Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Optimal Planning for Autonomous Air Vehicle Battle Management [1]

Gürdal Arslan, Jonathan D. Wolfe, Jeff Shamma, and Jason L. Speyer
Mechanical & Aerospace Engineering Department
University of California, Los Angeles
Los Angeles, CA 90095-1597, USA
(garslan, wolfe, shamma, speyer)@seas.ucla.edu

Abstract

We formulate a dynamic air vehicle assignment problem to sequentially determine the optimal allocation of air vehicle and ammunition resources to threat clusters in an air to ground campaign. A threat cluster includes a number of different types of threats, which are assumed to cooperate among themselves when they are engaged by a team of air vehicles. The objective is to efficiently allocate the assets to eliminate as many valuable threats as possible while minimizing the air vehicle attrition. This problem is formulated as an optimal control problem, whose exact solution can be obtained for small size problems by dynamic programming. For larger, more realistic problems, we investigate hierarchical control and potential function methods to solve the optimal control problem in a computationally efficient manner.

1 Introduction

In this paper, we study a resource allocation problem to optimally manage the air vehicle assets in suppressing an enemy air defense system. The enemy defense system utilizes a variety of resources, called threats, which are spread over a wide geographical area. When a number of different types of threats work in collaboration against a team of air vehicles, we say that those cooperating threats form a threat cluster. An example of a threat cluster is two coordinated Surface-to-Air-Missile (SAM) sites where each SAM site includes an Early Warning (EW) Radar, a Target Acquisition (TA) Radar, and a Transporter-Erector-Launcher (TEL) unit with a presumed infinite supply of munitions. From a top down perspective, we divide the enemy resources into a number of threat clusters, where the threat clusters do not cooperate but the threats within a cluster do cooperate. The objective of the campaign is to suppress the enemy defense system within a given time period by sequentially composing air vehicle teams of appropriate sizes and assigning them to the appropriate threat clusters. The air vehicles are equipped with a limited number of munitions, which may be of different types and hence may have different capabilities. Also, certain types of threats may be more valuable targets than the other types. Therefore the assignment of teams of air vehicles to threat clusters involves many considerations such as the values of the threats, the risk of losing an air vehicle, the number and types of munitions available, the locations of the threats, the duration of the campaign, etc. Once the teams of air vehicles are composed and assigned to different threat clusters, they move and engage their corresponding targets. An engagement is assumed to be a random event where a team of vehicles and the threats in the engaged cluster employ fixed coordinated strategies, which could also be optimal in some sense. The outcome probabilities of an engagement are based on either the strategies of the air vehicles and the threats or a worst case analysis by using game theoretical concepts. When the engagements are over, the team composition and assignment procedure is repeated, since the parameters of the campaign (the number of surviving threats and air vehicles, the number of remaining munitions, etc.) will change at the end of the engagements.

To suppress the enemy air defense system by efficiently utilizing air vehicle and ammunition resources, we formulate a finite horizon optimal control problem where the state is the assets of both sides and the control action is the allocation of the team of air vehicles to the threat clusters. The state is randomly updated at the end of the engagements (conditioned on the control action), where the engagements with different threat clusters are independent. The objective of the campaign is captured by an appropriate cost function in which the cost of moving air vehicles from one cluster to another can also be taken into account.

The optimal assignment of the air vehicles as a function of the surviving assets and the remaining time to complete the mission can be obtained by dynamic programming [1]. Because the dimension of the state

[1] This work was sponsored by the DARPA/MICA program through Alphatech Inc., Subcontract 01053-1641.

0-7803-7516-5/02/$17.00 ©2002 IEEE

space could be quite large for realistic problems, solving our optimal control problem by dynamic programming could be computationally prohibitive. This issue, namely the curse of dimensionality, calls for approximate methods; see [2, 3]. Hence, in this paper, we seek approximate solutions to our problem through hierarchical control and potential function approaches.

This paper is organized as follows: Section 2 formulates the resource allocation problem of assigning air vehicle teams to threat clusters, and presents the dynamic programming solution. In sections 3 and 4, we present the details of the hierarchical control and the potential function approaches, respectively. In section 5, we discuss the results of a simulation study. Section 6 concludes the paper.

