in Security and Defense Applications (CISDA 2009)

# Multiple UAV Teams for Multiple Tasks

P.B. Sujit, Joao Sousa, and Fernando Pereira

***Abstract*— In a search and prosecute mission, multiple heterogeneous unmanned aerial vehicles UAVs that carry different resources need to perform the classify, prosecute and battle damage assessment (BDA) tasks on targets sequentially. Depending on the target resource requirement, it may be necessary to deploy a coalition of UAVs to perform the action. In this paper, we propose coalition formation algorithms that have low computational overhead to determine coalitions for the prosecute and the BDA tasks. We also develop a simultaneous strike mechanism based on Dubins curves for the UAVs to prosecute the target simultaneously. Monte-Carlo simulation results are presented to show how the algorithms work and the effect of increasing the number of BDA tasks on the mission performance.**

## I. Introduction

Unmanned aerial vehicles (UAVs) with their ability to carry out tasks autonomously have changed the modus operandi of missions like surveillance, search, attack, monitoring, etc. In a *search-and-prosecute* mission, multiple heterogenous UAVs are deployed to carry out the target classification task, prosecuting the target task, and battle damage assessment (BDA) tasks sequentially. Currently, the UAVs do not possess sufficient intelligence to carry out the classification task autonomously. Therefore to classify a target, the UAVs use the help of the command and control center (C3) to classify a target.

When a target is classified with a confidence, the C3 allows the UAVs to strike the target. However, to prosecute a target completely a set of resources may be need that may not be available with a single UAV. Hence, the UAVs need to cooperate with each other and autonomously form a coalition satisfying the desired target resources that can prosecute the target *simultaneously* and in *minimum time*. Prosecuting the target simultaneously induces maximum damage to the target, while prosecuting the target in minimum time preserves fuel and accomplishes the mission quickly. We refer to this sub-team of UAVs as a *coalition*, UAVs in the coalition as *coalition members* and the UAV that detected the target as the *coalition leader*. The coalitions formed are temporary by nature; once the target is prosecuted, the coalition no longer exists and the coalition members can perform other tasks.

Once, the target is prosecuted, a BDA task is carried out. This task appears after prosecute task and has to be attended within a pre-defined time interval to acquire situational awareness (SA) of the target. Depending on the kind of resources pelted on the target, different types of sensors may be needed for SA. Therefore, again a coalition of agents may be required. If the target is not prosecuted to the desired requirement then the UAVs have to re-engage the target.

There is a close coupling associated with the tasks. Since some targets require BDA to be carried out within an interval, the UAVs need to ensure that a coalition can be formed for the BDA task before prosecuting the target. To carry out such a mission, where, tasks appear sequentially and are coupled is a difficult problem. The difficulty is further increased as the coalition formation problem is NP-Hard, the UAV resources deplete with use, and the UAVs have to form coalitions such that the target is prosecuted in minimum time and simultaneously.

A coalition is a group of team members that have agreed to cooperate with each other to execute a single task. Determining the optimal coalition from a group of agents is a computationally intensive task and is NP-hard [1] due to the size of the coalition structure. Fortunately, there are algorithms that provide approximate and near-optimal solutions [2]. The coalition formation algorithms developed in the multi-agent community [1] cannot be directly applied to multiple robot systems [2], since the resources cannot be transferred from one robot to another. Vig and Adams [2] developed a coalition scheme where the tasks act as agents and perform the function of an auctioneer for gathering bids and determining the coalition using RACHNA. Parker and Feng [3] present a coalition formation scheme where a coalition leader robot broadcasts the existence of a task and other robots reply by providing their availability. The leader robot evaluates all possible coalitions and sends an accept decision to the robots that it considers suitable. The task is executed by sharing the sensor information. Although [2] and [3], are closely related to our problem, both the approaches do not deal with sequential or close coupling tasks. Also, the coalition formation algorithm need to be computationally cheap to be implemented on UAVs that travel at higher velocity than robots.

The multiple UAV task assignment problems with multiple tasks have been addressed in [5] and [4], however the algorithms are offline while we are addressing an online task assignment problem. Once a coalition is determined, the coalition members have to prosecute the target simultaneously. Mclain and Beard [6] determine a time when the rendezvous should take place and using a consensus algorithm that changes the velocity to achieve the rendezvous. For rendezvous, the algorithm requires to communicate continuously. In our simultaneous strike scheme the coalition leader determines the agent with the latest strike time to the target and establishes this latest time as the rendezvous time. Once the coalition members receive the rendezvous time information, they replan their paths to meet the rendezvous

P.B. Sujit, Joao Sousa and Fernado Pereira are with the Department of Electrical and Computer Engineering, University of Porto, Portugal. (Email: sujit+jtasso+flp@fe.up.pt)

978-1-4244-3764-1/09/$25.00 ©2009 IEEE

constraint and there is no further communication between the agents. In the proposed scheme the communication happens only once thus reducing communication requirement in prosecuting the target.

Due to the presence of multiple tasks that the UAVs have to attend, we carry out the mission with two classes of vehicles namely, the prosecute class UAVs and the BDA class UAVs. The prosecute class UAVs perform the action of detecting the target and prosecuting the targets by forming the coalitions in association with the C3. While the BDA class of UAVs carry various sensors performing the BDA task. Earlier in [8], we developed a class of coalition formation algorithms that have low computational complexity for search and prosecute tasks that assume the target will be completely destroyed when prosecuted. The assumption may not be true always and requires a BDA to be carried out. With multiple tasks and due to their coupling it is difficult to solve the problem. In this paper, we use the results developed in [8], and develop a mechanism to solve the multiple tasks problem.

