Dynamic air tasking evaluation in a simulated network-centric battlespace

Madjid Tavana*

Management Information Systems,
La Salle University, Philadelphia, PA 19141, USA
E-mail: tavana@lasalle.edu
*Corresponding author

Matthew D. Bailey

Bucknell University,
Lewisburg, PA 17837, USA
E-mail: mdb025@bucknell.edu

Timothy E. Busch

Air Force Research Laboratory,
AFRL/IFSB, 525 Brooks Road,
Rome, NY 13441, USA
E-mail: timothy.busch@rl.af.mil

Abstract: Joint Air Operations (JAO) are historically designed through centralised planning using static air tasking that assigns air assets to mission packages for the purpose of achieving campaign objectives. The current methodology cannot anticipate changes in the battlespace nor take advantage of real-time information. In this study, we develop a simulation model of the battlespace that utilises a Multi-Criteria Decision Analysis (MCDA) model to assign vehicles to targets by considering four competing objectives (effort, effectiveness, efficiency, and connectivity). A Voronoi network is used to determine the paths of vehicles to their assigned targets and network optimisation is used to validate the quality of the assignments. The results indicate that dynamic air tasking is considerably more effective and more efficient than static air tasking.

Keywords: vehicle-target assignment; simulation; MCDA; multi-criteria decision analysis; network optimisation; multi-hop connectivity; Voronoi network.

Reference to this paper should be made as follows: Tavana, M., Bailey, M.D. and Busch, T.E. (2009) 'Dynamic air tasking evaluation in a simulated network-centric battlespace', *Int. J. Operational Research*, Vol. 5, No. 1, pp.1–25.

Biographical notes: Madjid Tavana is Professor of Management Information Systems and the Lindback Distinguished Chair of Information Systems at La Salle University where he served as the Chairman of the Management Department and Director of the Center for Technology and Management. He has been a distinguished faculty fellow at NASA's Kennedy Space Center,

NASA's Johnson Space Center, Naval Research Laboratory – Stennis Space Center, and Air Force Research Laboratory. In 2005, he was awarded the prestigious Space Act Award by NASA. He holds an MBA, a PMIS, and a PhD in Management Information systems and received his Post-Doctoral Diploma in Strategic Information Systems from the Wharton School of the University of Pennsylvania. He is the Editor-in-Chief for the *International Journal of Applied Decision Sciences* and the *International Journal of Strategic Decision Sciences*. He has published in journals such as *Decision Sciences, Interfaces, Information Systems, Information and Management, Computers and Operations Research*, *Journal of the Operational Research Society, and Advances in Engineering Software*, among others.

Matthew D. Bailey is an Assistant Professor of Management at Bucknell University. He received his PhD in Industrial and Operations Engineering from the University of Michigan. His primary research interests are the theory, computation aspects, and applications of stochastic optimisation. In particular, he is interested in modelling adversarial relationships as well as medical decision-making. He is a member of the Institute for Operations Research and the Management Sciences, and the Institute of Industrial Engineers.

Timothy E. Busch is a senior engineer with the Information Systems Research Branch of the Air Force Research Laboratory. He received his PhD in Electrical Engineering from the Binghamton University in the area of Control Theory. His primary research interest is in the application of model predictive control techniques to dynamic decision problems typical in high tempo operations. He is a research advisor to the National Research Council and an Assistant Professor in the Electrical Engineering Department of the State University of New York Institute of Technology. He is a senior member of the IEEE and also an Associate Editor for the *IEEE Transactions on Systems, Man, and Cybernetics Part C*.

1 Introduction

Joint Air Operations (JAO) involve joint air capabilities and forces in support of a military operation. Currently in JAO, vehicles are assigned predefined tasks through the Air Tasking Order (ATO) from the centralised command and control. The tasks are based on the needs of the mission and the limits of the vehicle capabilities. This results in a set of vehicles with static roles during an operation, such as surveillance and strike. With the advancement of information and communication technology and the increasing need for flexibility and responsiveness, vehicles are expected to perform multiple tasks and roles depending on situation needs and circumstances. In this paper, we expand our previous dynamic task allocation model (Tavana et al., 2008) to incorporate a simulated battlespace that can be used to reassign vehicles to various roles in a threat-filled environment.

Once the objectives of a mission are defined, Master Air Attack Planners (MAAPs) assign vehicles to targets and develop a plan of execution, the ATO. The ATO details how air power will support the overall mission in the presence of uncertainties and a hostile enemy. This plan transmitted to operators at the tactical level is a static plan. In addition, updates and changes to the ATO are time intensive and often require extensive coordination. As a result, the ability to exploit the continuous information

provided by the sensors is greatly hindered. These issues highlight the need for enhanced real-time information sharing and performance measurement that incorporates anticipative capabilities regarding alternative configuration scenarios and future uncertain events and a framework for evaluating such measurement. The Air Force is exploring various innovative infrastructures to increase the effectiveness of JAO. This study exploits this real-time information and proposes a MCDA model in a network-centric environment to assess different vehicle-target assignment scenarios and illustrate their effectiveness in a simulated battlespace.

The rigid air tasking process does not allow aircrafts to use real-time information to self-organise and more effectively and efficiently engage targets. We propose a dynamic environment where events such as the destruction of a target or loss of a vehicle trigger the assessment system. The system evaluated in this study, utilises real-time information to enhance information sharing, collaboration, shared situational awareness, and self-synchronisation fundamental to mission effectiveness in network-centric warfare. This tool is evaluated in an accompanying simulated battlespace.

The scope of our problem is related to the weapon-to-target assignment problem and vehicle task allocation. The weapon-to-target assignment problem deals with allocating weapons to targets in either a static or dynamic environment and has received considerable attention with surveys in Murphey (1999) and Voss (1999). Typically methods for solving these problems are mixed-integer linear programming (Schumacher et al., 2007), nonlinear programming (Ahuja et al., 2007), goal programming (Green et al., 1997), nonlinear network flow (Castanon, 1987), neural networks (Wachholder, 1989), and evolutionary genetic algorithms (Grant, 1993; Lee et al., 2003; Lee and Lee, 2003, 2005). Dynamic programming (Flint, 2003) and network optimisation (O'Rourke et al., 2000) have also been used to solve target assignment problems. Network optimisation modelling treats the individual vehicles as discrete supplies of single units while tasks are carried out as flows on arcs through the network. Tabu search is used to solve the difficult combinatorial network optimisation problems (O'Rourke et al., 2000; Toth and Vigo, 2002; Cullenbine et al., 2003). Although, these classic target assignment models consider static and dynamic assignment, they neglect multiple roles for vehicles and communication.

Task allocation is primarily discussed in the cooperative control of Umanned Air Vehicle (UAV) literature; see Chandler et al. (2002), Darrah et al. (2005), Rasmussen et al. (2002, 2003) and Schumacher et al. (2001, 2002). It includes assigning vehicles to various tasks such as strike and sensor. However, maintaining communication is not considered.

Discrete-event simulation has been used extensively for planning and model validation in a variety of applications (Robinson, 2005). Law and Kelton (2002) provide a comprehensive overview of the methodology and the applications areas. Within the context of military planning, discrete-event simulation has a long played a fundamental role with a detailed review provided by Hill et al. (2001). In addition to vehicle-target assignment, the discrete-event model developed in this study simulates the routing of vehicles in a threat-filled space. In the context of unmanned aerial vehicles, this problem has been investigated by Beard et al. (2002) and Pfeiffer et al. (2005); however, we focus on the methodology for assessing path threats in Beard et al. (2002) because of its computational efficiency.

We formulate the problem of assigning vehicles to targets as a MCDA problem with competing and conflicting objectives. This model utilises a weighted-sum measure of

performance for vehicle-target assignments by integrating several measures reflecting multiple objectives driven by problem situation and not by a particular method. Each objective is assigned a weighting value representing the relative importance of that objective, thus transforming the multiple objectives into an aggregated objective function. The details of the MCDA and optimisation models are presented in the next section followed by a description of the simulation model in Section 3. Section 4 presents the simulation data and results followed by the conclusion and future research directions presented in Section 5.

2 Dynamic air task allocation

2.1 MCDA model

While classical air tasking models are static, the model presented here utilises MCDA, Voronoi networks, and network optimisation to simulate and validate dynamic air tasking. We reallocate the roles and tasks of vehicles utilising a MCDA model that considers communication connectivity and area threats. This reallocation occurs on an as needed basis or when a reallocation event occurs. We define the reallocation events as:

- an unsuccessful or successful confirmation of a target
- a strike on a confirmed target
- confirmation of an unsuccessful or successful strike on a target
- confirmation of an unsuccessful or successful Battle Damage Assessment (BDA) of a target
- the loss of a vehicle.

