



# A tri-objective, dynamic weapon assignment model for surface-based air defence

DP Lötter\*      JH van Vuuren†

*Received: 7 October 2014; Revised: 11 May 2015; Accepted: 13 May 2015*

## Abstract

In a military surface-based air defence environment, a fire control officer typically employs a computerised weapon assignment decision support subsystem to aid him in the assignment of available surface-based weapon systems to engage aerial threats in an attempt to protect defended surface assets — a problem known in the military operations research literature as the *weapon assignment problem*. In this paper, a tri-objective, dynamic weapon assignment model is proposed by modelling the weapon assignment problem as a multi-objective variation of the celebrated vehicle routing problem with time windows. A multi-objective, evolutionary metaheuristic for solving the vehicle routing problem with time windows is used to solve the model. The workability of this modelling approach is illustrated by solving the model in the context of a simulated, surface-based air defence scenario.

**Key words:** Decision support, multi-objective optimisation, surface-based air defence, vehicle routing with time windows, weapon assignment.

## 1 Introduction

In the context of a military *surface-based air defence* (SBAD) environment, *weapon assignment* (WA) entails the assignment of available surface-based<sup>1</sup> *weapon systems* (WSs) by a human operator, known as a *fire control officer*, to engage aerial threats based on some pre-specified set of suitability criteria, in an attempt to neutralise these threats — a problem which is known in the military operations research literature as the *Weapon Assignment Problem* (WAP) [26].

---

\*Corresponding author: Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: [danielotter@sun.ac.za](mailto:danielotter@sun.ac.za)

†(Fellow of the Operations Research Society of South Africa), Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa

<sup>1</sup>In the context of SBAD, the term *surface* refers to the surface of a portion of land, lake or ocean.

A fire control officer operates under severely stressful conditions and typically has to solve the WAP within a very short time frame. Since poor assignment decisions may have catastrophic effects in terms of loss of life and other own-force assets, it is desirable to provide a fire control officer with some form of computerised decision support to aid him in assignment decisions. In 2014, Lötter and Van Vuuren [19] put forward a generic framework for the design of such a decision support system. This framework provides for a so-called *Weapon Assignment Model* (WAM) component in which four different classes of mathematical assignment models reside, ranging in different levels of complexity. It was suggested that the WA system should be designed in such a way that the fire control officer is able to configure a single WAM from one of these model classes to furnish him with WS-threat assignment pair proposals. The four model classes include single-objective static models, single-objective dynamic models, multi-objective static models, and multi-objective dynamic models. The term *static* refers here to models in which the numbers and locations of WSs and threats are known with certainty at some time instant  $\tau$  and a single assignment of WSs to threats is sought at time  $\tau$  such that all the WSs are committed [9]. Hence, a static WAM would typically be solved for each time stage within a suitable discretization of the time continuum. The term *dynamic* refers to the class of WA models in which suitable future time instants are sought at which to assign a subset of the available WSs to the threats observed [29]. Hence, a dynamic WAM would typically be able to solve the WAP for the entire time continuum at once. A graphical representation of these model classes is provided in Figure 1. The figure has been populated with references to model instances within these classes.

	Static	Dynamic
Single-objective	<ul style="list-style-type: none"> <li>● Manne (1957)</li> <li>● Bradford (1961)</li> <li>● Day (1966)</li> <li>● Ahuja <i>et al.</i> (2003)</li> <li>● Potgieter (2008a)</li> </ul>	<ul style="list-style-type: none"> <li>● Hosein &amp; Athans (1989)</li> <li>● Murphy (2000)</li> <li>● Du Toit (2006)</li> <li>● Potgieter (2008b)</li> <li>● Du Toit (2009)</li> <li>● Van der Merwe &amp; Van Vuuren (2014)</li> </ul>
Multi-objective	<ul style="list-style-type: none"> <li>● Lötter <i>et al.</i> (2013)</li> </ul>	No models

**Figure 1:** A grey-scaled representation of the four classes of WAMs, illustrating their levels of complexity, with darker quadrants portraying more complex WA models. The evolution of WAMs in each class is illustrated by listing their contributors in chronological order.

This paper is structured as follows. A detailed description of the model formulation may be found in §3. This is followed, in §4, by a discussion on how the model may be solved (approximately). A numerical example is provided in §5, illustrating the working of the

model in the context of a hypothetical, but realistic, case study. A discussion of the model results and conclusions in the context of this worked example follows, and the paper closes, in §6, with ideas for possible future work.

## 2 Literature review

The origin of WA modelling dates back as far as 1957 when the notion was first introduced by Flood [10] and formulated as a mathematical model by Manne [21] later that year. This formulation was based on the classical assignment problem of Votaw and Orden [30] of 1952, and it contains a single, linear objective function in which the aim is to determine WS-threat assignment pairs that minimise the total expected survival<sup>2</sup> values of the threats after the engagements. The model is applicable to a single instant in time and is hence static in nature.

This early model by Manne served as a catalyst for other WAP formulation variants which followed. Many authors have contributed single-objective WAP formulations, amongst them the dynamic programming formulation by Bradford [3] in 1961, the nonlinear programming formulation by Day [7] in 1966, the adapted formulation by Ahuja *et al.* [2] in 2003 which provides for WSs of different types or classes to be included in the model, and the adjusted  $k$ -WAP formulation of Potgieter [24] in 2008 which constrained the formulation of Manne by only allowing for a maximum of  $k$  WSs to be assigned to a threat. These formulations are all single-objective static WA models and hence reside in the first quadrant of Figure 1. They are typically easily solved approximately by employing standard metaheuristic solution methodologies from the operations research literature, but their disadvantage is that they have to be solved for each stage in the time continuum under consideration.

The next model class is the single-objective dynamic WAM class (*i.e.* WAMs in the second quadrant in Figure 1). In this class the WAP is solved by pursuing optimisation of a single objective over the entire time continuum during which the air picture unfolds. Apart from proposing WS-threat assignment pairs, these models are therefore able to propose subsets of future time stages, known as *time windows*<sup>3</sup>, during which these WS-threat pair assignments should occur, and hence incorporate a scheduling element over and above the assignment element of their earlier static counterparts.

One of the best known single-objective dynamic WAMs is the *shoot-look-shoot model* of Hosein and Athans [12], dating back to 1989. In this model the time continuum is partitioned into a number of shorter, discrete time stages and the aim is to minimise the total survival probabilities of the threats, weighted by their estimated threat priority values, at the end of the final stage. The model functions under the assumption that the outcome of the WS-threat pair assignments for each time stage (*i.e.* hits and misses) are observed (perfectly), before the WS-threat pair assignments for the next time stage are evaluated.

---

<sup>2</sup>The survival value of a threat is computed by subtracting the efficiency value achieved by a WS when assigned to the threat (also known as the *single shot hit probability* value) from unity.

<sup>3</sup>The time window associated with a specific WS-threat pair refers to the subset of time stages during which the WS should be committed to that threat in order for the engagement to achieve an acceptable level of effectiveness in terms of the corresponding single shot hit probability [12].

Although this formulation is generally considered to be a dynamic WAM, Huaiping *et al.* [13] rightly criticised the formulation by pointing out that it merely involves solving a static WAM class problem repeatedly. Hence, it is placed on the boundary between the static and dynamic single-objective model classes in Figure 1.

Another single-objective dynamic WAM is the *stochastic demand dynamic* WAM, due to Murphy [23] in 2000. In this model, the time continuum is also partitioned into shorter time stages, but with the difference that the number of threats is unknown in advance. During each time stage, only a subset of the threats and their positions are known. A WS may either be assigned to one of the known threats or the assignment may be postponed in favour of a threat that is as yet unknown, but which may be detected in the future. Prolonging the assignment of a WS is penalised in the objective function by cost coefficients which are monotonically increasing functions of time. The aim in this model is therefore to find a balance between proposing WS assignments to currently known threats immediately and reserving WS assignments for as yet undetermined, future threats.

In 2006, Du Toit [8] formulated the so-called *WA scheduling problem*, which is based on the stochastic demand dynamic WAM of Murphey [23]. The model is formulated under the assumption that the individual single shot hit probability values of the WSs are independent of each other and, hence, the objective function may be simplified by omitting the cost coefficients in the original formulation. In 2008, Potgieter [24] adjusted the WA scheduling problem formulation to include the use of non-renewable WSs and by allowing for a maximum of  $k$  WSs to be assigned to a threat over the whole temporal period under consideration. He also contributed another single-objective dynamic model formulation, known as a *multi-period WAP*, which is an extension of the  $k$ -WAP, and is based on the same fundamental approach as that of the shoot-look-shoot formulation of Hosein and Athans [12].

