2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Multiple UAV Coalition Formation

P.B. Sujit, J.M. George, and R.W. Beard

Abstract— **Unmanned aerial vehicles (UAVs) have the potential to carry munitions in support of battlefield operations, however they have limited sensor range and can carry only small quantities of resources. Often, to fully prosecute a target, a variety of assets may be required, and it may be necessary to deliver these assets simultaneously. Therefore, a team of UAVs that satisfies the target resource requirement needs to be assigned to a single target, and this team is called a coalition. Other desired requirements for the coalition are (i) minimize the target prosecution delay and (ii) minimize the size of the coalition. In this paper, we propose a two-stage optimal coalition formation algorithm that assigns appropriate numbers of UAVs satisfying the desired requirements. We developed a Dubins curves based simultaneous strike scheme. Simulation results are presented to show that the two-stage coalition formation algorithm has low computational overhead and can be applied in real-time.**

I. Introduction

In recent times there has been a significant increase in the use of low-cost unmanned aerial vehicles (UAVs) for military applications. These UAVs can be used for surveillance, search, attack, rescue missions, etc. Typically, for these missions, multiple UAVs are deployed to provide robustness through redundancy and to accomplish the mission faster. A search-and-destroy mission involves UAVs cooperating as a team to prosecute targets. Therefore, there are two kinds of tasks: (a) search tasks and (b) target prosecution tasks. Given a set of tasks, the task allocation problem maps tasks to agents. Grekey and Mataric [1], present a taxonomy of the multi-robot task allocation problem. Under their classification, our problem fall in the multi-task/multi-robot (MT-MR) task allocation category.

Many researchers have developed allocation algorithms for efficiently allocating UAVs to different tasks [2]-[5]. Most of the algorithms fall under the Multi-task/single-robot category and assume UAVs are (i) homogenous, (ii) carry resources that do not deplete with usage, (iii) can individually prosecute the target, and (iv) can destroy the target with any of its assets [2]-[4]. It is more realistic to assume that the targets can be of various types, and to destroy them, different types of assets are required. With these higher fidelity assumptions the solution must assign an appropriate number of UAVs to the same target for effective target prosecution.

This sub-group joins together to form a coalition. A coalition is a group of team members that have agreed to cooperate with each other to execute a single task [6]. Determining the optimal coalition from a group of agents is a computationally intensive task and is NP-hard [7] due to the size of the coalition structure. Fortunately, there are algorithms that provide approximate and near-optimal solutions [8]. Forming a coalition to achieve tasks has been an active field of research both in the multi-agent community [6], [7] and the multi-robot community [8], [9]. However, they do not address some of the issues that UAVs encounter.

In this paper, we present a two-stage algorithm that determines an optimal coalition with the required resources, plans an attack on the target in minimum time, and has a minimum number of agents in the coalition. The algorithm can handle dynamic changes in resources before the coalition formation process is instantiated. The coalition members change their path lengths such that they can strike simultaneously. The focus of this paper is to develop a coalition formation algorithm that generates the optimal coalition for a target with low computational overhead.

II. Problem Formulation

A. The mission

A search and destroy mission is carried out on a battlefield using N UAVs. We assume that there are M targets whose initial positions are unknown. The UAVs have the capacity to carry various munitions. Assume that UAV A_i can carry n types of resources represented by a capability vector $\mathcal{R}_i^A$ of the form:

$$\mathcal{R}_i^A =< R_{i1}^A, \ldots, R_{in}^A >, \quad i = 1, \ldots, N \tag{1}$$

where R_{ip}^A, $p = 1, \ldots, n$ represents the number of p-type resources held by agent A_i. For example, $\mathcal{R}_i^A =< 4, 2, 0, 6 >$ implies that agent A_i has four 1-type of resources($R_{i1}^A = 4$), two 2-type of resources($R_{i2}^A = 2$), zero 3-type resources($R_{i3}^A = 0$), and six 4-type of resources($R_{i1}^A = 6$). The UAVs perform a search task to detect targets. Once A_i detects a target T_j, we assume that the agent can also determine the type of resource required to prosecute the target. If m different types of resources are required to prosecute the target T_j, the capability vector of the target is represented as

$$\mathcal{R}_j^T =< R_{j1}^T, \ldots, R_{jm}^T >, \quad j = 1, \ldots, M \tag{2}$$

This work was partially funded by the National Science Foundation under Information Technology Research Grant CCR-0313056 and by NASA under STTR contract number NNA04AA19C to Scientific Systems Inc, and Brigham Young University and by the Air Force Office of Scientific Research award No. FA9550-04-0209.

