

The ability of the proposed system to successfully consider the highlighted issues was demonstrated using both a complex example scenario with a large concentration of threats, and a comparatively simple scenario. Successful solutions were found in a short space of time despite the exceedingly large problem space, thereby demonstrating the computational efficiency and hence usefulness of the proposed system. Further, both the underlying model and the generated strategies themselves were validated as reasonable through in-depth dissection and analysis in the context of the scenarios. Importantly, a number of complementary display methods were developed that succinctly, accurately, and effectively portray the developed strategies and their resultant interactions with the threats in the scenarios, which in turn can be used to convince human operators of their near optimality. Therefore, all the research objectives and questions of this work were indeed achieved, and answered in the affirmative. Not only that, but the ability of the system to generate a number of Pareto solutions that prioritise different strategy metrics was also demonstrated, along with its ability to generate solutions of different strengths at different rates using various stop criteria.

Thereafter, the various potential applications of this work were discussed. In its current form, the system can already be used for various training applications, where the optimised countermeasure strategies can be used as a benchmark against which a trainee can be evaluated in a quantitative manner, or as a tool to demonstrate and hence train humans to identify superior approaches to scenarios that otherwise would not be immediately obvious. Further, the display methods explored in this work can be used to create visual and interactive EW training software that builds an intuitive understanding of the EMS in all military personnel. Moreover, after further development this software could be used as a decision-support system for EW operators and decision makers in the field, both in terms of countermeasure strategy and cartridge load-out.

However, it is noted that this work exists in a void in the literature, where there is little to no published work in its field. As such, it is an exploratory work that attempts to build these numerous

capabilities in a very complex environment from the ground up. Therefore, a number of significant assumptions and simplifications were made in order to make this possible in a computationally efficient manner. Consequently, this system lays the groundwork for continued development of threat evaluation and countermeasure allocation systems that would slowly overcome its many limitations through the replacement of its simplified models and assumptions with improved ones. In this vein, a number of suggestions for future work and improvements were discussed. These included the better modelling of passive countermeasures, the inclusion of friendly platforms, and the inclusion of threat communication. Further, it was discussed how this system could be integrated into various other low-level physics- or parameter-based simulators to perform an initial coarse countermeasure optimisation step for these more slow and detailed systems, or even create additional functionalities such as route and manoeuvres optimisation. It was also discussed how the optimisation algorithm itself could potentially be improved, along with how this overall approach could potentially be modified to operate in real time so as to update the countermeasure strategy as the threat environment deviates from the model or programmed scenario. However, perhaps the most pressing future work would be to develop a method of using existing low-level simulators to automatically and accurately generate the lookup table and variable values for the specific platform and threats in the mission, as this would greatly improve the performance of this system.

# REFERENCES


[1] F. Johansson and G. Falkman, "Performance Evaluation of TEWA Systems for Improved Decision Support," in *Modeling Decisions for Artificial Intelligence*, V Torra, Y Narukawa, and M Inuiguchi, Eds. Berlin, Germany: Springer-Verlag, 2009, ch. 19, pp. 205-216.

[2] O. Karasakal, "Air Defense Missile-Target Allocation Models for a Naval Task Group," *Computers and Operations Research*, vol. 35, no. 6, pp. 1759-1770, June 2008.

[3] S. Noh and U. Jeong, "Intelligent Command and Control Agent in Electronic Warfare Settings," *International Journal of Intelligent Systems*, vol. 25, no. 6, pp. 514-528, June 2010.

[4] X. Zhai and Y. Zhuang, "IIGA Based Algorithm for Cooperative Jamming Resource Allocation," in *Asia Pacific Conference on Postgraduate Research in Microelectronics & Electronics*, Shanghai, China, 19-21 January 2009, pp. 368-371.

[5] M. Lv, D. Liu, N. Jiang, and Q. Chen, "Radar Jamming Resources Assignment Algorithm for EW Real-time Decision Support System of Multi-Platforms," in *International Conference on Intelligent Control and Information Processing*, Dalian, China, 13-15 August 2010, pp. 83-96.

[6] P. Wei, S. Ziming, and M. Lin, "Research on Force Assignment for Ground-to-Air Radar Jamming System based on Chaos Genetic Algorithms," in *The 27th Chinese Control and Decision Conference*, Qingdao, China, 23-25 May 2015, pp. 1215-1220.

[7] S. Kang et al., "Autonomously Deciding Countermeasures Against Threats in Electronic Warfare Settings," in *International Conference on Complex, Intilligent and Software Intensive Systems*, Fukuoka, Japan, 16-19 March 2009, pp. 177-184.