2 Problem Setup and An Exact Solution

We consider an enemy air defense system that consists of N clusters of threats, where each threat is one of d different types. For instance, a threat could be an EW Radar, or a TA Radar, or a TEL unit. The number of threats in the $i-$th cluster at time k is denoted by a d dimensional vector s_k^i, where the $j-$th element $s_k^{i,j} \in \{0, 1, ...\}$ represents the number of threats of type j. We assume that the threats in any cluster may be cooperating, but the clusters do not cooperate among themselves. Furthermore, no new threats will be added to any cluster during a campaign, i.e.,

$$s_{k+1}^{i,j} \leq s_k^{i,j}, \ \forall i \in \{1, ..., N\}, \ \forall j \in \{1, ..., d\}, \qquad (1)$$

$\forall k \geq 0$. The enemy defense system described above is to be suppressed by M Unmanned Air Vehicles (UAVs), where each UAV has a limited number of munitions. The state of the $i-$th UAV at time k is denoted by z_k^i, which is comprised of three components, i.e., $z_k^i = [p_k^i, (q_k^i)^T, r_k^i]^T$. The first component $p_k^i \in \{0, 1, ..., N\}$ is the index of the cluster engaged by the $i-$th UAV at time k, where $p_k^i = 0$ means that $i-$th UAV is not engaged with any cluster at time k. The second component q_k^i is a t dimensional vector, where the $j-$th element $q_k^{i,j} \in \{0, 1, ...\}$ of q_k^i represents the number of the $i-$th UAV's munitions of type j at time k. The third element $r_k^i \in \{0, 1\}$ of the state z_k^i is a flag indicating whether or not the $i-$th UAV at time k is alive, where $r_k^i = 1$ means that the $i-$th UAV is alive at time k. The UAVs will not be supplied with additional munitions during a campaign, and if a UAV does not engage any cluster, then its munitions will obviously remain the same. Therefore, we can write

$$\begin{aligned} & q_{k+1}^{i,j} \leq q_k^{i,j}, \text{ and } r_{k+1}^i \leq r_k^i, \\ & p_k^i = 0 \Rightarrow q_{k+1}^i = q_k^i, \text{ and } r_{k+1}^i = r_k^i, \end{aligned} \qquad (2)$$

for all $i \in \{1, ..., N\}$, for all $j \in \{1, \ldots, t\}$ and for all $k \geq 0$. The control input is the allocation of the UAVs to the clusters at each time instant, and is denoted by u_k at time k. The control input has M components and its $i-$th component $u_k^i \in \{0, 1, ..., N\}$ is the index of the cluster which will be engaged by the $i-$th UAV at time $k+1$. Therefore, by definition, we have

$$p_{k+1}^i = u_k^i. \qquad (3)$$

We assume that the control input has access to the campaign state vector x_k defined by

$$x_k := [(s_k^1)^T, ..., (s_k^N)^T, (z_k^1)^T, ..., (z_k^M)^T]^T.$$

We think of $\{x_k : k \geq 0\}$ as a controlled Markov chain, where the transition probability from x_k to x_{k+1} is determined by the control input u_k. In particular, (1), (2) and (3), which partially describe the evolution of the campaign state, are always true with probability one. To obtain the complete evolution of the campaign state, we assume that the events which take place at different clusters are independent. This implies

$$\begin{aligned} \mathrm{P}(x_{k+1}|x_k, u_k) \quad = \quad & \mathrm{P}(z_{k+1}^{J_0^k}|z_k^{J_0^k}, u_k^{J_0^k}) \times \qquad (4) \\ & \prod_{i=1}^{N} \mathrm{P}(s_{k+1}^i, z_{k+1}^{J_i^k}|s_k^i, z_k^{J_i^k}, u_k^{J_i^k}), \end{aligned}$$