P.B. Sujit and Randy Beard are with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, Utah 84604. sujit+beard@byu.edu

J. M. George is a Graduate student in the Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India. joel@aero.iisc.ernet.in

where $R^T_{jq},\ q = 1,\ldots,m$ and $m <= n$, represents the quantity of q-type resources required to prosecute the target T_j. If $m > n$, then for those targets, the needed resources $R_{jq}, (q > n)$ are not available, hence they will not be fully prosecuted. Therefore, the UAVs should carry all the types of resources required for the mission. For example: $\mathcal{R}^T_j =< 3, 0, 5 >$ indicates that to destroy target T_j, the agents need three 1-type resources ($R^T_{j1} = 4$), zero 2-type resources($R^T_{j2} = 0$), and five 3-type resources($R^T_{j3} = 5$).

The agent A_i broadcasts the required resource ($\mathcal{R}^T_i$), and the UAVs that have any of the required resource capability to strike target T_j will respond to A_i with their cost and resource capabilities. The cost is determined using Dubins curves from the UAV position to the target location.

The task for A_i is to select the coalition members such that the target is destroyed as well as satisfy other objectives which include minimizing the number of UAVs attacking the target, and minimizing the time to attack the target. Once the coalition is determined, the coalition members adjust their paths such that a simultaneous strike on the target can take place. When the agents approach the target, they deliver the resources. After deployment, depending on the quantity and the type of resource deployed, the agents reduce their capacities respectively.

We assume that the velocity of the UAVs is constant, therefore all the coalition members should travel the same distance to satisfy the simultaneous strike constraint. The maximum distance traveled by an agent in the coalition will determine the minimum time to attack the target. If there are N agents in the coalition, then the total distance traveled by these agents is $ND_{\max}$, where $D_{\max}$ is the maximum distance of any agent in the coalition. If the target resource constraint can be achieved with fewer agents (say N') then the total distance $N'D_{\max}$ is less than $ND_{\max}$. So, we use this idea to determine the minimum number of agents for the coalition. Let $\mathcal{D} = \{D_1, D_2, \cdots, D_N\}$ represents the set of distances for the N agents that are possible members of a coalition. The objective function that the coalition leader has to solve is given as:

$$\begin{aligned} &\text{Objective}: \min_{N'} N' \max \mathcal{D}' \\ &\text{subject to} \sum_{k=1}^{N'} R^A_{kp} \geq R^T_{jp},\ \text{for all } p = 1,\ldots,m \end{aligned} \tag{3}$$

where $N' = |\mathcal{D}'|$, and $|\mathcal{D}'|$ is the cardinality of the set $\mathcal{D}'$, and $\mathcal{D}' \subseteq \mathcal{D}$; D_k is the distance that A_k has to travel to target T_j. The N' agents selected should have their total resource capability match with that of the target in question as given in the constraint Equation (3) of the optimization problem. In the objective function, N' depends on the size of $|\mathcal{D}'|$, therefore they are coupled and it is very difficult to solve these coupled equations to obtain an optimal coalition. In this paper, we propose a two stage algorithm to determine the optimal coalition for a target with low computational overhead.

The UAVs are subjected to kinematic constraints and hence cannot turn instantaneously. We assume that the autopilots of the UAVs hold the altitude and maintain the ground speed. The kinematics of the UAVs are modeled using first order dynamics and are given by

$$\begin{aligned} \dot{x}_i &= v_i \cos\psi_i \\ \dot{y}_i &= v_i \sin\psi_i \\ \dot{\psi}_i &= k(\psi_d - \psi_i) \end{aligned} \tag{4}$$

where ψ_d is the desired heading of the UAV. We assume heading rate is constrained to $-\omega_{\max} \leq \dot{\psi} \leq \omega_{\min}$. The UAVs have to perform a search task to detect targets. When the targets are found, coalition with other agents have to be formed for target prosecution.