In 2009, Du Toit [9] formulated another WAM, which he called the *expected threat priority accumulation dynamic weapon target assignment* model and which is also based on the shoot-look-shoot model of Hosein and Athans [12]. The model formulation differs from the original one in the sense that earlier engagements by WSs are encouraged by assigning higher priorities to such engagements during the evaluation of the expected survival probabilities of the threats during each time stage, rather than only during the final time stage, as was the case with the original shoot-look-shoot model.

In 2014, Van der Merwe and Van Vuuren [29] put forward a single-objective, dynamic WAM which is able to schedule time windows during which WS-threat assignment pairs should occur. The formulation is based on the classical vehicle routing problem, where the WSs are modelled as vehicles having to deliver commodities (ammunition) to customers (threats) within pre-specified time windows (fire windows) with the aim of minimising the cost associated with the delivery (the survival probabilities of the threats).

The third class of WAMs contains multi-objective static WA models (*i.e.* the third quadrant in Figure 1). We are aware of only a single model in this class. In 2013, Lötter *et al.* [18] modelled the WAP in a bi-objective, static context by aiming to minimise the accumulated survival probability of the threats and to minimise the accumulated cost of the WS-threat assignment pairs simultaneously. In 2014, they extended this formulation by adding a third objective to the formulation which aims to maximise the number of

times that the least re-engageable WSs may be used during future assignments *after* the current decision window under consideration (in a bid to ensure that all WSs are reusable after the assignment) [19].

The final class of WAMs is the class of the multi-objective dynamic WAMs (*i.e.* the fourth quadrant in Figure 1). We are not aware of any models in this class and the aim in this paper is therefore to propose such a model.

### 3 A tri-objective, dynamic WA model

The multi-objective dynamic WAM formulation proposed here has its roots in the celebrated *vehicle routing problem with time windows* and is an extension of the single-objective dynamic WAM formulation of Van der Merwe and Van Vuuren [29].

The vehicle routing problem with time windows is a well-known combinatorial optimisation problem in which the aim is to find an optimal assignment of members of a fleet of heterogeneous vehicles, as well as an associated set of routes for these vehicles, in order to deliver commodities to customers with known demands [28]. The problem includes a central depot from which the vehicles are required to depart and return to on completion of their route(s). The depot is associated with a service time window, known as the scheduling horizon, during which customers are allowed to be served. Each customer is also associated with a service time window (*i.e.* a subset of consecutive time stages within the scheduling horizon) during which service by vehicles should take place. In addition, each customer is allocated a service time duration which indicates the number of time stages required to service the customer. A vehicle is allowed to service more than one customer along a particular route, but the time that service for a particular customer ends plus the time that the vehicle requires to travel to the next customer should not exceed the time at which service of the next customer starts. Furthermore, the accumulated demands of customers along the route assigned to a single vehicle should not exceed the capacity of the vehicle. A typical objective in the vehicle routing problem with time windows is to minimise the overall delivery cost incurred, which is usually a combination of the travel cost incurred by the vehicles and the cost of servicing the customers.

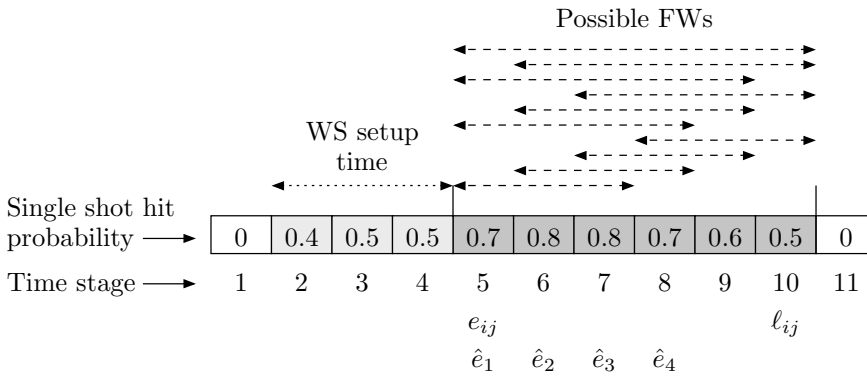
In the context of an SBAD environment, the multi-objective dynamic WA model may be formulated as a vehicle routing problem with time windows by modelling the WSs as the fleet of vehicles which have to deliver their ammunition (modelled as commodities) to the customers (modelled as the threats) — the WSs may be thought of as “vehicles” delivering ammunition to the threats [29]. A virtual depot is also included in the problem which represents the idle state during which no threats are engaged by any WSs. The system is required to start from this state and to return to this state again after WS-threat pair engagements are carried out.

Each threat is associated with a *fire window* (FW) (analogous to the service time window of a customer in the vehicle routing problem with time windows) during which WSs may engage the threat. Each FW involves a *first-time-to-fire* (FTTF) which represents the earliest time stage during which a WS may engage a threat, and *last-time-to-fire* (LTF) which represents the latest time stage during which a WS may engage a threat. WSs are

required to achieve a positive single shot hit probability during each time stage in the FW associated with a threat and if a WS can be assigned to more than one threat during the entire scheduling horizon, the FWs of these threats are not allowed to overlap. The set of fire windows associated with each WS-threat pair are therefore induced by a combination of terrain surface masking, meteorological conditions, position of the WS and predicted flight path of the threat. The number of time stages that have elapsed from the FTTF to the LTTF is called the length of the FW and a WS is allowed to engage a threat during any time stage within the FW, including the FTTF and LTTF.

Each WS is associated with a *weapon set-up time* (analogous to the time it takes for a vehicle to travel between customers in the vehicle routing problem), which refers to the accumulated period of time required for the fire control officer to respond to orders, to finalise an engagement order and to track a threat before the engagement commences. The amount of time remaining before the FTTF associated with a particular WS should be greater than or equal to the weapon setup time.

Consider the following example illustrating the notion of a FW for a WS-threat pair. Suppose that the period under consideration comprises eleven time stages, that the weapon setup time is three time stages and that the minimum length of a FW is three time stages. One possible FW has a FTTF at time stage<sup>4</sup> 5 and a LTTF at time stage 10. The length of the FW is six time stages, as illustrated graphically in Figure 2. This example situation also admits nine other FWs with (FTTF, LTTF)-pair values given by (5,7), (6,8), (7,9), (8,10), (5,8), (6,9), (7,10), (5,9) and (6,10), respectively. Note that although the FWs have various lengths, the length is always greater than or equal to the specified minimum length that a FW may be.



**Figure 2:** A number of possible FWs for a specific WS-threat pair.

From the FW example in Figure 2 it may be seen that there are a number of different combinations of FWs for every WS-threat pair. For each WS-threat pair  $(i, j)$  there exists a time interval, starting at the beginning of  $e_{ij}$  and ending at the end of  $l_{ij}$ , during which engagement of WS  $i$  to threat  $j$  may commence. Within this time interval, there exists a number of FWs having various FTTFs and LTTFs (as illustrated in the example in Figure 2) for the pair. Let  $L_{ij} = l_{ij} - e_{ij} + 1$  be the maximum length of any FW for the

<sup>4</sup>This is the smallest possible value of the FTTF, due to the weapon setup time length which is three time stages.

pair and let  $L_i^{min}$  denote the minimum length of a FW for WS  $i$ . Then, for each WS-threat pair  $(i, j)$ , there are  $F_{ij} = L_{ij} - L_i^{min} + 1$  different feasible FTTF stages  $\hat{e}_1, \hat{e}_2, \dots, \hat{e}_{F_{ij}}$  having values  $e_{ij}, e_{ij} + 1, \dots, \ell_{ij} - L_i^{min} + 1$ , respectively. For each different FTTF  $z$  there are  $\ell_{ij} + 1 - \hat{e}_z - (L_i^{min} - 1)$  different LTTF stages. It may be concluded that each WS-threat pair has

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

distinct possible FWs [29].

In 2013, Lötter *et al.* [18] conducted an electronic survey involving a group of military experts in order to identify important assignment objectives that may be used to populate the various WAM classes in Figure 1. Three of these objectives were chosen for use in a tri-objective dynamic WAM prototype in this paper. The first objective aims to minimise the accumulated survival probabilities of the threats, weighted by the respective priorities of eliminating them. The second objective aims to minimise the accumulated monetary cost of the WS-threat pair assignments (in terms of ammunition expended) and the third objective aims to maximise the number of times that the least re-engagable WS may re-engage after all assignments over the entire scheduling window have taken place.

The model functions under the assumption that only one unit of ammunition is delivered to a threat when engaged by a specific WS during any of the pair's FWs. This unit represents a single burst of ammunition which is typically one missile if a missile is fired, or a pre-determined number of rounds if a cannon is fired in burst mode. Furthermore, it is assumed that a WS may only be assigned at most once to a specific threat during the scheduling horizon and that a maximum number of  $\kappa$  WSs may be assigned to any single threat during the scheduling horizon.