[8] J. du Plessis, "Evaluating Chaff Fire Pattern Algorithms in a Simulation Environment," in *Aardvark AOC Little Crow Conference*, Simonstown, South Africa, 22 May 2014.

[9] N.R. Osner and W.P. du Plessis, "Threat Evaluation and Jamming Allocation," in *14th Little Crow Conference*, Pretoria, South Africa, 26 September, 2016.

[10] N.R. Osner and W.P. du Plessis, "Electronic Warfare Training Applications of Decision-Support Systems," in *19th Symposium of Operational Applications in Areas of Defence*, Sao Jose dos Campos, Brazil, 26-28 September 2017, pp. 130-135.

[11] W.P. du Plessis and N.R. Osner, "Cognitive Electronic Warfare (EW) Systems as a Training Aid," in *Fifth International Conference on Electronic Warfare India (EWCI)*, Bangalore,

India, 13-16 February, 2018.

[12] N.R. Osner and W.P. du Plessis, "Threat Evaluation and Jamming Allocation," *IET Radar, Sonar and Navigation*, vol. 11, no. 3, pp. 459-465, April 2017.

[13] N.R. Osner and W.P. du Plessis, "Countermeasure Allocation and Expendable Load-Out Automation," *IEEE Transactions on Aerospace and Electronic Systems*, submitted on 13 June 2017, revised version submitted on 27 November 2017.

[14] D. Adamy, *Introduction to Electronic Warfare Modelling and Simulation*, 1st ed. Norwood, USA: Artech House, 2003.

[15] F. Neri, *Introduction to Electronic Defense Systems*, 2nd ed. Raleigh, USA: SciTech Publishing, 2006.

[16] B. Sweetman, "Eye of the Storm," *Journal of Electronic Defense*, vol. 25, no. 7, pp. 61-63, 2002.

[17] Z. Laszlo, "Gripens in Hungary Spark EW Revival," *The Journal of Electronic Defense*, pp. 12-15, May 2006.

[18] J.S. Arora, *Introduction to Optimum Design*, 3rd ed. Waltham, USA: Academic Press, 2012.

[19] G.B.L. Shanwei and G.Q.Z. Na, "Genetic Algorithm Approach to the Jammer's Layout for EW," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 32, no. 8, pp. 933-966, August 2006.

[20] T. Wang and Y. Li, "Research on the Optimization Assignment Algorithms for Jamming Resources in Land Based Air Defence Countermeasure System," *Journal of CAEIT*, vol. 3, no. 5, pp. 441-445, October 2008.

[21] T. Blickle and L. Thiele, "A Comparison of Selection Schemes Used in Evolutionary Algorithms," *Evolutionary Computation*, vol. 4, no. 4, pp. 361-394, December 1996.

[22] R.L. Haupt, "Thinned Arrays using Genetic Algorithms," *IEEE Transactions on Antennas & Propogation*, vol. 42, no. 7, pp. 993-999, July 1994.

[23] K.K. Yit, P. Rajendran, R. Rainis, and W. Ibrahim, "Investigation on Selection Schemes and Population Sizes for Genetic Algorithm in Unmanned Aerial Vehicle Path Planning," in *International Symposium on Technology Management and Emerging Technologies (ISTMET)*, vol. 1, Kedah, Malaysia, 25-27 August 2015, pp. 6-10.

[24] R. Smith and E. Smuda, "Adaptively Resizing Populations: Algorithm, Analysis, and First Results," *Complex Systems*, vol. 9, no. 1, pp. 36-45, 1994.

[25] D.E. Goldberg, K. Deb, and J.H. Clark, "Genetic Algorithms, Noise, and the Sizing of Populations," *Complex Systems*, vol. 6, no. 1, pp. 333-362, 1991.

[26] V.K. Koumousis and C.K. Dimou, "The Effect of Oscillating Population Size on the Performance of Genetic Algorithms," in *4th GRACM Congress on Computational Mechanics*,

vol. 1, June 2002, pp. 8-15.

[27] V.K. Koumousis and C.P. Katsaras, "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 19-28, February 2006.

[28] J. Nocedal and S. Wright, *Numerical Optimisation*, 1st ed. New York, USA: SpringerVerlag, 1999.

[29] B. Manz, "Cognition: EW Gets Brainy," *The Journal of Electronic Defence*, pp. 32-39, October 2012.

[30] N. Willers, R. Willers, and A. de Waal, "Aircraft Vulnerability Analysis by Modelling and Simulation," in *Aardvark AOC Little Crow Conference*, Pretoria, South Africa, 17 November 2014, presented by K. Gopaul.