III. Coalition formation

When an agent detects a target it forms a coalition of agents depending on the target capability. But, many agents can detect targets in the region simultaneously. This results in deadlocks and to eliminate them we use a token mechanism.

In this approach, the coalition leaders first broadcast their token numbers and then arrange the received token number from other leaders in the descending order to determine their turn for coalition member request. When a coalition leader A_i announces the target information of all the agents that are not assigned to any target and also have the desired resources will respond to the leader. The responses may also include other coalition leaders. If these coalition leaders becomes a member of the coalition formed by other agents then also they will be the coalition leader for their detected target but will not participate in any coalition formation request.

Once the agent detects a target, it has to form a coalition based on its current resource level and its commitment to any other agent. When A_i detects a target T_j, which requires resources $\mathcal{R}^T_j = \{R^T_{j1}, R^T_{j2}, \ldots, R^T_{jm}\}$ and assume that the resource capability of A_i is $\mathcal{R}^A_i = \{R^A_{i1}, R^A_{i2}, \ldots, R^A_{in}\}$. The agent A_i checks if it has the required resources to attack the target. If $R^A_{ip} \geq R^T_{jp},\ \forall\ R^T_{jp} \in \mathcal{R}^T_j, p = 1,\ldots,m$, then A_i would attack target T_j without requesting a teaming arrangement with other UAVs.

However, when A_i has partial or no resources, then A_i which is also the coalition leader for T_j broadcasts the information about the target (i.e, its location, type and number of capabilities) to the other UAVs. The agents that have at least one type of the required resource will send their cost to arrive at the target, as well as the type and quantity of the resources available for deployment. The coalition leader takes all the requests and generates a coalition that can prosecute the target in minimum time with minimum coalition size. The selection of the team members for the coalition is carried out by solving the optimization problem given in Equation (3). Since, solving the optimization problem is computationally intensive due to coupling of N' and $\mathcal{D}'$, we developed a two-stage algorithm that will produce the optimal solution with low computational overhead.

A. Two-stage coalition formation algorithm

Determining the minimum time and the smallest coalition that successfully prosecutes the target requires two steps. In the first step, we determine the set of all UAVs that can achieve the minimum time requirement and then we use this set of UAVs in the second stage to achieve the minimum member coalition. The algorithm to achieve this task is given by Algorithm 1. In the first stage, the coalition leader sorts the responses in the ascending order of cost. Since we are assuming that the velocity of the UAV is constant, each agent in the coalition should travel the same distance to meet the simultaneous strike constraint. Therefore, we take one agent at a time and check if the resource constraint is met. When the constraint is not met, the agent is included in the coalition set, the current totalResources vector and $\mathcal{D}_c$ are updated. This process continues till the resource constraint is met. Once the target resource constraint is met, the rest of the agents that proposed are not included. As the agents are sorted in ascending order, the distance of agent A_k ($D(A_k)$) is greater than or equal to $D(A_{k-1})$, hence including A_k will not minimize the time to attack. The first stage guarantees that the coalition reaches the target in minimum time and is optimal.

Algorithm 1 Two-stage coalition formation algorithm

Stage 1:
Initialize:
coalition = [ ]; totalResources = [ ]
number_of_responses = N; $\mathcal{D}_c$=[];
\\ sorting of the responses by their distance in ascending order.
$\mathcal{D}$ = Sort($\{D_i\}$), $i = 1 \cdots, N$;
for $k = 1$ to N **do**
  for $r = 1 : m$ **do**
    totalResources(r) $\leftarrow$ totalResources(r) + agent(response(k)).resources(r)
  end for
  coalition $\leftarrow$ [coalition response(k)]
  $\mathcal{D}_c \leftarrow \mathcal{D}(k)$;
  if totalResources == $\mathcal{R}_j^T$ then BREAK
  else CONTINUE
end for
Stage 2:
$D_{\max} = \max \mathcal{D}_c$;
Solve:
Objective Function: $\min_{N'} N' D_{\max}$
Subject to $\sum_{v=1}^{N'} R_{vp}^A \geq R_{jp}^T$, for all $p = 1, \ldots, m$;
$v \in$ coalition

