JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 5, August 2008

Hierarchical Task Assignment and Communication Algorithms for Unmanned Aerial Vehicle Flocks

Yosi Ben-Asher* and Sharoni Feldman†
Haifa University, Israel

Pini Gurfil‡
Technion—Israel Institute of Technology, Haifa 32000, Israel

and

Moran Feldman§
Technion—Israel Institute of Technology, Haifa 32000, Israel

DOI: 10.2514/1.32957

This work develops distributed hierarchical task assignments and communication algorithms for flocks of unmanned aerial vehicles dispersed in an unknown theater while engaging multiple moving targets. The dynamical changes require distributed algorithms, without relying on a central agent, which may constitute a single point-of-failure and is exposed to communication breaks. Our methodology overcomes the typical deficiencies of a centralized solution by organizing the agents in spanning trees. Whenever possible, the spanning trees merge into a single tree that clusters the maximum number of agents. Using relaxation methods and inputs from its parent and children, every agent attempts to optimize its own solution for task assignment while communicating using an ad-hoc protocol. Simulation experiments show clear-cut advantages of using the proposed task assignment and communication algorithms.

I. Introduction

THE use of systems comprising multiple autonomous robots has been proposed to meet the requirements of complex missions [1]. The problem of design, development, and control of such multiagent systems has been studied in recent years for ubiquitous applications [2–6]. For example, Mataric [2] used the theory of multiagent systems to study the behavior of multiple robots through their mutual interactions, and referred to the resulting dynamics as group behavior. Reif and Wang [3] studied a very large-scale robotic system by adopting the method of potential fields. Beni and Wang [4] used the term "swarm" to define a multi-robot system. In this paper, we shall use the term "node" for each individual within the system, and the whole group will be referred to as a multi-agent system or a flock. In multiagent robotic systems, the term "behavior" is used to describe two different phenomena: the actions and movement of one agent with respect to its environment and states, and the global actions and movement of all agents. In this research, the term "agent behavior" is used in reference to the individual unmanned aerial

Received 21 June 2007; revision received 15 April 2008; accepted for publication 03 May 2008.

* Senior Lecturer, Computer Science Department

† Graduate Student, moranfe@cs.technion.ac.il, Computer Science Department

‡ Senior Lecturer, Faculty of Aerospace Engineering

§ Graduate Student, Computer Science Department vehicle (UAV) behavior and the term "emergent behavior" describes the behavior of the whole system of UAVs. This is consistent with Mataric [5] and Steels [6], who studied the overall behavior of a group of multiple autonomous robots.

Within the realm of multiagent robotic systems, cooperative UAVs constitute a persistently evolving research and development field [7]. The main effort is directed toward distribution of mission performance tasks among a number of collaborating vehicles. This methodology facilitates global data collection and their transformation into information allows optimized resource allocation and reduces the group's susceptibility to the loss of an individual vehicle.

To coordinate the UAV movement in a given theater, a coordination mechanism known as flocking (also referred to as swarming or formation flying) is required [8–14]. For example, Toner and Yuhai [8] suggest a rigorous theory for coordination of agents. Richards et al. [14] developed an optimal control scheme for the coordination of multiple UAVs. In the present paper, however, we shall follow the seminal work of Reynolds [15], who was the first to simulate flocking behavior based on cohesion (agents converge onto a given point), alignment (velocity matching to move at a given 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' flocking behavioral model to avoid collisions and improve mission execution by facilitating wide theater coverage.

To allow communication among the UAVs, ad-hoc networks are used. Ad-hoc networks constitute a natural solution for communication in wide-area theaters where no existing communication infrastructure exists. The ability of ad-hoc networks to preserve connectivity, even when the participating nodes are moving, has earned these networks a reputation as ubiquitous networks. Ad-hoc networks earned significant attention in the literature. In particular, routing in ad-hoc networks was studied by Blending-Royer [16], which developed the adaptive routing using clusters (ARC) mechanism. Rubin et al. [17] studied the ad-hoc on demand vector routing (AODV) method. Gu et al. [18] described a heterogeneous network wherein UAVs are used to bridge between ground mobile entities.

Hierarchical TA has been dealt with for a variety of applications [19,20]. For example, Walshand and Wellman [19] described a system for monitoring large collections of dynamically changing tasks while assuming that tasks are distributed over a virtual space. Other areas pertinent to task assignment (TA) in the distributed control scenario are market-based control themes and utility-based tasking [21,22].

In this work, we deal with the following mission execution problem, taken from the realm of multi-agent robotic systems: Assume that a cooperative group of UAVs, flying in coordinated manner, attempts autonomously to intercept a noncooperative group of targets. How should the UAVs assign themselves to targets while communicating using ad-hoc routing so as to minimize a given performance index?

The main contribution of this paper compared to previous works (e.g. Sujit and Beard [23] and Spry et al. [24]) is a development of a real-world, realizable, distributed communication routing protocol, used for propagating the coordination and task assignment information. Emphasis is given on designing a tree-based, computationally efficient and hierarchical structure for data propagation that is robust and reactive, thus creating a purely decentralized information processing. Additional contributions (compared to, for example, Ben-Asher et al. [25]) are:

1) A novel TA algorithm minimizing flying distances of the UAVs.
2) An improved flocking mechanism with emphasis on a distributed implementation.
3) Enhanced experimental validation including non-uniform distribution of the targets and an extended number of UAVs.

II. Problem Formulation

The first challenge is to determine how the UAVs should fly to search efficiently and attack the targets. The UAV formation should satisfy the following requirements:

1) Efficiently search for new targets in a field that is possibly much larger than the joint coverage area of the UAV's sensors. Thus, the UAV trajectories must be chosen so as to include a search procedure, as target location is unknown and the targets are constantly moving.
2) Fast and efficient interception of several targets, which are possibly centered in a small area. This implies that the UAVs must fly as a flock to commence immediate, multiple, attacks on a set of targets. The tradeoff

here is between two cases: a) All targets are randomly distributed in the field or b) all targets are concentrated in a small area.

3) Flying as a flock for a search mission requires a distributed algorithm that coordinates the UAVs movements such that:
   a) they all fly in the same direction while coordinating their velocities;
   b) they avoid collisions with each other and/or with obstacles;
   c) the coverage area of the flock's sensors is maximal;
   d) the flocking algorithm should continue working when UAVs start to engage in an interception course, allowing temporal distortions in the search pattern of the flock.

Assuming that the field is a rectangle, we chose a flocking pattern implementing a random coverage of the rectangle. In this pattern, the flock bounces back and forth from the theater edges. This mechanism favors (percentage-wise) the case of uniformly distributed targets, but it maintains some efficiency in case that the targets are grouped (this tendency is more pronounced in large theaters). We do not claim that this search procedure is optimal, but merely that it is a reasonable approach owing to its random coverage of the field. Hence, there must be an algorithm that attempts to group the set of UAVs in one relatively ordered flock.