## II. Problem Formulation

### A. The mission

A classify, prosecute and BDA mission is carried out on a battlefield using $N$ prosecute class UAVs (also called agents), $O$ BDA class UAVs and $M$ targets whose initial positions are unknown. Each prosecute class UAV has a unique token number $A_i, i = 1, \dots, N$, and the BDA class UAVs have token numbers $B_o, o = 1, \dots, O$. We assume that the prosecute class UAVs have the capacity to carry $n$ types of resources represented by a capability vector $\mathcal{R}_i^A$ of the form:

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

where $R_{ip}^A,\ p = 1, \dots, n$ represents the number of type-$p$ resources held by agent $A_i$. For example, $\mathcal{R}_i^A =< 4, 2, 0, 6 >$ implies that agent $A_i$ has four type-1 of resources ($R_{i1}^A = 4$), two type-2 of resources ($R_{i2}^A = 2$), zero type-3 resources ($R_{i3}^A = 0$), and six type-4 of resources ($R_{i4}^A = 6$). Similarly, the BDA class UAVs have the resource vector of the form:

$$\mathcal{R}_o^B =< R_{o1}^B, \dots, R_{oz}^B >, \quad o = 1, \dots, O. \tag{2}$$

Unlike prosecute UAVs, whose resources deplete with use, the BDA resources are constant as they are sensors.

We assume that the UAVs have limited sensor range ($r_s^i$) and they have to perform a search task to locate targets. Once $A_i$ detects a target $T_j$, it becomes the coalition leader for the target and with the help of the C3, $A_i$ classifies the target and also determines the type and quantity of resources required to prosecute the target. If $m-$different types of resources and quantities are required to engage target $T_j$, then the resource requirement vector is represented as

$$\mathcal{R}_j^T =< R_{j1}^T, \dots, R_{jm}^T >, \quad j = 1, \dots, M \tag{3}$$

where $R_{jq}^T,\ q = 1, \dots, m$ and $m <= n$, represents the quantity of type-$q$ resources required to prosecute the target $T_j$. For example: $\mathcal{R}_j^T =< 3, 0, 5 >$ indicates that to prosecute target $T_j$, the agents need three type-1 resources ($R_{j1}^T = 3$), zero type-2 resources ($R_{j2}^T = 0$), and five type-3 resources ($R_{j3}^T = 5$).

The coalition leader $A_i$ has to form a coalition to prosecute the target and a coalition to carry out the BDA task within the specified time interval. The coalition leader $A_i$ broadcasts the target resource requirement vector ($\mathcal{R}_j^T$).

When each agent receives a proposal to form a coalition, then the agent determines if it can contribute at least one resource to the coalition. If so, then the agent will respond to the coalition by sending their resources and the cost, otherwise the agent will not respond. The cost is the minimum distance taken to arrive at the target using the Dubins curves [7]. The task for $A_i$ is to select the coalition members such that the target is prosecuted satisfying other constraints that include (i) minimizing the time-to-attack the target and (ii) minimizing the number of UAVs attacking the target. The coalition formed by agent $A_i$ for target $T_j$ is represented as $C_j^i$ and the coalition resources are represented as $\mathcal{R}_j^{C^i}$. The objective (i) allows the mission to complete quickly with lower fuel consumption, while the objective (ii) ensures that the UAVs distribute their search effort in the search region so that the targets are detected quickly and the mission is accomplished faster.

Assume that the coalition leader $A_i$ receives $N$ proposals from potential coalition members. Let the set representing the cost of the potential coalition members be $\Lambda = \{\lambda_i, \dots, \lambda_N\}$, where $\lambda_k$ is the cost of the agent $A_k$ to arrive at location of target $T_j$. The optimization problem that the coalition leader has to solve is given as:

$$\text{Objective}: \quad \min_{\hat{\Lambda}} \max_{k \in \hat{\Lambda}} \lambda_k \tag{4}$$

$$\text{Subject to}: \quad \sum_{k=1}^{|\hat{\Lambda}|} R_{kp}^A \geq R_{jp}^T, \text{ for all } p = 1, \dots, m \tag{5}$$

where $\hat{\Lambda} \subseteq \Lambda$, represents the smallest coalition of $\Lambda$. The objective function determines the smallest coalition $\hat{\Lambda}$ that has the latest arrival time $\lambda_k$. We use the latest arrive time and not the earliest arrival time because, the latest arrival time of $\hat{\Lambda}$ determines the earliest simultaneous strike time for the coalition.

The solution to the optimization problem (Equations 4 and 5) can be computationally intensive as the number of agents increase. In this paper, we reduce the problem complexity by separating the problem into two-stages. In the first stage, we will determine a minimum time coalition set, and using this set we determine the smallest coalition that satisfies the constraints. We develop both polynomial time as well as optimal coalition algorithms.

Similar to the prosecute task, the coalition leader has to determine a coalition for the BDA task. Since, BDA and the prosecute tasks are coupled, we propose a mechanism where the coalition leader first determines a prosecute coalition, then determines if a BDA coalition can be formed. If both the coalitions can be formed, only then the coalition leader will broadcast the formation of the prosecute and BDA coalitions, otherwise it will dissolve the coalition formation request for the both the tasks.