Suppose that  $n$  threats are detected in the defended airspace and that there are  $m$  pre-deployed available WSs on the surface available for assignment to engage these threats. Let the depot be indexed by two artificial threats, threat 0 and threat  $n + 1$ , each having an engagement demand of zero. Let  $d_i$  denote the number of time stages it takes for WS  $i$  to "travel" between consecutive threats to which it is assigned, and denote the earliest possible FTTF value by  $e_{ijk}$  and the latest possible LTTF value by  $\ell_{ijk}$ , when WS  $i$  engages threat  $j$  during the pair's  $k^{th}$  FW. Also let  $f_{ij}$  be the number of distinct FWs for WS-threat pair  $(i, j)$ , as computed in (1), and let  $s_{ijk}$  denote the engagement time duration<sup>5</sup> associated with threat  $j$  when it is engaged by WS  $i$  during the pair's  $k^{th}$  FW, where  $s_{ijk} \leq \ell_{ijk} - e_{ijk} + 1$ .

Furthermore, let  $V_{j\tau}$  denote the priority value of eliminating threat  $j$  during time stage  $\tau$ , and let  $q_{ij\tau}$  denote the survival probability of threat  $j$  if it is engaged by WS  $i$  during

---

<sup>5</sup>The engagement time duration refers to the number of time stages required by a WS to guide its ammunition in the direction of the threat from the time stage during which the ammunition is launched (the time stage during which the trigger is pulled) until the time stage during which the ammunition hits the threat. This is only applicable to WSs which require operator guidance, such as a cannon which is able to fire multiple bursts with the pull of the trigger, but requires manual WS operator guidance to guide the ammunition in the direction of the threat. In some instances, WSs have on-board guidance systems, such as certain missiles which require no guidance from the WS operator, in which case the value  $s_{ijk}$  is assumed to be zero. The value  $s_{ijk}$  is included in the model formulation for the sake of flexibility.

time stage  $\tau$ . Then  $q_{ij\tau} = 1 - p_{ij\tau}$ , where  $p_{ij\tau}$  denotes the single shot hit probability value of WS  $i$  when engaging threat  $j$  during time stage  $\tau$ . A fixed mean approach<sup>6</sup> is adopted in order to associate a single survival value with the  $k^{\text{th}}$  FW of WS-threat pair  $(i, j)$ . Let  $M_{ijk}$  be the set of consecutive time stages to be used in calculating the fixed mean of the  $k^{\text{th}}$  FW for WS-threat pair  $(i, j)$ . Then the fixed mean value is

$$\mu_{ijk} = \frac{1}{|M_{ijk}|} \sum_{\tau \in M_{ijk}} q_{ij\tau}. \quad (2)$$

Finally, denote the cost associated with assigning WS  $i$  to a threat by  $C_i$  and let the number of bursts of ammunition available to WS  $i$  at the start of the scheduling horizon be  $A_i$ .

We incorporate a binary decision variable  $x_{ijk}$  in our model which takes the value 1 if WS  $i$  is assigned to threat  $j$  during the pair's  $k^{\text{th}}$  FW. A binary auxiliary variable  $y_{ihj}$  is also incorporated into the model which acts as a vehicle flow variable, taking the value 1 if threat  $h$  directly precedes threat  $j$  in the sequence of engagements by WS  $i$ , or the value 0 otherwise. The objectives in the model are to

$$\text{minimise} \quad \sum_{j=1}^n V_{j\tau} \prod_{i=1}^m \prod_{k=1}^{f_{ij}} (\mu_{ijk})^{x_{ijk}} \quad (3)$$

$$\text{minimise} \quad \sum_{i=1}^m C_i \sum_{j=1}^n \sum_{k=1}^{f_{ij}} x_{ijk} \quad (4)$$

$$\text{maximise} \quad \min_i \left\{ A_i - \sum_{j=1}^n \sum_{k=1}^{f_{ij}} x_{ijk} \right\} \quad (5)$$

$$\text{subject to} \quad \sum_{i=1}^m \sum_{h=0}^{n+1} y_{ihj} \leq \kappa, \quad j = 1, \dots, n, \quad (6)$$

$$\sum_{h=1}^{n+1} y_{i0h} = 1, \quad i = 1, \dots, m, \quad (7)$$

$$\sum_{h=1}^{n+1} y_{i,h,n+1} = 1, \quad i = 1, \dots, m, \quad (8)$$

$$\sum_{h=0}^{n+1} y_{ihj} - \sum_{h=0}^{n+1} y_{ijh} = 0, \quad \begin{cases} i = 1, \dots, m, \\ j = 0, \dots, n+1, \end{cases} \quad (9)$$

<sup>6</sup>The fixed mean approach entails considering a subset (of fixed, prespecified cardinality) of contiguous time stages within the FW and calculating the mean of the survival probabilities associated with the time stages in this subset. The time stages chosen for inclusion in the subset should be consecutive and should also include the time stage with the minimum survival probability of the entire set [29]. The procedure is carried out by repeating this procedure while moving through the FW for all such subsets, finally selecting the smallest mean obtained over all the subsets, and associating this mean as “survival probability” with the FW.



$$\sum_{k=1}^{f_{ij}} x_{ijk} = \sum_{\substack{h=1 \\ h \neq j}}^{n+1} y_{ihj}, \quad \begin{cases} i = 1, \dots, m, \\ j = 1, \dots, n, \end{cases} \quad (10)$$

$$\sum_{k=1}^{f_{ih}} (e_{ihk} + s_{ihk})x_{ihk} - \sum_{k=1}^{f_{ij}} e_{ijk}x_{ijk} + d_i < (1 - y_{ihj})L, \quad \begin{cases} i = 1, \dots, m, \\ j = 1, \dots, n, \\ h = 1, \dots, n, \end{cases} \quad (11)$$

$$\sum_{j=1}^n \sum_{k=1}^{f_{ij}} x_{ijk} \leq A_i, \quad i = 1, \dots, m, \quad (12)$$

$$y_{ihj} \in \{0, 1\}, \quad \begin{cases} i = 1, \dots, m, \\ j = 0, \dots, n + 1, \\ h = 0, \dots, n + 1, \end{cases} \quad (13)$$

$$x_{ijk} \in \{0, 1\}, \quad \begin{cases} i = 1, \dots, m, \\ j = 0, \dots, n + 1, \\ k = 1, \dots, f_{ij}. \end{cases} \quad (14)$$

Constraint set (6) ensures that no more than  $\kappa$  WSs are assigned to any threat during the scheduling horizon. Furthermore, constraint set (7) ensures that the system leaves the idle state exactly once if a WS is assigned to engage any threats, while constraint set (8) ensures that the system returns to the idle state exactly once after all engagements have taken place. Constraint set (9) ensures, if a WS is assigned to engage a threat, that it leaves the threat again so that the WS “moves” on to the next threat to which it is assigned. If WS  $i$  is scheduled to engage threat  $h$  directly before threat  $j$ , constraint set (10) ensures that WS  $i$  engages threat  $h$  during exactly one stage. In the same instance, constraint set (11) ensures that the time stage during which the engagement of threat  $h$  by WS  $i$  starts plus the time it takes to engage threat  $h$  plus the time it takes for WS  $i$  to “travel” to threat  $j$ , does not exceed the time stage during which the engagement of threat  $j$  starts, where  $L$  is a large number. Constraint set (11) also ensures, if WS  $i$  is assigned to engage threat  $j$ , that the assignment is scheduled during a feasible FW for the WS-threat pair. Constraint set (12) ensures that WS  $i$  is assigned at most  $A_i$  times for engagement, while constraint sets (13)–(14) enforce the binary nature of the decision and auxiliary variables.

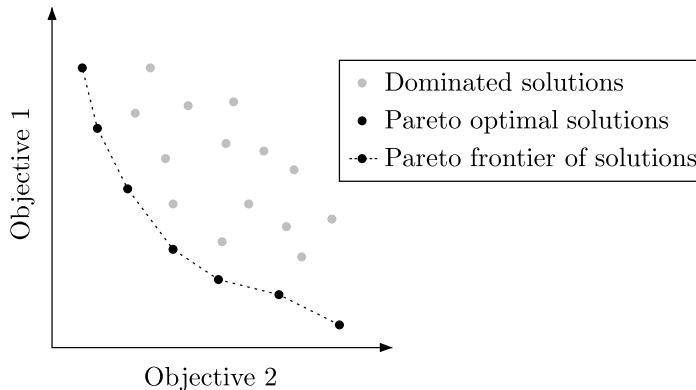
## 4 Model implementation

In multi-objective decision problems, a number of conflicting objectives have to be optimised simultaneously. In contrast to a single-objective optimisation problem where the aim is to approximate a high-quality single solution to the problem, a single solution which optimises the adopted set of objectives in a multi-objective problem simultaneously typically does not exist, because of the conflicting nature of the objectives. Instead, the aim is to find a set of high-quality solutions which achieve acceptable compromises between the objectives [16].