An additional challenge considered here is finding an attack plan, defined as an algorithm that computes TA to each UAV, that is, allocating targets to the UAVs as follows:

1) The algorithm generates a constant flow of updated target allocation plans distributed to each UAV. In this way, the flock responds to any change in the target location, available set of UAVs, and their available ammunition.
2) The attack plans must allocate targets to UAVs based on how close is each UAV to a given target while avoiding assigning several UAVs to the same target and by considering the amount of available ammunition of each UAV. This implies that the target allocation plan must be computed globally based on data collected from all the UAVs.
3) On the other hand, a UAV should not wait for the results of a global computation but should rather launch an attack based on its current state and the last attack plan received based on a preceding global computation. In general, for each UAV there must be several levels of attack plans. Each level should include updated data from a wider surrounding set of neighbors. Consequently, each UAV is autonomous in the sense that it always follows some locally updated attack plan. However, it is also coordinated with the rest of the flock through incoming attack plans from the different levels.

Both the flocking and attack plan computations run in parallel. In our treatment, we do not idealize the communication system; rather, we acknowledge the fact that the communication links between the UAVs are constantly disrupted owing to dynamic changes in the theater (i.e., new UAVs join in or move out of communication range). Thus, the communication algorithm for coordination and TA must maintain its performances under these dynamical changes. To propagate the TA and flocking information, we selected the ad-hoc metrical routing algorithm (MRA) [26]. As described in Sec. III, the MRA naturally supports the organization needed to implement the algorithms of flocking and TA in parallel.

The solution to the mission execution problem proposed herein is to maintain coverage of the UAVs by a minimal set of spanning trees. There is a constant flow of updated attack plans and flocking commands. Each node (UAV) constantly merges the incoming streams while sending a stream of updated TA and flocking commands to both its parent and children. The root of the tree sends updated streams of TA and flocking instructions to its children. In this way, the tree implements the basic requirement of computing updated TA and flocking in levels corresponding to the wider surroundings of neighbors.

Figure 1 depicts the mechanism in which updated streams of TA and flocking commands are computed. Full integration of information from all nodes occurs only at the root. However, there are many distributive computations of updated TA and flocking at a given time instant. For example, if there are two distinct events (in two distant subtrees), then two updated TA and flocking commands will be separately computed by the nodes. Thus, the updated TA/flocking of each event will be distributed to all the relevant nodes without first reaching the root. Thus, the computation of an updated TA/flocking information combining events from all parts of the trees must occur eventually at some node; hence, the root of the tree should not be regarded as a centralized bottleneck. In addition, each node constantly merges

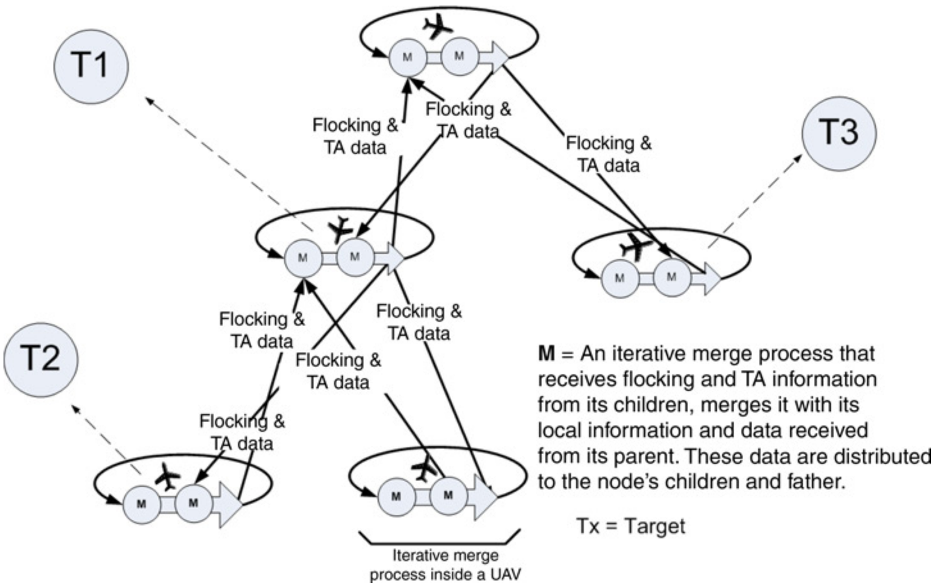
T1
T3
T2
M
Flocking & TA data
Flocking & TA data
Flocking & TA data
Flocking & TA data
Flocking & TA data
Flocking & TA data
Flocking & TA data
M = An iterative merge process that receives flocking and TA information from its children, merges it with its local information and data received from its parent. These data are distributed to the node's children and father.
Tx = Target
Iterative merge process inside a UAV

Fig. 1 Updating streams in a spanning tree.

the updated streams from its children and parent. Therefore, the size of the data that the root of any sub-tree receives is approximately the same (i.e., there is no doubling of incoming information owing to the merging).

The main contribution of this work is the development of a reactive distributed TA algorithm capable of efficiently engaging multiple dynamical targets using unstable communication links. The algorithm is designed to optimize the usage of munitions and, in a lesser priority, to minimize the flight distances to the targets. Our heuristic real-time distributed TA algorithm leads towards an optimal solution in the following sense: If we "freeze" the targets, the TA algorithm will find the optimal target allocation and attack plan for each UAV, minimizing some instantaneous performance index. Although we "freeze" the targets momentarily, the algorithm reevaluates the situation when new, updated information regarding the UAVs and/or the targets arrives. This notion of optimality is weak, as it approximates the dynamic case by an equivalent set of static cases. Ideally, optimality should constitute a deliberative algorithm that has perfect knowledge of the future movements of all targets. Such an algorithm, however, seems impractical for complex, real-world scenarios.

The global performance of the new TA algorithm shows substantial increase in efficiency (the number of targets hit at a given time instant for a constant number of munitions in the flock) vis-à-vis the baseline cases, where no flocking and/or TA algorithms are used. The experiments show that when the targets are uniformly distributed, more successful hits are obtained and, percentage-wise, flocking is more significant; and that TA is more significant when the targets are grouped at a given area in the theater. In a large theater, TA and flocking improve performance by about 30% (uniform target distribution) and 40% (targets grouped).

III. UAV Communication via Metrical Routing Algorithm

In this section, we describe the MRA [26], used as an ad-hoc communication protocol among the UAVs for communicating target lists, flocking information, and tree formation processes.

To explain the communication topology, let G denote the communication graph whose edges correspond to the current communication links between UAVs $U_1, \ldots, U_n$. An existence of an edge in G indicates that two nodes are in transmission range and can thus send/receive messages using some wireless MAC protocol (IEEE 802.11 was used in our experiments). Generally speaking, the MAC protocol may prevent collisions owing to the existence of overlapping transmissions from multiple neighbors.

The graph G is not fully connected. Conversely to other ad-hoc routing algorithms, the MRA attempts constantly to embed G in a minimal set of rooted trees that preserve geographical distances; viz. distances on the rooted trees

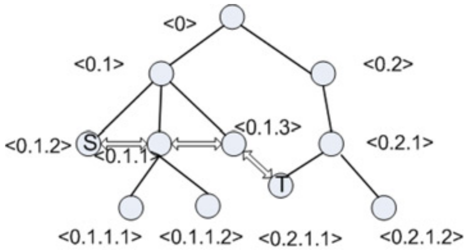

Fig. 2 Sample tree with a shortcut.

