Hierarchical Decision and Control for Cooperative Multi-UAV Systems using Ad-Hoc Communication[*]

Yosi Ben-Asher, Sharoni Feldman
Computer Science Department, Haifa University, Israel
{yosi, sharoni}@cs.haifa.ac.il

Pini Gurfil
Faculty of Aerospace Engineering
Technion - Israel Institute of Technology, Haifa 32000, Israel
pgurfil@technion.ac.il

Moran Feldman
Computer Science Department, Haifa University, Israel

Abstract

This works develops a novel hierarchical algorithm for task assignment (TA), coordination and communication of multiple UAVs engaging multiple targets and conceives an ad-hoc routing algorithm for synchronization of target lists utilizing a distributed computing topology. Assuming limited communication bandwidth and range, coordination of UAV motion is achieved by implementing a simple behavioral flocking algorithm utilizing a tree topology for target list routing. The TA algorithm is based on a graph-theoretic approach, in which a node locates all the detectable targets, identifies them and computes its distance to each target. The node then produces an attack plan that minimizes the sum of distances of the UAVs in the subtree of a given node to the targets. Simulation experiments show that the combination of flocking and TA algorithms gives the best performance. Clear-cut advantages of the TA algorithm are shown to exist in cases where the hit probability tends to one. An improvement of efficiency, representing the ratio between killed targets and the number of missile launches, is obtained for larger numbers of UAVs only if the engagement times are long enough, utilizing the improved coverage achieved by more UAVs.

[*] Provisional US Patent Filed.

1. Introduction

The problem of design, development and control of multi-agent systems has been studied in recent years for many applications. In particular, the use of systems consisting of multiple autonomous robots or unmanned aerial vehicles (UAVs) has been proposed in order to meet the requirements of complex missions [1]. Control, communication and decision support systems for UAVs constitute rapidly evolving research and development fields, as indicated by the Department of Defense UAV Roadmap 2002-2027 [2]. The use of groups of cooperating UAVs in order to perform various missions is currently studied throughout the world and is considered a main research goal by the United States Air Force Research Laboratory (AFRL) [3].

Using a cooperative group of UAVs for autonomously attaining a given goal requires that each agent assume a certain task at a given time. The global result of the group of UAVs acting together is the execution of a mission, e.g. *search and attack*, comprising the tasks of *search* (look for a suspected target), *identification* (determine that the suspected target is a legitimate target), *track* (update the target location) and *attack* (launch munitions).

Assigning multiple UAVs to perform these tasks according to their capabilities is a challenge that requires the development of specialized algorithms. These algorithms may be classified into two main types: *optimal* and *heuristic*. While optimal algorithms yield better results in terms of task assignment [11]-[13], they are usually more sensitive to system uncertainties, enemy behavior, and environment changes. Heuristic algorithms [14]-[19], on the other hand, are usually sub-optimal but more robust. In this work, we develop a heuristic task assignment (TA) algorithm, which is robust to changes in the UAV group properties.

An issue strongly related to cooperative UAV motion is *flocking* (also referred to as *formation flying*), which has been extensively studied in the last two decades [20]-[26], following the seminal work of Reynolds [9]. Reynolds simulated biological flocking behavior based on *cohesion* (agents converge onto a given point), *alignment* (velocity matching in order to move at a give direction), *collision avoidance* (preventing an agent from colliding with other agents), *obstacle avoidance* (steering the agents away from obstacles) and *migration* (path following). In this work, we implement Reynolds' behavioral flocking model in order to avoid collisions and improve mission execution by facilitating wide theater coverage.

In order to propagate the TA and flocking information, we develop a wireless ad-hoc communication media wherein the communication between the UAVs is constantly disrupted due to dynamic changes in the field (i.e., new UAVs join in or move out of communication range). Thus, the proposed algorithm for coordinating the UAVs must maintain its performances under the dynamic changes resulting from both the ad-hoc communication and the changes due to the movements of the UAVs and targets.

We use the Metrical Routing Algorithm (MRA) [27] for ad-hoc communication. This algorithm maintains communication by dynamically connecting the underling UAVs with a minimal set of rooted spanning trees (RSTs). The proposed algorithms for coordinating the UAVs uses the RST structure as a black-box building block. Thus, the dynamic communication details are masked out by the concomitant dynamics of the underlying set of RSTs. We thus mainly focus on coordination and decision support rather than on details related to the ad-hoc communication. The ad-hoc communication level is only reflected in the experimental part, where we study its effect on the performances of the UAVs.

Consequently, while most researchers assume a given system, propose a new control algorithm, and examine the algorithm's performance compared to other known algorithms, we take a novel approach and develop a hierarchical UAV decision and control system comprising *all* three layers: *flocking*, *communication* and *task assignment*. We subsequently consider a search and attack mission, where a group of autonomous armed UAVs is dispatched onto a theater environment in order to detect and attack time-critical targets. Targets are detected using a Ground Moving Target Indicator (GMTI) and are attacked once within the field-of-view (FOV) of an electro-optical (EO) payload. We evaluate the system's performance by examining *efficiency*, measured by the ratio between the number of killed targets and the number of munitions launched by the group of UAVs at a given time.

2. The Flocking Layer: Heuristic Relative UAV Control

This chapter discusses the heuristic control algorithm for UAV coordination based on Reynolds' flocking.

2.1 Preliminaries

Let us embark on our endeavor by first recalling a few basic concepts of graph theory, to be used in the sequel. The most basic definition of a *graph* is a pair $G=(V,E)$ of sets such that the elements of E are 2-element subsets of *V*. The elements of *V* are the *vertices*, and the elements of *E* are the *edges*. The *degree* of a vertex *v*, $d_G(v)$, is the number of edges at *v*. If all vertices of *G* have the same degree, then *G* is *regular*. A *path* is a non-empty graph $P=(V,E)$ of the form $V=\{v_1,v_2,\ldots,v_N\}$, $E=\{v_1v_2,v_2v_3,\ldots,v_{N-1}v_N\}$.

A *bipartite graph*, (or a *bigraph*), is a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent. A *directed graph* (or *digraph*) is a pair (V,E) of disjoint sets of vertices and edges together with two maps assigning to every edge ε an *initial vertex* $\mathrm{init}(\varepsilon)$ and a *terminal vertex* $\mathrm{ter}(\varepsilon)$. The edge ε is said to be directed from $\mathrm{init}(\varepsilon)$ to $\mathrm{ter}(\varepsilon)$. Consequently, a path is always directed.

If $P=v_1,\ldots,v_N$ is an acyclic directed path with $\mathrm{init}(v_{k-1}v_k)=v_{k-1}$, $\mathrm{ter}(v_{k-1}v_k)=v_k$, and $d_G(v_k)=1$, then the graph *P* is called a *path graph*; the path graph is a *tree*, which is an acyclic (directed) graph. A *rooted tree* is a tree in which a special (*labeled*) node is singled out. This node is called the *root* of the tree. A *subtree* is a tree G' whose graph vertices and graph edges form subsets of the graph vertices and graph edges of a given tree *G*. A *spanning tree* of a graph is a subset of $n-1$ edges that form a tree. A *child* is a node which is one graph edge further away from a given node, the *father*, in a rooted tree. Finally, a *leaf* of an unrooted tree is a node *v* of vertex degree 1, $d_G(v)=1$. Note that for a rooted tree, the root node is generally not considered a leaf node, whereas all other nodes of degree 1 are.

2.2 Reynolds' Behavioral Algorithm

The flocking algorithm in use is an implementation of Reynolds' behavioral algorithm [9]. Let $U_k \in G$ denote some UAV constituting a node in the graph G. U_k calculates its desired velocity as follows:

$$\mathbf{v}_d^k = \sum_{i=1}^{4} w_i^k \mathbf{v}_i^k \qquad \mathbf{v}_d^k \in ß^3, \tag{1}$$

where w is a constant scalar weight function, k is the UAV index and i is the algorithm law index, defined by the following *flocking rules*:

$i = 1$: *Cohesion*; commands the UAV to converge onto the center of the flock, computed by each UAV from the data communicated to it by the other UAVs. We denote the desired cohesion velocity for U_k by $\mathbf{v}_1^k$, and by $\mathbf{x}^k$ the position vector of U_k. The cohesion velocity command may be therefore written as

$$\mathbf{v}_1^k = \left\| \mathbf{v}^k \right\| \cdot \frac{\left(\mathbf{x}_{avg}^k - \mathbf{x}^k \right)}{R_{ref}}, \tag{2}$$

where $\|¥\|$ denotes the Euclidian vector norm, R_{ref} is a reference distance, usually related to the maximum payload detection range, representing the effective area of the UAV payload, and

$$\mathbf{x}_{avg}^k = \frac{\sum_{j=1}^{n} r_1^{jk} \mathbf{x}^j}{\sum_{j=1}^{n} r_1^{jk}}, \tag{3}$$

where r_1^{jk} is the cohesion rule weight for U_k relative to U_j, given by

$$r_1^{jk} = r_1(t, \mathbf{x}^k, \mathbf{x}^j). \tag{4}$$

In Eq. (3) and the subsequent equations, n denotes the number of nodes of some subtree $G' \in G$, and not necessarily the total number of UAVs, to be denoted by *N*.