The concept of multi-objective optimisation naturally gives rise to the notion of *domination* which entails comparing candidate solutions in the solution space with one another with

the aim of finding a set of solutions which dominates the remaining solutions. This set of solutions is known as a *nondominated* set of solutions. The notion of domination may be defined as follows for a multi-objective problem in which  $\omega$  objectives have to be minimised simultaneously. Consider two solutions  $\mathbf{a}$  and  $\mathbf{b}$  which achieve the values  $a_i$  and  $b_i$  for the  $i$ -th objective function, respectively, where  $i \in \{1, \dots, \omega\}$ . Solution  $\mathbf{a}$  dominates solution  $\mathbf{b}$  if each element in  $[a_1, \dots, a_\omega]$  is less than or equal to the corresponding element in  $[b_1, \dots, b_\omega]$  and at least one element in  $[a_1, \dots, a_\omega]$  is strictly less than the corresponding element in  $[b_1, \dots, b_\omega]$ ; that is  $a_i \leq b_i$  for all  $i \in \{1, \dots, \omega\}$  and  $a_j < b_j$  for some  $j \in \{1, \dots, \omega\}$  [6].

The nondominated solutions are superior to the remaining solutions in the solution space, but they are not superior (nor are they inferior) to any other solutions in the nondominated set — no solution in the nondominated set is considered to be better or worse than any other solution in the set. The nondominated solutions are together called the *Pareto optimal solutions* and form a so-called *Pareto optimal frontier* when plotted in objective space. An example of a Pareto optimal set of solutions for a bi-objective problem in which both objectives have to be minimised is illustrated graphically in Figure 3, in which the objective function trade-offs between solutions are easily discernible.



**Figure 3:** The notion of Pareto optimality in objective space for a problem in which two objectives are to be minimised.

If the complexity or dimensions of a multi-objective optimisation problem is such that it cannot be solved exactly within an acceptable time frame (as is the case in the model<sup>7</sup> (3)–(14)), exact solution approaches are often abandoned in favour of metaheuristic solution approaches. The aim in multi-objective metaheuristic optimisation is to approximate a good spread of solutions along the Pareto frontier. A number of multi-objective metaheuristic solution methodologies in the operations research literature may be employed for this purpose, such as the *Nondominated Sorting Genetic Algorithm II* (NSGA II) of Agarwal *et al.* [1] and the partial swarm optimisation technique of Moore and Chapman [22], to name but two. In this paper, a multi-objective evolutionary algorithm based on the nondominated sorting notion used in the NSGA II is adopted to solve the tri-objective dynamic WA model (3)–(14).

<sup>7</sup>The air picture in the WAP is updated at a frequency dictated by the radar refresh rate, which typically ranges between two and four seconds, hence resulting in a very short timeframe during which instances of the model (3)–(14) must be solved.

Evolutionary algorithms are based on Darwin’s theory of evolution by means of natural selection [4]. In these algorithms, a population of candidate solutions is allowed to evolve over time in a carefully controlled environment. Parent solutions in the population are selected, based on a pre-specified set of criteria (known as the fitness measure<sup>8</sup>), to produce offspring solutions which populate the next generation of solutions. The offspring solutions are produced by applying so-called selection and recombination operators to the selected parent solutions. This process is iterated over a number of solution generations until significantly fitter solutions can no longer be found or until a pre-specified number of generations is reached.

The evolutionary algorithm adopted here is based on the algorithm designed by Bullinaria and Garcia-Najera [4] for solving the vehicle routing problem with time windows. A solution to the model (3)–(14) is encoded by a three-dimensional matrix with threats, WSs and time as its dimensions — each cell in this matrix corresponds to a decision variable of the tri-objective WA model (3)–(14). The solutions in the initial parent population are generated stochastically by first selecting (according to a uniform distribution) a WS-threat pair  $(i, j)$  that have not yet been assigned to each other and then assigning the pair during any one of their feasible FWs (*i.e.* FWs which do not overlap with other assignments already made) — again according to a uniform distribution. If no feasible FW exists for the pair, another WS-threat pair is selected randomly and the procedure is repeated until all the WSs have been considered for assignment. This random solution generation procedure is implemented in such a way that none of the constraints (6)–(14) are violated.

Once the initial parent population has been populated, the objective function values of each solution are calculated and a fitness measure is evaluated for each solution. The fitness measure we employ is based on the domination status of a solution and the fast nondominated sorting algorithm of Agarwal *et al.* [1] is used to sort the population of solutions into various fronts in objective space, based on the number of solutions that they dominate. Solutions which are not dominated by any other solutions are contained in the first front (these are the approximately Pareto optimal solutions), while solutions dominated once are contained in the second front, and so on. The front in which a solution resides is taken as the solution’s fitness value (*e.g.* a solution in the first front has a fitness value of 1).

A standard tournament selection procedure is implemented in the algorithm in order to select parent solutions [11]. The aim in this selection procedure is to allow for fitter parent solutions to have a greater probability of being selected for reproduction than lesser fit solutions. Lesser fit parent solutions should nevertheless have a small chance of being selected so as to avoid premature convergence [4].

The selection procedure implemented in the algorithm employs two selection criteria. The first parent is selected based on its fitness value (*i.e.* the number of the front in objective space in which the solution resides) while the second parent is chosen based on a diversity preservation measure known as a *similarity measure*. This measure evaluates how similar

---

<sup>8</sup>The fitness of a solution is an indication of the quality of the solution with respect to the other solutions. A typical fitness measure employed in genetic algorithms for maximisation problems is the objective function value of a solution.

a solution is to the rest of the solutions in the population. Solutions with low similarity values are preferred over solutions with high similarity values in a bid to explore a more diverse region of the solution space. The similarity measure adopted here is based on Jaccard's similarity coefficient [14], which measures the similarity between two sets  $X$  and  $Y$  as the ratio

$$J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}. \quad (15)$$

If both sets share the same elements, then  $|X \cap Y| = |X \cup Y|$  and so  $J(X, Y) = 1$ . If, at the other extreme, the sets share no elements, then  $J(X, Y) = |X \cap Y| = 0$ . In all other cases,  $J(X, Y) \in (0, 1)$ . The similarities of the solutions in the population are similarly calculated as the ratio of the number of assignments shared by both solutions to the total number of assignments present in both solutions combined. Hence, if both solutions are exactly the same (*i.e.* the same WSs are assigned to the same threats during the same FWs), they will achieve a similarity value of 1, whereas if they are totally distinct in all their assignments, they will achieve a similarity value of 0. The similarity value of a single solution with respect to the rest of the solutions in the population is taken as the average of the similarities achieved by the solution with respect to all the other solutions in the population [4].

The recombination procedure consists of applying a so-called crossover operator and a mutation operator. The crossover operator randomly selects a single WS in turn and selects the WA schedule associated with that WS from either the first parent or the second parent for inclusion in the corresponding offspring solution — there is a probability of 0.5 that the WA schedule is taken from the first parent for each WS. During the crossover procedure, care is taken to ensure that all the constraints in (6)–(14) are satisfied, by only generating feasible offspring solutions.

Next, a mutation operator is applied to the offspring solutions to allow for local changes to a small portion of the solutions so as to promote exploration of new regions of the solution space. Our mutation procedure randomly chooses a WS in the offspring solution and unassigns all the threats assigned to it. The WS is then randomly assigned to any other threat to which it was not previously assigned. Only feasible mutations are allowed.

Once the offspring population has been created, it is combined with the parent population to form a larger intermediate population of solutions. The fitness of each of these solutions is evaluated as described earlier and the next generation of solutions is populated by including all the solutions from the first front of this intermediate population, followed by all the solutions in the second front, and so on, until the pre-specified population size is reached. If the population size would be exceeded when a next front of solutions were to be included, the similarity measure of each solution is calculated with respect to the other solutions in that front and those solutions achieving the lowest similarity values are included until the desired population size is reached.

A further diversity preservation measure, known as a *multistart procedure*, was also implemented in the algorithm. The procedure allows for the algorithm to be executed in a pre-specified number of parallel runs in order to obtain a number of approximately Pareto optimal sets of solutions. These sets are then combined to form a larger set of solutions and the fast nondominated sorting algorithm of Agarwal *et al.* [1] is again used to filter out

the dominated solutions so as to obtain a new single approximately Pareto optimal set of solutions. The idea is that the procedure allows for the approximation of different approximations of the true Pareto optimal set of solutions by using different randomly generated initial solutions. In this way a wider exploration of the solution space is promoted.