where $J_i^k \subseteq \{1, ..., M\}$ represents the set of UAVs engaged with the $i-$th cluster for $i = 0, 1, ..., N$ at time k, and $z_k^{J_i^k}$ is the collection of the states of UAVs engaged with the $i-$th cluster at time k ($u_k^{J_i^k}$ is defined analogously). From (2) and (3), the transition probability of the states of UAVs not engaging any cluster (the first term on the right-hand-side of (4)) can be written as

$$\mathrm{P}(z_{k+1}^{J_0^k}|z_k^{J_0^k}, u_k^{J_0^k}) = \begin{cases} 1 & \begin{array}{c} p_{k+1}^j = u_k^j \text{ and} \\ q_{k+1}^j = q_k^j \text{ and} \\ r_{k+1}^j = r_k^j, \ \forall j \in J_0^k \end{array} \\ 0 & \text{else} \end{cases}.$$

The transition probabilities of the states of the other UAVs, however, could be quite general (as long as they satisfy (1), (2) and (3)), and need to be computed based on the rules of a coordinated engagement of multiple UAVs with a cluster. The tactics employed by the threat clusters and the UAVs in our simulations are outlined in section 5.

The objective of the campaign, which will last T_f units of time, is to eliminate as many valuable threats as possible while minimizing the UAV attrition. This can be achieved by allocating the UAVs to different clusters in such a way that the following cost function is minimized:

$$E\{G(x_{T_f}) + \sum_{k=0}^{T_f-1} g(x_k, u_k)|x_0 = x\}, \qquad (5)$$

where the terminal cost function $G(.)$ and the one-stage cost function $g(.,.)$ are defined as

$$G(x_{T_f}) := \sum_{i=1}^{N}\sum_{j=1}^{d} e_j s_{T_f}^{i,j} - \sum_{i=1}^{M}[ar_{T_f}^i + \sum_{j=1}^{t} m_j q_{T_f}^{i,j}],$$

$$g(x_k, u_k) := \sum_{i=1}^{M} c(p_k^i, u_k^i),$$

where e_j, $j \in \{1, ..., d\}$, m_j, $j \in \{1, ..., t\}$, and a are positive constants representing the values of different types of threats, different types of munitions, and the UAVs respectively, and c is a positive-valued function that represents the cost of moving a UAV from one cluster to another. The first term in the terminal cost function is the total value of the surviving threats, whereas the second term is the total value of the surviving UAVs and the remaining munitions. To obtain the optimal allocation of UAVs to the clusters at each time instant, we use the dynamic programming procedure; see Section 5 for the results of a simulation study using this approach.

We are also interested the infinite horizon version of the finite-horizon problem formulated above. More precisely, we are interested in minimizing the cost function

$$\lim_{T_f\to\infty} E\{\sum_{k=0}^{T_f-1} \alpha^k \bar{g}(x_{k+1}, x_k, u_k)|x_0 = x\},$$

for an appropriate choice of the discount factor $\alpha \in (0, 1]$, where, to capture the essence of the finite-horizon cost, we introduced the one-stage cost function

$$\bar{g}(x_{k+1}, x_k, u_k) := G(x_{k+1}) - G(x_k) + g(x_k, u_k),$$

for the infinite-horizon problem. It is well known that a stationary controller $\mu^* : X \mapsto U$ minimizes the infinite-horizon cost, and such an optimal controller can be obtained by several different methods such as the Value Iteration and the Policy Iteration Algorithms, and the Linear programming Method; see [2]. However, all of these methods as well as the dynamic programming procedure suffer from the well-known "curse of dimensionality", which motivates us to consider the hierarchical control approach explored in the following section.

3 Hierarchical Control Approach

Here, we first assume that the enemy resources are divided into different Areas of Responsibilities (AOR) in which there are (possibly) different number of threat clusters. The significance of the concept of AOR is that the air vehicles sent to an AOR can only engage with the threat clusters within that particular AOR throughout the campaign. Therefore, it is important to make a judicious allocation of the air vehicle and munition resources to different AORs at the beginning of the campaign. Let us denote the number of AORs by L, the total number of air vehicles by M, and the total number of munitions of type j by Q_j, $j = 1, ..., t$. Furthermore, let us assume, for the time being, that the optimal expected return from assigning M^i air vehicle and $Q^i = [Q_1^i, ..., Q_t^i]$ munitions of different types to different AORs, denoted by $C_i(M^i, Q^i)$ for the $i-$th AOR, is known. Then, a reasonable allocation of the air vehicle and munition resources would be obtained by solving the following minimization problem.