In our study, we focus on reallocations motivated by the occurrence of one of the above events. We assume that a vehicle is assigned to one and only one target; however, multiple vehicles could be assigned to a target. We consider a problem with T known targets and V available vehicles. We let h_{vt} represent the assignment of the vth vehicle to the tth target. Where $h_{vt} = 1$ to indicate assignment of the vth vehicle to the tth target, otherwise $h_{vt} = 0$. The resulting assignment matrix is denoted as H where the vth row corresponds to the vth vehicle and the tth column corresponds to the tth target, i.e., $[H]_{vt} = h_{vt}$. The vehicle's role is dependent on the current status of a target. Although the locations of the targets are assumed to be known, the status of a target must be confirmed before a strike. Once a target has been confirmed, the vehicle is assigned the 'strike' role, until that time the vehicle is assigned the 'sensor' role. Tavana et al. (2008), has proposed a MCDA model with four competing objectives (F_r) to determine the allocation of the roles and targets: *Effort* (F_1), *Effectiveness* (F_2), *Efficiency* (F_3), and *Connectivity* (F_4).

All four objectives are normalised to a real number between 0 and 1. An overall objective function value F, where $0 \le F \le 1$, is constructed through a weighted linear combination of the four objectives, $F = \sum_{r=1}^{4} w_r F_r$, where w_r is the relative importance (weight) of each objective $(0 \le w_r \le 1)$ such that $\sum_{r=1}^{4} w_r = 1$ and F_r is the performance measure of each objective. Upon the occurrence of a reallocation event, we use this measure of performance for the evaluation of M selected scenarios detailing

the alternative vehicle-target assignment. We denote these scenarios as S_x (for $x = 1, 2, \ldots, M$) and $s_{x,r}$ is the measure of performance of scenario S_x in terms of objective F_r. Based on these definitions F^x is the weighted overall performance of scenario x. We briefly review the various components that are combined to compute each of the above performance measures.

Effort (F_1) is the measure of the weighted proportion of vehicles assigned to the various targets. This weight is based on the relative importance of each target. We define X_t as the number of vehicles assigned to target t, where $X_t = \sum_{v=1}^{V} h_{vt}$. In addition, we define p_t as the proportion of total vehicles assigned to target t, i.e., $p_t = X_t/V$. We use are given normalised weights to the target w_t' where $\sum_{t=1}^{T} w_t' = 1$. Using these weights, we find $\overline{p}_t$, the weighted proportion of total vehicles assigned to target t as $\overline{p}_t = w_t' p_t$. We then find the normalised weighted proportion of total vehicles assigned to target t, $\hat{p}_t = \overline{p}_t / \sum_{j=1}^{T} \overline{p}_j$. After the vector $\hat{p} = (\hat{p}_1, p_2, ..., \hat{p}_T)$ has been computed, we employ a Shannon entropy measure (Shannon 1948), $E(\hat{p}) = -(1/\ln(r)) \sum_{t=1}^{T} \hat{p}_t \ln \hat{p}_t$ and evaluate the amount of dissonance among the weighted proportion of vehicles assigned to different targets. This entropy measure takes a value between 0 and 1. Its value is zero if all vehicles are assigned to a target with maximum weight and its value is one if the weighted proportion assigned to each target is identical. We then define our efficiency measure, $F_1 = 1 - E(\hat{p})$ where $0 \le F_1 \le 1$. The higher F_1, the more vehicles are assigned to higher-valued targets.

The second measure, effectiveness (F_2), determines the proportion of the total value of targets that are assigned at least one vehicle. If any vehicle is assigned to target t, i.e., $h_{vt} = 1$ for some v, we define $b_t = 1$ otherwise $b_t = 0$. Based on these indictors we compute $F_2 = \sum_{t=1}^{T} w_t' b_t$ using the target weights from above and then $0 \le F_2 \le 1$.

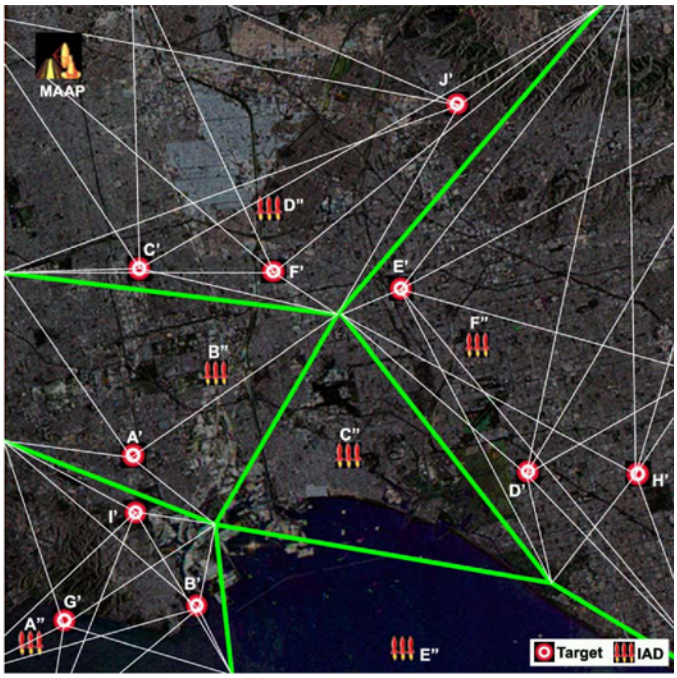
The third measure, efficiency (F_3), is a measure of the proximity of vehicles to the known targets balanced by the proximity of IADs. We define d_{vt} as the distance of the vth vehicle to the tth target. Since all measures are being maximised, we define $\overline{d}_{vt}$, $\overline{d}_{vt} = 1/d_{vt}$ and $\hat{d}_{vt}$, the normalised measure of performance of distance of the vth vehicle to the tth target, as $\hat{d}_{vt} = \overline{d}_{vt} / \sum_{t=1}^{T} \sum_{v=1}^{V} \overline{d}_{vt}$. In addition, we consider A known IADs and define d_{va} as the distance of the vth vehicle to the ath IAD. The average distance of the vth vehicle to the A IADs, $\overline{d}_v$, is $\overline{d}_v = \sum_{a=1}^{A} d_{va} / A$. Further defining d_{ta} as the distance of the tth target to the ath IAD, we find the average distance of target t to the A IADs, $\overline{d}_t$, as $\overline{d}_t = \sum_{a=1}^{A} d_{ta} / A$. The measure of performance of assigning vth vehicle to the tth target in terms of average distance from the IAD, $\overline{\overline{d}}_{vt}$, is $\overline{\overline{d}}_{vt} = \overline{d}_v + \overline{d}_t / 2$. We find the normalised measure of this performance as $\hat{\overline{d}} = \overline{\overline{d}}_{vt} / \sum_{t=1}^{T} \sum_{v=1}^{V} \overline{\overline{d}}_{vt}$. Next, we average $\hat{d}_{vt}$ and $\hat{\overline{d}}$ to find d_{vt}^*, the combined measure of travel and threat for assigning the vth vehicle to the tth target $\left(d_{vt}^* = (\hat{d}_{vt} + \hat{\overline{d}}_{vt}) / 2 \right)$. Using these values, we find the normalised measure of travel and threat for assigning the vth vehicle to the tth-target, $d_{vt}^{**} = d_{vt}^* - \underset{v,t}{\text{Min}}\, d_{vt}^* / \underset{v,t}{\text{Max}}\, d_{vt}^* - \underset{v,t}{\text{Min}}\, d_{vt}^*$. The overall measure of travel and threat for a given assignment, F_3, is $F_3 = \sum_{v,t} h_{vt} d_{vt}^{**} / V$ where $0 \le F_3 \le 1$.

Our final measure, connectivity (F_4), assesses the quality of the current communication network by determining the extent of multi-hop connectivity (Miller, 2001). Multi-hop connectivity provides a measure of the number of vehicles (or communication nodes) through which information must be relayed for vehicles to

communicate. To determine this measure, we create the $V \times V$ multi-hop connectivity matrix C where the (i, j)th entry in the matrix, c_{ij} corresponds to the minimum number of hops for vehicle i to relay information to vehicle j and a 0 indicates that no connectivity exists between i and j. In this instance, we replace 0 with $V+1$ to penalise the disconnectivity. As in Miller (2001), this matrix is derived from an adjacency matrix A which consists of indicators for whether a direct link between vehicles exists. In this matrix, a 1 in the (i, j)th entry indicates that vehicle i is directly linked to vehicle j and a 0 indicates that no direct connection exists. Once the multi-hop connectivity matrix C has been derived we compute the value β where $\beta = \sum_{i=1}^{V} \sum_{j=1}^{1} c_{ij} / V(V-1)$ and $0 \le \beta \le V+1$. For fully-connected networks, β is the average hop distance; and for disconnected networks, β is the penalised hop distance (penalised for disconnectivity). Therefore, $F_4 = 1 - \beta / V + 1$ provides the measure of the connectivity of the current network $(0 \le F_4 \le 1)$.