For example, assume agent A_1 detects a target whose resources are $\mathcal{R}_1^T = \{4, 1\}$. Suppose that the UAVs that responded to the coalition leader with their resources and cost are given in the Table I. As per the first step of the algorithm, the possible coalition members are sorted in ascending order based on cost and this new list is also shown in Table I. The ordered set of coalition members is $E_1 = \{A_2, A_3, A_1, A_4, A_5\}$. The coalition is formed by recruiting members from E_1 starting with the first member. The coalition formation process is as follows. Initially A_2, the first member in the ordered set is added to the coalition making the coalition $C_1 = \{A_2\}$ with $\mathcal{R}_1^C = \{1, 3\}$ resources. Since, $\mathcal{R}_1^C$ does not have sufficient resources to effectively prosecute the target, the next agent A_3 in E_1 is added to C_1 forming $C_1 = \{A_2, A_3\}$ with combined resources of $\mathcal{R}_1^C = \{1, 4\}$. Since $\mathcal{R}_1^C < \mathcal{R}_1^T$, the next UAV A_1 is added to the coalition, making $C_1 = \{A_2, A_3, A_1\}$ with total resources of $\mathcal{R}_1^C = \{3, 5\}$, which still does not meet the target resource requirement. Therefore, the fourth member A_4 in E_1 is recruited into the coalition, resulting in $C_1 = \{A_2, A_3, A_1, A_4\}$. This coalition has combined resource of $\mathcal{R}_1^C = \{5, 5\}$, which satisfies the target resource requirement. The first stage of the algorithm terminates once the agent determines the coalition satisfying the target resource constraint. The distance for simultaneous rendezvous is determined by the coalition member whose distance is the longest. In the present example, the distance of A_4 is the longest (172).

Initial list of agents with their resources and cost		
UAV	Resources	Cost
A_1	2, 1	123
A_2	1, 3	47
A_3	0, 1	63
A_4	2, 0	172
A_5	3, 2	207

List of agents after sorting		
UAV	Resources	Cost
A_2	1, 3	47
A_3	0, 1	63
A_1	2, 1	123
A_4	2, 0	172
A_5	3, 2	207

TABLE I
AGENTS WITH THEIR RESOURCES BEFORE AND AFTER FIRST STAGE.

Theorem 1: Stage 1 of the two-stage algorithm generates the optimal minimal time coalition.

Proof: Let N be the number of agents that responded to a coalition formation request to attack target T_j with cost D_i . The target requires $\mathcal{R}_j^T$ resources to be prosecuted while the agents have $\mathcal{R}_i^A$ resources. The first step in stage 1 of the algorithm is to sort the proposals in increasing order of distances. This results in an ordered set $\mathcal{C} = \{A^1, A^2, \cdots, A^N\}, v = 1, \ldots, N$ with the property that $D_v \leq D_{v+1}$, where A^v represents the v^{th} agent in the coalition set $\mathcal{C}$ and D_v represents the distance of agent A^v to the target.

We assume that a feasible coalition can be formed and hence $\sum_{i=1}^{N} \mathcal{R}_i^A \geq \mathcal{R}_j^T$. Define $\mathcal{C}_k \subset \mathcal{C}$ to be the first k elements in $\mathcal{C}$, i.e., $\mathcal{C}_k = \{A^1, \ldots, A^k\}$. The outer *for loop* of stage 1 computes $\mathcal{C}_k$ for each k. Let s be the minimum integer that satisfies $\sum_{v=1}^{s} \mathcal{R}_v^A \geq \mathcal{R}_j^T$. Choose any index $y, y < s$, then $\mathcal{C}_y$ will not form a feasible coalition. This implies that a feasible solution should include at least one agent from the set $\mathcal{C} \backslash \mathcal{C}_{s-1} = \{A^s, A^{s+1}, \ldots, A^N\}$. Since A^s is the agent that has the shortest distance to target T_j, A^s is included in the set $\mathcal{C}_y \leftarrow [\mathcal{C}_y \ s]$ resulting in $\mathcal{C}_s$. If we include A^{s+1} in $\mathcal{C}_s$ then either $D_{s+1} > D_s$ or $D_{s+1} = D_s$.