Although r_1^{jk} may be time dependant, it is more likely that it would be directly dependant upon the relative position, increasing as the relative distance between the UAVs decreases, or remain constant.

$i = 2$:*Alignment*; matches the UAV's velocity vector to the mean velocity vector of the group. Alignment therefore attempts to steer the UAVs to fly in parallel to each other. We denote the desired alignment velocity for U_k by $\mathbf{v}_2^k$, and let r_2^{jk} be the alignment weight for U_k relative to U_j, so that

$$\mathbf{v}_2^k = \mathbf{v}_{avg}^k = \frac{\sum_{j=1}^{n} r_2^{jk} \mathbf{v}^j}{\sum_{j=1}^{n} r_2^{jk}} \tag{5}$$

and

$$r_2^{jk} = r_2(t, \mathbf{x}^k, \mathbf{x}^j). \tag{6}$$

Similarly to r_1, r_2 may be constant, time-dependant, or a function of the relative distance between U_k and U_j.

$i = 3$:*Collision avoidance*; restricts the UAV from colliding with its nearest neighbors. To that end, U_k calculates its desired collision avoidance velocity, $\mathbf{v}_3^k$, relative to the other UAVs according to the formula

$$\mathbf{v}_3^k = \left\| \mathbf{v}^k \right\| \frac{\sum_{j=1}^{n} r_3^{jk} \cdot (\mathbf{x}^j - \mathbf{x}^k) / R_{ref}}{\sum_{j=1}^{n} r_3^{jk}}, \quad \forall j \neq k, \tag{7}$$

where r_3^{jk} is the collision avoidance rule weight of U_k avoiding collision with U_j, and $\mathbf{x}^j$ and $\mathbf{x}^k$ are the position vectors of U_j and U_k, respectively. The weight function r_3^{jk} is likely to be dependant upon the relative distance between U_k and U_j, equaling 1 for the closest neighbor to UAV k and 0 for all other UAVs.

The desired velocity (1) is translated into acceleration using the following kinematic equation:

$$\mathbf{a}^k(t) = \frac{[(\mathbf{v}^k \times \mathbf{v}_d^k) \times \mathbf{v}^k]}{\left\|\mathbf{v}^k\right\|^2 \left\|\mathbf{v}_d^k\right\|} g\sqrt{(n_{max}^k)^2 - 1}, \tag{8}$$

where $\mathbf{v}^k \in$ **ß**^{3} and $\mathbf{v}_d^k \in$ **ß**^{3} are the current and desired velocity of U_k, respectively, and n_{max}^k is the maximal load factor of U_k. The term $\mathbf{v}^k \times \mathbf{v}_d^k$ in Eq. (8) yields a vector perpendicular to the plane defined by the velocity vectors $\mathbf{v}^k$ and $\mathbf{v}_d^k$. This perpendicular vector is then vector multiplied again by $\mathbf{v}^k$ to define the direction, perpendicular to $\mathbf{v}^k$, in which the UAV will accelerate in order to reach the desired velocity $\mathbf{v}_d^k$. The quotient defines a unit vector in the desired maneuver direction, and then multiplied by the UAV maximal load factor to give the maneuver magnitude. This acceleration is integrated into velocity and position using the kinematic model:

$$\begin{aligned} \mathbf{v}^k[t(i)+\Delta t] &= \mathbf{v}^k[t(i)] + \mathbf{a}^k[t(i)]\Delta t \\ \mathbf{x}^k[t(i)+\Delta t] &= \mathbf{x}^k[t(i)] + \mathbf{v}^k[t(i)]\Delta t \end{aligned} \tag{9}$$

Figure 1 depicts an example flocking scenario, implementing cohesion, alignment and collision avoidance.

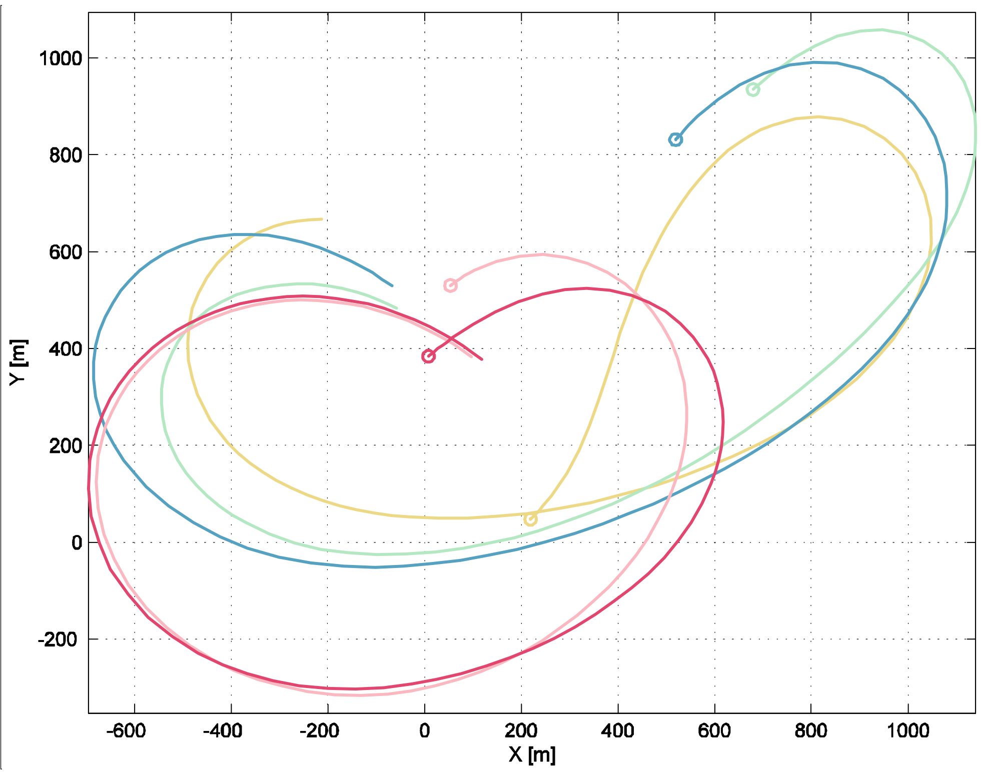


Figure 2: 5-UAV flocking

2.3 UAV Sensors for Target Detection

Every UAV is equipped with two types of sensors. A Ground Moving Target Indicator (GMTI) that detects vehicle movement and an Electro Optical (EO) sensor used to track the target and guide the missiles. The detection radius of the GMTI is larger than the detection radius of the EO sensor. Figure 3 depicts the field-of-view (FOV) of two UAVs with a radio connection and illustrates the sensor detection radius.

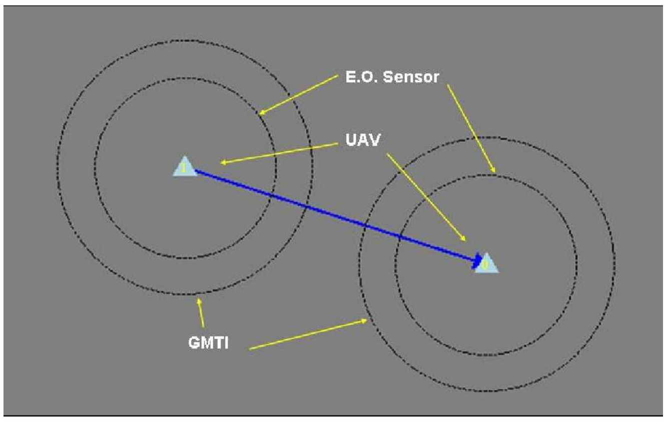


Figure 3: Field-of-view of the UAV sensors

3. The Communication Layer: Metrical Routing Algorithm

In this section, we describe the metrical routing algorithm (MRA), used as an ad-hoc communication protocol between the UAVs for communicating target list and flocking information.