### B. UAV Kinematics

Both the classes of UAVs are subjected to kinematic constraints preventing instantaneous course changes. We assume that each UAV is located at different unique altitudes hence collision avoidance is not an issue. We also assume that each UAV has a different velocity $v_i$ and the autopilots of the UAVs hold the altitude and maintain the ground speed. The kinematics of the UAVs are modeled using first order kinematics as

$$\begin{aligned} \dot{x}_i &= v_i \cos\psi_i \\ \dot{y}_i &= v_i \sin\psi_i \\ \dot{\psi}_i &= k(\psi_d - \psi_i) \end{aligned} \tag{6}$$

where $\psi_d$ is the desired heading of the UAV, $v_i$ is the ground track, and $k$ is the autopilot gain. We assume heading rate is constrained to

$$-\omega_{\max} \leq \psi \leq \omega_{\max} \tag{7}$$

The UAVs have different velocities and carry various resources of different types. Hence these UAVs are heterogenous that form coalitions to execute the assigned tasks cooperatively.

## III. Coalition Formation Algorithms

The sequence of tasks that the UAVs have to carry out are classify, prosecute and BDA. Since, the classification is carried out with the help of C3, the UAV have to autonomously carry out the prosecute and BDA tasks. An interesting aspect of the prosecute and BDA task is the coupling. Unless a target is prosecuted, BDA task is not released. But if there are no agents to carry out the BDA within a given interval of executing the prosecution task, then the prosecute task should not be performed.

On the other hand, if we form the BDA coalition and then create the prosecute coalition for the target, then there may be situations where prosecute coalition may not be formed. Thus complicating the situation. In order to carry out the mission smoothly we perform the following sequence of actions.

Step 1 The coalition leader broadcasts for a coalition. The rest of the agents respond to the coalition formation broadcast.

Step 2 The coalition leader determines if a coalition can be formed or not for a prosecute task.

Step 3 If a prosecute coalition cannot be formed then the coalition is dissolved, otherwise the coalition leader broadcasts for a BDA task. Since, the coalition leader knows the time the coalition will strike the target, it will use this time and determines a time interval during which the BDA has to be carried out. Note, the coalition leader has not yet broadcasted to the coalition members about the prosecute coalition decision. This process if required to determine Whether a BDA coalition is desired or not.

Step 4 The BDA agents will respond to the request.

Step 5 If no BDA coalition can be formed with the desired resources and within the pre-determined interval, then the coalition leaders dissolves both the BDA coalition and the prosecute coalition. Otherwise, it will broadcast the coalition formation for the prosecute task and BDA task. Note, only after confirming the availability of agents with resources to perform both the tasks, the coalition leader broadcasts its decision.

Using steps1-4, the agents determine whether the coalitions for the prosecute and BDA tasks will be formed or not. These steps show the way agents process the sequential and coupled tasks. Initially, we will present the coalition formation algorithms for the prosecute task and then show how the same algorithms can be extended for BDA task.

### A. Coalition formation algorithms for prosecute task

Forming a coalition for a target depends on (i) the coalition leader may have sufficient resources to prosecute the target by itself and (ii) the coalition leader does not have sufficient resources in that case it has to form a coalition with other agents.

When the agent $A_i$ detects target $T_j$ that requires $\mathcal{R}_j^T$ resources. If $R_{ip}^A \geq R_{jp}^T, \ \forall \ p = 1, \ldots, m$, then $A_i$ would attack target $T_j$ without requesting a coalition with other UAVs. When an agent $A_i$ detects target $T_j$ and has insufficient resources then the coalition leader has to form a coalition. The coalition leader broadcasts the information about the target (i.e, its location and required resources) to the other UAVs to form a coalition. The agents that have at least one type of the required resource will send their cost to arrive at the target and the type and quantity of the available resources. There may be situations where the agent can receive many coalition formation requests from different coalition leaders. Once simple mechanism is that the agent can send its proposal to the coalition leader that is the nearest. However, this mechanism does not guarantee that a coalition will be formed for all the coalition leaders and a deadlock will not be formed. In order to ensure that at least one coalition is formed, we design a rule that the agent will send response to that coalition leader whose agent Id is the highest. This rule avoids a situation where none of the agents can determine a coalition that results in delaying the mission completion time.

The coalition leader considers all the responses and forms a coalition. The selection of the team members for the coalition is carried out by solving the optimization problem (Equations 4 and 5). Since solving the optimization problem is computationally intensive, we developed two coalition formation algorithms: (i) Polynomial time coalition (that is sub-optimal) and (ii) Optimal coalition. Both the algorithms use a two-stage mechanism to produce solutions that have low computational complexity. The optimal coalition is a coalition formed for a single target and it is not optimal for the entire mission. These two algorithms were proposed in [8].

*1) Polynomial time coalition formation algorithm (PTCFA):* Determining the minimum time and the smallest coalition that would successfully prosecute the target can be accomplished in two stages. In the first stage, we determine the set of all UAVs that can achieve the minimum time requirement and then we prune this set to achieve the minimum member coalition in the second stage. The process to achieve this task is shown by Algorithms 1 and 2. We assume that agent $A_i$ is the coalition leader and it has detected target $T_j$ that requires $\mathcal{R}_j^T$ resources.