In the former case, $\mathcal{C}_{s+1}$ is not the minimum time coalition as the task can be achieved with $\mathcal{C}_s$. For the latter case, since including the agent does not change the time to prosecute the target and the mission can be accomplished without A^{s+1}, $\mathcal{C}_s$ is the minimum time coalition. □

Once the minimum time coalition is formed, we need to eliminate those members whose resources are not required to satisfy the strike requirement. In the second stage, we formulate an integer programming problem as given in Algorithm 1 to determine the minimum number of agents that have sufficient resources to attack the target. The optimization problem yields optimal solutions. Since the first stage provides optimal minimum time and the second stage generates an optimal solution of minimum number of agents, the two-stage algorithm is an optimal coalition. From the above example, $D_{\max}$ is 172, and the optimal coalition after solving the integer programming problem yields $C_1 = \{A_4, A_1\}$.

B. Complexity analysis

Now we analyze the complexity of the proposed two-stage optimal coalition formation algorithm. In the first stage of the algorithm major computational blocks are the comparison of the resources and the sorting of the received proposals.

Theorem 2: The computational complexity of the first stage of the algorithm is $O(N(\log N + m))$.

Proof: Assume that N agent send their proposals and the target T_j requires m types of resources to prosecute. In the first stage, the complexity of sorting the proposals takes $O(N \log N)$. In the worst case, all of the N members can be part of the coalition and the *coalition* vector in the algorithm has to make m comparisons. Hence Nm computations have to be carried out to include all the N members. Therefore, the computational complexity of the first stage is $O(N \log N) + O(Nm)$, which is equal to $O(N(\log N + m))$. □

In the second stage, we use the optimum minimal time coalition to get optimal minimum size coalition using an integer programming technique. Although solving an integer programming problem is NP-hard, there are pseudo-polynomial algorithms to generate feasible solutions [11]. To solve the integer programming problem in the second stage, we used the $bintprog$ command in MATLAB that in turn uses a branch and bound technique to compute the solution.

The computational time depends on two factors: the number of agents and the quantity of their resources. During the initial phase, the resources of the UAVs are full and hence the coalition leader may receive a higher number of proposals. Since the UAVs are full of resources a lower number of members can perform the task. As N' (the selected number of members) is small, the computational time will be small. The case is reversed during the final stages where the UAVs may have fewer resources causing the coalition size may be large. As the coalition size becomes large the computational time increases. Through the above analysis, the computation time for the coalition formation mainly depends on the distribution of the UAV resources.

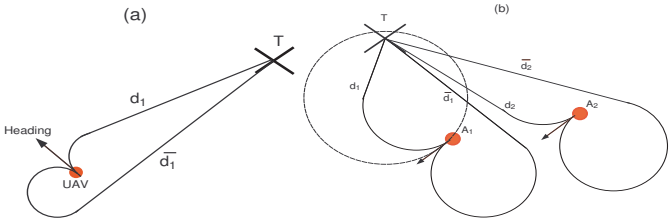

Fig. 1. (a) Dubins curves, where d_1 is the Dubins shortest distance, while $\bar{d}_1$ is the Dubins longest distance (b) An example of selecting Dubins distance.

The coalition leader at times may not receive a sufficient number of proposals that will satisfy the resource requirement of the target. In that case, the UAV will abort the coalition formation request and will re-broadcast the request after delta time units. Also, sometimes the coalition leader may not receive any response from other agents for its coalition formation request. This can happen either because the agents do not have required resources or they are busy in executing some other agent task. Even in this case, the formation request is aborted and new request will be broadcast after delta time units.

C. Simultaneous strike

The coalition leader A_i determines the agent that takes the longest route, distance P. The coalition members adjust their paths so that each path length is equal to P.