3.1 The MRA Protocol

The MRA protocol presented herein is a hybrid ad-hoc protocol in the sense that some traffic control is used to maintain the mapping of the communicating nodes. The small overhead of the MRA protocol used to maintain the mapping is a worthy investment, as the MRA is capable of handling successfully a demanding traffic load under a high node density and fast node movement. The MRA organizes the nodes in rooted trees in order to find short session paths between nodes on the tree. The algorithm attempts to minimize the number of trees by fusing separate adjacent trees into a single tree. As long as any node in one tree is not in the transmission range of any node in the other trees, the trees will function autonomously. As soon as a radio connection is created between two nodes, the trees will be fused into a single tree.

All nodes run the same protocol implementing the MRA. As nodes emerge, disappear and move in or out of range of other nodes, there is need to update the trees. A primary goal of the algorithm is to identify these changes and adapt the tree structure to the new state. In the following discussion, we shall present an elaborate description of the MRA protocol, to be ultimately employed for a simulation study of the mobile ad-hoc networks routing performance.

The MRA algorithm organizes the nodes in the field in rooted trees. Only nodes that belong to the same tree can create sessions among themselves. To ensure maximal connectivity, all nodes will try to organize themselves in a single tree. Every node in the field has a unique node-ID (similar to a phone number or an IP address) and virtual coordinates that may change depending on the changes in the tree structure. Every tree is identified by a “tree name” which is the ID of the root node. Nodes periodically send beacons, termed *hello* messages. Every node that receives a beacon checks whether the node that sent the beacon belongs to a different tree. If the nodes belong to different trees, they initiate a fusion process that fuses the separate trees into a single tree. The fusion protocol should satisfy the following properties:

1. The protocol should not cause active sessions to break;
2. Eventually (assuming no dynamic changes occur) all trees with nodes within transmission range must fuse into a single tree;
3. When two trees are being fused, most updates should be made to the nodes of the smaller tree (in terms of the number of nodes);
4. The protocol should maximize the number of nodes that migrate from one tree to another in every step (yielding parallel fusion);
5. The protocol is fully distributed with no “central" bottlenecks, namely it is defined at the level of pairs of nodes.

Initially, every node forms a separate tree of size 1. Every node in the tree can autonomously migrate to a neighboring tree regardless of the node position in the tree. The migrating node gets new coordinates in its new tree according to the node’s new position. Naturally, when a node migrates from one tree to a new tree, it may carry along its neighboring nodes (since it belongs now to a bigger tree). In the macro view, the migration of the single nodes looks like a fusion of smaller trees into larger ones. Figure 4 presents two stages of the tree fusion process: The initial state and final organization to trees (assuming no significant node movements occurred during this process). Note that the two separate trees in Figure 4B cannot fuse because there are no two nodes within transmission range interconnecting the trees.

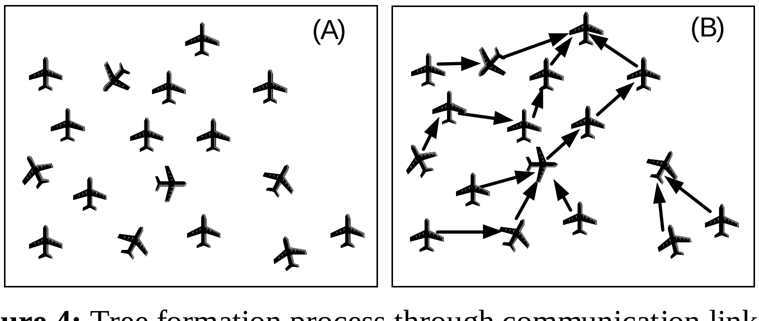


Figure 4: Tree formation process through communication links

The fusion process of two trees is parallel, that is, at any given time step multiple nodes of the smaller tree join the larger tree, as depicted by Figure 5. The implementation of the flocking and TA algorithms is based on the tree structure. Every tree runs these algorithms autonomously, as it cannot necessarily communicate with other trees. Existence of such communication will initiate a merge process that will ultimately result in a single tree.

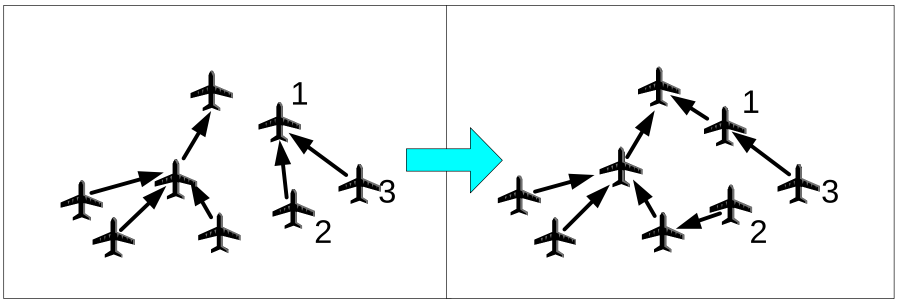


Figure 5: Fusion of trees. Parallel interconnection of nodes takes place at every given time step.

3.2 The Distributed Communication Topology

As mentioned above, the flocking algorithm controls the velocity and heading of the UAVs. However, each UAV communicates with its closest neighbors *only* and is unable to get a global view of the heading and velocity of the entire flock. The control information including the flocking data propagates from node to node using the tree management protocol.

The tree structure created by the MRA algorithm renders the root as the node that can gather data from its subtrees, calculate global variables and distribute the global variables back to its subtrees. We defined two data streams in the MRA protocol: *up stream*, moving from the tree leaves or subtrees towards the root; and *downstream*, moving data from the root towards the leaves.

The streams move stepwise from the father node to its children or vice-versa, as shown in Figure 6. Every sub-tree root (node *St* in Figure 6) gets the following values from its children (nodes *St*-1, *St*-2 … *St*-n in Figure 6): W_i – the weight of *St-i*, which is the number of nodes forming sub-tree *St-i*; and $\mathbf{x}_i$ – the average position vector of sub-tree *St-i*. In addition, let $\mathbf{x}_s$ be the position vector of node *St*. The average position vector that node *St* will send to its father is therefore given by

$$\mathbf{x}_{St} = \frac{\mathbf{x}_s + \sum_i W_i \mathbf{x}_i}{1 + \sum_i W_i} \tag{10}$$

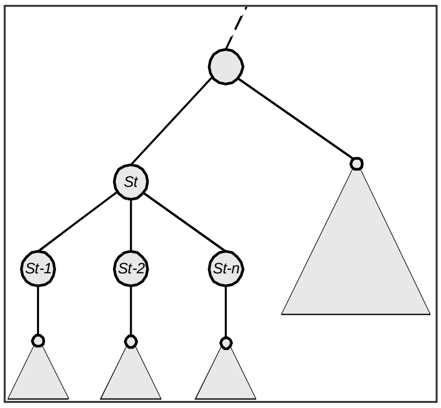


Figure 6: Subtree layers

The iterative process terminates at the root node. The root calculates its own flocking velocity, which is also the flocking velocity for the tree, and communicates it downstream. If the root node is lost, then a new root will be declared. The new root will then take over the tasks of the lost root. The up and down streams in our simulator propagate the information in both directions in a nominal rate of 30Hz.

4. The Task Assignment Layer: Target List Management

The TA layer relies on the arrangement of the UAVs in trees and on the inter-communication capabilities among the UAVs using ad-hoc routing. Every UAV is *autonomous*, performing autonomous decisions and behaving according to the changes in the theater. However, when a UAV becomes a node in a tree created by the MRA, it upgrades its behavior and acts as a member of a group.

In this section we consider the problem of computing a targeting plan for a set of moving agents $G = \{U_1, U_2, \ldots, U_N\}$ (UAVs in our case) attacking moving targets $A = \{T_1, T_2, \ldots, T_m\}$ (vehicles in our scenario). We focus on a distributed solution over a special setting where the communication among $U_1, U_2, \ldots, U_N$ is carried out by an ad-hoc network, as described in the previous section. Using ad-hoc communication yields a complex and challenging setting wherein the following factors should be considered:

- Ad-hoc communication implies that communication links among $U_1, U_2, \ldots, U_N$ are constantly changing. Thus, there is no guarantee that a given subset of *G* that was previously connected will remain connected.
- At any stage new information regarding (a) new targets, (b) changes in the location of known targets and (c) new U_i's that are closer to a given target can pop-up.
- It is desired not to fix a targeting plan (i.e., assign targets to each U_i) in advance, but rather adopt the reactive setting wherein at any time step only a portion of the targets are assigned to some subset $G' \in G$.
- Centralized algorithms where all the data (location of $U_1, U_2, \ldots, U_N$ and $T_1, T_2, \ldots, T_m$) is collected and then processed may fail to obtain good solutions due to disrupted communication and long communication delays.