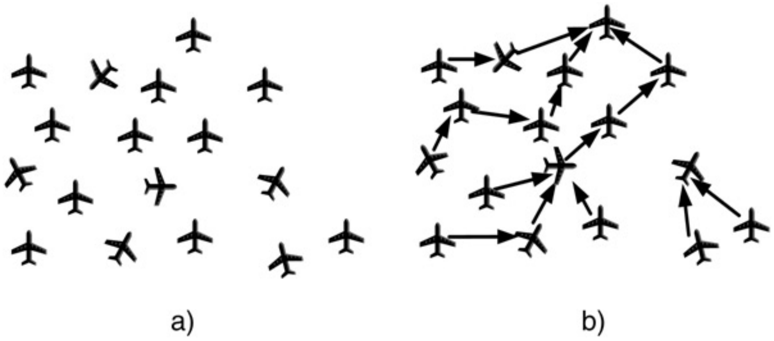

Fig. 3 Tree formation process.

are proportional to the original distances in G. The rooted trees created by the MRA can naturally be used for both distributed computing (of, for example, the flocking data) as well as for communication, in addition to propagation and computation of the TA commands.

Figure 2 depicts a sample tree created by the MRA. The tree is used to express distance metrics among the network's nodes. These metrics constitute a representation of the nodes in the real, physical, space by a labeled graph, so that the distances in space are approximated by the distances on the graph. For example, the path $\langle 0.1.2\rangle\langle 0.1.1\rangle\langle 0.1.3\rangle\langle 0.2.1.1\rangle$ in Fig. 2 represents a session path based on a communication link between the endpoints S and T (this path can be created in a variety of other manners as well).

The MRA 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 between two nodes has been created, the trees will be fused into a single tree.

The 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, then they will initiate a fusion process that combines the separate trees into a single tree. The resulting protocol is thus fully distributed. Comparing to other ad-hoc routing algorithms, this protocol uses less control messages for establishing routing paths.

Figure 3 illustrates two stages of the tree fusion process: The initial state and the final organization into trees (if no significant node movements occurred). The two separate trees in Fig. 3b cannot fuse because there are no nodes within transmission range.

Fusion of two trees is a parallel process, wherein at any given stage, more than one node of the smaller tree joins the larger tree, as depicted in Fig. 4. The implementation of the TA and flocking algorithms is based on this tree structure. Every tree autonomously runs these algorithms as it does not necessarily posses communication with other trees.

The complexity to add a new node to the tree is $O(1)$ and the average complexity to remove a node is $O[\log(n)]$. Removing a node from a tree requires updating all the nodes in its sub-tree. If the tree is created randomly (adding a new child to a random node) then most of the subtrees will have less than $O[\log(n)]$ nodes. Every node is required to store the ID and the address of every neighboring node. This information requires less than 10 bytes per node. In

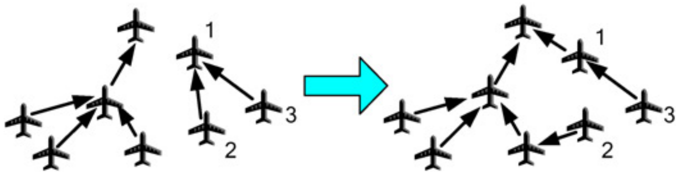

Fig. 4 Fusion of trees.

addition, the hash nodes, which are the direct children of the root, store mappings of node IDs and addresses for all the nodes in the network. In spite of the UAVs' speed and variable heading, the trees maintained by the MRA remain stable. This is because the UAV dynamics are slow compared to the communication speeds. Hence, the tree fusion can be completed in spite of constant changes in the graph G without a significant loss of messages. In addition, the transmission range of a UAV is large relatively to its instantaneous position change; thus, changes in the MRA trees are not very frequent. The locality and the relatively low overhead of adding/removing nodes also contributes to the ability to maintain stable trees. Finally, the use of flocking contributes to the stability of the MRA's trees.

The tree fusion works in the communication level of the MRA. The proposed distributed TA and the distributed flocking rely only on the ability to send/receive relatively short messages between adjacent nodes in the spanning tree. Changes in the tree therefore correspond to failures in receiving a message (updated TA or flocking information) sent from one node to another. The flocking and the TA algorithms are designed so as not to include any form of delay; hence these algorithms are not affected by the frequent changes in the tree topology.

IV. Reactive Distributed Flocking Algorithm

The flocking algorithm for coordinating UAV motion uses Reynolds' behavioral algorithm [15], which computes a desired velocity command generated based on the three basic flocking rules of cohesion,alignment, and collision avoidance. This velocity command is then used to steer the UAVs, or, in other words, to generate commanded acceleration which determines the turn rate of each UAV. Complete details are provided reference [25].

Although Reynolds' flocking algorithm is a well-known mechanism for motion coordination of multiagent robotic systems, the implementation of this algorithm in our work has a number of novel features. First, every UAV communicates only with its parent and children in the tree and is unable to obtaint a global view of the position and velocity of the entire flock. The tree structure created by the MRA forces the root to gather data from its subtrees, calculate the global values, and distribute them back to its subtrees. We defined two data streams on the spanning trees: upstream, moving from the tree leaves or subtrees towards the root; and downstream, moving data from the root towards the leaves. The data flow and the calculations are performed in a purely distributed manner, namely the root UAV performs tasks that are similar to the tasks of any other UAV in the tree.

The streams move stepwise form the parent node to its children and vice-versa, as shown in Fig. 5. In the upstream direction, every UAV communicates the size of its subtree, the cohesion, alignment and collision avoidance velocities,

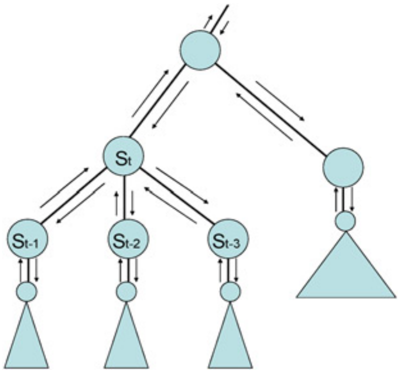

Fig. 5 Streams that move across subtrees are used to transport TA and flocking information and commands.

and the position of its subtree. The UAV calculates these values based on the values received from its children and their own subtrees.

The calculation of the desired velocity vector is done in a similar way. Every subtree root (node *St* in Fig. 5) receives the following values from its children (nodes S_{t-1}, S_{t-2}, S_{t-3} in Fig. 5): m_i – the size of S_{t-i}, which is the number of nodes forming subtree S_{t-i}; and $\boldsymbol{x}_i$—the average position vector of subtree S_{t-i}. In addition, if $\boldsymbol{x}_s$ is the position vector of node *St*, then the average position vector that node *St* will send to its parent is therefore given by

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

The subtree size that *St* will send to its parent is $1 + \sum_i m_i$. The iterative process ends in the root node. The root calculates its own subtree position and velocity, which is also the position and velocity for the tree, and communicates it downstream. These values will be spread from a parent to its children until they reach the leaves. If the root node is lost, a new root will be declared; the new root will then take over the tasks of the lost root.