$$\min_{(M^0,Q^0),...,(M^L,Q^L)} \sum_{i=0}^{L} C_i(M^i, Q^i), \text{ subject to:}$$

$$\sum_{i=0}^{L} M^i = M, \quad \sum_{i=0}^{L} Q_j^i = Q_j, \ \forall j \in \{1, ..., t\},$$

where $C_0(M^0, Q^0)$ represents the cost of not assigning to any AOR. We note that this minimization problem can be converted to an integer programming problem, and can be solved when L, M, and Q_j, $j = 1, ..., t$, are reasonably small.

To obtain the optimal costs $C_0, ..., C_L$, one ideally needs to solve an optimization problem of the type introduced in section 2. However, we know that such an optimization will likely be computationally overwhelming when $T_f > 1$. Therefore, for the purpose of computing the costs $C_0, ..., C_L$, we will assume that the air vehicles will play based on some possibly suboptimal policy. Such a suboptimal policy could be, for instance, the greedy policy, which is obtained by solving an optimization problem of the type introduced in section 2 under the assumption that $T_f = 1$, even though in actuality $T_f > 1$. It may also be possible to obtain a *good* suboptimal policy by heuristics. Although the one-stage cost of a suboptimal policy can be obtained relatively easily, the computation of its long-term expected cost may still not be an easy task. In such cases, the long-term expected return of a suboptimal policy μ_s for an initial condition x_0 can be estimated by summing up its one-stage costs for an appropriate sequence of states $y_0, ..., y_{T_f-1}$, where this sequence of states is generated by the recursion $y_{k+1} \approx E[x_{k+1}|x_k = y_k, u_k = \mu_s(y_k)]$, $k = 0, ..., T_f - 2$, with $y_0 = x_0$.

Once the high-level optimization problem described above is solved and the allocation of the resources to AORs are determined, we are faced with multiple optimization subproblems of the type introduced in section 2, one for each AOR. We already stated that each of these subproblems, which we also call mid-level optimization problems, can be solved by dynamic programming when the dimensions of the state and control spaces are sufficiently small. However, it is possible that all of the resources can be allocated to a single AOR or the number of AORs can be small compared to the resources; hence we may end up with several computationally intractable subproblems for each AOR. For such instances, we introduce a simplified version of the optimization problem formulated in section 2. The essence of this reformulated problem is that the objects of the same type, threat clusters, air vehicles, or munitions, are indistinguishable regardless of their locations. Therefore, the state variable consists of the number of threat clusters of different types, the number of armed air vehicles with different types of combinations of munitions, and the total number of munitions of different types. The mid-level optimization problem is to determine the distribution of the air vehicle and munition resources to the threat clusters sequentially. This simpler reformulation has a smaller state space than our original formulation, and it can also be solved by dynamic programming; see Section 5 for the results of a simulation study using this approach.

Finally, we discuss the issue of routing the air vehicles to their targets, which will require us to solve a low-level optimization problem. Since the solution to the mid-level optimization problem tell us only the optimal distribution of the air vehicles, we somehow need to determine the actual destination of each air vehicle. We would like to do this routing in such a way that the total distance traversed by the air vehicles is minimized subject to the satisfaction of the optimal distribution determined by the mid-level optimizer. This low-level optimization problem can be converted to an integer programming Problem as follows: Let us assume, for the sake of simplicity, that there are M identical (armed) UAVs and N identical threats, where the locations of UAVs, the base station, and the threats are denoted by $U = [U_1, ..., U_M]$, S_0, and $S = [S_1, ..., S_N]$, respectively. Let us further assume that the mid-level optimizer required the satisfaction of any permutation of the distribution vector $D = [D_1, ..., D_N]$, where $\sum_{i=1}^{N} D_i \leq M$. Then, our low-level optimization problem can be stated as:

$$\begin{aligned} &\min_x \sum_{i=1}^{M} \sum_{j=0}^{N} x_i^j \|U_i - S_j\|, \text{ subject to:} \\ &\sum_{j=0}^{N} x_i^j = 1, \ \forall i \in \{1, ..., M\}, \\ &\sum_{i=1}^{M} x_i^j = \sum_{i=1}^{N} D_i y_i^j, \ \forall j \in \{1, ..., N\}, \\ &\sum_{i=1}^{N} y_i^j = \sum_{j=1}^{N} y_i^j = 1, \ \forall i, \forall j \in \{1, ..., N\} \end{aligned}$$

where x_i^j, $i = 1, ..., M$, $j = 0, ..., N$, are the binary decision variables representing the destinations of the air vehicles, and y_i^j, $i = 1, ..., N$, $j = 1, ..., N$, are the (binary) auxiliary decision variables enforcing the satisfaction of the desired distribution of the air vehicles. When a solution to this low-level optimization problem is obtained, then the $i-$th UAV is sent to the $j-$th threat for which $x_i^j = 1$. The generalization to the case where there are multiple types of threats and resources are straight-forward but somewhat messy; hence we omit such a generalization here. We end this section by pointing out an implementation issue, that is, even though the high-level and mid-level optimization problems can be solved off-line and the solutions can be stored for every possible state, the low-level optimization problem will typically be solved on-line. The reason for this is that the number of possible locations for the UAVs throughout the campaign will be enormous even for a few UAVs, unless the locations of the UAVs are approximated by using a very coarse grid on the AORs. Moreover, an on-line implementation of this routing algorithm will make our hierarchical controller completely independent of the locations of the threats. This will sometimes require us to obtain the best feasible solution within a given time limit. Such a feasible solution will typically be a better solution than an arbitrary one, even though it may be sub-optimal.

4 Potential Function Approach

Exact solutions by dynamic programming are difficult to obtain for problems with many agents. One technique that may be used to reduce computational complexity at the expense of some degree of optimality involves using potential functions [4] to determine the movements of the UAVs. The potential function we use is

$$\begin{aligned} J(u_k^1, \ldots, u_k^M, x_k) = &\sum_{i=1}^{M} \frac{1}{1 + \sum_{j=1}^{d} e_j s_k^{u_k^i, j}} \\ &+ \sum_{i=1}^{N} \Big(\sum_{j=1}^{M} I\{u_k^j = i\} - \beta \Big)^2 + \sum_{i=1}^{M} c(p_k^i, u_k^i), \end{aligned}$$

where $(u_k^1, \ldots, u_k^M)$ are the UAV allocations at time k, x_k is the current state, e_j is the value of a threat of type j, $s_k^{i,j}$ is the total number of threats of type j at location i at time k, $I\{.\}$ is the indicator function, β is the preferred number of UAVs to be assigned to each cluster, and c is a matrix reflecting the cost of moving a UAV from one cluster to another. The parameter β is especially interesting, because varying it provides an easy way to adjust the aggressiveness of the controller (low β allows more aggressive individual behavior, while high β encourages the UAVs to better their odds in an encounter through numerical superiority). At each time step, the UAV allocations are determined by a one step optimization of the above cost function. Because the optimization occurs over only one time step, the computational complexity is low, and information constraints are easy to accommodate. Note that the potential function may be interpreted as an approximation of the optimal return function for a multi-stage dynamic program, and it may be advantageous to adjust the parameters c, e_j and β through an adaptive "learning" process.

5 A Numerical Example

First, we explain the tactics employed by the threat clusters and the UAVs in our simulation using the notation borrowed from Khalak *et. al.* [5].