The notion of an "optimal" solution in this setting is therefore ambiguous, as a few feasible solutions may be contemplated. To illustrate this observation, let d_{ij} denote a distance metric between some $U_i \in G$ and $T_j \in A$. Different bipartite graphs resulting from minimization of distinct performance measures can be considered:

1. GA_1, resulting from minimizing the maximum distance between $U_i \in G$ and $T_j \in A$, assuming that all targets are attacked by the UAVs (each UAV is assigned to a different target),

$$GA_1 : d_1^* = \min \max_{i \in G, j \in A} d_{ij} \tag{11}$$

2. GA_2, resulting from minimizing the sum of distances between $U_i \in G$ and $T_j \in A$, assuming again that all targets are attacked by the UAVs (each UAV is assigned to a different target),

$$GA_2 : d_2^* = \sum_{i \in G, j \in A} d_{ij} \tag{12}$$

In certain scenarios, one may choose to attack only a subset of all targets, $A' \in A$. In this case Eqs. (11) and (12) should be modified into

$$GA'_1 : d_1^{*}{}' = \min \max_{i \in G,\, j \in A'} d_{ij} \tag{13}$$

and

$$GA'_2 : d_2^{*}{}' = \sum_{i \in G,\, j \in A'} d_{ij}\ , \tag{14}$$

respectively.

Consider, for example, the bipartite graph of Figure 7, showing the distance metrics between four UAVs (U_1, U_2, U_3, U_4) and targets (T_1, T_2, T_3, T_4). The optimal solution in the sense of Eq. (11) is $GA_1 = \{U_1 \to T_2, U_2 \to T_1, U_3 \to T_3, U_4 \to T_4\}$ with $d_1^{*} = 10,\ d_2 = 29$; the optimal solution in the sense of Eq. (12) is $GA_2 = \{U_i \to T_i,\ i = 1, \mathbf{h}, 4\}$, with $d_1 = 12,\ d_2^{*} = 27$; and the optimal solution in the sense of Eq. (13) is $GA_1' = \{U_1 \to T_1, U_2 \to T_3, U_3 \to T_4\}$, with $d_1^{*} = 9,\ d_2 = 16$. In the last case, targeting T_2 can be determined at the next stage/step where possibly there will be new distance metrics to choose from.

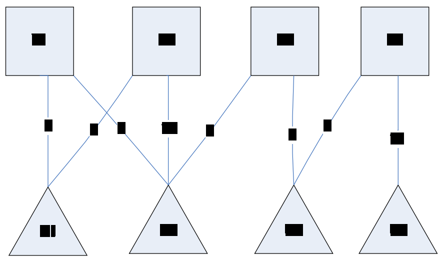


Figure 7: A bipartite graph showing distances of UAVs and targets

We hereby suggest a novel TA algorithm, which is supported by the ad-hoc communication protocol. In this algorithm, target assignments are communicated among the UAVs using the MRA protocol described above. Unlike other ad-hoc routing algorithms, the MRA attempts to connect $G = \{U_1, U_2, \mathbf{h}, U_N\}$ (or a subtree thereof) by a minimal set of rooted trees that preserves geographical distances, namely distances on the rooted trees are usually proportional to the distances of $U_1, U_2, \mathbf{h}, U_N$ in the given engagement theater.

More formally, let $G(t)$ be the graph at time *t* wherein each two nodes U_i, U_j that can communicate have an edge in $G(t)$. The MRA algorithm attempts to cover $G(t)$ by a minimal set of spanning trees. These rooted trees can be naturally used for both distributed computing (of, e.g., the flocking layer) as well as for communication, in addition to propagation and computation of the TA layer.

The proposed TA algorithm for $U_i \in G$ using the MRA protocol can be thus summarized as follows:

1. Each node U_i in a tree (or a subtree) locates all the detectable targets, identifies them and computes its distance to each target. The target ID is the target location. Note that computing a unique target ID is not always straightforward, since it may require fusion of the target location taken by several UAVs in adjacent time steps and locations.
2. At each time step $t(i)$, a node *v* constructs a weighted bipartite graph $B_v[t(i)]$ representing the distances between each U_i and T_j related to the subtree rooted at *v*. There are three events that lead to the creation of a new bipartite graph $B_v[t(i+1)]$:
   - A new $B_u[t(i+1)]$ is received from one of *v*'s children.
   - A new $B_{Fv}[t(i+1)]$ is received from *v*'s father.
   - There is a change in the target list **L** of *v*, i.e., *v* detects a new target or an old target disappears or destroyed.

   In each of these events, a new $B_v[t(i+1)]$ is computed by merging $B_u[t(i+1)]$ or $B_{Fv}[t(i+1)]$ or **L** into $B_v[t(i+1)]$.

3. The node v computes a minimum weighted matching $M_v[t(i+1)]$ of $B_v[t(i+1)]$ obtaining an attack plan that minimizes the sum of distances of the UAVs in the subtree of v to their targets,

$$d^* = \sum_{i,j \in B_v[t(i+1)]} d_{ij} \tag{15}$$

4. $B_v[t(i+1)]$ is sent to the father F_v of v and $M_v[t(i+1)]$ is sent to all the children of v.
5. When a node v receives an attack plan $M_{Fv}[t(i+1)]$ from its father, it checks to see if it is assigned a new target; if so, it "leaves" its current target and starts to engage the recommended target.
6. The attack plan $M_{Fv}[t(i+1)]$ is sent to all the children of the current node.

Note that in case a target is destroyed or disappears, it will be removed from each $B_v(t)$, since these are propagated only up the MRA trees.

The implementation of the target selection algorithm uses a single data structure to transfer the bipartite graph $B_v(t)$ and the attack graph $M_v[t(i+1)]$. This data structure is the *Target List* (TL). A simplified TL model is depicted by Figure 8. This model ignores the parallelism in the TL flow between the UAVs. It explains the decision-making process and the decision overruling performed by higher levels of UAVs in the tree with broader views.

Figure 8A presents the initial phase where U_4 and U_5 have detected target T_1, U_6 and U_7 have detected T_2, and U_8 detected T_3. Every UAV that has one or more targets will *autonomously* select a target from the possible targets in its TL and will commence a pursuit. The current state depicted in Figure 8A is that U_4 and U_5 are in pursuit after T_1, U_6 and U_7 prosecute T_2 and U_8 will pursuit T_3. The pursuit process of the UAVs is independent of other UAV activities. Note that this is the initial phase, where the targets were detected by the GMTI detector but are not yet within the range of the UAV missiles launch distance (i.e. within the FOV of the EO payload). Every UAV stores a TL comprising all targets known to the UAV and indications on the target state. Every UAV then sends its TL to its father and children. U_4, U_5 and U_6, constituting leaves in the tree, send their TLs toward node 1, which is the subtree father. The decision taken by U_1 arrives to U_4, which is also the UAV attacking T_1. U_4 continues its attack while U_5 receives the same TL from its father, and finds out that it should abort its attack on T_1. U_5 will look for an alternative target without an owner in the TL that is within its GMTI range, or will search for a new target that might emerge.

Figure 8B presents a situation in which U_1 had analyzed the TLs and decided that sub-tree *D* will be responsible to attack T_1, sub-tree *E* will not attack T_1 and sub-tree *F* will attack T_2. The decisions of U_1 are sent to its father node 0 and its children. A similar process takes place in the other parts of the tree.

In Figure 8C, the root distributes the results of its decisions to its children. The decisions are embedded in its TL. The decision of U_0 is that subtree *C* will assume the responsibility to attack T_2, while subtree *A* will abandon its attack. These decisions will be distributed by every subtree towards its children until they reach the leaves. In the meantime, U_6 and U_7 continue their pursuit after T_2.

Figure 8D presents a situation where the root decisions arrived to the attacking UAVs and U_6 stopped its attack on T_2 while U_7 continues its attack. The upstream and downstream flow of TLs is not affected by changes in the tree structure or by appearance of new targets.