The input data required by the algorithm include the ammunition available to each WS, the required WS setup times, the cost of a burst of ammunition for each WS, the minimum length of a FW for each WS, the earliest and latest times for each WS to engage a threat, the threat priority values of the threats for each time stage (as estimated during a prior process of threat evaluation), the survival probability values of the threats for each time stage with respect to each WS and the maximum number of WSs that may be assigned to any threat during the scheduling horizon. The parameter values required by the algorithm include the population size, the tour size in the tournament selection procedure, the mutation probability and the maximum number of generations over which the algorithm should be iterated.

## 5 A worked example

The working of the tri-objective, dynamic WA model (3)–(14) is illustrated in this section by solving it in the context of a simulated SBAD scenario using the evolutionary algorithm described in §4. A detailed description of the simulated SBAD test scenario is given in §5.1 after which the numerical results obtained are described in §5.2.

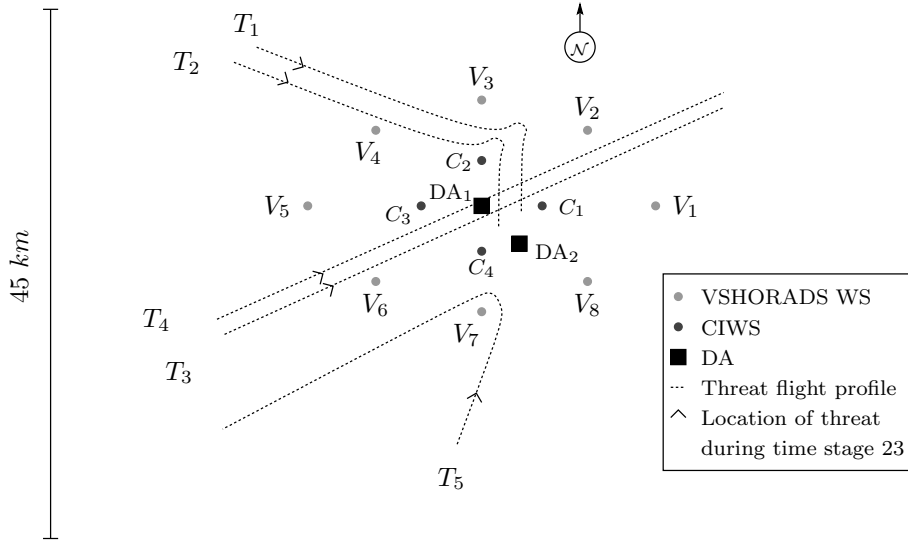
### 5.1 Simulated SBAD test scenario description

The SBAD test scenario adopted for illustrative purposes in this section was originally designed by Roux [25] and mimics a real-life SBAD deployment in which two *Defended Assets* (DAs) (labelled  $DA_1$  and  $DA_2$ , respectively) are afforded protection by two sets of SBAD WSs which are deployed in concentric circles surrounding  $DA_1$ . The first set comprises four *Close-In WSs* (CIWSs), labelled  $C_1$ ,  $C_2$ ,  $C_3$  and  $C_4$ , which are spaced evenly and deployed at approximately half the distance of their ranges from  $DA_1$  (they are deployed closest to the DAs). The second set comprises eight *Very Short Range Air Defense* (VSHORAD) WSs, labelled  $V_1$ ,  $V_2$ ,  $\dots$ ,  $V_8$ , which are deployed further away from the DAs in such a way that their primary fire arcs cross. A top-view of the locations of the DAs and the deployment of the two sets of WSs is shown in Figure 4.

An attack scenario in which five aerial threats, labelled  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  and  $T_5$ , are detected in the defended airspace is considered and they enter the defended airspace in three groupings, offset at different time stages. The first group comprises threats  $T_1$  and  $T_2$  and they approach from a north-westerly direction at an average velocity of 251 km/h and at an altitude of 430 m. They aim to attack  $DA_1$  by executing a so-called *combat hump dive attack technique*.<sup>9</sup> The second group comprises threats  $T_3$  and  $T_4$  and they approach from

---

<sup>9</sup>The combat hump dive attack technique involves a threat approaching the DA at a low altitude. When the threat is approximately 3–9 km away from the DA (depending on the type of weapons carried by the threat), it pulls up and turns in to the DA to release its ammunition at approximately 800–2500 m from the DA. After the weapon has been released by the threat, it pulls up from the dive in order to move away



**Figure 4:** Top view of the simulated SBAD scenario.

a south-westerly direction during a later stage at an average velocity of 246 km/h and at an altitude of 4 400 m. They are considered as fly-overs and act as decoys to the system in the sense that they do not aim to attack any of the DAs. Finally, the third group consists of threat  $T_5$  only which enters from a southerly direction during the same stage as the second group at an average velocity of 249 km/h and at an altitude of 90 m. It aims to attack  $DA_2$  by executing a so-called *toss bomb attack technique*.<sup>10</sup> A top view of the flight profiles<sup>11</sup> executed by the five threats is shown in Figure 4.

The time continuum of the scenario is subdivided into 120 shorter time intervals, called *time stages*, of four seconds each. During each of these time stages the future flight path of each threat is predicted for the remaining stages by employing a straight line prediction model (*i.e.* it is predicted that each threat will continue in a straight line, while maintaining its current velocity at its current altitude). The efficiency values achievable by the WSs with respect to the threats for each of these time stages are discretised into a 3-dimensional *Engagement Efficiency Matrix* having threat, WS and time as its dimensions. The engagement efficiency matrix used in the scenario is not filtered for any environmental conditions such as inclement weather or terrain obstructions (*i.e.* ideal engagement conditions and a flat earth are assumed for simplicity). The priority values of eliminating each threat (typically obtained from a *Threat Evaluation (TE)* subsystem) during each of the 120 time stages, as well as the future predicted engagement efficiency matrix values for each time stage, are available online [17].

from the DA as quickly as possible [25].

<sup>10</sup>The toss bomb attack technique involves a threat approaching the DA at low altitude. The threat pulls up at a pre-determined position (approximately 4.5–6 km from the DA) and starts aiming at the DA. It releases its ammunition at a slant distance of approximately 3.5–4.7 km from the DA. After the ammunition has been released, the threat moves away from the DA as quickly as possible [25].

<sup>11</sup>These flight paths represent the actual locations of the threats as a function of time, and not the predicted flight paths of the threats. Neither these flight paths nor the intended targets of the aerial threats are, of course, known in advance by the surface force.

Time stage 23 in the scenario was chosen as the time stage during which to solve the tri-objective dynamic WA model (3)–(14) over the remaining time stages due to the diversity in the locations of the threats during this time stage. The approximate locations of the threats during this time stage are illustrated graphically by arrows in Figure 4. The cost of the ammunition of the WSs (in South African Rand values) are assumed to be  $R\ 34\ 000$  for a single burst of a CIWS and  $R\ 1\ 000\ 000$  for a single VSHORAD missile<sup>12</sup>. Furthermore, the ammunition available to each WS during time stage 23 is given in Table 1. Finally, it is assumed that the maximum number of WSs that may be assigned to any threat during the entire time continuum (*i.e.* the value of  $\kappa$ ) is two, that the setup time of each WS is three time stages, that the minimum length of a FW is four time stages and that the number of time stages to include in the fixed mean calculation in (2) be fixed to the minimum length of a FW. The earliest and latest times during which a WS may engage a threat (*i.e.* the values of  $e_{ijk}$  and  $\ell_{ijk}$ ) are given in Table 2.

WS	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	$V_7$	$V_8$	$C_1$	$C_2$	$C_3$	$C_4$
$A_i$	7	6	5	5	5	7	8	9	5	6	4	6

**Table 1:** The ammunition available to each WS at the start of the hypothetical SBAD scenario depicted in Figure 4.

WS	Threat				
	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$
$V_1$	72	72	32	33	86
$V_2$	0	0	35	33	98
$V_3$	34	36	0	0	0
$V_4$	19	18	0	0	0
$V_5$	0	0	0	0	0
$V_6$	0	0	0	0	0
$V_7$	0	0	0	0	34
$V_8$	83	80	0	0	57
$C_1$	66	64	19	17	85
$C_2$	53	52	0	19	0
$C_3$	0	0	3	1	0
$C_4$	0	0	5	4	0

(a) Earliest engagement stages

(b) Latest engagement stages

**Table 2:** (a) The earliest and (b) the latest stages during which WSs may engage threats for the hypothetical SBAD scenario depicted in Figure 4.