Once the various scenarios have been evaluated and a selection has been made, the simulation progresses as the vehicles perform their assigned roles. In order to achieve this, we determine the selected path for a vehicle and the probabilities of various reallocation events. To determine the paths of vehicle to their assigned targets, we measure the threat based on a set of preferred paths through a Voronoi network (Aurenhammer and Klein, 2000) which considers the volume of threats and their proximity to the links in the Voronoi network. The network is created by taking the locations of the IADs and creating a network around them such that any point on a link is equidistant from the IADs on either side of it. The Voronoi network for the example simulation run with ten targets and six IADs is presented in Figure 1.

Figure 1 The Voronoi graph of the example simulation run (see online version for colours)



This network provides a set of feasible paths through the battlespace that maximises the distance from individual IADs. We then connect each target to the network by determining the closest vertices, thus creating a connected network with a set of nodes N and links L. Although pilots will select their own paths to minimise exposure, these path approximations are used for computationally efficient target assignment valuation. As in Beard et al. (2002), an efficient measure of the threat along an arc l of this network is computed based on its proximity to the set of defense sites. For an arc l in L of length d_l, we use three points along the arc ($d_l/6$, $3d_l/6$, $5d_l/6$) to estimate the threat over that link. Using the distance from IAD a to the point 1/6 of the length along arc l, defined as $d_{1/6,l,a}$, and the similarly defined $d_{3/6,l,a}$ and $d_{5/6,l,a}$, we compute a cumulative threat measure over arc l, $\text{threat}_l = d_l / 3 \sum_{a=1}^{I} (1/d_{1/6,l,a}^4 + 1/d_{3/6,l,a}^4 + 1/d_{5/6,l,a}^4)$.

We then find the closest node to the vehicle based on the vehicle's current location and determine a 'shortest path' to the assigned target using a weighted average of the length of each arc and the above threat measure for that arc, $m_a = \gamma d_l + (1-\gamma)\text{threat}_l$, where $\gamma \geq 0$. The weight γ will be determined by the assumed preference of a pilot to balance a vehicle's proximity to a target and the threat over that path.

For the simulation to progress we must also address the occurrence of the other reallocation events. We assume a vehicle confirms a target in its sensor range in a region with a given probability. Although the actual probability of target confirmation is based on several factors, such as the sensor footprint and angle to the target, we simplify this by defining the probability that a vehicle confirms a target as a fixed p_c. After a target is confirmed, a vehicle can be assigned to strike the target. For a strike, we utilise a probability p_s as the probability that a vehicle destroys a target. Once a target has been attacked a vehicle can be assigned for BDA which is a sensor role. We assume that there is a known probability of success for a vehicle's BDA. Using these vehicle and target specific probabilities, the simulation progresses until the next reallocation event upon which the dynamic air tasking model is utilised to update the task assignments.

2.2 Network optimisation model

In order to validate the quality of the allocation proposed by the above MCDA and the simulation, we compared our solution to that generated by another target selection and task allocation model. This model is a network optimisation model based on Bailey et al. (2006). At each reallocation event or decision we assign a feasible task to each vehicle. To determine the task allocations for each vehicle, an extension of the classic assignment problem (Kuhn, 1955; Munkres, 1957) is utilised. Similar to the allocation of strike and sensor roles, we define $h_{vt} = 1$ if vehicle i is allocated to target t and 0 otherwise. Additionally, we define $r(v, t)$ as the reward for assigning the vth vehicle to the tth target to confirm a target (sensor), strike, or perform BDA (sensor). These rewards are based on the status of the target at the time of reallocation and whether it is unconfirmed, requiring a strike, or requiring BDA. Assuming the system rewards are separable, these definitions result in the following binary integer program,

$$NC = \max_h \sum_{v,t} r(v,t)h(v,t). \tag{1a}$$

Subject to

$$\sum_{i=2}^{T} h(v,t) \le 1, \quad \text{for } v = 1,\ldots,V, \tag{1b}$$

$$\sum_{v=1}^{V} h(v,t) \le 1, \quad \text{for all unverified targets,} \tag{1c}$$

$$\sum_{i=1}^{V} h(v,t) \le 1, \quad \text{for all verified targets,} \tag{1d}$$

$$\sum_{i=1}^{V} h(v,t) \le 1, \quad \text{for all targets with an attempted strike,} \tag{1e}$$

$$h(v,t) \in \text{IB} \quad \text{for } i = 1,\ldots,v \quad \text{and} \quad t = 1,\ldots,T. \tag{1f}$$

Constraints (1b) restrict each vehicle to be assigned to at most one target. Constraints (1c) restrict the confirmation (sensor) role for each target t to at most one vehicle. Constraints (1d) and (1e) are defined similarly for strike and BDA roles. The rewards for the above model are based on the probabilities of confirmation and strike as defined below in the simulation. Similar to Schumacher et al. (2002), we define the benefits for vehicle to target and task allocation based on the target status.

For a target that has not been confirmed the reward for assigning vehicle v to this target t is

$$\begin{aligned} r(v,t) &= \text{The expected reward from confirming target } t \text{ and subsequently destroying} \\ &\quad \text{target } t \text{ with vehicle } v - \text{the threat/distance of vehicle } v \text{ travelling to target } t, \\ &= \alpha^m p_c p_d w_t' - \beta c(v,t). \end{aligned}$$

For a target that has been confirmed, but not yet attacked the reward is

$$\begin{aligned} r(v,t) &= \text{The expected reward for striking confirmed target } t \text{ with vehicle } v \\ &\quad - \text{the threat/distance of vehicle } i \text{ travelling to target } t, \\ &= \alpha^m p_d w_t' - \beta c(v,t). \end{aligned}$$

For a target that has been attacked, but whose destruction has not been confirmed the reward for an assignment is,

$$\begin{aligned} r(v,t) &= \text{The expected reward for confirming the post-strike status of target } t \\ &\quad \text{by vehicle } v - \text{the threat/distance of vehicle } v \text{ travelling to target } t, \\ &= \alpha^m (1 - p_d) w_t' - \beta c(v,t) \end{aligned}$$

where m is the amount of time elapsed since a full communication network connectivity was in place and α is a discount factor, $0 \le \alpha \le 1$ to account for this lack of communication network connectivity. The β is a scaling factor to relate the path threat values and the expected rewards. The term $c(v, t)$ represents the shortest path for vehicle v to travel to target t using the m_a values, described above, as arc lengths. The values w_t' are identical to the weights utilised in the MCDA. The other terms are probabilities of

events as defined in the associated simulation. We describe our simulation model in the next section.

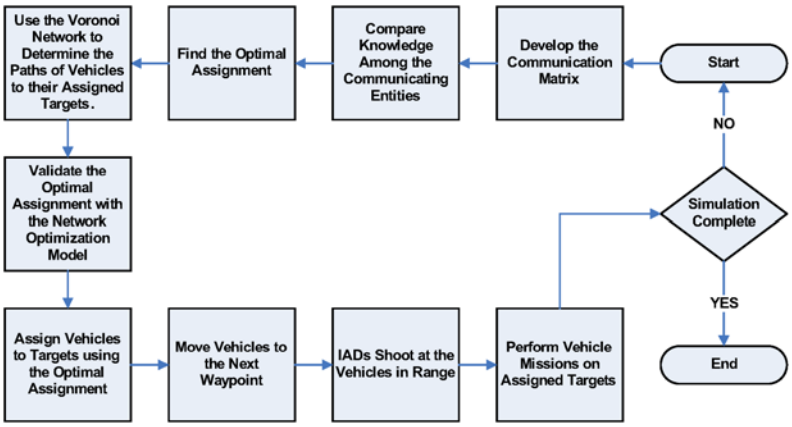
3 Simulation model

The simulation model presented here depicts a JAO in which an air package is assembled and assigned to a geographical location.

3.1 Simulation framework

The airbase where the MAAP is located is the staging point for all air packages. Once the mission objectives are defined, the MAAP assigns vehicles (aircrafts) to destroy or confirm targets while avoiding IADs. A vehicle is assigned to one and only one target while multiple vehicles could be assigned to a target. Targets have predetermined priorities in the simulation. The priority of a destroyed target is set to zero and the remaining active targets inherent this priority through a normalisation process. Each IAD has a pre-specified attack range with a known probability that determines if the vehicle will be shot down. A graphical representation of the simulation framework is presented in Figure 2.

Figure 2 Simulation framework (see online version for colours)



Vehicles are the only moving entities in the simulation. They travel throughout the operations theatre at a pre-specified constant velocity. Initially, all vehicles are assigned to active or confirmed targets with a known location. After completing each assignment, vehicles obtain a new order from the MAAP. Each vehicle travels to its assigned target and attempts to complete its *strike* or *sensor* mission depending on whether the target is *Confirmed*, *Unconfirmed*, or *Destroyed*. Strike missions are assigned to confirmed targets and BDA or sensor missions are assigned to unconfirmed targets to determine whether they are active or destroyed. Destroyed targets are those targets that are taken out in strike missions and confirmed as being destroyed in a following sensor missions. Depending on the success of the mission and whether or not the vehicle is in communication range of the airbase, the MAAP will reassign the vehicle. If a vehicle completes its mission and is

outside communication range of the airbase, it will return to base by default for new orders. Vehicles have limited munitions. If a vehicle expends all of its munitions it will no longer be able to strike targets and will only assume a sensor role.