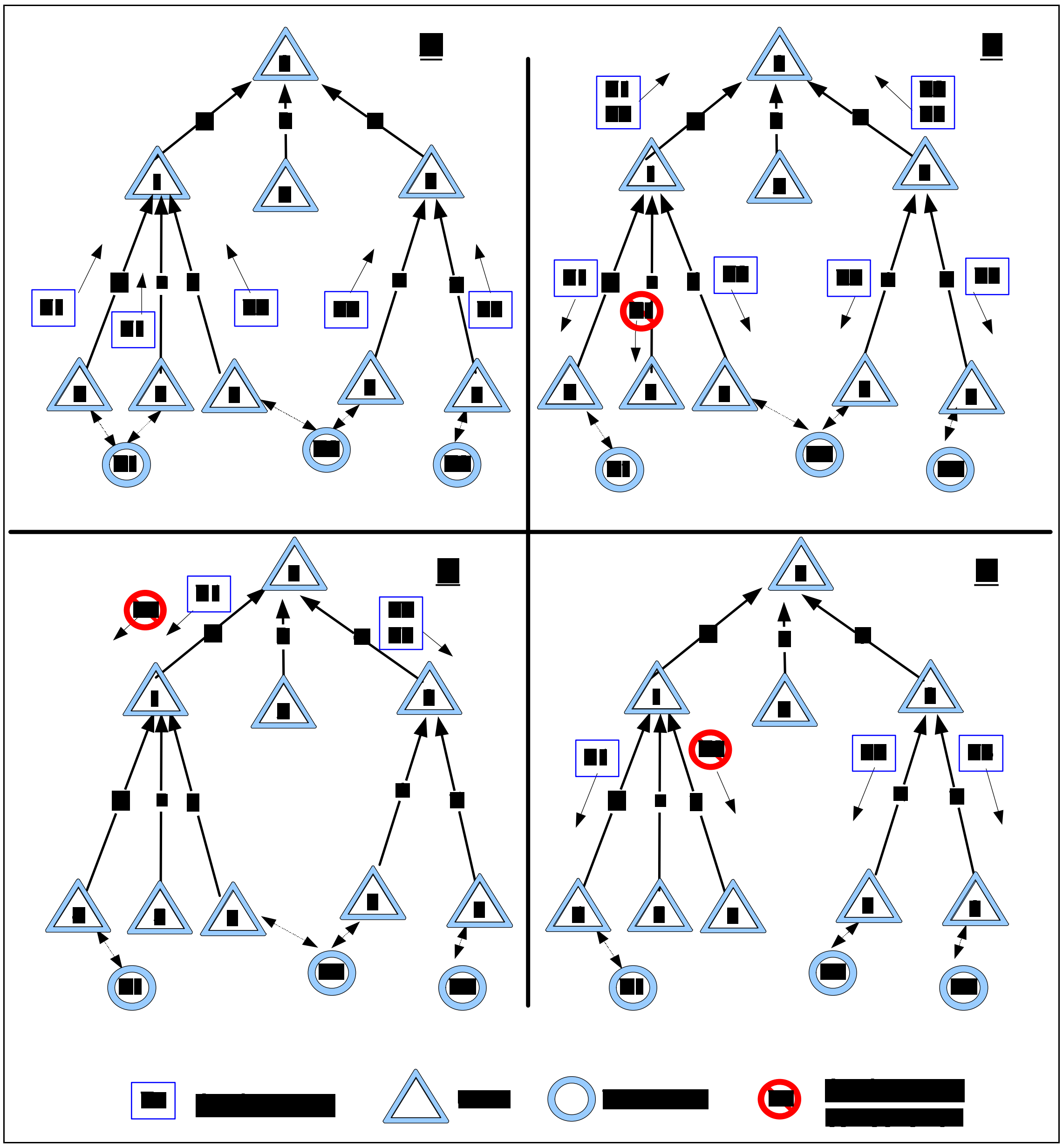


Figure 8: Target List flow

In addition to the data describing the relations between UAVs and potential targets, the TL is used to transfer data regarding missiles that are in the course of an interception process. The distribution of this data is essential to minimize the probability that a UAV launches its last missile towards a target and declares itself as an empty UAV. A second UAV will interpret this situation as a “permission” to launch again a missile to the same target. The exchange of data regarding flying missiles will prevent the second UAV from launching a new missile to the same target until it gets a message that the missile failed to hit the target.

5. Simulation and Visualization

5.1 Theater Representation

Figure 9 presents a snapshot of the theater as created by the simulator. The UAVs are identified by their position in the tree, where R is the root, R.1 is one of the children of the root and R.1.1 is a child of R.1. Figure 9A presents the detection footprint of R.1.1 while Figure 9B presents the detection footprint of R.1.

R.1.1 in Figure 9A detected two potential targets. One of the targets, marked by a cross, was selected as the target to be attacked and this is the 1st priority target for this UAV. R.1 in Figure 9B also detected two targets, where one of the targets is observed by both UAVs. The target marked with a cross was selected as the target to be attacked. Note that this UAV is attacking the farthest target and not the close one. This decision was taken after analyzing the velocities and headings of the entities participating in this pursuit or by a decision of the upper layer in the tree (node R).

Figure 10 presents the TL of R.1. This TL stores the list of all potential targets observed by the UAVs forming the tree. The *enemy list* which is a part of the Target List contains the current coordinates of every target. These coordinates are updated regularly as the targets move. The rightmost column of the TL lists the UAVs that observe the target selected from the enemy list. The target is now under pursuit by R.1, which has the status *true* while R.1.1 has the status *false.* This target appears in Figure 9B as the target marked by a cross.

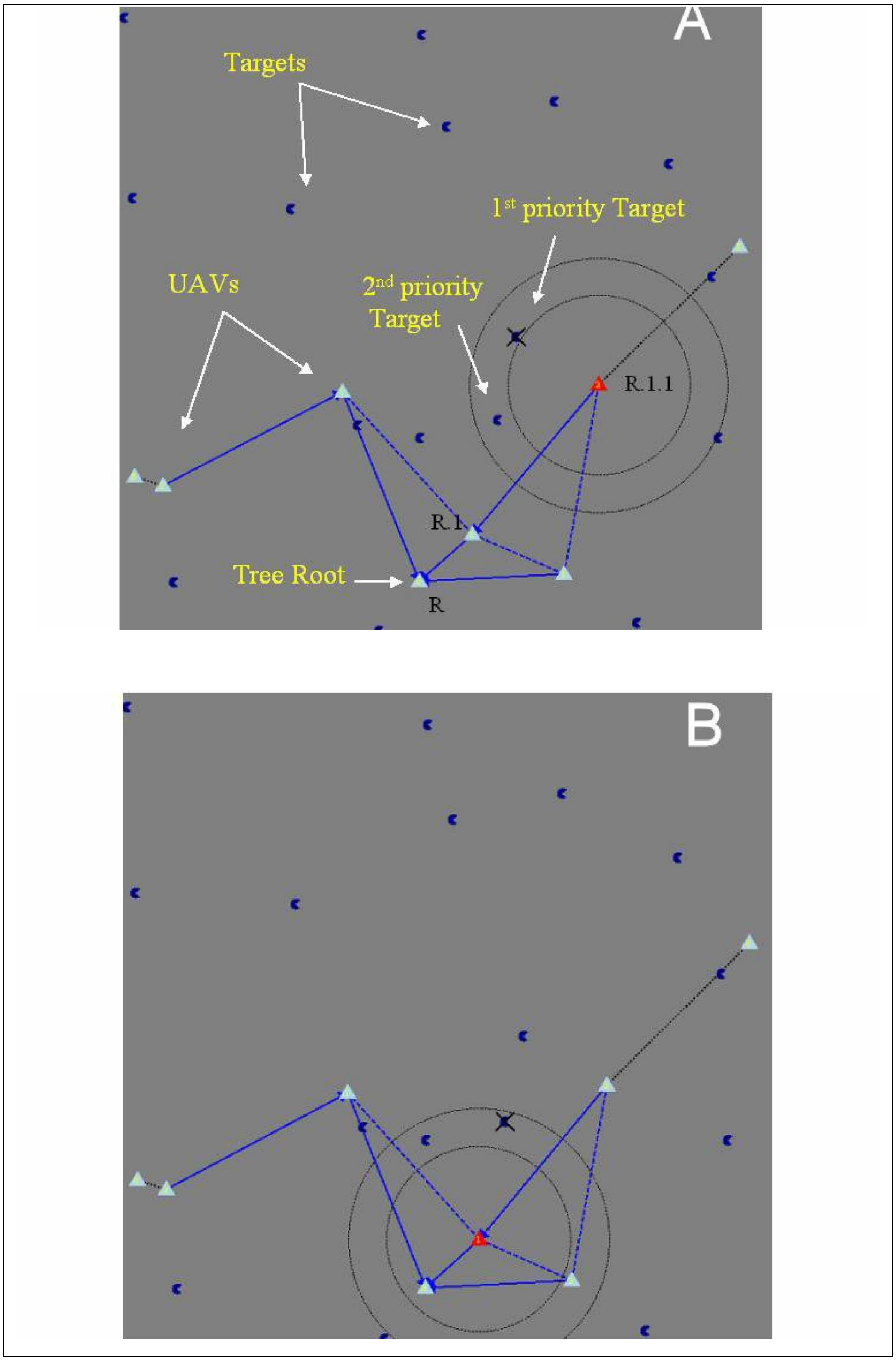
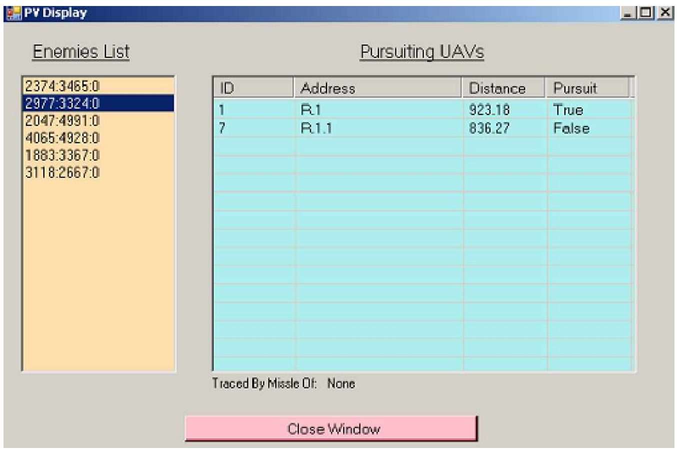