V. Task Assignment Algorithm

The TA algorithm 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 acting according to the changes in the field (UAV movements, target movements, munitions launching, and target elimination). However, when a UAV constitutes 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, $S = \{U_1, U_2, \ldots, U_N\}$, attacking moving targets, $A = \{T_1, T_2, \ldots, T_M\}$. As previously indicated, we focus on a distributed solution in a special setting wherein the communication among $U_1, U_2, \ldots, U_N$ is carried out by an ad-hoc network (using ad-hoc communication yields a complex and challenging setup [25]).

In previous works, the TA problem was solved using variants of the weighted matching algorithm (WMA) [25]. In the present work, we suggest a new algorithm, the look-ahead algorithm (LAA). This algorithm, the advantages of which are to be described shortly, is activated in every UAV whenever a new event is detected. A new event is declared in the following cases: a new target appears in the detection area of the UAV; a potential target disappears; a target list (TL) was accepted from one of the children or the parent; a UAV launches a missile at a target. Every UAV may overrule the decisions of its children, but it must obey the decisions taken by its parent. For example, after running the LAA, the parent may decide to assign another subtree to a target and it may prohibit the subtree currently intercepting the UAV from proceeding with the interception of this specific target.

A. Look-Ahead Assignment Algorithm

The LAA is a distributed algorithm activated in every UAV once it detects a new event. Every UAV runs the algorithm on an internal model with an input comprising a list of UAVs that appear in the TL.

Two concepts are essential for understanding the main idea of the LAA: The potential time and potential zone. The potential time, E, of a UAV is the elapsed time as the last target engagement (if the UAV was never engaged, the potential time equals the actual time). The UAV could move in any direction during the potential time. The potential zone is a circle, centered at the UAV, with a radius equal to the potential time multiplied by the instantaneous speed. The LAA can be summarized as follows:

```
1.  Set the potential zone of every UAV to 0
2.  While there are more targets:
    2.1  For each UAV and target pair do
         2.1.1  Calculate $E$, the potential time of the UAV
         2.1.2  Roll back the target to its location $E$ seconds earlier
         2.1.3  Calculate $T$ , the time required by $U$ to intercept $P$
         2.1.4  Roll forward the target to its current location
    2.2  Subtract $E$ from $T$
    2.3  Let $(U, P)$ be a UAV-target pair yielding the minimum interception time
```

```
2.4   Assign $P$ as the next target for $U$
2.5   Remove $P$ from the current target list
2.6   Update the location of every target $P$ to its predicted location based on its current location, speed and
      direction.
2.7   Update the location of $U$ to an interception position for the $P$
2.8   For each UAV other then $U$
      2.8.1   Add $T$ to the potential of the UAV
2.9   Set $E$=0.
```

The following Lemma states that the potential time can never exceed the value of the interception time:

Lemma: Let $T_i(U, P)$ be the time required for UAV U to intercept target P at iteration i, and let $E_i(U)$ be the potential time of UAV U at iteration i. Then

$$T_i(U, P) \geqslant E_i(U) \tag{2}$$

Proof. See Appendix. ■

Example: Consider the situation depicted by Fig. 6. The LAA checks for the optimal target to be intercepted based on minimizing the distances between the UAVs and the targets, while taking into account the velocities of the UAVs and targets. The LAA predicts the developments in the theater until U1 reaches the point where it can launch a missile at T1. Note that target movement is predicted during this timeframe, and, every UAV can move in any direction relative to its initial position. The radius of the potential zone of U2 is 3 and the distances from the surface of the potential zone to T2 and T3 are 4 and 3, respectively, as depicted in Fig. 6. The LAA will assign U2 to intercept T3. The LAA predicts the developments in the theater until U2 reaches the point it can launch a missile towards T3. The potential zone, created around U1 for predicting the possible positions of U1 after launching the missile towards U1, renders U1 the best candidate to intercept T2. The final attack plan for our example is: U1 attacks T1, U2 attacks T2 and T3.

Figure 7 shows an example of the upstream and downstream TA flow. In Fig. 7a, UAVs U3 and U4 detected a potential target T2. Both UAVs decide to intercept T2 and commence the interception process. Independently, the UAVs send their own TA to U2, which is the subtree UAV of U3 and U4. U2 analyzes the TAs received from U3 and U4 and selects U4 to attack T2. When another target T1 emerges, U1 and U2 are able to intercept it. Independently, both UAVs which are not engaged, decide to attack T1. U2 sends its TA upstream. This TA describes the decisions taken by U2 regarding its subtree and itself. The decisions of U2 are that U4 will intercept T2 and U2 will attack T1. The decision of U2 is that U4, and not U3, will intercept T2. This will halt the interception of T2 by U3.

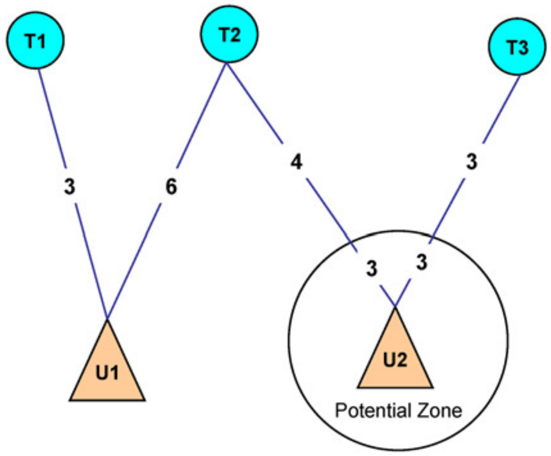

Fig. 6 Task assignment using the look-ahead algorithm.

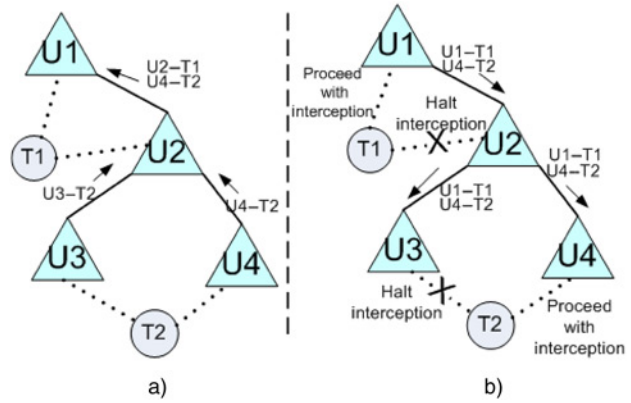

Fig. 7 Upstream and downstream TA flow.

Figure 7b depicts the opposite streams, where the TA flows towards the leaves of the tree. The downstream flow is independent of the upstream flow and the decisions are taken autonomously by every UAV. The decisions taken by all UAVs translate into an attack plan which is embedded in the TAs and distributed to all UAVs.