Assume that agent A_i is the coalition leader and let the coalition formed after using Algorithm 1 be $\mathcal{C}$. Also assume that the maximum distance (path length) is due to an agent $A_{k'} \in \mathcal{C}$, which has to travel a distance $D_{k'}$ to the target. In this case, A_i specifies this path length to the members of the coalition. Each member $A_k \in \mathcal{C}, k = 1, \ldots, |\mathcal{C}|$ has to adjust their path lengths accordingly so that D_k is equal to $D_{k'}$. In order to achieve that, the agents need to find $\underline{r}_k$, the radius of turn (greater than or equal to the radius of minimum turn), such that $D_k = D_{k'}$. Since, $\underline{r}_k$ cannot be calculated using a closed form solution, we calculate $\underline{r}_k$ iteratively until the condition $P = D_{k'}$ is satisfied.

Cost: The selection of the coalition members in the first stage of the coalition formation algorithm is based on the distance of the agent from the target. The distance is used as the cost function by the agents. Since the agents have kinematic constraints, we need to use Dubins curves [10] to compute the length of P.

Given an agent position and heading angle, we can determine two Dubins distances to the target: (i) Dubins shortest distance and (ii) Dubins longest distance, as shown in Figure 1. Either of these two distances can be the metric for cost which the coalition leader uses to make decisions. However, consider the example shown in Figure 1, where agent A_1 is the coalition leader and A_2 is a member of the coalition. If we choose the Dubins shortest distance, then according to

the coalition, A_1 has to adjust its path length such that d_1 is equal to d_2. But, it may happen that d_1 is impossible to obtain if A_1 has to move along the minimum radius circle with a too large of a radius (shown in dotted circle). On the other hand $\bar{d}_1 > d_2$, therefore A_1 cannot travel along $\bar{d}_1$. To eliminate this discontinuity, we always use the Dubins longest path as the metric for cost. In this case, A_1 can easily modify its radius such that it can meet the distance constraint of A_2. By adjusting the path length of the other members of the coalition, simultaneous strike condition can be obtained.

IV. Results

The performance of the coalition formation algorithm is carried out using Monte-Carlo simulations. The complexity of the optimization problem increases with increase in the number of agents and the number of targets. Using Monte-Carlo simulations we will analyze their effect on the two-stage algorithm.

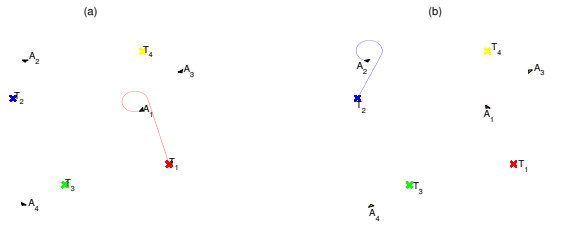

Fig. 2. (a)The route that A_1 follows to attack target T_1 (b)The route that A_2 follows to attack target T_2

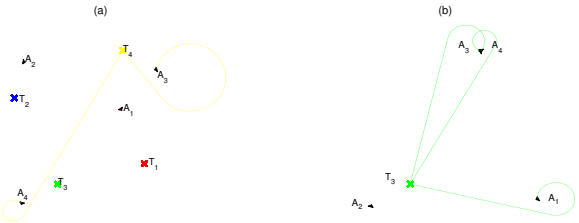

Fig. 3. (a) The route that A_3 and A_4 follow to attack target T_4. (b) The trajectories of the agents attacking their assigned targets (T_4 and T_3).

We consider a sample mission scenario with 4 UAVs and 4 targets on a $1000\text{m} \times 1000\text{m}$ area and analyze the computation time, and the time required to accomplish mission. The initial position of the UAVs and targets are shown in Figure 2(a). The velocity of the UAVs is 10m/s and the sensor range is 300m. The resources of the UAVs are $\mathcal{R}_1^A = \{2,3,4\}$, $\mathcal{R}_2^A = \{2,1,3\}$, $\mathcal{R}_3^A = \{3,2,4\}$, $\mathcal{R}_4^A = \{2,2,0\}$ and those of the targets are $\mathcal{R}_1^T = \{1,1,2\}$, $\mathcal{R}_2^T = \{3,2,4\}$, $\mathcal{R}_3^T = \{2,1,2\}$, $\mathcal{R}_4^T = \{3,4,1\}$.