## 5.2 Numerical results obtained

The multi-objective dynamic WA model (3)–(14) was solved by implementing the multi-objective evolutionary algorithm described in §4 for a population size of 100, a tour size in the selection procedure of two and a multistart procedure of 20 runs. Only two WSs were allowed to be assigned to a threat during the entire scheduling window and a particular WS was allowed to be assigned to a specific threat only once during the scheduling window. A

<sup>12</sup>These values were realistic in 2012 as the costs of a burst of ammunition for a 35 mm cannon and an Umkhonto missile, respectively.

mutation probability of 0.07 and a stopping criterion of 100 generations were implemented. This set of parameter values were decided upon after extensive experimentation since they produce an acceptable spread of solutions in the Pareto frontier. The algorithm was able to solve the model in 139 seconds on an Intel Core i7-4770 processor with 8GB of random access memory operating at 3.40 GHz.

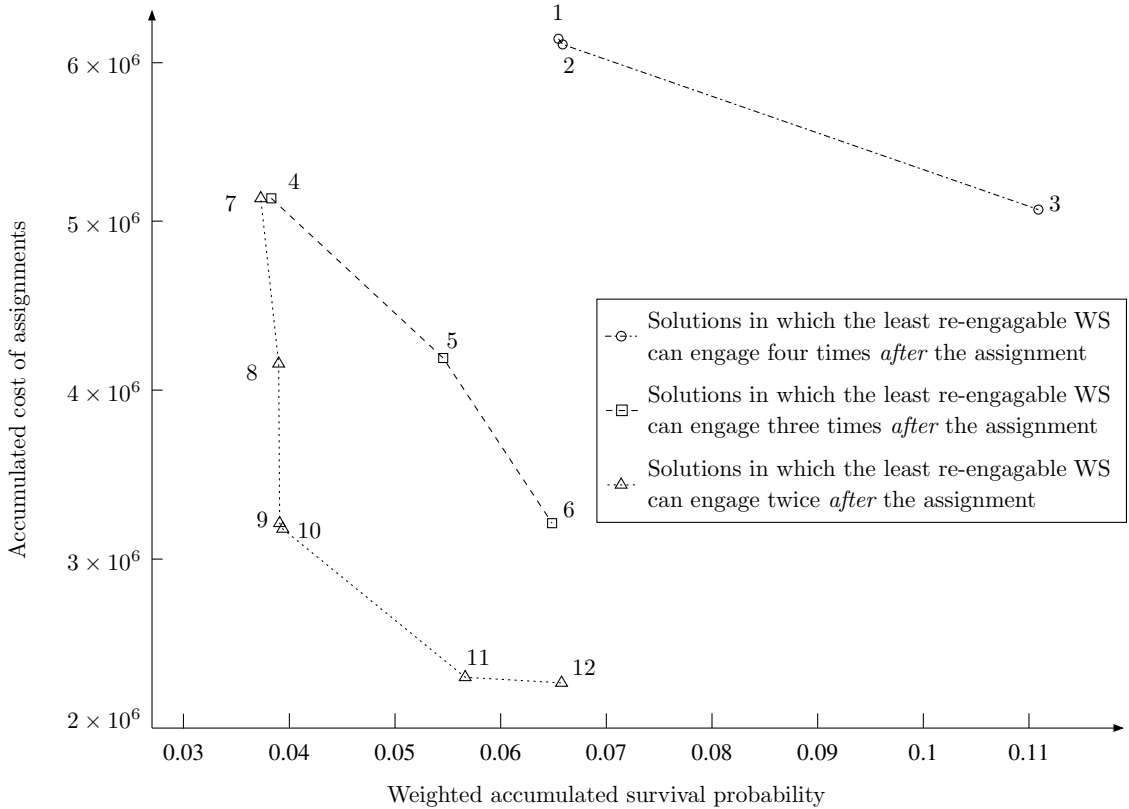
Sol	WS-threat pair assignment schedules	Obj 1	Obj 2	Obj 3
1	$V_3 [40,55], C_2 [58,64] \rightarrow T_1; V_4 [21,40], V_1 [78,88] \rightarrow T_2;$ $C_4 [14,20], V_2 [39,67] \rightarrow T_3; C_4 [6,10], C_1 [22,33] \rightarrow T_4;$ $V_7 [38,61], V_8 [60,85] \rightarrow T_5$	0.0662	R 6 136 000	4
2	$V_3 [35,53], C_2 [58,64] \rightarrow T_1; V_4 [22,28], V_8 [81,103] \rightarrow T_2;$ $C_4 [8,19], V_2 [39,47] \rightarrow T_3; C_1 [22,33] \rightarrow T_4; V_7 [37,55],$ $V_8 [57,67] \rightarrow T_5$	0.0666	R 6 102 000	4
3	$V_3 [35,53], C_2 [58,64] \rightarrow T_1; V_4 [23,40] \rightarrow T_2; C_4 [9,19],$ $V_2 [39,47] \rightarrow T_3; C_1 [22,33] \rightarrow T_4; V_7 [35,47],$ $V_8 [60,85] \rightarrow T_5$	0.1120	R 5 102 000	4
4	$C_2 [56,64], V_8 [83,101] \rightarrow T_1; V_4 [20,27], C_1 [69,76] \rightarrow T_2;$ $C_4 [13,23], V_2 [38,45] \rightarrow T_3; C_3 [8,12], C_1 [20,34] \rightarrow T_4;$ $V_8 [61,74], V_2 [98,112] \rightarrow T_5$	0.0388	R 5 170 000	3
5	$C_1 [71,76], V_8 [85,100] \rightarrow T_1; C_2 [63,67], V_1 [78,92] \rightarrow T_2;$ $C_3 [7,16], C_4 [9,21] \rightarrow T_3; C_1 [21,33], C_2 [22,28] \rightarrow T_4;$ $V_8 [61,71], V_2 [107,113] \rightarrow T_5$	0.0552	R 4 204 000	3
6	$C_1 [66,77], V_8 [85,100] \rightarrow T_1; C_2 [63,67] \rightarrow T_2; C_3 [7,16],$ $C_4 [10,23] \rightarrow T_3; C_1 [20,32], C_2 [22,28] \rightarrow T_4; V_8 [61,71],$ $V_2 [103,113] \rightarrow T_5$	0.0656	R 3 204 000	3
7	$V_3 [42,55], C_2 [57,69] \rightarrow T_1; C_1 [71,81], V_1 [75,86] \rightarrow T_2;$ $C_4 [6,17], V_2 [39,64] \rightarrow T_3; C_1 [23,34], V_1 [35,38] \rightarrow T_4;$ $C_1 [90,99], V_2 [100,116] \rightarrow T_5$	0.0378	R 5 170 000	2
8	$V_3 [42,55], C_2 [57,69] \rightarrow T_1; C_1 [71,81], V_8 [93,101] \rightarrow T_2;$ $C_4 [6,17], V_2 [39,64] \rightarrow T_3; C_1 [23,34] \rightarrow T_4; C_1 [90,99],$ $V_2 [100,116] \rightarrow T_5$	0.0395	R 4 170 000	2
9	$V_3 [42,55], C_2 [57,66] \rightarrow T_1; C_1 [71,81] \rightarrow T_2; C_4 [6,17],$ $V_2 [39,64] \rightarrow T_3; C_2 [19,27], C_1 [23,34] \rightarrow T_4; C_1 [90,99],$ $V_2 [100,116] \rightarrow T_5$	0.0396	R 3 204 000	2
10	$V_3 [42,55], C_2 [57,69] \rightarrow T_1; C_1 [71,81] \rightarrow T_2; C_4 [6,17],$ $V_2 [38,65] \rightarrow T_3; C_1 [23,34] \rightarrow T_4; C_1 [90,99],$ $V_2 [104,114] \rightarrow T_5$	0.0399	R 3 170 000	2
11	$C_2 [64,67], C_1 [66,75] \rightarrow T_1; C_2 [54,59], V_8 [86,102] \rightarrow T_2;$ $C_3 [4,14], C_4 [13,24] \rightarrow T_3; C_2 [19,24], C_1 [21,36] \rightarrow T_4;$ $V_8 [58,64], C_1 [85,98] \rightarrow T_5$	0.0573	R 2 272 000	2
12	$C_2 [63,68], C_1 [67,76] \rightarrow T_1; C_2 [53,57], V_8 [83,101] \rightarrow T_2;$ $C_4 [6,18], C_3 [7,18] \rightarrow T_3; C_1 [23,36] \rightarrow T_4; V_8 [57,77],$ $C_1 [90,101] \rightarrow T_5$	0.0665	R 2 238 000	2

**Table 3:** The set of approximately Pareto optimal solutions together with their WS-threat pair assignment schedules and their respective objective function values in Figure 5. Obj 1 represents the accumulated survival objective function values, Obj 2 represents the cost objective function values and Obj 3 represents the least re-engagable WS objective function values. The assignment schedules are given as the WS, followed by the FW in brackets, followed by a right-arrow indicating to which threat the WS should be assigned.