**Algorithm 1** First stage of the PTCFA.

```
 1: Initialize:
 2: $C_j^i$ = [ ]; $\mathcal{R}_j^{C^i}$ = [ ]; agents_responded = $\{A_1, A_2, \cdots, A_N\}$
 3: arrival_times = $\{\lambda_1, \cdots, \lambda_N\}$; $\Lambda_c = []$;
 4: Stage 1:
 5: $[\Lambda_u,\ \Lambda_a] = Sort$ (arrival_times); % $\Lambda_u \leftarrow$ sorted ETAT, $\Lambda_a \leftarrow$ corresponding agent index of $\Lambda_u$
 6: for $k = 1$ to |agents_responded| do
 7:     $A_q = \Lambda_a(k)$;
 8:     $C_j^i \leftarrow$ append $A_q$ to $C_j^i$
 9:     $\mathcal{R}_j^{C^i} \leftarrow \mathcal{R}_j^{C^i} + \mathcal{R}_q^A$
10:     $\Lambda_c \leftarrow$ append $\Lambda_u(k)$;
11:     if $k > 1$ then
12:         if $R_{jp}^{C^i} >= R_{jp}^T,$ for all $p$ and $\Lambda_u(k) > \Lambda_u(k-1)$ then
13:             BREAK
14:         else
15:             CONTINUE
16:         end if
17:     else
18:         if $R_{jp}^{C^i} >= R_{jp}^T,$ for all $p$ then
19:             BREAK
20:         else
21:             CONTINUE
22:         end if
23:     end if
24: end for
```

**Algorithm 2** Second stage of the PTCFA.

```
 1: Stage 2:
 2: $\hat{C}_j^i$ = $C_j^i$;
 3: for $k = 1 : |C_j^i|$ do
 4:     $A_q = C_j^i(k)$;
 5:     $\hat{\mathcal{R}}_j^{C^i} = \mathcal{R}_j^{C^i} - \mathcal{R}_j^{C^q}$
 6:     if $R_{jp}^{C^i} >= R_{jp}^T,\ \forall\ p$ then
 7:         $\hat{C}_j^i \leftarrow$ remove $A_q$ from $\hat{C}_j^i$
 8:         $\mathcal{R}_j^{C^i} = \hat{\mathcal{R}}_j^{C^i}$
 9:     end if
10: end for
11: $C_j^i$ = $\hat{C}_j^i$;
```

The Algorithm 1 begins with initializing the coalition set and the coalition resources set to empty sets (line 2). First the coalition leader sorts the responses in the ascending order of cost (line 5). To determine the coalition, we take one agent ($A_q$) at a time (line 7), include $A_q$ in the coalition $C_j^i$ (line 8), update the coalition resources set $\mathcal{R}_j^{C^i}$ (line 9) and the coalition set $\Lambda_c$ (line 10). Then check if the target resource constraint is met by the (line 12 or 18). That is, $R_{jp}^{C^i} \geq R_{jp}^T$, for all $p$. When the constraint is not met, then the process of including the next agent and its resources and verifying the resource constraint continues till the target resource constraint is met. The minimum time to strike the target is determined by $\lambda_{\max} = \max \Lambda_c$.

*Theorem 1:* Algorithm 1 generates the optimal minimal time coalition set.

*proof:* The proof is given in [8].□

Once the minimum time coalition is formed by Algorithm 1, we need to prune those members whose resources are not required to form a minimalist coalition. This process is carried out using algorithm 2. In the second stage, we check if the resource of agent $A_q \in C_j^i$ are required for the coalition or not by removing its resources from $\mathcal{R}_j^{C^i}$ (line 5). If $A_q$ resources are not required (line 6), then the agent $A_q$ is removed from $C_j^i$ (line 7) and its resources are deducted from the $\mathcal{R}_j^{C^i}$ (line 8), otherwise $A_q$ and its resources are not removed. This process is carried out for all the agents $A_q \in C_j^i$. The process of the second stage is described in Algorithm 2.

*2) Optimal coalition formation algorithm (OCFA):* To determine the optimal coalition, we begin with the minimum time coalition generated by Algorithm 1 and formulate an integer programming problem to determine the least number of agents that have sufficient resources to prosecute the target. The formulation of the problem is given in Algorithm 3.

**Algorithm 3** Optimal coalition formation algorithm.

```
1: Stage 1:
2: Use Algorithm 1 to determine $\Lambda_c$
3: Stage 2:
4: $\lambda_{\max} = \max \Lambda_c$;
5: Solve:
```

$$\begin{aligned} Objective\ Function : \min_{N'} N'\lambda_{\max} \\ \text{subject to } \textstyle\sum_{i=1}^{N'} R_{ip}^A \geq R_{jp}^T, \forall p = 1, \ldots, m; A_i \in C_j^i \end{aligned} \quad (8)$$

In order to strike simultaneously, each agent of the coalition must travel the same distance. The maximum distance to be traveled is given in Step: 4 of Algorithm 3. Assuming $N'$ agents are present in the coalition set $C_j^i$, the cost of the coalition is $N'D_{\max}$. By minimizing $N'$, we can minimize the cost of the objective function. Hence, the optimization problem in Algorithm 3 will yield the optimal minimum member coalition, while the Algorithm 3 determines the optimal minimum time and minimum member coalition.

*3) Complexity analysis:* Now we analyze the complexity of the PTCFA and OCFA algorithms. The PTCFA produces a sub-optimal solution that has polynomial time complexity.

*Theorem 2:* The computational complexity of the PTCFA is $O(N(\log N + 2m))$.
*proof:* See [8] for proof. □

The computational complexity for the optimal coalition algorithm has a polynomial time complexity for the first stage. But we use an integer programming technique in the second stage. Although solving an integer programming problem is *NP*-hard, there are pseudo-polynomial algorithms to generate optimal solutions [9]. To solve the integer programming problem in the second stage, we used the *bintprog* command in MATLAB. If $N$ is large, then the computational time depends the number of agents and the quantity of their resources. During the initial phase, the UAV resources are abundant 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. However, in the final stages, the UAVs may have fewer resources resulting in larger coalition size. As the coalition size becomes large the computational time increases. The above analysis indicates that the computational time for the coalition formation mainly depends on the distribution of the UAV resources.