The agents have limited sensor range and must search for targets. Once a target is detected, the agent forms a coalition. At $t = 1.2$ seconds, T_1, T_2 and T_3 are detected by A_1, A_2 and A_4 respectively. Agents A_1 and A_2 form single member coalition to attack T_1 and T_2 respectively, but A_4 cannot

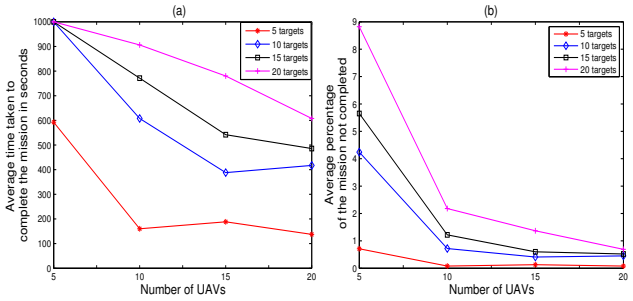

Fig. 4. (a) Comparison of mean mission completion time with increase in number of agents and targets (b) Mean number of targets not prosecuted due to lack of sufficient resources.

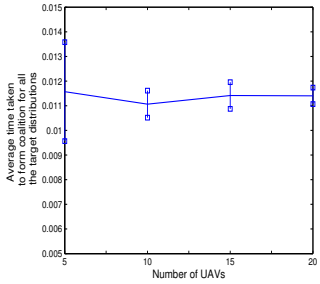

Fig. 5. Average computational time taken for given number of targets and varying number of agents.

form a coalition due to the lack of resource by its neighbors. At time $t = 1.8$, A_3 detects T_4 and forms a coalition with A_4. The routes taken by these agents are shown in Figures 2(a), 2(b), and 3(a). Later A_2 detects T_3 and agents A_1, A_3 and A_4 are assigned to the coalition. The routes that these agents follow are shown in figure 3(b). The time taken by the UAV to accomplish the mission was 230 seconds, while the simulation was completed in 7.53 seconds on a 1 GHz Laptop.

Effect of increase in number of agents and targets using two-stage algorithm

A set of 100 simulations were carried out on a $1000\text{m} \times 1000\text{m}$ area to study the effect on increase in number of agents and targets. We assume the velocity of the UAVs is 10m/s, and the minimum turning radius is 50m. For each simulation, the target positions and resources and the UAV positions and resources are randomly generated. The sensor range of the UAVs is 100m. For a given number of targets, we find the performance in terms of mission completion time by varying the number of agents to 5, 10, 15 and 20, while changing the number of targets 5, 10, 15 and 20. Each simulation was carried out for 1000 seconds.

The performance of the mission is determined as follows: (i) the agents are placed with some resources in the search space; (ii) a coalition is formed to attack targets as they are detected; (iii) when all the targets are prosecuted, the simulation is stopped and the time to complete the mission is recorded (iv) When the agents do not have sufficient

resources to prosecute all the targets then they accomplish as much as they can and the simulation will last for 1000 seconds.

The performance curves for the Monte-Carlo simulations are shown in Figure 4. In Figure 4(a) there are four curves and each curves represents the average time taken for different number of agents to prosecute given number of targets in 100 simulations. The curves in Figure 4(b) shows the average percentage of the mission not completed, where the average percentage is calculated as

$$\% \text{ mission not completed} = \frac{\text{Number of targets left}}{M} \times 100$$

Consider first the performance of the 5 targets curve for different numbers of UAVs as shown in 4(a). From the figure we can see that with increased agents the time to complete the mission decreases. When the number of UAVs are 5, the percentage mission not completed is around 0.8%, and the average time to accomplish the mission is high. But, for 10 and 20 UAVs, the time to prosecute all the targets is low because the percentage of the mission not completed is around 0.08%, which is very low. Intuitively, performance curves should monotonically decrease as the number of agents increases, but the average time for 15 UAVs is greater than 10 UAVs. This is because for 15 UAVs the percentage of the mission not complete is 0.13% which is greater than the percentage mission not completed for 10 UAVs (0.08%), hence the average time is greater than 10 UAVs. The percentage mission not completed in higher because the agents did not have sufficient resources. Therefore, the time to complete the mission depends on the resource distribution given to the agents.