Owing to the distributed nature of the decision algorithm, the decision of U2—namely, that U4 will intercept T2 and not U3—is forwarded to U3 and U4, but also to U1. However, the decisions of lower levels in the tree can be overruled in some cases by UAVs in upper levels. This refinement may optimize the global performance and is complementary to the local and autonomous decision-making process of every UAV.

B. WMA and LAA Comparison

The WMA attempts to minimize the total distance of the UAVs to their targets over a single step [25],

$$\min_{A} \sum_{(i,j\in A)} [\mathrm{dist}(U_i, T_j)], \tag{3}$$

where the attack plan, A, is a set of ordered pairs, in which the first item is the index of a UAV, and the second item is the index of the target to be intercepted.

The LAA is a greedy algorithm, in which the optimum TA solution is sought over a given number of predicted time steps, namely

$$\min_{A} \left\{ \max_{[i,(j_h)_{h=1}^{n}]\in A} \left[\mathrm{dist}(U_i, T_{j_1}) + \sum_{h=1}^{n-1} \mathrm{dist}(T_{j_h}, T_{j_{h+1}}) \right]\Bigg] \right\} \tag{4}$$

In the LAA case, a pair in A, which represents the plan for a single UAV, is superimposed with an ordered tuple containing the indexes of the targets in their interception order.

Based on Eqs. (3) and (4), one can calculate the computational complexity of both TA algorithms. The computational complexity of the WMA is given by

$$O[(T+U)^2 \times (T \times U)] \tag{5}$$

while the complexity of the LAA is given by

$$O(T^2 \times U) \tag{6}$$

Clearly, the complexity of the WMA is higher than the complexity of the LAA.

VI. Simulation Results

The simulation experiments are aimed at evaluating the contribution of the TA and flocking algorithms to the performance of the UAVs using MRA-based ad-hoc communication. The TA algorithm was selected to be LAA, as it exhibits marked advantages over the WMA in terms of computational complexity and performance, as explained above.

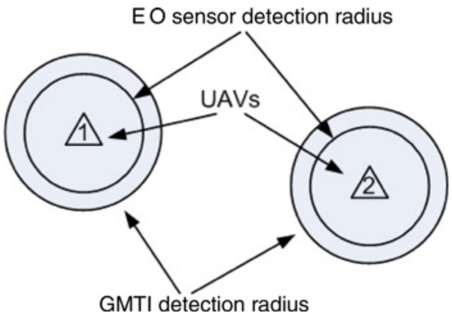

Fig. 8 Simulation snapshot: field-of-view of the UAVs.

A. UAV Sensors for Target Detection

We assume that 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 on the ground and guide the missiles. The detection radius of the GMTI is larger than the detection radius of the EO sensor. Figure 8 depicts a simulation snapshot showing the field-of-view (FOV) of two UAVs with a radio connection and illustrates the sensor detection radius.

B. Main Experiments and Simulation Models and Parameters

The main experiments comprise the following benchmarks:

1) Reference simulations performed without employing the TA and flocking algorithms;
2) Simulations used to evaluate the contribution of the TA and flocking algorithms separately;
3) Combined tests where both algorithms are employed simultaneously;
4) Simulations targeted to evaluate the load on the communication queues;
5) Scalability simulations aimed at examining the efficiency of the TA algorithm when the number of UAVs grows;
6) Evaluation of the effect of target distribution in the theater on the number of intercepted targets.

We have performed two sets of experiments. The first set compared the number of targets hit between two cases: Uniform distribution of the targets in the entire theater and a dense distribution of targets at a give area in the theater. The second set investigated the effect of interception time and hit probability on performance when the targets are uniformly distributed in the theater. Table 1 presents the simulation parameters used in both test sets.

Table 1 Main simulation parameters

Theater dimensions	10 km × 10 km in 1st set and 5 km × 5 km in 2nd set
UAV initial speed	100 km/h–120 km/h (uniformly distributed)
UAV radio transmission range	1.7 km
Target speed	50 km/h–80 km/h (uniformly distributed)
No. of targets	20 in 1st set and 16 in 2nd set
No. of UAVs	10 in 1st set and 8–16 in 2nd set
Missiles	20 missiles in 1st set. Fixed value of 16 in 2nd set. 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.
TA synchronization rate	Nominal rate of 30 Hz for both up and down streams. This rate is sufficient to allow both the computations of the TA and follow the rate of changes in the UAVs location.

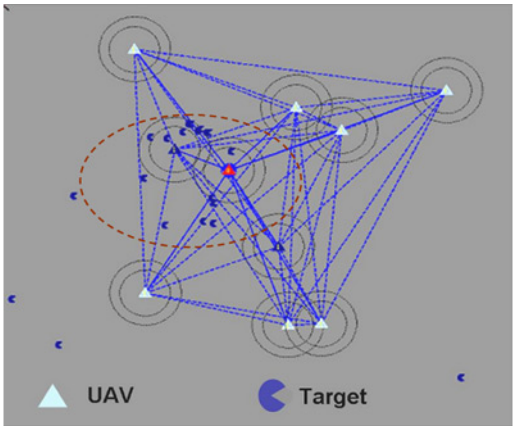

Fig. 9 A group of targets surrounded by UAVs.

1. Set 1: Effect of Target Distribution in the Theater

In the first set of simulations, we evaluated the effect of target distribution in a 10 km × 10 km field on the number of successful hits. In this set, every experiment includes 50 simulation runs, each lasts 120 s. The missile hit probability, p_h, was set to 1. We compared between two scenarios: Uniform distribution of the targets versus a dense distribution, where the targets are concentrated at a given area in the field, as shown in Fig. 9 (the target area is marked by the dashed ellipse – the external circle around every UAV represents the detection radius of the UAV and the internal circle around the UAVs represent the effective munitions range). In both cases, the initial position and velocity of each UAV were randomly selected.

In the latter case, the UAVs close on the targets while moving from the outer area into the inner field, where the majority of the targets are concentrated. Large flocks are quickly created. The main contribution of flocking in this case is that the UAV groups uniformly scan segments of the field.

Figure 10 depicts the results of the first set of experiments. Each data point in Fig. 10b represents the cumulative number of hits. In Fig. 10b, we show normalized results, where in the baseline case neither TA nor flocking are employed. The following observations can be drawn from the results:

1) The UAVs intercepted more targets when the targets were uniformly distributed.
2) Both flocking and TA contribute to the overall performance; the TA contributes less than the flocking in the uniform case; however, this tendency turns over in the grouped case.
3) In both cases, the combination of flocking and TA gives the best results. Compared to the case when flocking and TA are disabled, there is an increase of 42% in the number of successful hits in the grouped case and 29% in the uniform case.

An interesting question that stems from the results is why the number of hits in the uniform case is greater than in the grouped case. The analysis of the interception process shows that in the grouped case the UAVs surrounding the group of targets start to move in the direction of the centroid of the target group. During this movement, the UAVs hit most of the targets. However, this flock departs the other UAVs. In the uniform case, the UAVs hit targets while flying uniformly (on average) in the field, thus intercepting any target that is within the sensor FOV.