Twelve approximately Pareto optimal solutions were thus obtained. These solutions are listed in Table 3 together with their WS-threat pair assignment schedules (*i.e.* the FWs during which WA assignments may occur) as well their corresponding objective function



values (the weighted accumulated survival probability, the cost of the assignments and the number of ammunition available at the least re-engagable WS in the solution after the assignment). The assignment schedules for each WS-threat pair is given as the WS, followed by the FW during which the assignment should occur in brackets, followed by a right-arrow and the threat to which it is assigned (*e.g.*  $V_3 [40,55] \rightarrow T_1$  implies that WS  $V_3$  should engage threat  $T_1$  during any time stage from time stage 40 to time stage 55, inclusive).

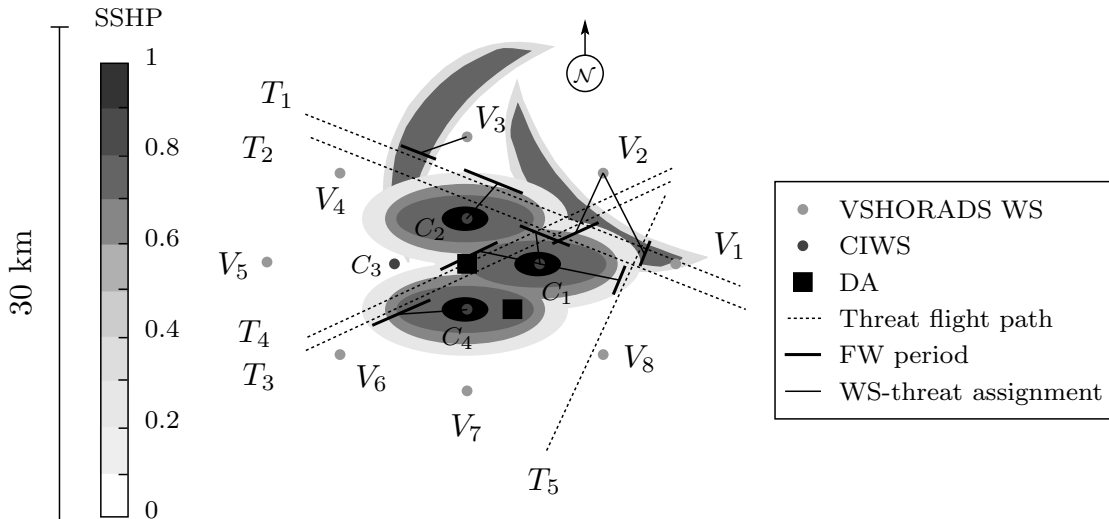


**Figure 5:** The approximately Pareto optimal frontier in solution space.

The approximately Pareto optimal solutions are also presented graphically in objective space in Figure 5. The solutions are numbered 1, ..., 12. For solutions 1, 2 and 3 (represented by open circles) the least re-engagable WS can engage four times after the assignment, for solutions 4, 5 and 6 (represented by open squares) the least re-engagable WS can engage three times after the assignment and for solutions 7, ..., 12 (represented by open triangles) the least re-engagable WS can engage twice after the assignment. Figure 5 clearly illustrates how much of one objective is compromised for a gain in another — the trade-offs between the objective function values may be observed when moving from one solution to an adjacent solution. These solutions may be presented to the fire control officer after which he may choose one of them (based on his knowledge and expertise) for implementation.

In order to illustrate the WS-threat pair assignment schedules in Table 3, consider So-

lution<sup>13</sup> 10. This solution corresponds to eight WS-threat assignment pairs with an accumulated survival probability of 0.0399 at an assignment cost of R3 170 000. The least re-engagable WSs in this solution (WS  $C_1$ ) can re-engage twice after the assignment. WS  $V_3$  should engage Threat  $T_1$  during time stages 42 to 55 (inclusive). The same threat should again be engaged by WS  $C_2$  during time stages 57 to 69 (inclusive). WS  $C_1$  should also engage Threat  $T_2$  during time stages 71 to 81 (inclusive). WS  $C_4$  should engage Threat  $T_3$  during time stages 6 to 17 (inclusive) and the threat should again be engaged by WS  $V_2$  during time stages 38 to 65 (inclusive). WS  $C_1$  should engage Threat  $T_4$  during time stages 23 to 34 (inclusive). Finally, WS  $C_1$  should engage Threat  $T_5$  during time stages 90 to 99 (inclusive), and the threat should again be engaged by WS  $V_2$  during time stages 104 to 114 (inclusive). A graphical representation of this assignment schedule is shown in Figure 6. The predicted threat flight paths are denoted by dotted lines, the fire windows are denoted by thick solid lines and the WS-threat pair assignments are denoted by thin solid lines.



**Figure 6:** Top view graphical illustration of the assignment schedule corresponding to Solution 10 in Table 3 and Figure 5.

## 6 Conclusion and possible future work

The multi-objective WA model (3)–(14) seems to yield plausible results when considering the solutions in Table 3. Figure 5 also shows an acceptable spread of Pareto optimal solutions which achieve satisfactory trade-offs between their objective function values. There are, however, two shortcomings to our modelling approach which require further research.

The first problem is the amount of time it takes for the algorithm to solve the model (139 seconds for the scenario described in §5.1). In a real-life SBAD scenario, the algorithm should be able to solve the model in a much shorter time span since the speed at which

<sup>13</sup>Only Solution 10 is discussed in detail here in order to avoid repetition.

enemy aircraft travel through the defended airspace to reach the DAs allows for very small time frames during which fire control officers must make WS-threat pair assignment decisions. The algorithm was, however, implemented in a high-level programming language (Java) in order merely to illustrate the working of the proposed WA model (3)–(14). Speed improvements of a factor of up to 100 may be obtainable with a low-level implementation [5]. Furthermore, a smaller number of parallel multistart runs of the algorithm may be selected and more restrictive stopping criteria may also be enforced in a bid to accelerate the solution procedure even further. Such restrictions may admittedly lead to a deterioration in the quality of the final solutions contained in the approximate Pareto optimal front, but the scheduling problem faced by the fire control officer is so complex that it is anticipated that even in such a restricted setting, the model (3)–(14) may yield results that are far superior to what even an experienced human operator can come up with manually within a few seconds.

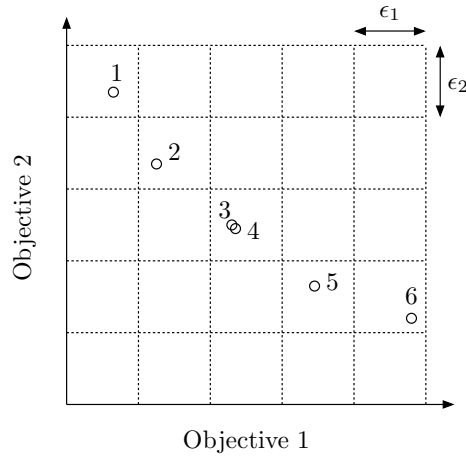
The second problem is the number of solutions that are presented to the fire control officer as decision support and the way in which these solutions are presented to the fire control officer. The possibility of overwhelming the fire control officer with information was identified by Lötter and Van Vuuren [20] as an implementation challenge in 2014 and they suggested that the human machine interface through which the results are communicated to the fire control officer should be configured in such a way that the fire control officer is presented with only the necessary information actually required by the operator.<sup>14</sup> One way of reducing the number of solutions presented to the fire control officer, is to eliminate some solutions involving objective function values which are very close to one another in all the objectives (*i.e.* solutions lying very close together in solution space, such as Solutions 1 and 2, Solutions 7 and 4, and Solutions 9 and 10, respectively, in Figure 5). The operator may well be indifferent between pairs of solutions that are so close together in solution space, in which case only one solution of each pair may actually be presented to the fire control officer in order to reduce decision support clutter.

This filtering approach may be achieved in an automated fashion by superimposing an *indifference grid* with an indifference granularity of  $\epsilon_i$  in the direction of the  $i$ -th objective over the solutions in objective space, where the values  $\epsilon_1, \dots, \epsilon_\omega$  are user-specified parameters. Two solutions residing within the same cell in such a grid may then be considered to be indistinguishable in terms of their solution quality, with the result that only one of these solutions may be presented to the fire control officer. Consider the example in Figure 7 for a bi-objective problem in which both objectives have to be minimised. There are six solutions in the Pareto optimal frontier labelled 1, . . . , 6 in the figure, and solutions 3 and 4 have objective function values which are very close to one another. Suppose an indifference grid with indifference granularity values of  $(\epsilon_1, \epsilon_2)$  has been specified by the fire control officer, as shown in the figure. Then Solutions 3 and 4 reside within the same grid cell and hence only one of them may be presented as decision support to the fire control officer (together with Solutions 1, 2, 5 and 6).