Figure 9: Theater view during engagement



2374:3465:0
2977:3324:0
2047:4991:0
4065:4928:0
1883:3367:0
3118:2667:0

ID	Address	Distance	Pursuit
1	R.1	923.18	True
7	R.1.1	836.27	False

Traced By Missle Of: None

Figure 10: Target List of UAV R.1

5. 2 Synchronization of Target Lists

A global theater view requires that the UAVs synchronize their TLs in a high rate. The synchronization rate of the TLs in the following simulations is performed nominally at 30 Hz. This synchronization rate ensures an effective TA process, yielding the following merits:

1. Better utilization of the limited number of missiles. The missiles are a limited resource and the TA algorithm prevents a simultaneous launch of two or more missiles towards a single target. Figure 11 presents a typical case when the TL is not synchronized. The TA algorithm does not have a global picture of the theater, viz. every UAV autonomously handles its own tasks. As a result, two UAVs simultaneously launch missiles at a single target.

2. Reducing the total interception time by a better task-sharing among UAVs. A UAV can handle only a single interception at a given time. By eliminating the multiple launching of missiles, the UAVs that are not engaged with an interception can launch missiles towards other targets.

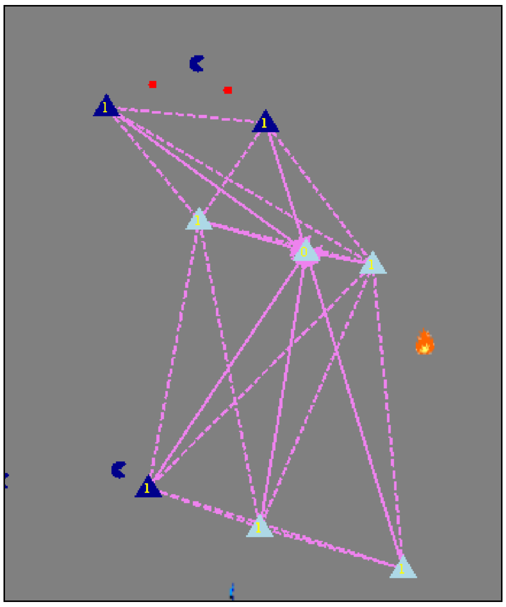


Figure 11: Absence of TL synchronization induces launch of multiple missiles (dots) at a single target (“packman” symbol, upper part) by two UAVs (triangles)

6. Experimental Results

The simulation experiments are aimed at evaluating the contribution of the flocking and TA algorithms to the performance of the UAVs using MRA-based ad-hoc communication.

6.1 Main Experiments and Simulation Models and Parameters

The main experiments comprise the following benchmarks:

1. Reference Monte-Carlo simulations performed without employing the flocking and TA algorithms;
2. Monte-Carlo simulations used to evaluate the contribution of the flocking algorithm and TA algorithm separately; and
3. Combined Monte-Carlo tests where both algorithms are employed simultaneously.

Each simulation run in Cases 1-3 has been performed assuming two different models for the missile hit probability, p_h. The first model assumes that p_h depends upon the initial missile-target range, as detailed by Table 1. The second model, used for reference, assumes that $p_h = 1$ within some operational initial missile-target range.

Initial missile-target range [m]	0 - 200	201 - 500	501- 4000	4001 - 5000
Hit probability, p_h	0	0.4	0.95	0.4

Table 1: Missile interception probability model

All simulations were performed using two time intervals - $t_f = 60$ seconds and $t_f = 120$ seconds. The 60-second simulations differ significantly from the 120-second simulations as the interception process in the first 60 seconds demands a very short target interception time. The 120-second case enables to pursue targets that are initially far from the UAVs and are not within the initial interception range. Additional simulation parameters are listed in Table 2.

Theater dimensions	5Km x 5Km
UAV initial speed	100Km/h - 120Km/h (uniform distribution)
UAV radio transmission Range	1.7 Km
Target speed	50Km/h - 80Km/h (uniform distribution)
No. of targets	16
No. of UAVs	8-16
Missiles	Fixed value of 16. In case of 8 UAVs, every UAV carries 2 missiles. In case of 16 UAVs, every UAV carries 1 missile. For all other cases, every UAV carries randomly 1 or 2 missiles.
TL synchronization rate	30 Hz

Table 2: Main simulations parameters

The main performance evaluation measure is the *efficiency*,η , defined as the ratio between the number of successful hits and the number of launched missiles *for a given UAV group size,* and calculated as the ensemble average over 50 Monte-Carlo runs. The mean efficiency, $\bar{\eta}$, is defined as the average of η *over all UAV group sizes.*

Table 3 summarizes the Monte-Carlo simulation batches used to evaluate the overall performance of the system.

Simulation Batch	Flocking Algorithm	Task Assignment Algorithm	Hit probability	Time (seconds)
Simulation Batch 1	active	active	1	60
	active	not active		
	not active	active		
	not active	not active		
Simulation Batch 2	active	active	< 1	
	active	not active		
	not active	active		
	not active	not active		
Simulation Batch 3	active	active	1	120
	active	not active		
	not active	active		
	not active	not active		
Simulation Batch 4	active	active	< 1	
	active	not active		
	not active	active		
	not active	not active		

Table 3: Simulations batches

6. 2 Simulation Results

The results of the Monte-Carlo simulations are elaborated herein.

6.2.1 Simulation Batch 1

Figure 12 depicts the results for Simulation Batch 1. In this figure, as well as in Figs. 14-16, the *x*-axis is the UAV group size and the *y*-axis is the efficiency, η . Each point on this graph and the following graphs represents an ensemble average of 50 Monte-Carlo runs.