2. Set 2: Time-Critical Simulations

In this set, each simulation run was performed in a smaller theater, 5 km × 5 km, while assuming that the missile hit probability $P_h = 1$. The smaller theater reduces the effect of TA and flocking. All simulations were performed using two time intervals: $t_f = 60\,\mathrm{s}$ and $t_f = 120\,\mathrm{s}$. The 60-s simulations differ significantly from the 120-s simulations, as the interception process in the first 60 s demands a very short target interception time. The 120-s 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 1.

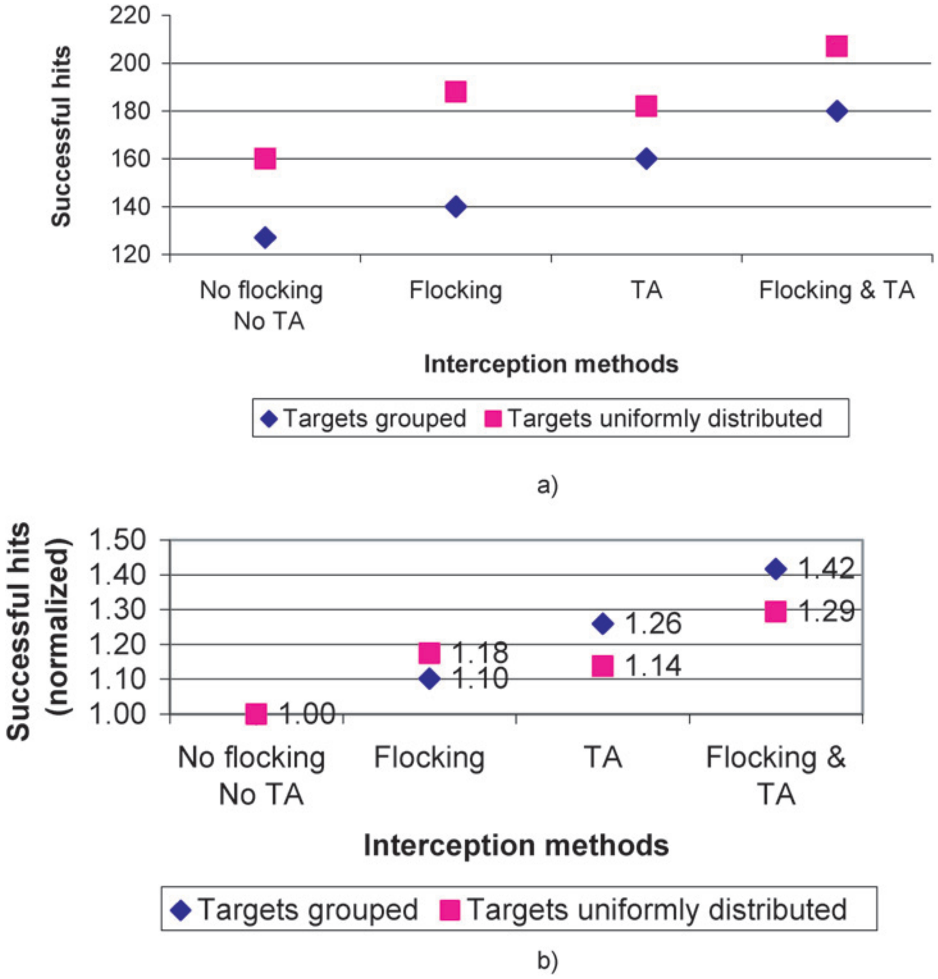

Fig. 10 The effect of flocking and TA: (a) more sucessful hits are obtained when the targets are uniformly distributed; (b) flocking is more significant in the uniform case, while TA is more significant in the grouped case.

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 runs. The mean efficiency, $\bar{\eta}$, is defined as the average of η over all UAV group sizes.

Table 2 summarizes the simulation batches used to evaluate the overall performance of the system.

Table 2 Simulations batches

Simulation batch	Flocking algorithm	Task assignment Algorithm	Time, s
Simulation batch 1	Active	Active	
	Active	Not active	60
	Not active	Active	
	Not active	Not active	
Simulation batch 2	Active	Active	120
	Active	Not active	
	Not active	Active	
	Not active	Not active	

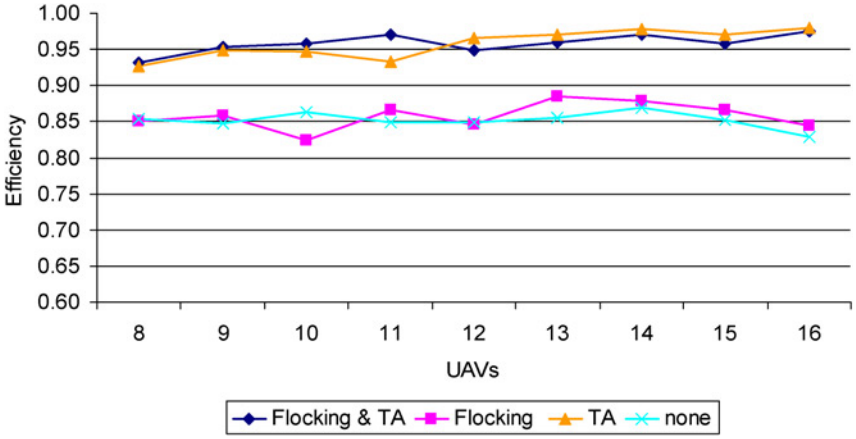

Fig. 11 Simulation results for a $t_f = 60\,\mathrm{s}$ and $p_h = 1$.

a) Simulation batch 1

Figure 11 depicts the results for simulation batch 1. In this figure, as well as in Fig. 15 (see later), the x-axis is the UAV group size and the y-axis is the efficiency, η. Each point on this graph represents an ensemble average of 50 Monte-Carlo runs.

The results exhibit two distinct regions. The upper lines result from simulations in which TA was 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 12 shows a simulation snapshot illustrating 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.

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.

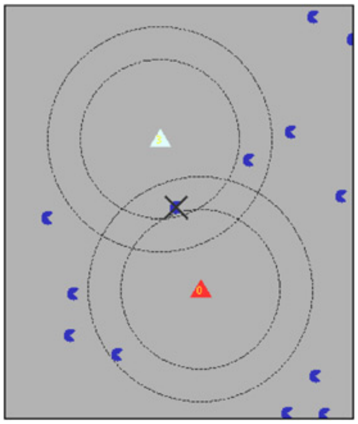

Fig. 12 Simulation snapshot: Multiple attacks on single target (crossed) by multiple UAVs (triangles) result in efficiency degradation.