We assume that there is only one type of threat cluster which we refer to as a SAM site. The tactics of a SAM site are fixed. The Early-Warning (EW) radar scans 360° with period t_s. After each scan, there is a probability ρ_s that a UAV in the scan volume is detected. Once a UAV has been detected, a SAM is launched towards it, and the Target Acquisition (TA) radar attempts to track the detected UAV. The TA radar has a scan rate s_{fc}, and a TA radar sweep hits a UAV with probability ρ_{fc}. If the TA radar misses two successive pings from the UAV, then the TA has lost track of the target. When this happens the target is presumed lost and must be detected by the EW again. The SAM takes t_m units of time to arrive at the UAV. When the SAM arrives at the UAV location, it destroys the UAV with probability ρ_d, provided that the TA has maintained track of the target during the entire flight time of the SAM. If the UAV survives, it is assumed that the SAM site has lost track of it and must attempt to detect it again. In our simulations we assume that each SAM site has an infinite ammunition supply, and that each SAM site can track and attack only one UAV at a time.

The tactics of the UAVs are also fixed. It is assumed that a UAV knows the position of the SAM site that the UAV is commanded to attack perfectly, and flies directly towards it with velocity V_{UAV}. If and when a UAV gets within r_f of the target, it launches an air-to-ground missile. The missile lands t_f later, and destroys the threat with probability ρ_k. When the missile detonates, a perfect battle damage assessment is performed instantaneously. We assume that each UAV has a limited ammunition supply.

To design our controllers, the outcome probabilities of the encounters need to be computed based on the tactics explained above and the following parameter values we picked for our simulations: $r_{fc} = 150$ km, $t_s = 10$ s, $s_{fc} = 1$ s^{-1}, $\rho_s = 0.01$, $\rho_{fc} = 0.69$, $\rho_d = 0.25$, $\rho_k = 0.86$, $r_f = 40$ km, $t_f = 60$ s, $V_{UAV} = 250$ m/s, $t_m = 30$ s. This computation can be done quite accurately through a Markov Chain analysis with some simplifying assumptions. However, for additional simplicity, we prefer to construct the outcome probabilities of the encounters based on a simple encounter where there is only one UAV with one missile and one SAM site. We obtain the outcome probabilities of this simple encounter through Monte-Carlo simulations as $(p_s, p_u) = (0.05, 0.75)$, where (p_s, p_u) are the probabilities of the events that the SAM site destroys the UAV and the UAV destroys the SAM site, respectively, which are assumed to occur independently of each other. We further assume, for the purpose of calculating the outcome probabilities, that each SAM site is a threat cluster, and hence does not cooperate with other SAM sites. Therefore, teams of UAVs are assumed to engage only one SAM site, and these engagements are assumed to be independent. In such an engagement, before the UAVs are reallocated by the controller, the SAM site picks one of the UAVs randomly and destroys it with probability p_s, and the team of UAVs destroy the SAM site with probability $1-(1-p_u)^M$, where M is the team size. As a result of the various independence assumptions we made above, we compute the outcome probabilities of all possible encounters based on an engagement of a team of UAVs with a single SAM site.

We now discuss the results of our simulations for two different cases. In the first case, we simulate a campaign with three UAVs, each of which has two missiles, and four SAM sites, whose locations are randomly picked at the beginning. We set the value of a SAM site and a UAV to unity, the cost of moving a UAV from one location to another to 0.05% of the distance traversed in kilometers, and the value of a missile to zero. We then compute and store the exact solution to the problem formulated in section 2 for three time steps, and a decentralized controller based on the potential function approach setting the parameter β to unity. In our implementation, we reallocate the UAVs only when all the engagements corresponding to an allocation end, in order to make the implementation as close to the formulation of section 2 as possible. The cost, as defined in (5), achieved in simulations by the controllers based on the exact solution and the potential function solution averaged over 20 runs are $J_{Exact} = -1.4752$, and $J_{Pot} = -0.8007$, respectively. We note that the cost achieved by the exact solution is somewhat lower than the cost achieved by the controller based on the potential function approach, as expected. However, the cost achieved by both controllers are quite higher than the optimal expected cost $J^* = -2.2541$ predicted by dynamic programming. We explain the high costs achieved by the two controllers with the synchronization problem we observed under both controllers. The synchronization problem arises because, very often, the engagements do not end simultaneously, and those UAVs that completed their mission must wait for all the engagements to end before being reallocated to their new targets. This may cause the controllers to perform poorly because of inefficient utilization of the air vehicle resources. Although it is difficult to avoid the synchronization problem completely, the hierarchical control approach, which is illustrated below, addresses this issue in a satisfactory way. We also note that the dimension of the state space for this case is 128000, and that this is the case with the largest number of threats and air vehicles for which we could obtain an exact solution by dynamic programming.