[31] G.W. Collins, *The Foundations of Celestial Mechanics*. United States of America: Pachart Publishing House, 1989.

[32] S. Orfanidis, *Electromagnetic Waves and Antennas*. Rutgers, United States of America: Rutgers University, 2016.

[33] C.W. Trueman, S.J. Kubina, S.R. Mishra, and C.L. Larose, "RCS of Small Aircraft at HF Frequencies," in *Symposium on Antenna Technology and Applied Electromagnetics*, Ottawa, Canada, 3-5 August 1994, pp. 151-157.

[34] Eurofighter Jagdflugzeug GmbH. (2013) Eurofighter Typhoon Technical Guide.

[35] B.R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*. United States of America: Chapman & Hall/CRC, 2000.

[36] A. Maslovskiy, M. Legenkiy, and M. Antyufeyeva, "BSP Step for Complex Target RCS Measuring or Calculation," in *9th International Kharkiv Symposium on Physics and Engineering of Microwaves, Millimeter, and Submillimeter Waves*, Kharkiv, Ukraine, 20-24 June 2016.

[37] M.S. Pavlovic, M.S. Tasic, B.L. Mrdakovic, and B.M. Kolundzija, "WIPL-D: Monostatic RCS Analysis of Fighter Aircrafts," in *10th European Conference on Antennas and Propogation*, Davos, Switzerland, 10-15 April 2016.

[38] A. Golden, *Radar Electronic Warfare*, 1st ed. Washington DC, Maryland: American Institute of Aeronautics and Astronautics, 1987.

[39] R. Reddy, private communication, March 2017.

# GLOSSARY

**Combinatorial Optimisation** The optimisation of problems where the set of feasible solutions is discrete in nature, thus preventing the use of traditional derivative-based optimisation processes.

**Danger Value** A numerical value that is representative of the level of danger presented by a threat to the platform in the time interval being examined. It is based on the platform RCS perceived by that threat, along with a number of threat characteristics: radar stage, accuracy, projectile time to platform, time to next radar stage, and the probability of encountering that threat in that time interval.

**Distributed Threat Area** The physical area in which a threat is expected to occur. It is characterised by a central point location, and a radius around that point in which the threat is expected to occur.

**Hit** In this work, a hit is assumed to occur when a projectile either hits the platform directly, or when a warhead detonates sufficiently close to the platform that it is fatally wounded.

**Illumination** A term used to describe when the EW system of the platform acts as a beacon for the attacking threats, allowing it to be more easily detected or tracked.

**Jamming Allocation** The process whereby specific countermeasure techniques are allocated to each of the ECM channels of the platform for each time interval. This term originates as a mirrored term from published TEWA systems.

| | |
|---|---|
| **Jamming Factor** | A multiplicative factor used to account for the effect of countermeasures on the danger value of a threat. A jamming factor of less than one indicates effective jamming, and consequently reduces the danger presented by a threat, and vice versa. |
| **Load-Out** | The specific combination of flare and chaff cartridges loaded into the platform prior to the commencement of a mission. |
| **Platform** | The friendly platform whose countermeasures are being optimised in this work. |
| **Player** | Any system, friendly or adversary, on the battlefield. |
| **Post-Jamming Danger Value** | The danger presented by a threat in a particular time interval, after the effects of the applied countermeasures are taken into account. It is calculated as a product of a threat's danger value and its jamming factor. |
| **Radar Range** | The ground range of the radar system of a threat. This is the average range of the radar system under ordinary conditions. |
| **Radar Stage** | The radar mode of a threat in the time interval being examined, where in this work threats are assumed to progress through a series of radar stages from an initial search stage, through acquisition and tracking stages, before culminating in a guidance stage. |
| **Threat** | Any adversary platform or system. In this work threat weapon and radar systems are assumed to be co-located. |
| **Threat ID** | The identity number of a threat used to differentiate between the specific adversary players in a scenario. A set of ascending numbers is separately allocated to both the IR threats and the RF threats. |
| **Threat Type** | The specific weapon and radar system combination of a threat, where an ascending set of numbers is allocated to each system type in a mission to differentiate between them. A separate set of numbers is allocated to the IR and the RF threats, where the numbers are allocated in ascending frequency- |

band-usage order.

**Threat Evaluation** The process of allocating a danger value to each threat in a time interval. This serves to allow for the prioritisation of threats according to the level of danger they present to the platform in the time interval being examined. This term originates from TEWA systems.

**Threat Number** The threat type number of a threat.

**Weapon Range** The ground range of the weapon system of a threat. This is the average range if the weapon system under ordinary conditions.