### B. Coalition formation for BDA task

Once the coalition leader determines a coalition for prosecute task, then it has to determine the coalition for the BDA task. The coalition broadcasts the resources required to carry out the BDA task given by $\mathcal{R}_j^{TB}$ and the time interval ($[\underline{\Gamma_j}\ \overline{\Gamma_j}]$) between which the BDA agents have to carry out the task.

Since, the target $T_j$ will be prosecuted at time $\lambda_{\max}$, the coalition leader determines the minimum time to carry out the BDA as $\underline{\Gamma_j} = \lambda_{\max} + \delta$, where $\delta$ represents the time the target is available after prosecute task and the maximum time to complete the BDA task as $\overline{\Gamma_j} = \underline{\Gamma_j} + \Delta$, where $\Delta$ is the desired interval. The BDA agents that are free in the region and have the required resources $R_{io}^B \in \mathcal{R}_j^{TB}$ for some $o, o = 1, \ldots, O$ will respond to the coalition request.

Now, the coalition leader has to choose the BDA coalition members for the task. Since, the task has to be carried out within an interval, the coalition leader does not have to carry out the first stage of the PTCFA or OCFA algorithms. The coalition leader can use the second stage and determine the minimum number of agents required to perform the task. However, there may be a necessary to reduce the time to perform the BDA task and reduce the number of agents performing the task by using PTCFA or OCFA algorithms. In this paper, we use OCFA to determine the minimum time and minimum member coalition for the BDA task as well.

Once, the coalition for the BDA is determined, the coalition leader broadcasts the selection of the members for the prosecute task as well as the BDA task. During the process of determining the BDA coalition, if the prosecute agents receive coalition formation requests from other agents, then they will not respond. Although this is a hard rule, it will not complicate the process of forming coalitions. If the coalition leader was unable to determine a BDA coalition then it will dissolve the prosecute coalition request and the BDA coalition request, since these two tasks are coupled.

When a coalition for both the tasks can be formed, the coalition leader broadcasts the selection of the coalition members and the desired ETAT to the prosecute class of UAVs and BDA agents respectively. Once, the coalition members receive acceptance to be part of the coalition, they replan their path to meet the simultaneous strike condition. Both, the prosecute coalition and the BDA coalition members replan their paths using the simultaneous strike mechanism based on dubins curves as described in the next section.

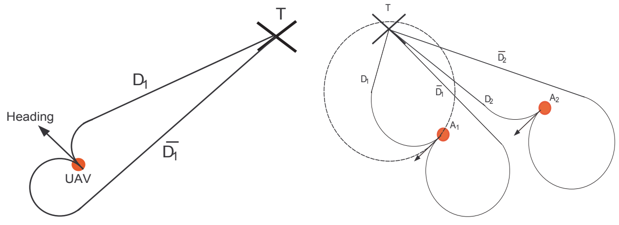
## IV. Simultaneous strike

The coalition leader $A_i$ determines the agent that takes the latest time $\lambda_{\max}$ to arrive at the target location and broadcasts this information for simultaneous strike. Each coalition member adjusts its path such that the time taken using the path is equal to $\lambda_{\max}$. Since, we use Dubin curves for simultaneous strike, therefore the radius $\underline{r}$ of the Dubins curve can be altered such that the time taken by the agent is equal to $\lambda_{\max}$. Similarly, the BDA agents alter their turn radius such that they meet $\lambda_{\max}$ constraint given by OCFA.

Given an agent position and heading angle, we can determine two Dubins paths to the target: (i) Dubins shortest path ( $D_1$) and (ii) Dubins longest path ($\bar{D_1}$) as shown in Figure 1(a). The agents use Dubins path for determining the earliest arrival time to the target and simultaneously strike condition. If we choose (i) as the metric then the agent can reach the target in the shortest time. But, for the simultaneous strike condition, the agent has to increase its radius. When this process is carried out there may be situations where the radius is large enough that it encircles the target as shown in Figure 1(b). Hence, using Dubins shortest path not the correct method. Therefore, we choose Dubins longest path as the metric as well as the mechanism for simultaneous arrival condition. Unlike (i), (ii) does not suffer any situation is encircling the target.

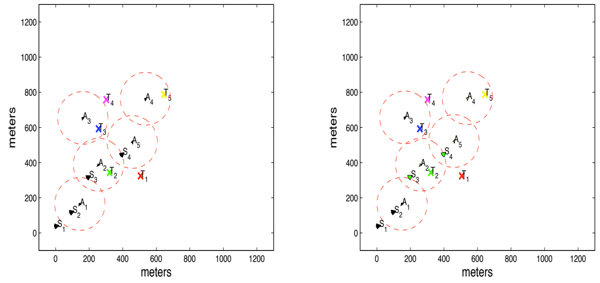
Assume that agent $A_i$ is the coalition leader for target $T_j$ and let the coalition be $\mathcal{C}_j^i$. Assume that agent $A_{k'} \in \mathcal{C}_j^i$ is the agent that has the latest arrival time be $\lambda_{\max}$). Agent $A_i$ broadcasts this latest time $\lambda_{\max}$ to the members of the coalition as the time to simultaneously prosecute the target $T_j$. Each member $A_k \in \mathcal{C}_j^i, k = 1, \ldots, |\mathcal{C}_j^i|$ have to adjust their path length accordingly so that $\lambda_k$ is equal to $\lambda_{\max}$. In order to achieve that, the agents need to determine the required turning radius $\underline{r}_k$ such that $\lambda_k = \lambda_{\max}$. Since, $\underline{r}_k$ cannot be calculated using a closed form solution, we calculate $\underline{r}_k$ iteratively until the condition $\lambda_k = \lambda_{\max}$ is satisfied.