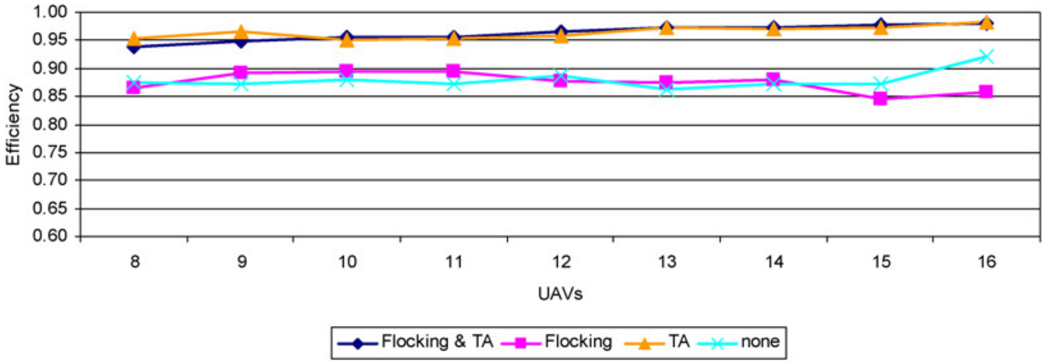

Fig. 13 Simulation results for a $t_f = 120$ s and $p_h = 1$.

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. Owing to the delay in building the trees and synchronizing the TLs, multiple missiles are launched towards targets that are detected by multiple UAVs. To mitigate this phenomenon we added a concomitant time delay enabling initial TL synchronization after which the UAVs are allowed to launch missiles.

b) Simulation batch 2

Figure 13 presents the results for $t_f = 120$ s and $p_h = 1$.

Simulation batch 2 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.

3. *Communication Queue Analysis*

Figure 14 presents the relation between the message queue size of every UAV and the normalized interception efficiency, defined as the ratio between a given efficiency and the efficiency obtained with non-communicating UAVs. The message queue is used by the MRA to manage the communication and handle the arrangement of the UAVs in trees. A zero-size queue prevents any communication between the UAVs. As a result, the UAVs behave noncooperatively. The maximal queue size used in this test was 10 messages. A queue of this size is sufficient and no messages are lost owing to queue overflow.

As shown in Fig. 14, the normalized efficiency remains almost constant when the queue size is reduced from a buffer of 10 messages to a buffer of five messages. A moderate decrease in the normalized efficiency is obtained when the queue size is reduced from a five-message buffer to a three-message buffer. A significant drop in the normalized efficiency is observed when the queue size is smaller than three messages.

The load on the queues is proportional to the number of UAVs in the theater and the message transmission rate. The TA data are synchronized among the UAVs using a frequency of 30 Hz. Although this synchronization frequency creates a load on the queues, loss of sporadic messages owing to queue overflow does not significantly reduce the normalized efficiency as long as the queue buffer can contain at least three messages.

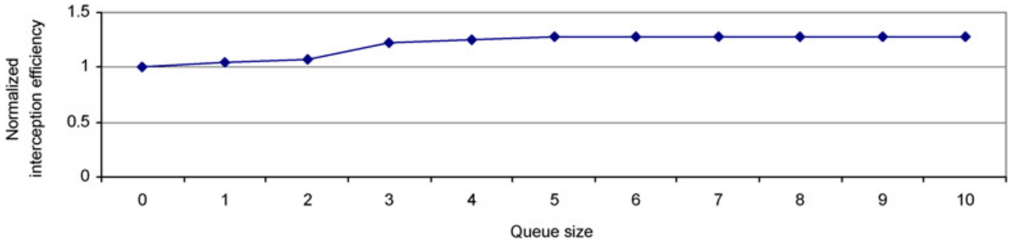

Fig. 14 Normalized efficiency as a function of UAV queue size. The reference efficiency, defined as the efficiency obtained from a noncommunicating group, is 1.

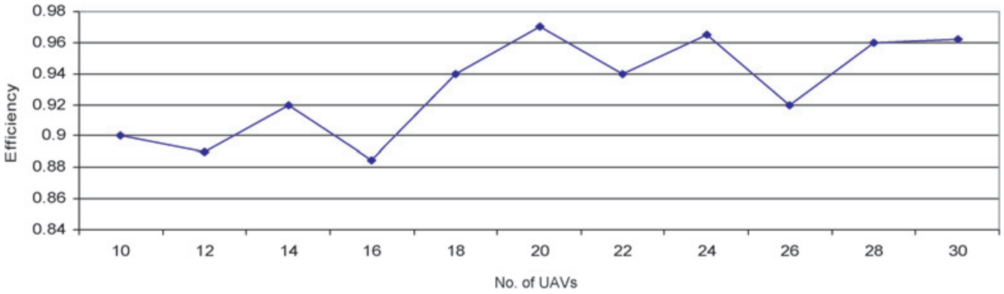

Fig. 15 Efficiency under variable number of UAVs.

4. Scalability

Simulations were performed to evaluate the scalability of the LAA TA algorithm so as to ensure that the performance does not decrease as the number of UAVs in the theater grows. More UAVs and targets yields a larger attack plan; an extra load on the communication links is then obtained owing to 1) more UAVs that detect more targets, and 2) additional targets yield larger data blocks to be exchanged among the UAVs.

We carried out a series of simulations for 10–30 UAVs. For a given number of UAVs, we kept the density of the UAVs constant by adapting the theater size to the number of UAVs. The number of targets was twice the number of UAVs.

Figure 15 shows the simulation results. The efficiency remains relatively constant until the number of UAVs grows beyond 16. The efficiency then grows until the number of UAVs reaches 20 and stays relatively stable for 20–30 UAVs.

The main conclusion from Fig. 15 is that the efficiency does not decrease as the number of UAVs grows. Moreover, a small improvement is obtained when the number of UAVs increases to 20. This is because 1) the likelihood that the communication links among groups of UAVs are lost decreases. As a result, the UAVs operate in a more stable communication environment, which improves the utilization of munitions; and 2) the probability that the targets assume “shelters” in the corners of the theater lessens. Targets that remain in the corners of the square theater are less exposed to the UAVs. However, adding more UAVs to the theater renders these “shelters” smaller, so fewer targets ultimately elude the UAVs.

VII. Conclusions

We developed a hierarchical algorithm for task assignment and communication of multiple UAVs engaging multiple targets in an arbitrary theater, and implemented an ad-hoc routing algorithm for synchronization of target lists 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.

The distributed TA algorithm for autonomous target interception was conceived based on a graph-theoretic approach. This LAA, was based on a finite-horizon prediction of the situation in the theater. The relatively low complexity of the LAA algorithm renders this algorithm suitable for implementation on actual operational flight computers.

Our simulation 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-cut advantage of the task assignment algorithms have been shown to exist in cases where the hit probability tends to one.