The criteria used in the decision as to which solution from a number of alternatives in the same indifference grid cell to present to the fire control officer should be implemented as

---

<sup>14</sup>The fire control officer should also have the option to view any WA related information, should he choose to.



**Figure 7:** An  $(\epsilon_1, \epsilon_2)$ -grid for a bi-objective problem in which both objectives have to be minimised.

a default setting in the WA decision support system. An example of such a criterion may be to choose the solution which achieves the smallest accumulated survival probability of the threats. The fire control officer should be able to configure these default selection criteria pre-deployment and should also have the option of overriding the default criteria in real time. Suppose, for the example in Figure 7, that the system is configured in such a way that the solution which achieves the smallest value in objective function 1 should be chosen. Solution 3 should then be chosen for presentation to the fire control officer from the alternative solutions 3 and 4.

Care should, of course, be taken in selecting appropriate values for the indifference granularities  $\epsilon_1, \dots, \epsilon_\omega$  since values that are too large may result in unnecessarily discarding solutions, while values that are too small may result in removing no or very few solutions from the recommended set and hence, again, overwhelming the fire control officer with information.

Although the indifference grid procedure may seem to be a good approach in trying to avoid overwhelming the fire control officer with decision support alternatives, it is sensitive in the way the indifference granularity values  $\epsilon_1, \dots, \epsilon_\omega$  are chosen. Solutions which lie close to one another, but lie in different cells in the grid are not considered as lying close to one another in the indifference grid method, and hence, both solutions will be presented to the fire control officer and, again, may overwhelm him/her with decision support alternatives.

Another approach that may be considered to resolve this problem is to use an operator-specified distance measure between solutions to filter away unnecessary solutions. The fire control operator should specify an allowable distance for each of the objectives (similar to the indifference granularity values  $\epsilon_1, \dots, \epsilon_\omega$  for each objective in the indifference grid method). Solutions should be compared with one another and if the distance between two solutions exceeds the specified value in all the objectives, both solutions remain in the set of Pareto optimal solutions, otherwise, only one of the two solutions remain in the Pareto set of solutions. The fire control should specify which one of the two solutions he/she prefers.

In addition, operator-specified threshold values for each objective function may be implemented in such a way that the system automatically filters out solutions that exceed these threshold values.

## 7 Acknowledgments

The authors would like to express their gratitude towards Dr Martin Kidd [15] for his assistance and valuable suggestions with respect to the implementation of the solution procedure for the multi-objective WAM (3)–(14), as described in §4.

## References

- [1] AGARWAL S, DEB K, MEYARIVAN T & PRATAP A, 2002, *A fast and elitist multi-objective genetic algorithm: NSGA II*, IEEE Transactions on Evolutionary Computation, **6(2)**, pp. 182–197.
- [2] AHUJA RK, KUMAR A, JHA KC & ORLIN JB, 2003, *Exact and heuristic algorithms for the weapon target assignment problem*, MIT Sloan, Working Paper No. 4464-03.
- [3] BRADFORD JC, 1961, *Determination of the optimal assignment of a weapon system to several threats*, Vought Aeronautics, AER-EITM-9, p. 14.
- [4] BULLINARIA JA & GARCIA-NAJERA A, 2010, *An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows*, Computers & Operations Research, **38(2011)**, pp. 287–300.
- [5] BURGER AP, 2014, Expert in Combinatorial Programming Implementations, Department of Logistics, Stellenbosch University, [Personal Communication], Contactable at [apburger@sun.ac.za](mailto:apburger@sun.ac.za).
- [6] COELLO COELLO CA, LAMONT GB & VAN VELDHUIZEN DA, 2007, *Evolutionary algorithms for solving multi-objective problems*, 2<sup>nd</sup> Edition, Springer Science, New York (NY).
- [7] DAY RH, 1966, *Allocating weapons to target complexes by means of nonlinear programming*, Operations Research, **14**, pp. 992–1013.
- [8] DU TOIT FJ, 2006, *Design of a weapon assignment system*, Honours Year Project, Stellenbosch University, Stellenbosch.
- [9] DU TOIT FJ, 2009, *The dynamic weapon target assignment problem in a ground-based air defence environment*, MSc Thesis, Stellenbosch University, Stellenbosch.
- [10] FLOOD M, 1957, *Target-assignment model*, Proceedings of the Princeton University Conference on Linear Programming, Princeton (NJ).
- [11] GOLDBERG DE & DEB K, 1991, *A comparative analysis of selection schemes used in genetic algorithms*, pp. 69–93 in RAWLINS GJE (ED), *Foundations of genetic algorithms*, Morgan Kaufman, San Mateo (CA).
- [12] HOSEIN PA & ATHANS M, 1989, *The dynamic weapon-target assignment problem*, (Unpublished) Technical Report, Massachusetts Institute of Technology, Cambridge (MA).
- [13] HUAIPING C, JINGXU L, YINGWU C & HAO W, 2006, *Survey of the research on dynamic weapon-target assignment problem*, Journal of Systems Engineering and Electronics, **17(3)**, pp. 559–565.
- [14] JACCARD P, 1901, *Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines*, Bulletin de la Société Vaudoise des Sciences Naturelles, **37**, pp. 241–272.

- [15] KIDD MP, 2014, Postdoctoral Fellow, Department of Electrical, Electronic and Information Engineering, *University of Bologna*, [Personal Communication], Contactable at [martin.philip.kidd@gmail.com](mailto:martin.philip.kidd@gmail.com).
- [16] LI H, SUGANTHAN PN, QU BY, ZHANG Q, ZHAO SZ & ZHOU A, 2011, *Multi-objective evolutionary algorithms: A survey of the state of the art*, *Swarm and Evolutionary Computation*, **1(2011)**, pp. 32–49.
- [17] LÖTTER DP, 2014, *A repository of data for a simulated SBAD scenario*, [Online], [Cited August 5<sup>th</sup>, 2014], Available from <http://www.vuuren.co.za> → **Repositories**
- [18] LÖTTER DP, NIEUWOUDT I & VAN VUUREN JH, 2013, *A multiobjective approach towards weapon assignment in a ground-based air defence environment*, *ORiON*, **29(1)**, pp. 31–54.
- [19] LÖTTER DP & VAN VUUREN JH, 2014, *Weapon assignment decision support in a surface-based air defence environment*, Submitted.
- [20] LÖTTER DP & VAN VUUREN JH, 2014, *Implementation challenges associated with a threat evaluation and weapon assignment system*, *Proceedings of the 2014 Operations Research Society of South Africa Annual Conference*, pp. 27–35.
- [21] MANNE AS, 1958, *A target-assignment problem*, *Operations Research*, **6**, pp. 346–351.
- [22] MOORE J & CHAPMAN R, 1999, *Application of particle swarm to multi-objective optimisation*, (Unpublished) Technical Report, Department of Computer Science and Software Engineering, Auburn University, Auburn (AL).
- [23] MURPHEY RA, 2000, *Target-based weapon target assignment problems*, pp. 39–53 in PARDALOS PM & PITSOULIS LS (EDS), *Nonlinear assignment problems*, Kluwer Academic Publishers, Dordrecht.
- [24] POTGIETER JJ, 2008, *Real-time weapon assignment in a ground-based air defence environment*, MSc Thesis, Stellenbosch University, Stellenbosch.
- [25] ROUX JN, 2010, *Design of a threat evaluation subsystem in a ground-based air defence environment*, PhD Dissertation, Stellenbosch University, Stellenbosch.
- [26] ROUX JN & VAN VUUREN JH, 2007, *Threat evaluation and weapon assignment decision support: A review of the state of the art*, *ORiON*, **23(2)**, pp. 151–187.
- [27] ROUX JN & VAN VUUREN JH, 2008, *Real-time threat evaluation in a ground-based air defence environment*, *ORiON*, **24(1)**, pp. 75–101.
- [28] TOTH P & VIGO D, 2001, *The vehicle routing problem*, *SIAM Monographs on Discrete Mathematics and Applications*, Society for Industrial and Applied Mathematics, Camden (PA).
- [29] VAN DER MERWE M & VAN VUUREN JH, 2013, *The weapon assignment scheduling problem in a ground-based air defence environment*, Submitted.
- [30] VOTAW DF & ORDEN A, 1952, *The personnel assignment problem*, *Symposium on Linear Inequalities and Programming*, *SCOOP* **10**, pp. 155–163.