Vehicles and the MAAP have a pre-specified communication range and have knowledge of the targets and their status. Although knowledge is localised to each vehicle or the MAAP, it is shared every time two knowledge carrying entities are within communication range of each other. The communicating entities compare their simulation timestamps and the most current knowledge with the latest timestamp is updated. New knowledge is obtained by vehicles when they strike targets or perform BDA of the targets. When a vehicle strikes a target, its local knowledge for that target changes from confirmed to unconfirmed. If the BDA is successful, the status of the target changes to destroyed or confirmed and if the BDA is unsuccessful, the vehicle's local knowledge of the target remains unconfirmed. Communication within vehicles and between the vehicles and the MAAP is not limited to direct communication. Communicating entities can communicate with each other over multi-hop networks. If a vehicle is outside of direct communication range of the MAAP, it can still communicate with the MAAP if it is within range of another vehicle which is in range of the MAAP. This instantaneous chaining of communication works over multiple intermediate hops in the simulation.

The MAAP is the brains of the simulation and issues ATO to the vehicles. In this study, we have developed two different simulation models for static and dynamic air tasking. The static air tasking model depicts the current system employed by the US Air Force where vehicles are assigned to the next confirmed target from a pre-specified ATO that remains unchanged throughout the simulation. In the dynamic air tasking model presented in this study, the MCDA model described earlier is used to determine the values associated with Effort (F_1), Effectiveness (F_2), Efficiency (F_3), Connectivity (F_4), and the overall measure of performance (F) after the occurrence of each event. The optimal solution choice is identified using the threat-path determination on the created Voronoi graph described earlier. The network optimisation model described in the previous section is used to validate the results. A new ATO is issued and if a vehicle's assignment is changed and the vehicle is in the communication range, it aborts its previous assignment and carries out the new assignment. When a vehicle's assignment changes and it is outside of a communication range, the vehicle will continue with its previous assignment until it completes its assignment or re-enters a communication chain.

3.2 Simulation flowchart

This simulation model is developed in Java, using Java SE 6 with the following parameters:

- *RandomSeedNumber*: the seed used to generate random numbers
- *RunNumber*: the number of simulation runs representing the maximum duration of a mission
- *ObjectiveWeights*: the weights associated with the four objectives
- *RegionSize*: the width and the height of the simulation region in miles

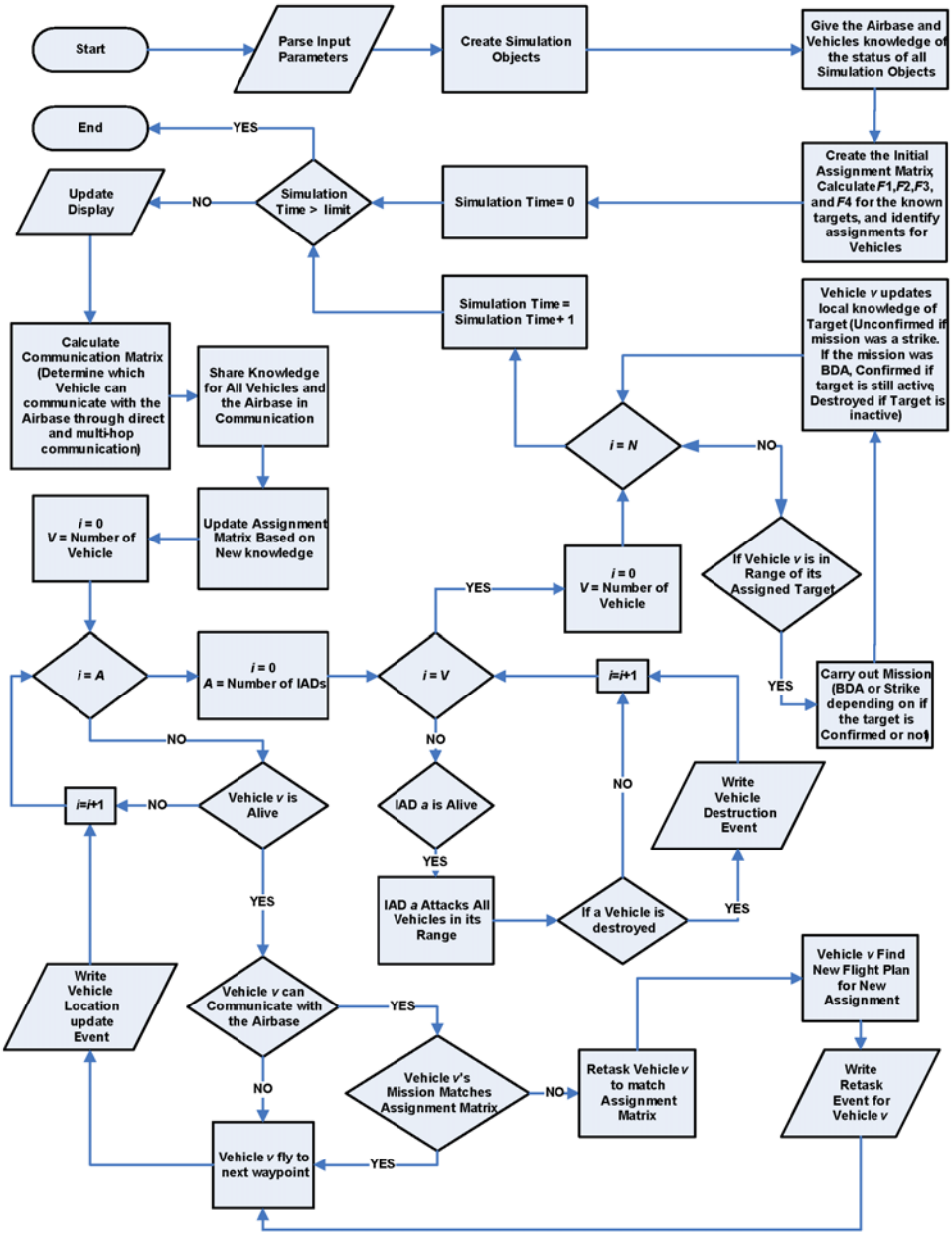
- *SleepNumber*: simulation time step duration (ms) for controlling execution speed
- *DurationNumber*: the maximum number of time steps for a single simulation run
- *SensorProbability*: the probability of success for a BDA assignment
- *StrikeProbability*: the probability of success for a strike assignment
- *BaseLocation*: the x and y coordinates of air base
- *BaseRange*: the communication range of the air base (miles)
- *VehicleSpeed*: the velocity of the vehicles (mph)
- *VehicleRange*: the communication range of the vehicles (miles)
- *VehicleMunitions*: the number of munitions of the vehicles
- *TargetLocation*: the x and y coordinates of the targets
- *TargetPriority*: the importance weight of the targets
- *IADLocation*: the x and y coordinates of the IADs
- *IADRange*: the attack range of the IADs (miles)
- *IADProbability*: the successful attack probability of the IADs.

As it is shown in the simulation flowchart presented in Figure 3, after the scenario file has been parsed, all entities are initialised and the vehicles begin at the airbase. The MAAP and the vehicles are given knowledge of the entities in the system including vehicles, targets and IADs. Once the simulation is initialised, the simulator checks to see if the number of simulation runs has elapsed. If so, the simulator exits. Otherwise, the simulator proceeds to the execution loop. Each pass through the execution loop represents one time step in the simulation. Every time step is meant to represent one minute of actual time. The display shows the actual physical location of every entity in the simulation as well as some additional knowledge such as the status of vehicles and targets. The display is updated to properly reflect the current state of the simulation and the simulator determines which of the entities can communicate with each other through direct or multi-hop communication. The simulator implements the communication model described earlier to establish a communication paradigm among the entities. The communication paradigm is used to update the knowledge of all entities capable of communicating with each other. This task is accomplished by iterating through each knowledge carrying entity and comparing its knowledge with all other knowledge carrying entities. If an entry has a more recent time stamp, the newer entry is kept by both entities and the old one is discarded.