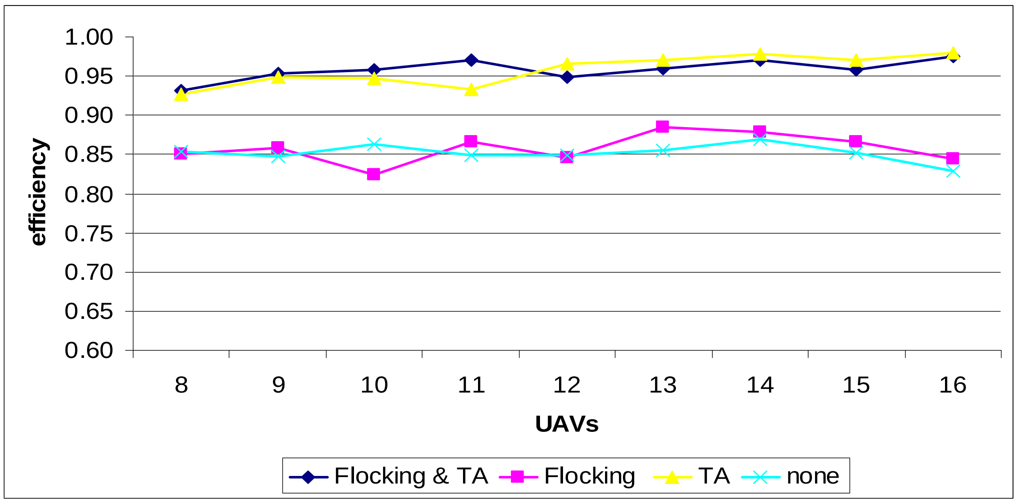


Figure 12: Monte-Carlo simulation results for a $t_f = 60$ sec and $p_h = 1$

The results exhibit two distinct regions. The upper lines result from the simulations in which TA is used with and without flocking. The lower lines are obtained without incorporating the TA algorithm. Using the TA algorithm has a significant contribution to efficiency, as $\bar{\eta}$ increases from 0.86 without TA to 0.96 with TA. In addition, η increases moderately with the number of UAVs when the TA algorithm is used. Note, however, that $\eta, \bar{\eta} < 1$ although in the current batch $p_h = 1$. This stems from the following reasons:

1. The existence of separate trees of UAVs that cannot merge into a single tree. UAVs that belong to separate trees independently launch missiles at a single target. Figure 13 illustrates a case where two UAVs, which are not within communication distance and hence are unaware of each other, initiate multiple attacks on a single target.

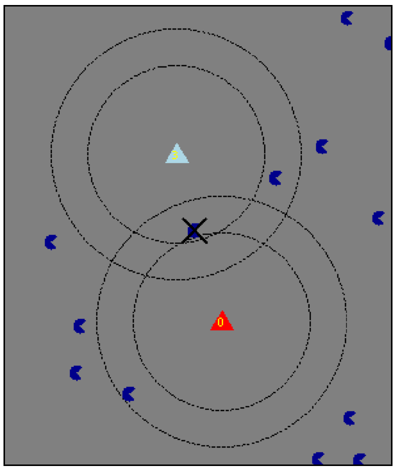


Figure 13: Multiple attacks on a single target (crossed) by multiple UAVs (triangles) result in efficiency degradation

   This situation is possible if the transmission range is smaller than twice the GMTI sensor detection distance.

2. Communication breaks between UAVs that belong to one tree. These breaks can lead to a situation wherein the TL is momentarily desynchronized. In this case, multiple launches towards a single target are possible.

3. TL synchronization requires an adequate bandwidth for communicating the TLs and the flocking information between the nodes in addition to the data needed to

handle the trees by the MRA protocol. The size of an average TL message is 850 bytes. In addition, a temporal burst of messages may cause some messages to be lost due to queue limitations.

4. Initially, the UAVs and targets are distributed randomly in the theater. Some of the targets initially fall inside the detection area of multiple UAVs. The UAVs respond by launching missiles towards the detected targets. Due to the delay in building the trees and synchronizing the TLs, multiple missiles are launched towards targets that are detected by multiple UAVs. In order to mitigate this phenomenon we added a concomitant time delay enabling initial TL synchronization after which the UAVs are allowed to launch missiles.

6.2.2 Simulation Batch 2

Figure 14 presents the results for Simulation Batch 2, where p_h is a function of the initial missile-target range (cf. Table 1).

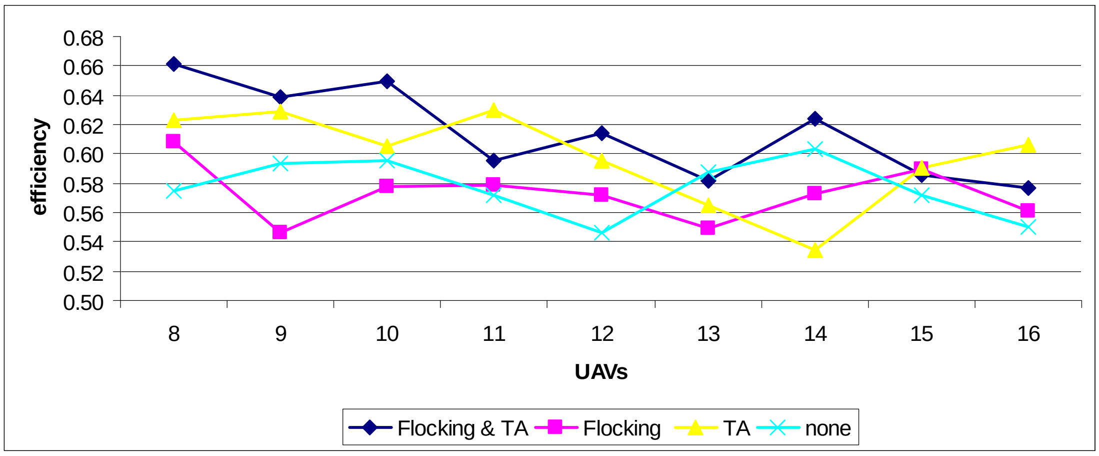


Figure 14: Monte-Carlo simulation results for a $t_f = 60$ sec and $p_h < 1$

Calculation of the average efficiency for the four combination yields $\bar{\eta}=0.62$ for flocking and TA, $\bar{\eta}=0.60$ for TA only, $\bar{\eta}=0.58$ for no flocking and no TA, and $\bar{\eta}=0.57$ for flocking only. Thus, the case $p_h < 1$ is significantly different from the ideal hit probability case. In particular (a) η decreases when the number the number of UAVs increases from 8 to 16; (b) The clear distinction between the different cases deteriorates as the number of UAVs in the field grows; (c) The combination of the TA algorithm and the flocking algorithm yields the best results; and (d) The use of the flocking algorithm only gives the worst results.

6.2.3 Simulation Batch 3

Figure 15 presents the results for $t_f = 120$ sec and $p_h = 1$.

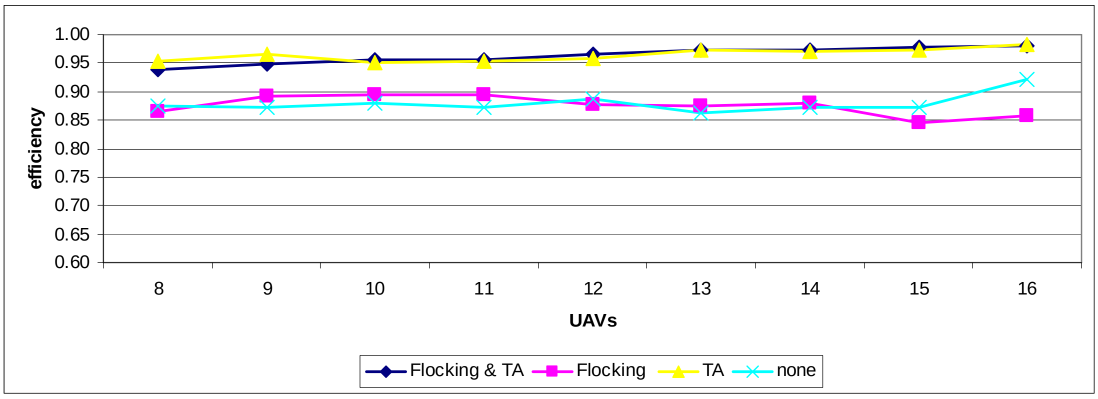


Figure 15: Monte-Carlo simulation results for a $t_f = 120$ sec and $p_h = 1$

Simulation Batch 3 yields results similar to Simulation Batch 1. Again, the distinction between the simulations that use the TA algorithm and simulations that do not use the TA algorithm is clear. In contrast to the results of Simulation Batch 1, however, the gap between the TA-based simulations and the non-TA simulations increases as the number of UAVs increases from 8 to 16, so TA-based interceptions improve with more UAVs and no-TA-based interceptions worsen as the number of UAV grows. The combination of flocking and TA gives again the best results; however, the incorporation of flocking into TA-driven UAVs yields only a miniscule improvement in η .

6.2.4 Simulation Batch 4

Figure 16 presents the results for $t_f = 120$ sec and $p_h < 1$ (cf. Table 1 for the probability model).