Appendix: Proving that Potential Time does not Exceed Interception Time

We distinguish between two cases:

1) U is the UAV that was assigned a target in the $i-1$ iteration then based on 2.10, $E_i(U) = 0$. The interception time is always positive and therefore $T_i(U, P) \geqslant E_i(U) = 0$.
2) U is not the UAV that was assigned a target in the i-1 iteration. Let V be the UAV that was assigned a target in the i-1 iteration and let P' be the target that was assigned to V. Based on 2.3

$$T_{i-1}(V, P') - E_{i-1}(V) \leqslant T_{i-1}(U, P) - E_{i-1}(U), \tag{A1}$$

because otherwise the algorithm would not have assigned P' to V. Eq. (A1) is equivalent to

$$T_{i-1}(U, P) \geqslant E_{i-1}(U) + [T_{i-1}(V, P') - E_{i-1}(V)]. \tag{A2}$$

At the beginning of iteration i, UAV U remained stationary. However, the target P moved during the $i-1$ iteration to its location t seconds later

$$t = T_{i-1}(V, P') + E_{i-1}(V), \tag{A3}$$

but the potential time of U also increased by t seconds, and therefore the interception time of U and P as calculated in 2.1.3 does not change between the $i-1$ and the$i$th iterations

$$T_i(U, P) = T_{i-1}(U, P). \tag{A4}$$

Using Eqs. (A2) and (A4) we obtain

$$T_i(U, P) \geqslant E_{i-1}(U) + [T_{i-1}(V, P') - E_{i-1}(V)] \tag{A5}$$

In 2.8.1, we increase the potential time of every UAV by the difference between the interception time and the potential time of the UAV that was assigned to a target in the current iteration. Therefore

$$E_i(U) = E_{i-1}(U) + [T_{i-1}(V, P') - E_{i-1}(V)]. \tag{A6}$$

The right-hand side of Eq. (A6) equals the right side of Eqs. (A5) and (2) immediately follows.

References

[1] Dudek, G., Jenkin, M., Milios, E., and Wilkes, D., "A Taxonomy for Multi-agent Robotics," *Autonomous Robots*, Vol. 3, No. 4, 1996, pp. 375–397.

[2] Mataric, M. J. "Designing and Understanding Adaptive Group Behaviors," *Adaptive Behavior*, Vol. 4, No. 1, 1995, pp. 51–80. doi: 10.1177/105971239500400104

[3] Reif, J., and Wang, H., "Social Potential Fields: A Distributed Behavioral Control for Autonomous Robots," *Robotics and Autonomous Systems*, Vol. 27, 1999, pp. 171–194. doi: 10.1016/S0921-8890(99)00004-4

[4] Beni, G., and Wang, J., "Swarm Intelligence in Cellular Robotic Systems," *Proceedings of the NATO Advanced Workshop on Robotics and Biological Systems*, Tuscany, Italy, 1989.

[5] Mataric, M. J., “Behavior Based Control: Examples from Navigation, Learning, and Group Behavior,” *Journal of Experimental and Theoretical Artificial Intelligence*, IEEE, NJ, USA, Vol. 9, No. 2-3, 1997, pp. 323–336.

[6] Steels, L., “The Artificial Life Roots of Artificial Intelligence,” *Artificial Life*, Vol. 1, 1994, pp. 75–110.

[7] Banda, S. S., “Future Directions in Control for Unmanned Aerial Vehicles,” Slides, April 2002, [online] http://www.cds.caltech.edu/ murray/cdspanel/fdc-apr02/banda-26apr02.pdf.

[8] Toner, J., and Yuhai, T., “Flocks, Herds, and Schools: A Quantitative Theory of Flocking,” *Physical Review E*, Vol. 58, No. 4, 1998, pp. 4828–4858.

[9] Crowther, B., “Rule-based Guidance for Flight Vehicle Flocking,” *Proceedings of the I Mech E Part G Journal of Aerospace Engineering*, Vol. 218, No. 2, 2004, pp. 111–124.

[10] Crowther, B., “Flocking of Autonomous Unmanned Air Vehicles,” *Aeronautical Journal*, Vol. 107, No. 1068, 2003, pp. 99–110.

[11] Czirok, A., Vicsek, M., and Vicsek, T., “Collective Motion of Organisms in Three Dimensions,” *Physica A*, Vol. 264, 1999, pp. 299–304.
doi: 10.1016/S0378-4371(98)00468-3

[12] Balch T., and Arkin, R. C., “Behavior-based Formation Control for Multiagent Robot Teams,” *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, 1998, pp. 926–939.

[13] Burchan, B. O., Lien, J.-M., and Amato, N. M., “Better Flocking Behaviors in Complex Environments using Global Roadmaps,” Texas A&M University, Technical Report TR02-003, May 2002.

[14] Richards, A., Bellingham, J., Tillerson, M., and How,J. P., “Co-ordination and Control of Multiple UAVs,” *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, 2002.

[15] Reynolds, C. W., “Flocks, Herds, and Schools: A Distributed Behavioral Model,” *Computer Graphics*, Vol. 21, 1987, pp. 25–34.
doi: 10.1145/37402.37406

[16] Blending-Royer, E. M., “Hierarchical Routing in Ad hoc Mobile Networks,” *Wireless Communication and Mobile Computing*, Vol. 2, No. 5, 2002, pp. 515–532.

[17] Rubin, I., Behzad, A., Ju, H., Zhang, R., Huang, X., Liu, Y., and Khalaf, R., “Ad Hoc Wireless Networks using Mobile Backbones,” in *Emerging Location Aware Broadband Wireless Ad Hoc Networks*, edited by R. Ganesh, S. Kota, K. Pahlavan and R. Agust, Kluwer Academic Publishers, 2004, Chapter 9, pp. 141–161.

[18] Gu, D. L., Pei, G., Ly, H., Gerla, M., Zhang, B., and Hong, X., “UAV-Aided Intelligent Routing for Ad-Hoc Wireless Networks in Single-area Theater,” *Proceedings of the IEEE Wireless Communication and Networks Conference*, Chicago, IL, 2000.

[19] Walshand, W. E., and Wellman, M. P., “Efficiency and Equilibrium in Task Allocation Economies with Hierarchical Dependencies,“*Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Santa Fe Institute, Santa Fe, NM, USA, Stockholm, Sweden, 1999, pp. 520–526.

[20] Yadgar, O., Kraus, S., and Ortiz, C., ”Hierarchical Organizations for Realtime Large-scale Task and Team Environments,” *Proceedings of the Autonomous Agents and Multi-Agent Systems Conference*, ACM Press, New York, NY, USA, Barcelona, Spain, 2002.

[21] Padhy, P., Dash, R. K., Martinez, K., and Jennings, N. R., “A Utility-based Sensing and Communication Model for a Glacial Sensor Network,” *Proceedings of the Autonomous Agents and Multi-Agent Systems Conference*, ACM Press New York, NY, USA, Hakodate, Japan, 2006.

[22] Clearwater, S. (ed.), *Market-based Control of Distributed Systems*, World Scientific Press, Singapore, 1995.

[23] Sujit, P. B., and Beard, R., “Multiple MAV Task Allocation using Distributed Auctions,” *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, 2003, AIAA Paper 2007-6452.

[24] Spry, S. C., Girard, A. R., and Hedrick, J. K., “Convoy Protection using Multiple Unmanned Aerial Vehicles: Organization and Coordination,” *Proceedings of American Control Conference*, IEEE, NJ, USA, June 2005, pp. 3524–3529.

[25] Ben-Asher, Y., Feldman, S., Gurfil, P. and Feldman, M., “Distributed Decision and Control for Cooperative UAVs using Ad-Hoc Communication,” *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 3, 2008, pp. 511–517.

[26] Ben-Asher, Y., Feldman M., and Feldman, S., “Ad-Hoc Routing using Virtual Coordinates Based on Rooted Trees”, *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Taiwan, 2006.

Ella Atkins
Associate Editor