Next, the simulation develops a new assignment matrix. In the static model, each vehicle is assigned to the next target on its list. If all targets on a vehicle's list are destroyed, the vehicle returns to the airbase. In the dynamic air tasking model, a new assignment matrix is developed based on the four objective functions: F_1, F_2, F_3, and F_4; and the threat-path determination on the created Voronoi graph. Following the calculation of the optimal assignment, the solution is validated with the network optimisation model and the simulator informs all vehicles within the communication loop about their new assignments. If a vehicle's assignment is changed, the simulation assigns a role to match

the status of the new target (i.e., sensing if the target is unconfirmed or strike if the target is confirmed). If the vehicle has expended all of its munitions it will take up a sensing role. The vehicles start moving towards their new assignments and the simulator checks the IADs to see if there are any vehicles within their range. If there are any vehicles within an IAD's shooting range, it attacks the vehicle with a pre-specified probability of success. The statuses of the vehicles that are shot downed by IADs is changed from active to destroyed and are they are no longer controlled by the MAAP.

Figure 3 Simulation flowchart (see online version for colours)



Finally, the simulation checks every vehicle's target range. If a vehicle is within range of its target, it attempts to complete its assignment. If the vehicle has a sensing role, it will attempt to scan the unconfirmed target with a previously specified probability of success. If the BDA is successful, the true status of the target will be revealed (either destroyed or confirmed). Otherwise, the target will remain unconfirmed. If the vehicle has a strike role, it will attempt an attack on the target expending ammunition. The strike will or will not be successful depending on a random probability of the success of an attack. After the strike, the local knowledge of the vehicle for the target is changed to unconfirmed whether or not the strike is actually successful. The simulation increments the time step and proceeds to the beginning of the main execution loop. This continues until either all the targets have been confirmed destroyed and all active vehicles have returned to the airbase or the duration limit for the mission has passed.

3.3 Simulation display

All inputs to the simulation model are parsed from either an input file or a graphical user interface. Once all input parameters are set, the simulation creates all the objects including the airbase, vehicles, targets, and IADs. Images are updated throughout the simulation run. Vehicles, represented by planes are either strike (), sensor (), or destroyed (). Targets, represented by bull's-eyes, are either confirmed (), unconfirmed (), or destroyed (). IADs are represented by missiles () images and the airbase is represented by a small airstrip/radar image (). Three example simulation displays are provided in the next section.

The airbase and vehicles must communicate throughout the simulation to share their local knowledge in order to create a global view of the environment. Each vehicle and airbase has a communication range, represented as a blue circle that surrounds that object. Vehicles have an additional red circle around them representing their shooting range for the role of strike, and BDA range for the role of sensor. The red circles around the IADs represent their attack range. If a vehicle flies within this range, there is a probability that that vehicle will be shot down. The longer a vehicle stays in this range, the greater the chance it will be destroyed. Dotted yellow lines show the vehicle-target assignments.

The next section presents a comprehensive evaluation of our MCDA dynamic air tasking model in the proposed simulation environment. As will be shown, dynamic air tasking is considerably more effective than static air tasking.

4 Simulation results

In this section, we present a comprehensive evaluation of the dynamic air tasking model described earlier using the proposed simulation environment. As will be shown, dynamic air tasking is considerable more effective and more efficient than static air tasking. We begin by describing the model through a sample test case followed by the data collected from 100,000 randomly generated static air tasking cases and 100,000 randomly generated dynamic air tasking cases.

The sample test case is based on a scenario with 5 vehicles (A thru E), 10 targets (A' thru J'), and 6 IADs (A'' thru F''). All targets and IADs are randomly located on a map. In the static air tasking case, each vehicle which is capable of destroying two

targets, is randomly assigned to two targets. Vehicle A is assigned to targets I' and D', vehicle B to G' and E', vehicle C to C' and F', vehicle D to B' and H', and vehicle E to A' and J'. Table 1 shows various vehicle-target assignments time stamped from $t + 000$ (start) to $t + 285$ (end) for the above static air tasking.

Table 1 The vehicle-target assignments for the example simulation run using static air tasking

	Vehicle				
Time	*A*	*B*	*C*	*D*	*E*
$t + 000$	I'	G'	C'	B'	A'
$t + 009$	I'	G'	C'	B'	A'
$t + 018$	I'	G'	F'	B'	A'
$t + 026$	I'	G'	F'	B'	A'
$t + 038$	I'	G'	–	B'	A'
$t + 049$	I'	G'	–	B'	A'
$t + 057$	I'	G'	–	B'	J'
$t + 063$	I'	G'	–	B'	J'
$t + 074$	D'	G'	–	B'	J'
$t + 085$	D'	G'	–	B'	J'
$t + 098$	D'	E'	–	B'	J'
$t + 122$	D'	E'	–	B'	J'
$t + 131$	D'	–	–	H'	J'
$t + 148$	D'	–	–	H'	J'
$t + 165$	D'	–	–	H'	J'
$t + 178$	D'	–	–	H'	–
$t + 192$	D'	–	–	H'	–
$t + 246$	D'	–	–	H'	–
$t + 254$	–	–	–	H'	–
$t + 285$	–	–	–	H'	–

As shown in Table 1, the simulation starts at $t + 000$. At $t + 009$, vehicle C destroys target C'; at $t + 026$, C destroys F'; at $t + 049$, E destroys A'; at $t + 063$, A destroys I'; at $t + 085$, B destroys G'; at $t + 122$, B destroys E' and D destroys B'; at $t + 165$, E destroys J'; at $t + 246$, A destroys D'; and at $t + 271$, D destroys H'. The static air tasking simulation for this scenario is terminated at $t + 285$. All targets are destroyed and no vehicles are lost. It should also be noted that the initial vehicle-target assignments remains unchanged throughout the static air tasking.

Next, we run our dynamic air tasking simulation model for the same problem with five vehicles, ten targets, and six IADs. All simulation parameters are identical for both experiments. Table 2 presents vehicle-target assignments throughout this simulation run. Figure 4 presents a simulation snapshot of the example simulation run using dynamic air tasking at $t + 056$.

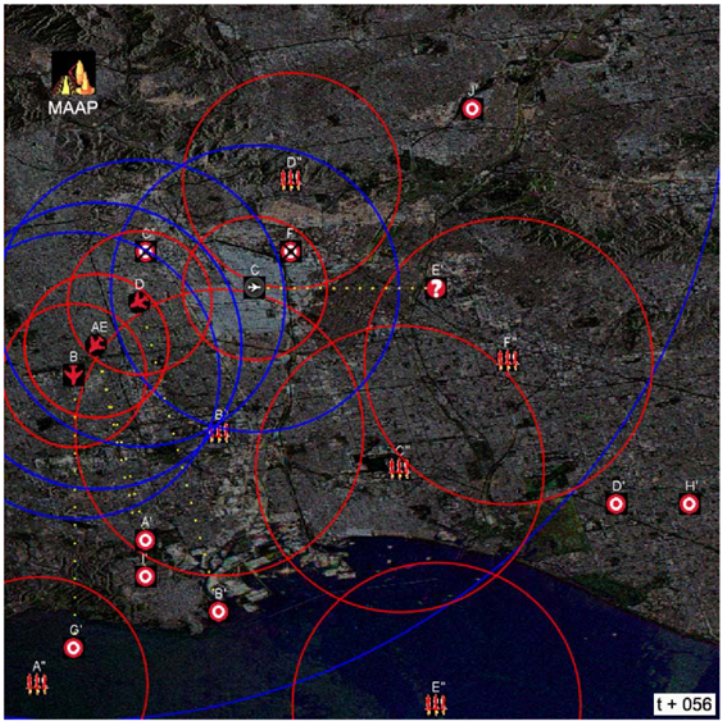
Table 2 The role and status of the vehicles for the example simulation run using dynamic air tasking (see online version for colours)

Time	*Vehicle*				
	A	*B*	*C*	*D*	*E*
$t + 000$					
$t + 012$					
$t + 023$					
$t + 037$					
$t + 048$					
$t + 056$					
$t + 067$					
$t + 079$					
$t + 088$					
$t + 098$					
$t + 112$					
$t + 123$					
$t + 135$					
$t + 142$					
$t + 161$					
$t + 175$					
$t + 184$					
$t + 192$					
$t + 201$					
$t + 201$					
$t + 212$					

Strike Vehicle Sensor Vehicle Destroyed Vehicle

As it is shown in Figure 4, at $t + 056$, vehicles *A*, *B*, *D*, and *E* have assumed the strike role while vehicle *C* has assumed the BDA role. The role and status of the vehicles for different timestamps throughout the dynamic air tasking simulation are presented in Table 2. Table 2 also shows the simulation starts at $t + 000$ and terminates at $t + 212$. The dynamic air tasking has reduced the simulation run from 285 minutes to 212 minutes resulting in a 25.6% reduction in the mission completion time.

Figure 4 Simulation snapshot of the example simulation run using dynamic air tasking at $t + 056$ (see online version for colours)



As it is shown in this table, vehicles are assigned either a strike or a BDA role, depending on the solution derived from the MCDA and the network optimisation models. For example, Vehicle *C* is assigned several alternating strike and sensing roles throughout the simulation while vehicle *D* is assigned to the strike role almost the entire simulation run except for the last few minutes from $t + 201$ to $t + 212$ where it was assigned to BDA. None of the vehicles were lost throughout the dynamic air tasking run.