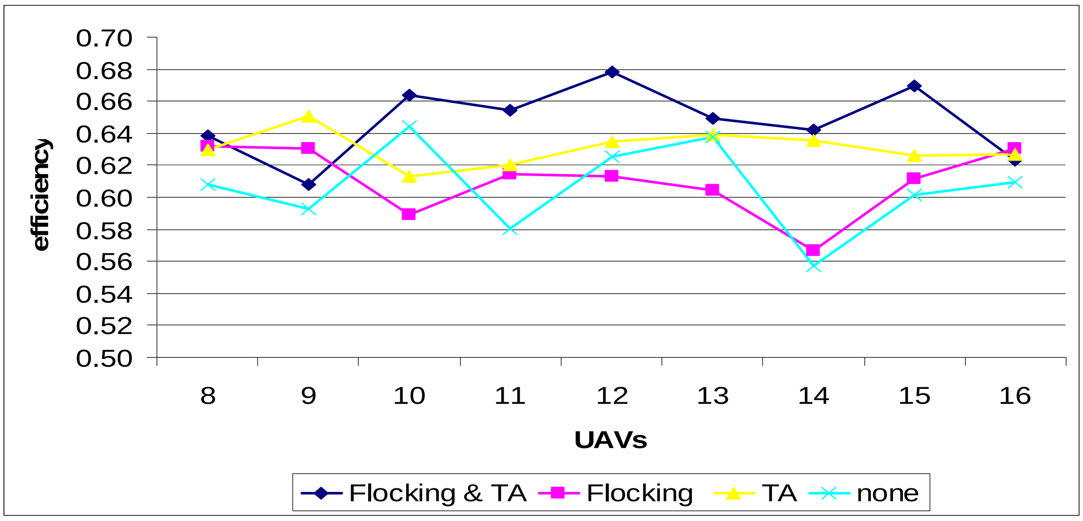


Figure 16: Monte-Carlo simulation results for a $t_f = 120$ sec and $p_h < 1$

Similarly to Simulation Batch 2, the incorporation of $p_h < 1$ dramatically changed the results. Here, the use of both the flocking and TA algorithms clearly and distinctively yields the best results with $\bar{\eta} = 0.65$. The use of TA only yields $\bar{\eta} = 0.63$, and the use of flocking only or neither algorithms results yields $\bar{\eta} = 0.61$.

7. Conclusions

We developed a hierarchical algorithm for task assignment, coordination and communication of multiple UAVs engaging multiple targets in an arbitrary theater, and implemented an ad-hoc routing algorithm for synchronization of target lists and ego-motion information based on distributed communication and computing theory.

Assuming limited communication bandwidth and limited communication range, coordination of UAV motion was achieved by implementing a simple behavioral flocking algorithm utilizing a tree topology for target list routing. This algorithm achieved feasible theater coverage while preventing collisions and enabling coordinated motion of multiple UAVs. The heuristic-reactive nature of the flocking algorithm reduces computational complexity and is robust to initial uncertainties in target location and theater characteristics. We conclude that it may be sufficient to implement low-level flocking on distributed UAV systems, and doubt the need for optimal relative position control and optimal coverage algorithms.

The task assignment algorithm was developed based on a graph-theoretic approach. In this algorithm, a node in a tree (or a subtree) locates all the detectable targets, identifies them and computes its distance to each target. At each time step, a node constructs a weighted bipartite graph representing the distances between each UAV and target the subtree rooted at the node. The node then computes a minimum weighted matching obtaining an attack plan that minimizes the sum of distances of the UAVs in the subtree of a given node to the targets. Although the task algorithm is not optimal, it was shown to be ideally suited for routing in ad-hoc networks with limited communication range.

Our simulations experiments raise a number of important conclusions. First, we conclude that the embedding of flocking and task assignment gives the best performance, which is much improved relative to the performance obtained with flocking only, and somewhat improved compared to the case of task assignment only. Clear advantages of the task assignment algorithms have been shown to exist in cases where the hit probability tends to one. An improvement of efficiency, representing the ratio between killed targets and the number of missile launches, is obtained for larger numbers of UAVs only if the engagement times are long enough, enabling utilization of the better coverage achieved by more UAVs. This improvement is possible only if the UAVs are capable of task collaboration and target list exchange. Otherwise, increasing the number of UAVs will result in a worse performance.

Refernces

[1] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, pp. 375–397, 1996.

[2] Department of Defence, *UAV Roadmap 2002-2027*, 2002.

[3] S. S. Banda, "Future directions in control for unmanned aerial vehicles," Slides, April 2002, available at http://www.cds.caltech.edu/ murray/cdspanel/fdc-apr02/banda-26apr02.pdf.

[4] M. J. Mataric, "Designing and understanding adaptive group behaviors," *Adaptive Behavior*, vol. 4, pp. 51–80, 1995.

[5] J. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Workshop on Algorithmic Foundations of Robotics (WAFR94')*, 1994.

[6] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," *Proceedings of the NATO Advanced Workshop on Robotics and Biological Systems*, 1989.

[7] M. J. Mataric, "Behavior based control: Examples from navigation, learning, and group behavior," *Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Physical Agents*, vol. 9, 1997, edited by H. Hexmoor et al.

[8] L. Steels, "The artificial life roots of artificial intelligence," *Artificial Life*, vol. 1, pp. 75–110, 1994.

[9] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, pp. 25–34, 1987.

[10] J. M. Usher and Y.-C. Wang, "Intelligent agent architectures for manufacturing control," in *Industrial Engineering Research Conference 2000*, 2000.

[11] K. Passino, M. Polycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint, and M. Baum, "Cooperative control for autonomous air vehicles," *Proceedings of the Cooperative Control Workshop, Florida*, December 2000.

[12] C. Schumacher, P. Chandler, and M. Pachter, "UAV task assignment with timing constraints," *AFRL-VA-WP-TP-2003-315*, 2003, united States Air Force Research Laboratory.

[13] S. Rasmussen, P. Chandler, J. W. Mitchell, C. Schumacher, and A. Sparks, “Optimal vs. heuristic assignment of cooperative autonomous unmanned air vehicles,” *Proceedings of the AIAA Guidance, Navigation & Control Conference*, 2003.

[14] H. V. D. Parunak, M. Purcell, and R. O’Connell, “Digital pheromones for autonomous coordination of swarming uavs,” *Proceedings of the AIAA’s First Technical Conference and Workshop on Unmanned Aerospace Vehicles*, vol. 3446, 2002.

[15] C. A. Lua, K. Altenburg, and K. E. Nygard, “Synchronized multi-point attack by autonomous reactive vehicles with simple local communication,” in *IEEE Swarm Intelligence Symposium*, April 2003.

[16] P. R. Chandler, M. Pachter, D. Swaroop, J. M. Fowler, J. K. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard, “Complexity in uav cooperative control,” in *Proceedings of the American Control Conference*, May 2002, pp. 1831–1836.

[17] W. Guo and K. Nygard, “Combinatorial trading mechanism for task allocation,” in *Proceedings of the 13th International Conference on Computer Applications in Industry and Engineering*, June 2001.

[18] J. W. Schlecht, “Mission planning for unmanned air vehicles using emergent behavior techniques,” McNair Scholars Day, NDSU, Fargo, ND, April 2001.

[19] C. T. Cunningham and R. S. Roberts, “An adaptive path planning algorithm for cooperating unmanned air vehicles,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, South Korea, May 2001.

[20] J. Toner and T. Yuhai, “Flocks, herds, and schools: A quantitative theory of flocking,” *Physical Review E*, vol. 58, Number 4, pp. 4828–4858, 1998.

[21] B. Crowther, “Rule-based guidance for flight vehicle flocking,” *Proceedings of the I MECH E Part G Journal of Aerospace Engineering*, vol. 218, no. 2, pp. 111–124(14), April 2004.

[22] ——, “Flocking of autonomous unmanned air vehicles,” *Aeronautical Journal*, vol. 107, no. 1068, pp. 99–110, February 2003.

[23] A. Czirok, M. Vicsek, and T. Vicsek, “Collective motion of organisms in three dimensions,” *Physica A*, vol. 264, pp. 299–304, 1999.

[24] T. Balch and R. C. Arkin, “Behavior-based formation control for multiagent robot teams,” in *IEEE Transactions on Robotics and Automation*, vol. 14(6). IEEE, December 1998, pp. 926–939.

[25] B. O. Burchan, J.-M. Lien, and N. M. Amato, “Better flocking behaviors in complex environments using global roadmaps,” Texas A&M University, Technical Report TR02-003, May 2002.

[26] A. Richards, J. Bellingham, M. Tillerson, and J. P. How, “Coordination and control of multiple UAVs,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, no. 2002–4588, Monterey, CA, 2002.

[27] Y. Ben-Asher, S. Feldman, “Ad-hoc routing using virtual coordinates-based on rooted trees”, *http://cs.haifa.ac.il/YOSI/papers.html*