For 10 targets, the average time to complete the mission decreases monotonically for 5, 10, and 15 UAVs, but the average time increases for 20 UAVs. In Figure 4(b) we can see for the 15 targets case that the percentage mission not completed for 20 UAVs is marginally higher than 15 UAVs case, and hence the average time to complete the mission with 20 UAVs is higher than 15 UAVs case. For 15 and 20 targets case, the performance curves have monotonic response but they have higher average mission time of 1000 sec for 5 UAV case because the average percentage mission not completed is high due to lack of sufficient resources.

Hypothesis: When a coalition leader receives proposals for possible coalition members then it has to use Algorithm 1 to determine the optimal coalition members for the task. The most computationally intensive part of Algorithm 1 is stage 2. Intuitively, with an increase in number of agents the computational time should increase. But this may not always be true. With an increase in the number of agents the resources can also differ, therefore there is high probability that with fewer number of agents the mission can still be accomplished.

In order to show evidence for the above hypothesis, we recorded the time taken to form a coalition for a given number of UAVs and different target distribution, as shown in Figure 5. From the figure we can see that the computational time for a given number of UAVs is almost constant for different target distributions. Similarly, the computational time to form a coalition is almost constant for different UAVs with small standard deviation. This shows that the number of agents entering into stage 2 is low and hence low computational time. This result also shows that the two-stage algorithm is scalable for large number agents and target distributions. The standard deviation is low which also shows that the time taken to form a coalition is almost constant.

V. Conclusion

We proposed a two-stage optimal coalition formation algorithm for a multiple UAV search and prosecute mission. The UAVs search for targets in the search space and when found a coalition of UAVs is formed whose total resources meet the target resource requirement. Once a coalition is formed, the coalition members adjust their paths using Dubins curves to prosecute the target simultaneously that results in inducing maximum damage. Monte-Carlo simulation results are presented to show that the computational time taken by the coalition formation algorithms is almost constant with increase in number targets and UAVs. The results also show that for a given number of targets, increase in number of agents results in lower mission completion time.

References

[1] B. Gerkey, and M.J. Mataric: A Formal Framework for the Study of Task Allocation in Multi-Robot Systems, *International Journal of Robotics Research*, Vol. 23, No.9, Sep 2004, pp. 939-954.

[2] C. Schumacher and P. Chandler: UAV task assignment with timing constraints via mixed-integer linear programming, *AIAA Unmanned Unlimited Technical Conference, Workshop and Exhibit*, Chicago, Illinois, Sept. 2004, AIAA-2004-6410.

[3] M. Alighanbari and J. How: Robust decentralized task assignment for cooperative UAVs, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, Aug. 21-24, 2006.

[4] P.B. Sujit, A. Sinha, and D. Ghose: Multi-UAV task allocation using team theory, *Proc. of the IEEE conference on Decision and Control, and European Control Conference*, Seville, Spain, Dec. 2005, pp. 1497 - 1502.

[5] D.B. Kingston and C.J. Schumacher: Time-dependent cooperative assignment, *Proc. of the American Control Conference*, Portland, Oregon, June 2005, pp. 4084- 4089.

[6] O. M. Shehory: Methods for task allocation via agent coalition formation, *Artificial Intelligence*, 1998, Vol. 101, No. 12, pp. 165200.

[7] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme: Coalition structure generation with worst case guarantees, *Artificial Intelligence*, 1999, Vol. 111, No. 12, pp. 209238.

[8] L. Vig and J. A. Adams: Multi-robot coalition formation, *IEEE Transactions on Robotics*, Vol. 22, No. 4, Aug. 2006, pp. 637 - 649.

[9] L. E. Parker and F. Tang: Building multi-robot coalitions through automated task solution synthesis, *Proc. of the IEEE special issue on Multi-Robot Systems*, Vol. 94, No. 7, 2006, pp. 1289-1305.

[10] L. E. Dubins: On curves of minimal length with a constraint on average curvature and prescribed initial and terminal positions and tangents, *American Journal of Mathematics*, 1957, Vol. 79, pp. 497-516.

[11] C.H. Papadimitriou and K. Steiglitz: *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.