Vehicle-target assignments for $t + 056$ are also visible in Figure 4. The dotted yellow lines show vehicle *A* is assigned to strike target *I′*, vehicle *B* is assigned to strike target *G′*, vehicle *C* is assigned to target *E*' for BDA, vehicle *D* is assigned to strike target *B′*, and vehicle *E* is assigned to strike target *A′*. The vehicle-target assignments for different timestamps throughout the dynamic air tasking simulation run are presented in Table 3. Table 3 also shows that initially vehicles *A*, *B*, *C*, *D*, and *E* were assigned to strike targets *I′*, *G′*, *C′*, *B′*, and *A′*. Vehicle *C* destroys targets *C′*, *F′*, and *E′* at $t + 012$, $t + 037$, and $t + 056$. Vehicle *B* destroys target *I′* at $t + 079$. Vehicle *E* destroys targets *A′*, *B′*, and *G′* at $t + 098$, $t + 123$, and $t + 142$. Vehicle *A* destroys targets *J′* and *H′* at $t + 175$ and $t + 192$. Finally, vehicle *D* destroys target *D′* at $t + 201$.

The status of targets at $t + 056$ are also noticeable in Figure 4. Targets *A′*, *B′*, *D′*, *G′*, *H′*, *I′*, and *J′* al all confirmed while targets *C′* and *F′* are destroyed and target *E′* is unconfirmed. The status of the targets for different timestamps throughout the dynamic air tasking simulation is presented in Table 4.

Table 3 The vehicle-target assignments for the example simulation run using dynamic air tasking

Time	*Vehicle*				
	A	*B*	*C*	*D*	*E*
t + 000	*I′*	*G′*	*C′*	*B′*	*A′*
t + 012	*I′*	*G′*	*C′*	*B′*	*A′*
t + 023	*I′*	*G′*	*F′*	*B′*	*A′*
t + 037	*I′*	*G′*	*F′*	*B′*	*A′*
t + 048	*I′*	*G′*	*E′*	*B′*	*A′*
t + 056	*I′*	*G′*	*E′*	*B′*	*A′*
t + 067	*J′*	*I′*	*G′*	*B′*	*A′*
t + 079	*J′*	*I′*	*G′*	*B′*	*A′*
t + 088	*J′*	*J′*	*G′*	*B′*	*A′*
t + 098	*J′*	*J′*	*G′*	*B′*	*A′*
t + 112	*J′*	*J′*	*J′*	*G′*	*B′*
t + 123	*J′*	*J′*	*J′*	*G′*	*B′*
t + 135	*J′*	*J′*	*J′*	*J′*	*G′*
t + 142	*J′*	*J′*	*J′*	*J′*	*G′*
t + 161	*J′*	*J′*	*J′*	*H′*	*D′*
t + 175	*J′*	*J′*	*J′*	*H′*	*D′*
t + 184	*H′*	*D′*	*D′*	*D′*	*D′*
t + 192	*H′*	*D′*	*D′*	*D′*	*D′*
t + 201	*D′*	*D′*	*D′*	*D′*	*D′*
t + 212	*D′*	*D′*	*D′*	*D′*	*D′*

Table 4 The status of the targets for the example simulation run using dynamic air tasking (see online version for colours)

Time	*Target*									
	A′	*B′*	*C′*	*D′*	*E′*	*F′*	*G′*	*H′*	*I′*	*J′*
t + 000										
t + 012										
t + 023										
t + 037										
t + 048										
t + 056										
t + 067										
t + 079										
t + 088										

Table 4 The status of the targets for the example simulation run using dynamic air tasking (see online version for colours) (continued)

	Target									
Time	*A′*	*B′*	*C′*	*D′*	*E′*	*F′*	*G′*	*H′*	*I′*	*J′*
t + 098	?	◎	⊗	◎	⊗	⊗	◎	◎	⊗	◎
t + 112	⊗	◎	⊗	◎	⊗	⊗	◎	◎	⊗	◎
t + 123	⊗	?	⊗	◎	⊗	⊗	◎	◎	⊗	◎
t + 135	⊗	⊗	⊗	◎	⊗	⊗	◎	◎	⊗	◎
t + 142	⊗	⊗	⊗	◎	⊗	⊗	?	◎	⊗	◎
t + 161	⊗	⊗	⊗	◎	⊗	⊗	⊗	◎	⊗	◎
t + 175	⊗	⊗	⊗	◎	⊗	⊗	⊗	◎	⊗	?
t + 184	⊗	⊗	⊗	◎	⊗	⊗	⊗	◎	⊗	⊗
t + 192	⊗	⊗	⊗	◎	⊗	⊗	⊗	?	⊗	⊗
t + 201	⊗	⊗	⊗	?	⊗	⊗	⊗	⊗	⊗	⊗
t + 212	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗

? Unconfirmed Target ◎ Confirmed Target ⊗ Destroyed Target

The initial target priorities specified were 0.10, 0.20, 0.05, 0.05, 0.10, 0.05, 0.25, 0.05, 0.10, and 0.05 for targets *A′*, *B′*, *C′*, *D′*, *E′*, *F′*, *G′*, *H′*, *I′*, and *J′* respectively. However, as the simulation progresses, targets are destroyed and the remaining targets inherent the priorities of the destroyed targets through a normalisation process. The priorities of the targets for different timestamps throughout the dynamic air tasking simulation are presented in Table 5.

The scores of the four objectives, *Effort* (F_1), *Effectiveness* (F_2), *Efficiency* (F_3), and *Connectivity* (F_4) for different timestamps throughout the dynamic air tasking simulation are presented in Table 6. A graphical representation of these scores is also provided in Figure 5.

The rise and fall of F_1 during a mission may be attributed to number of available targets during various points of a mission. While there are few targets, the assignment of several vehicles to the remaining targets is easily accomplished. For a larger number of targets, the MCDA can be selective and focus on the high value targets. Once the high priority targets are eliminated then the vehicles are spread among the remaining (potentially equal-valued) targets, so F_1 decreases until the number of targets is reduced.

As for F_4, communication causes problems when targets have a level of status, but a vehicle does not have updated information. Initially, the information at each vehicle is up-to-date for all the targets. This information may slowly become incorrect, however as the targets are eliminated the scope of information is reduced and more vehicles are assigned to fewer targets, so that we see that metric F_4 is related to F_1. A larger number of targets causes the model to focus force on high priority targets (so close together implying good communication) while with fewer targets the force has to be focused

together for a better level of F_1 and F_4. As a result, it is the in-between times that F_1 and F_4 are reduced.

Table 5 The priority of the targets for the example simulation run using dynamic air tasking

	Target									
Time	*A′*	*B′*	*C′*	*D′*	*E′*	*F′*	*G′*	*H′*	*I′*	*J′*
$t + 000$	0.100	0.200	0.050	0.050	0.100	0.050	0.250	0.050	0.100	0.050
$t + 012$	0.100	0.200	0.050	0.050	0.100	0.050	0.250	0.050	0.100	0.050
$t + 023$	0.105	0.211	0.000	0.053	0.105	0.053	0.263	0.053	0.105	0.053
$t + 037$	0.105	0.211	0.000	0.053	0.105	0.053	0.263	0.053	0.105	0.053
$t + 048$	0.111	0.222	0.000	0.056	0.111	0.000	0.278	0.056	0.111	0.056
$t + 056$	0.111	0.222	0.000	0.056	0.111	0.000	0.278	0.056	0.111	0.056
$t + 067$	0.125	0.250	0.000	0.063	0.000	0.000	0.313	0.063	0.125	0.063
$t + 079$	0.125	0.250	0.000	0.063	0.000	0.000	0.313	0.063	0.125	0.063
$t + 088$	0.143	0.286	0.000	0.071	0.000	0.000	0.357	0.071	0.000	0.071
$t + 098$	0.143	0.286	0.000	0.071	0.000	0.000	0.357	0.071	0.000	0.071
$t + 112$	0.000	0.333	0.000	0.083	0.000	0.000	0.417	0.083	0.000	0.083
$t + 123$	0.000	0.333	0.000	0.083	0.000	0.000	0.417	0.083	0.000	0.083
$t + 135$	0.000	0.000	0.000	0.125	0.000	0.000	0.625	0.125	0.000	0.125
$t + 142$	0.000	0.000	0.000	0.125	0.000	0.000	0.625	0.125	0.000	0.125
$t + 161$	0.000	0.000	0.000	0.333	0.000	0.000	0.000	0.333	0.000	0.333
$t + 175$	0.000	0.000	0.000	0.333	0.000	0.000	0.000	0.333	0.000	0.333
$t + 184$	0.000	0.000	0.000	0.500	0.000	0.000	0.000	0.500	0.000	0.000
$t + 192$	0.000	0.000	0.000	0.500	0.000	0.000	0.000	0.500	0.000	0.000
$t + 201$	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
$t + 212$	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 5 The graphical view of the objectives during the example simulation run using dynamic air tasking (see online version for colours)