Note that each UAV's Dubins longest path for the simultaneous strike or BDA task can be found by adjusting its radius and it is simple to compute. One may argue that the Dubins longest may not be the minimal path towards the target as we can modify the shortest Dubins path such that the new path is shorter than the longest path. Although it is true, but the number of rules that are required to carry out such



(a) (b)

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



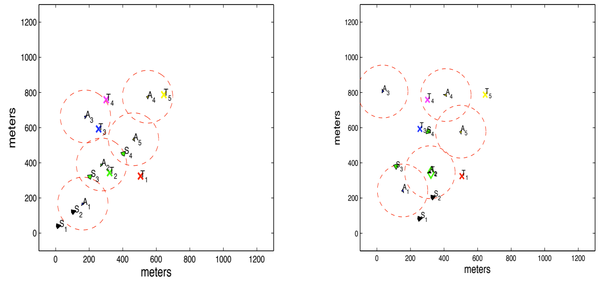
(a) (b)

Fig. 2. (a) Initial configuration of the agents and the targets (b) Prosecute coalitions of $A_2$ to $T_2$ and agents $A_4$ and $A_5$ to target $T_5$ are formed, while BDA coalition of agents $S_3$ and $S_4$ is formed for $T_2$ at time $t = 0.3s$.

modifications are higher compared to the proposed the simple strategy. We are interested in designing a generic model with a single rule, hence we adopted the Dubins longest path. In the worst case, the time needed to prosecute target $T_j$ increases by $\frac{2\pi r_{\min}}{V}$ seconds as compared to the time taken using Dubins shortest path.

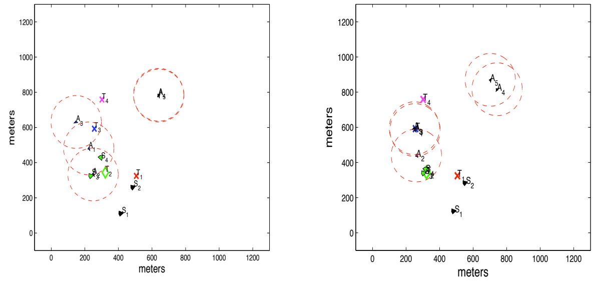
## V. Simulation Results

The performance of the coalition algorithms for search-prosecute-BDA tasks is evaluated using simulations. Initially, we will present simulations to show how the coalitions are formed and then present Monte-Carlo simulations with varying number of BDA targets.



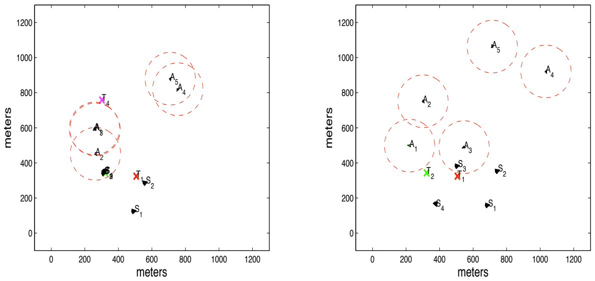
(a) (b)

Fig. 3. (a) Prosecute coalition consisting of agents $A_3$ and $A_1$ formed for target $T_4$ at time $t = 0.9s$ (b) Target $T_2$ prosecuted by $A_2$ and a BDA task is initiated at time $t = 19.6s$.



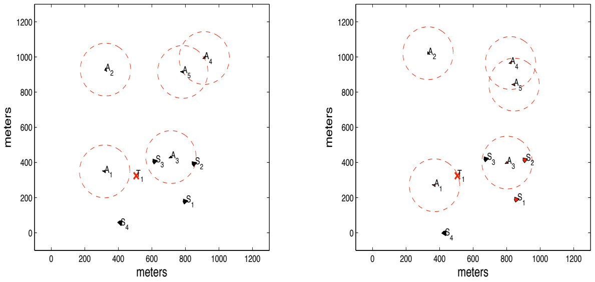
(a) (b)

Fig. 4. (a) Target $T_5$ prosecuted successfully at time $t = 32s$ (b) Target $T_3$ is successfully executed at time $t = 37.4s$.



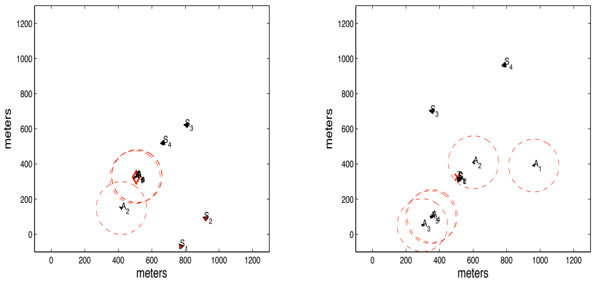
(a) (b)

Fig. 5. (a) BDA task on $T_2$ is completed as the target has to be re-engaged with additional resources (b) Target $T_4$ is prosecuted by $A_2$ at time $t = 53.1s$.