In the second case, we simulate a campaign with 20 UAVs, each of which has four missiles, and 25 SAM sites, under a hierarchical controller and a controller based on the potential function approach. The locations of the SAM sites are again randomly picked at the beginning, and the values of a UAV and a SAM site are set to unity. The moving cost in this case is set to 0.025% of the distance traversed in kilometers. To compute the hierarchical controller, we divide the area into two AORs with 15 and 10 SAM sites in them, respectively. We use the long-term expected cost of a heuristic policy as the high-level costs to solve the high-level optimization problem. The solution to the high-level optimization problem assigns 11 UAVs to the first AOR and 9 UAVs to the second AOR. We then solve the mid-level optimization problems for the two AOR for an infinite time horizon with a discount factor $\alpha = 0.99999$. After solving the high and mid-level optimization problems off-line and storing the results, we implement the hierarchical controller by solving the low-level optimization problem online after each reallocation of the UAVs. We reallocate the UAVs whenever a SAM site or a UAV is destroyed, or a UAV runs out of ammunition. We also compute and implement a decentralized controller based on the potential function approach in the same way as in the first case with $\beta = 1$. The cost $J_H = -3.5592$ achieved by the hierarchical controller turns out to be appreciably lower than the cost $J_{Pot} = 0.6692$ achieved by the potential function approach, both averaged over 20 runs. We note that, in the second case, it is virtually impossible to obtain the exact optimal cost and/or the exact solution to the problem introduced in section 2 by dynamic programming because the dimension of the state space and the control space would be approximately 10^{51} and 10^{28}, respectively. Hence, the hierarchical control and potential function approaches appear to be sensible alternatives to an exact solution by dynamic programming for the problems with large number of threats and air vehicles. Finally, we observed in the second case that while the controller based on the potential function approach continued to suffer from the synchronization problem, the hierarchical controller was able to reallocate those air vehicles that completed their mission to new targets.

6 Conclusions

In this paper, we formulate a dynamic air vehicle assignment problem to determine the sequence of optimal allocations of air vehicles to threat clusters in an air to ground campaign. The objective is to minimize a cost function which captures our original objective, that is to destroy as many valuable threats as possible while limiting air vehicle attrition. In principle, the minimization of the cost function can be carried out by using the dynamic programming procedure, and the sequence of optimal allocations can be computed exactly. However, this approach quickly becomes impractical as the size of the problem grows. Hence, for more realistic problems, we explore the hierarchical control and potential function methods to deal with the issue of dimensionality. To compute our hierarchical controller, we introduce three levels of hierarchy and solve an optimization problem at each level. The high-level optimization yields the allocation of the resources to different AORs. The sequential distribution of the resources to the threats within the AORs is achieved by solving a mid-level optimization problem for each AOR. Finally, the destinations of the UAVs are determined by solving a low-level optimization problem online for each AOR such that the desired distribution of the resources to the threats is achieved while minimizing the moving cost. We show through simulations, that the hierarchical control approach not only allows us to solve problems with larger number of threats and air vehicles but also provides a satisfactory solution to the synchronization problem by reallocating and rerouting the UAVs to new targets as soon as they complete their mission. We also illustrate by simulations that the potential function approach is an intuitive and computationally efficient method to tackle the problems with large number of threats and air vehicles. Furthermore, because the potential function approach only conducts optimization over one time step, it allows us to easily incorporate the communication constraints into the controller design.

References

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control: Vol. 2.* Athena Scientific, MA, 1995.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming.* Athena Scientific, MA, 1996.

[3] R. S. Sutto and A. G. Barto, *Reinforcement Learning: An Introduction.* MIT Press, MA, 1998.

[4] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” IEEE Control Systems Magazine, vol. 22, pp. 52–67, June 2002.

[5] A. Khalak, B. Bank, T. Sliski, and J. Wohletz, *SHARC challenge problem version 0.3.* 2002.