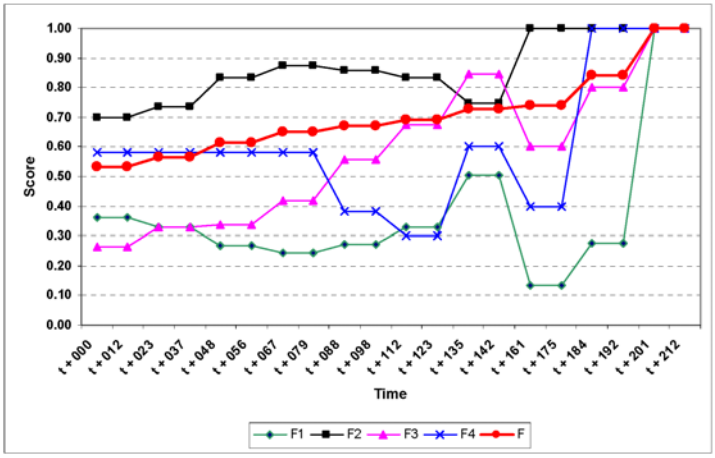


Table 6 The score of the objectives for the example simulation run using dynamic air tasking

Time	Objective				
	Effort (F_1)	*Effectiveness* (F_2)	*Efficiency* (F_3)	*Connectivity* (F_4)	*Overall* (F)
$t + 000$	0.362	0.700	0.264	0.583	0.534
$t + 012$	0.362	0.700	0.264	0.583	0.534
$t + 023$	0.331	0.737	0.329	0.583	0.566
$t + 037$	0.331	0.737	0.329	0.583	0.566
$t + 048$	0.267	0.833	0.338	0.583	0.612
$t + 056$	0.267	0.833	0.338	0.583	0.612
$t + 067$	0.245	0.875	0.417	0.583	0.651
$t + 079$	0.245	0.875	0.417	0.583	0.651
$t + 088$	0.271	0.857	0.557	0.383	0.671
$t + 098$	0.271	0.857	0.557	0.383	0.671
$t + 112$	0.330	0.833	0.675	0.300	0.692
$t + 123$	0.330	0.833	0.675	0.300	0.692
$t + 135$	0.504	0.750	0.845	0.600	0.729
$t + 142$	0.504	0.750	0.845	0.600	0.729
$t + 161$	0.135	1.000	0.602	0.400	0.741
$t + 175$	0.135	1.000	0.602	0.400	0.741
$t + 184$	0.278	1.000	0.800	1.000	0.842
$t + 192$	0.278	1.000	0.800	1.000	0.842
$t + 201$	1.000	1.000	1.000	1.000	1.000
$t + 212$	1.000	1.000	1.000	1.000	1.000

The sensitivity of the weights associated with the four objectives for different timestamps throughout the dynamic air tasking simulation run is presented in Table 7. The lower-bound/upper-bound limits indicate how far a weight can be altered before the optimal solution for a particular timestamp is changed. For example, at $t + 056$, reducing the pre-specified weight of 0.15 for F_1 is not feasible. However, the optimal solution will change if the weight for F_1 is increased beyond the 0.281 threshold.

Table 7 The sensitivity of the weights associated with the four objectives in the example simulation run using dynamic air tasking

Time	$w_1 = 0.15$		$w_2 = 0.55$		$w_3 = 0.25$		$w_4 = 0.05$	
	Lower bound	*Upper bound*	*Lower bound*	*Upper bound*	*Lower bound*	*Upper bound*	*Lower bound*	*Upper bound*
$t + 000$	0.147	0.161	0.528	0.660	0.248	0.253	0.036	0.052
$t + 012$	0.147	0.161	0.528	0.660	0.248	0.253	0.036	0.052
$t + 023$	NF	0.249	0.490	0.594	0.215	0.305	0.034	0.121
$t + 037$	NF	0.249	0.490	0.594	0.215	0.305	0.034	0.121
$t + 048$	NF	0.281	0.484	NF	NF	0.285	0.028	0.131

Table 7 The sensitivity of the weights associated with the four objectives in the example simulation run using dynamic air tasking (continued)

	$w_1 = 0.15$		$w_2 = 0.55$		$w_3 = 0.25$		$w_4 = 0.05$	
Time	*Lower bound*	*Upper bound*	*Lower bound*	*Upper bound*	*Lower bound*	*Upper bound*	*Lower bound*	*Upper bound*
$t + 056$	NF	0.281	0.484	NF	NF	0.285	0.028	0.131
$t + 067$	0.149	0.155	0.545	0.572	0.248	0.255	0.038	0.052
$t + 079$	0.149	0.155	0.545	0.572	0.248	0.255	0.038	0.052
$t + 088$	0.146	0.155	0.545	0.561	0.245	0.253	0.036	0.057
$t + 098$	0.146	0.155	0.545	0.561	0.245	0.253	0.036	0.057
$t + 112$	NF	0.275	0.391	0.643	0.157	0.510	NF	0.258
$t + 123$	NF	0.275	0.391	0.643	0.157	0.510	NF	0.258
$t + 135$	0.058	0.278	0.319	0.605	0.202	0.682	NF	0.282
$t + 142$	0.058	0.278	0.319	0.605	0.202	0.682	NF	0.282
$t + 161$	NF	0.245	0.462	1.000	0.030	0.410	NF	0.195
$t + 175$	NF	0.245	0.462	1.000	0.030	0.410	NF	0.195
$t + 184$	NF	0.269	0.413	1.000	0.000	0.527	NF	1.000
$t + 192$	NF	0.269	0.413	1.000	0.000	0.527	NF	1.000
$t + 201$	NF	NF	NF	NF	NF	NF	NF	NF
$t + 212$	NF	NF	NF	NF	NF	NF	NF	NF

NF: Not Feasible.

Next, we present the data collected from 180,000 randomly generated air tasking cases for nine different scenarios in Table 8. Each scenario consisted of 20,000 randomly generated cases. 10,000 cases were simulated using the dynamic air tasking model presented in this study and 10,000 cases were simulated using static air tasking model currently employed by the US Air Force. The first scenario was constructed with two vehicles, two targets, and two IADs and the last scenario was based on nine vehicles, nine targets, and nine IADs.

As it is shown in Table 8, the number of targets destroyed has increased 7.85% on average using dynamic air tasking. This difference between the average number of targets destroyed with dynamic air tasking and the average number of targets destroyed with static air tasking is statistically significant ($\alpha = 0.05$). Furthermore, the number of vehicles destroyed has increased 1.89% on average using dynamic air tasking. This difference between the average number of vehicles destroyed with dynamic air tasking and average number of vehicles destroyed with static air tasking is not statistically significant ($\alpha = 0.05$). Finally, mission completion time has decreased 20.62% on average using dynamic air tasking. This difference between the average mission duration time with dynamic air tasking and the average mission duration time with static air tasking is statistically significant ($\alpha = 0.05$). The results also indicate that the gains in larger scenarios are much more significant than the gains in the smaller scenarios with fewer vehicles, targets, and IADs.

Table 8 Simulation results for dynamic and static air tasking

Scenario	*Number of vehicles*	*Number of targets*	*Number of IADs*	*Number of targets destroyed with static air tasking*	*Number of targets destroyed with dynamic air tasking*	*Change in number of targets destroyed* (%)	*Number of vehicles destroyed with static air tasking*	*Number of vehicles destroyed with dynamic air tasking*	*Change in number of vehicles destroyed* (%)	*Mission completion with static air taking* (min)	*Mission completion with dynamic air tasking* (min)	*Change in mission completion time* (%)
1	2	2	2	9,462	9,978	+5.45*	14.00	15.00	+7.14*	25.10	21.25	–15.32*
2	3	3	3	9,376	9,972	+6.36*	19.00	19.00	+0.00**	35.65	29.95	–16.00*
3	4	4	4	9,295	9,965	+7.21*	28.00	29.00	+3.57*	59.91	49.73	–17.00*
4	5	5	5	9,267	9,953	+7.40*	45.00	46.00	+2.22**	104.19	85.13	–18.29*
5	6	6	6	9,211	9,944	+7.96*	56.00	57.00	+1.79**	157.15	125.96	–19.85*
6	7	7	7	9,168	9,936	+8.38*	63.00	64.00	+1.59**	217.59	170.50	–21.64*
7	8	8	8	9,103	9,925	+9.03*	72.00	74.00	+2.78**	321.12	245.22	–23.64*
8	9	9	9	9,067	9,911	+9.31*	87.00	87.00	+0.00**	416.09	308.76	–25.80*
9	10	10	10	9,022	9,903	+9.77*	93.00	95.00	+2.15**	556.42	400.17	–28.08*
Mean	6	6	6	9,219.00	9,943.00	+7.85*	53.00	54.00	+1.89**	210.36	159.63	–20.62*

*Significant ($\alpha = 0.05$).