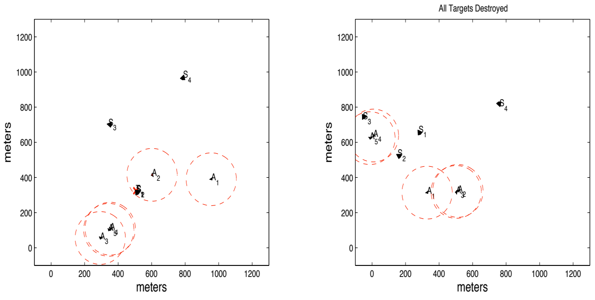
(a) (b)

Fig. 6. (a) $A_1$ prosecutes target $T_2$ at time $t = 62s$ (b) Prosecute coalition of $A_1, A_3, A_4, A_5$ is formed with the BDA coalition members of $S_1$ and $S_2$ at time $t = 66.7s$.



(a) (b)

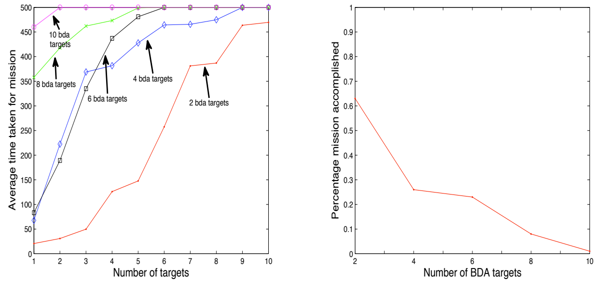
Fig. 7. (a) Target $T_1$ is prosecuted and a BDA task is instantiated at time $t = 117.9s$ (b) BDA task is carried out on $T_1$ and additional resources are required to completely prosecute it at time $t = 153.2s$.



(a)

(b)

Fig. 8. (a) Prosecute coalition consisting of $A_2$ and $A_3$ is formed to prosecute $T_1$ at time $t = 153.5s$ (b) Target $T_1$ is prosecuted successfully completing the mission in time $t = 185.1s$



(a)

(b)

Fig. 9. (a) Average time taken to accomplish the mission for different number of BDA targets (b) Average mission accomplished for different number of BDA targets.

### A. Sample scenario

We present a sample scenario to show how the coalitions for prosecute and BDA tasks are formed using the coalition formation algorithms presented in Section III. We consider a classify, prosecute and BDA mission with five prosecute class UAVs, four BDA class UAVs. The prosecute class agent resources are $\mathcal{R}_1^A = \{3,2,3\}, \mathcal{R}_2^A = \{6,5,3\}, \mathcal{R}_3^A = \{2,1,4\}, \mathcal{R}_4^A = \{4,2,1\}, \mathcal{R}_5^A = \{1,2,2\}$. The sensor range $R_i^s$ is 100m and the velocity of the vehicles is $20m/s$. The BDA class of vehicles have resources $\mathcal{R}_1^B = \{0,1,1\}, \mathcal{R}_2^B = \{1,1,1\}, \mathcal{R}_3^B = \{\}, \mathcal{R}_4^B = \{\}$ and the velocity of the vehicles is $15m/s$. The region of interest is $1000 \times 1000\ m^2$ area with five targets having resources $\mathcal{R}_1^T = \{3,2,1\}, \mathcal{R}_2^T = \{2,3,1\}, \mathcal{R}_3^T = \{4,1,0\}, \mathcal{R}_4^T = \{3,1,2\}, \mathcal{R}_5^T = \{2,1,3\}$. We assume that, only the targets $T_1$ and $T_2$ need BDA after prosecuting them.

The initial location of the UAVs and the targets is shown in Figure 2(a). The targets are marked as 'X' while the prosecute targets are labeled as $A_1 - A_5$, and the BDA agents are labelled as $S_1 - S_4$. When a coalition is formed, depending on the color of the targets the UAV colors are changed to show their assignment. At the current moment, $A_2$ can detect $T_2$, $A_4$ can detect $T_5$, and $A_3$ can detect $T_3$, hence there are three coalition leaders. Since, $A_2$ has sufficient resources to prosecute $T_2$, it does not send a request for coalition, but $T_2$ requires BDA task to be performed hence it sends a BDA coalition formation request to the BDA agents. The agents $A_4$ and $A_3$ do not have sufficient resources, hence they broadcast for coalition formation request.

At t=0.1s, the agents receive the coalition requests from $A_3$, and $A_5$. Since, the potential coalition members receive more than one proposal, they use the rule defined in Section III, and send proposal to $A_5$ as it has the highest token number and none send their proposal to $A_3$. For the BDA coalition formation request, only agents $S_3$ and $S_4$ respond as $S_1$ and $S_2$ cannot perform the desired tasks in specified interval.

At time t= 0.2s, the coalition leader $A_3$ dissolves the coalition formation request, while agent $A_5$ determines a coalition with agents $A_4$ and $A_5$, while agents $A_1$ and $A_3$ receive a reject decision. At the same time, $A_2$ forms a coalition with $S_3$ and $S_4$ and broadcasts the decision. The coalitions are formed by these at time $t = 0.3s$ and shown in Figure 2(b). Following a similar process, $A_3$ determines a coalition with $A_1$ at time $t = 0.9s$ as shown in Figure 3(a).

The agent $A_2$ prosecutes the target $T_2$ at $t = 19.6s$, since $T_2$ requires a BDA task, the 'X' symbol is changed to a 'diamond' symbol as shown in Figure 3(b). In the figure we can the two BDA agents closing on $T_2$. At $t = 32s$, agents $A_4$ and $A_5$ prosecute target $T_5$ as shown in Figure 4(a). The target $T_5$ does not require the BDA task, hence it is completely destroyed and is not shown in the figure. At $t = 37.4s$, $A_1$ and $A_3$ prosecute $T_3$ completely while $S_3$ and $S_4$ are close to the target $T_2$, to perform the BDA task as shown in Figure 4(b). These BDA agents perform the task at $t = 37.7s$ as shown in Figure 5(a). The agent $A_2$ detects the availability of the target $T_2$ that needs a re-engagement to completely prosecute with resources $\{1,0,1\}$. A single agent coalition is formed by $A_2$ with agent $A_1$ at time $t = 38s$ (not shown in the figures).