**Not Significant ($\alpha = 0.05$).

5 Conclusion and future research directions

The current static vehicle-target assignment model does not allow aircrafts to use real-time information to self-organise and more effectively and efficiently engage targets. The air tasking model proposed in this study is dynamic and capable of considering the value of connectivity in a networked environment. Our model is dynamic, since the outcome of previous engagements and current position impact the future assignments; flexible, since it allows multiple roles over time for vehicles; and communicative, since it takes into account the value of connectivity and communication.

We have evaluated our dynamic air tasking model in a simulated network-centric battlespace. The results indicate that dynamic air tasking is considerably more effective and more efficient than static air tasking. We showed that there is no difference between the numbers of vehicles lost in dynamic air tasking vs. static air tasking. However, dynamic air tasking is considerably more effective and more efficient than static air tasking since more targets are destroyed in a shorter period of time.

The MCDA and network optimisation models presented in this study has wider applications in problems where multiple roles are dynamically assigned to multiple resources (i.e., rescue operations, law enforcement, and disaster recovery). Future work will expand this model to allow the simultaneous assignment of roles. In addition, further enhancements will partition the battlespace to only solve localised versions of the problem and then combine these results into a large-scale role allocation solution.

Acknowledgements

This research was supported by the US Air Force Research Laboratory grant number FA8750-07-2-0043. The authors are grateful to the Editor and anonymous reviewers for their valuable and constructive comments and suggestions.

References

Ahuja, R.K., Kumar, A., Jha, K. and Orlin, J.B. (2007) 'Exact and heuristic algorithms for the weapon-target assignment problem', *Operations Research*, Vol. 55, No. 6, pp.1136–1146.

Aurenhammer, F. and Klein, T.R. (2000) 'Voronoi diagrams', in Sack, J.R. and Urrutia, J. (Eds.): *Handbook of Computational Geometry*, North-Holland, Amsterdam, The Netherlands, pp.210–290.

Bailey, M.D., Tavana, M. and Busch, T.E. (2006) 'Communication role allocation for joint air operations in a network-centric environment', *International Journal of Computer Science and Network Security*, Vol. 6, pp.165–170.

Beard, R.W., McLain, T.W. and Goodrich, M.A. (2002) 'Coordinated target assignment and intercept for unmanned air vehicles', *IEEE Transaction on Robotics and Automation*, Vol. 18, pp.911–922.

Castanon, D.A. (1987) *Advanced Weapon-Target Assignment Algorithm Quarterly Report (TR-337)*, ALPHATECH Inc., Burlington, Massachusetts.

Chandler, P.R., Pachter, M., Nygard, K.E. and Swaroop, D. (2002) 'Cooperative control for target classification', in Murphey, R. and Pardalos, P.M. (Eds.): *Cooperative Control and Optimization*, Kluwer Academic Publishers, Boston, Massachusetts, pp.1–19.

Cullenbine, C.A., Gallagher, M.A. and Moore, J.T. (2003) 'Assigning nuclear weapons with reactive tabu search', *Military Operations Research*, Vol. 8, pp.57–69.

Darrah, M.A., Niland, W.M. and Stolarik, B.M. (2005) *Multiple UAV Dynamic Task Allocation Using Mixed Integer Linear Programming in a SEAD Mission*, Available at: http://www.isr.us/pdfs/publishedpapers/05_AIAA_Infotech.pdf

Flint, M. (2003) 'Stochastic models of cooperative autonomous UAV search problem', *Military Operations Research*, Vol. 8, pp.13–32.

Grant, K.E. (1993) 'Optimal resource allocation using genetic algorithms', *Naval Review*, Naval Research Laboratory, Washington DC, pp.174, 175.

Green, D.J., Moore, J.T. and Borsi, J.J. (1997) 'An integer solution heuristic for the arsenal exchange model (AEM)', *Military Operations Research*, Vol. 3, pp.5–15.

Hill, R.R., Miller, J.O. and McIntyre, G.A.M. (2001) 'Applications of discrete event simulation models to military problems', in Peters, B.A., Smith, J.S., Medeiros, D.J. and Rohrer, M.W. (Eds.): *Proceedings of the 2001 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Washington DC, pp780–788.

Kuhn, H.W. (1955) 'The Hungarian method for the assignment problem', *Naval Research Logistic Quarterly*, Vol. 2, pp.83–97.

Law, A.M. and Kelton, W.D. (2000) *Simulation Modeling and Analysis*, 3rd ed., McGraw-Hill, New York.

Lee, Z-J. and Lee, C-Y. (2003) 'A hybrid search algorithm of ant colony optimization and genetic algorithm applied to weapon-target assignment problems', in Liu, J., Cheung, Y. and Yin, H. (Eds.): *Intelligent Data Engineering and Automated Learning*, Springer, pp.278–285.

Lee, Z-J. and Lee, C-Y. (2005) 'A hybrid search algorithm with heuristics for resource allocation problem', *Informatics and Computer Science*, Vol. 173, pp.155–167.

Lee, Z-J., Lee, S-G. and Lee, C-Y. (2003) 'Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics', *IEEE Transactions on Systems Man and Cybernetics Part B*, Vol. 33, pp.113–121.

Miller, L.E. (2001) *Multihop Connectivity of Arbitrary Networks*, Available at: http://w3.antd.nist.gov/wctg/netanal/ConCalc.pdf

Munkres, J. (1957) 'Algorithms for assignment and transportation problems', *Journal of the Society for Industrial and Applied Mathematics*, Vol. 5, pp.32–38.

Murphey, R.A. (1999) 'Target-based weapon target assignment problems', in Pardalos, P.M. and Pitsoulis, L.S. (Eds.): *Nonlinear assignment problems: Algorithms and Applications*, Kluwer Academic Publishers, Boston, Massachusetts, pp.39–53.

O'Rourke, K.P., Bailey, T.G., Hill, R. and Carlton, W.B. (2000) 'Dynamic routing of unmanned aerial vehicles using reactive tabu search', *Military Operations Research*, Vol. 6, pp.33–42.

Pfeiffer, B., Batta, R., Klamroth, K. and Nagi, R. (2005) *Path Planning for UAVs in the Presence of Threat Zones using Probabilistic Modeling*, Available at: http://www.am.uni-erlangen.de/~pfeiffer/documents/prob120705.pdf

Rasmussen, S.J., Chandler, P.R., Mitchell, J.W., Schumacher, C.J. and Sparks, A.G. (2003) 'Optimal vs. heuristic assignment of cooperative autonomous unmanned air vehicles', *Proceedings of the 2003 AIAA Guidance, Navigation, and Control Conference*, pp.2003–5586.

Rasmussen, S.J., Schumacher, C.J. and Chandler, P.R. (2002) 'Investigation of single vs. multiple task tour assignments for UAV cooperative control', *AIAA Guidance Navigation and Control Conference and Exhibit*, August, Monterey, CA, AIAA 2002–4675.

Robinson, S. (2005) 'Discrete-event simulation: from the pioneers to the present, what next?', *Journal of the Operational Research Society*, Vol. 56, pp.619–629.

Schumacher, C., Chandler, P.R. and Rasmussen, S.J. (2001) 'Task allocation for wide area search munitions via network flow', *Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference*, AIAA 2001–4147.

Schumacher, C., Chandler, P.R. and Rasmussen, S.J. (2002) 'Task allocation for wide area search munitions via iterative network flow', *Proceedings of the 2002 AIAA Guidance, Navigation, and Control Conference*, AIAA 2002–4586.

Schumacher, C., Chandler, P.R., Pachter, M. and Pachter, L.S. (2007) 'Optimization of air vehicles operations using mixed-integer linear programming', *Journal of the Operational Research Society*, Vol. 58, pp.516–527.

Shannon, C. (1948) 'A mathematical theory of communication', *The Bell System Technical Journal*, Vol. 27, pp.379–423.

Tavana, M., Bailey, M.D. and Busch, T.E. (2008) 'A multi-criteria vehicle-target allocation assessment model for network-centric joint air operations', *International Journal of Operational Research*, Vol. 3, No. 3, pp.235–254.

Toth, P. and Vigo, D. (2002) 'The vehicle routing problem', *SIAM Monographs on Discrete Mathematics and Applications*, Vol. 9, SIAM, Philadelphia, Pennsylvania.

Voss, S. (1999) 'Heuristics for nonlinear assignment problems', in Pardalos, P.M. and Pitsoulis, L.S. (Eds.): *Nonlinear Assignment Problems: Algorithms and Applications*, Kluwer Academic Publishers, Boston, Massachusetts, pp.175–215.

Wachholder, E. (1989) 'A neural network-based optimization algorithm for the static weapon-target assignment problem', *ORSA Journal on Computing*, Vol. 4, pp.232–246.