At $t = 53.1s$, $A_2$ destroys target $T_4$ as shown in Figure 5(b). In the same figure we can that $A_1$ is approaching target $T_2$ to eliminate it completely and it performs that action at time $t = 62s$ as shown in Figure 6(a). The only remaining agents is target $T_1$ that also requires BDA task.

The agent $A_1$ after prosecuting the target $T_2$ detects target $T_1$ and forms a coalition with agents $A_1, A_4, A_3$ and $A_5$ to prosecute $T_1$ at $t = 66.7s$. Since, this target also requires BDA task, a BDA coalition with agents $S_1$ and $S_2$ is formed by $A_1$ as shown in Figure 6(b). The prosecute coalition prosecutes the $T_1$ at $t = 117.9s$ as shown in Figure 7(a). After prosecution, the BDA task for the target is released and hence the the target symbol is changed to diamond. In the figure we can see that the BDA agents $S_1$ and $S_2$ approaching $T_1$ from below. At $t = 153.2s$, the BDA task is executed on $T_1$ as shown in Figure 7(b) and the new resources required to eliminate the target is determined as $\{0,1,1\}$.

The agent $A_2$ whose sensor range captures the detection of $T_1$ that requires a re-engage to completely destroy it. A coalition consisting of $A_2$ and $A_3$ is formed at $t = 153.5$ as shown in Figure 8(a) and the target is destroyed at $t = 185.1s$ as shown in Figure 8(b). From Figures 2(a)-8(b) we can see how the coalitions are formed for the prosecute and BDA tasks in a systematic fashion.

### B. Monte-Carlo simulations

In order to know the performance of the algorithm it is necessary to conduct Monte-Carlo simulations. For the simulations there are various parameters that can be studied, like, the increase in number of agents, number of targets, number of BDA UAVs. It is a natural phenomena that, for a given number targets and its resources, with increase in number of agents, the mission time decreases correspondingly. When we fix the number of UAVs and increase the number of targets, then the mission time will increase. However, the amount of increase in mission time depends on the UAV resource distribution. Since, these analysis can be concluded based of various studies of increasing agents and targets in the UAV task allocation literature, we will not attempt to carry out these simulation. Instead, we will study the effect in increasing the number of BDA targets with fixed number of targets and UAVs. This study will provide additional insight into the resource distribution.

We carried out Monte-Carlo simulations with 10 targets, five UAVs for search and prosecute task, five BDA UAVs and varying number of BDA targets from 2 to 10. The different number of resources required for the targets is three and the resources for the targets and the UAVs were randomly generated. The mission time for each simulations was fixed to 500 seconds. Figure 9(a) shows the average time taken for the agents to accomplish the mission. From the figure we can see that, with increase in number of BDA targets, the time to accomplish the increases. Because, effectively the number of tasks that are needed to perform also increase considerably. Figure 9(b) shows the average percentage mission accomplished in terms of the number of targets destroyed. From the figure we can see that when the number of BDA tasks are equal to the number of targets, there was no mission that was accomplished completely within the specified mission time. This happened due to non-availability of BDA UAVs within the specified interval of time to carry out the BDA task. These results show that we need to provided resources to UAVs depending on the number of available targets and the number of BDA tasks to be accomplished.

## VI. Conclusion

In this paper, we developed a mechanism of cooperation between two different types of UAVs that can interact to decouple the tasks. However, for BDA and prosecute tasks, we need a coalition of agents to perform the task. Hence, we used our previously developed coalition formation algorithms to solve for individual tasks. For simultaneous strike, the proposed rendezvous mechanism using Dubins curve is also an attractive scheme that does not require constant communication between the agents to achieve rendezvous. In order to carry out the closely couple prosecute and BDA tasks, we developed a mechanism where the prosecute coalition is determined initially and then the BDA coalition is determined. If both the coalitions are formed only then the coalition leader allows the prosecute and BDA tasks to be executed by the coalition members otherwise the coalition is dissolved. This process enables the UAVs to perform the tasks with ease. Theoretical results are established to show that the coalition algorithms have low complexity and a simulation showing the operation of the entire mission is described. Monte-carlo simulations are carried out that show with increase with number of BDA for a given target and UAV distribution, the mission also increases correspondingly.

## References

[1] O. M. Shehory: Methods for task allocation via agent coalition formation, *Artificial Intelligence*, 1998, Vol. 101, No. 12, pp. 165-200.

[2] L. Vig and J. A. Adams: Market-based multi-robot coalition formation, *Proc. of the International Symposium on Distributed Autonomous Robotic Systems*, M. Gini and R. Voyles (Eds.), Springer, 2006, Minneapolis, MN, pp. 227-236.

[3] 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.

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

[5] T. Shima, S. Rasmussen, A. Sparks, and K. M. Passino: Multiple task assignments for cooperating uninhabited aerial vehicles using genetic lgorithms, *Computers and Operations Research*, Vol. 33, No. 11, 2005, pp. 3252-3269.

[6] T. W. McLain, and R. W. Beard: Coordination variables, coordination functions, and cooperative timing missions, *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 1, January, 2005, pp. 150-161.

[7] 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.

[8] P.B. Sujit, J.M. George, and Randy Beard: Multiple UAV coalition formation, *Proc. of the American Control Conference*, Seattle, Washington, June 2008.

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