Coalition Formation for Multiple Heterogeneous UAVs Cooperative Search and Prosecute with Communication Constraints*

LIU Zhong, GAO Xiao-guang, FU Xiao-wei

School of Electronics and Information, Northwestern Polytechnical University, Xi’an, 710072, China
E-mail: 15829732829@163.com

Abstract: To improve the cooperative search and attack effectiveness of multiple heterogeneous unmanned aerial vehicles (UAVs) in unknown environment, a novel coalition formation method with communication constraints is presented in this paper. First, the coalition formation model is established on the basis of minimizing the target attack delay and minimizing the coalition size with the constraint of required resources and simultaneous strike. Second, considering communication constrains such as limited communication ranges and communication delays, a mechanism was developed in order to find the potential coalition members within a maximum number of hops over a dynamic UAV network. Third, to reduce the huge computational complexity in coalition formation optimization solution we propose a Multistage Sub-Optimal Coalition Formation Algorithm (MSOCFA) with low computational complexity. Furthermore, in order to enable multiple cooperative UAVs accomplish the search and prosecute missions autonomously, a distributed autonomous control strategy is proposed which is based on the Finite-State Machine (FSM). Comparison simulations are carried out to demonstrate that how the potential coalition members finding technique impact on the coalition achieved by MSOCFA. The effects of number of maximum allowed hops for a message and hop delay are studied by employing Monte-Carlo method. The experimental reveals that, in the cases of large communication delay, forming a coalition from the immediate neighbors is sufficient for a good performance in term of the mission completion time. Under smaller delays, including neighbors up to a few hops will increase performance, and any additional increase in hop count will degrade performance.

Key Words: multi-UAV; cooperative search and prosecute; coalition formation; communication constraints

1 Introduction

In recent time, the application of unmanned aerial vehicle (UAV) in the field of search-and-prosecute missions of unknown environment has gained a growing development. Apart from the obvious advantage of avoiding the risk of human life, the nonparticipation of human pilots can also guarantee significant weight decrease, cost reductions as well as the possibility of new flight operations. But a single UAV can only carry limited payload, making a deployment of UAVs team being necessary. Because the total munitions of a team is sufficient for the attack on the all targets. Aiming at enhancing the whole team performance, a higher efficient tasks allocation algorithm has to be designed [1]. This paper is devoted to the problem of real-time task assignment for multiple UAVs cooperative searching and attacking targets in unknown environment.

The problem of task assignment for cooperative UAVs addresses at determining the task sequence and precise timing for each team member, which can satisfy all the constraints and minimize an overall objective function for the whole team meanwhile [1]. The problem about assigning the tasks to the UAVs can be interpreted as a complex combinatorial optimization problem. Different authors have developed several task assignment methods for UAVs, such as dynamic network flow optimization (DNFO)[2], mixed integer linear programming (MILP)[3], multidimensional multiple choice knapsack problem (MMKP)[4], contract net[5], satisfying decision theory[6]. However, most of the solutions in these previous works assume that (i)the UAVs are homogeneous, or the UAVs have limited resources that deplete with use; (ii)the task assignment algorithms are computationally intensive, but the task assignment methods with low computational complexity are required for real-time applications; (iii)the number locations and required resources of targets are all known in advance, but in a unknown environment, the distribution of targets cannot be got fully by the UAV ahead of time. Consequently, the algorithms considered in this paper cannot be applied directly.

In the multi-agent system (MAS), when a single agent cannot complete the tasks, a coalition formed by a sub-group of agents will perform the tasks through mutual cooperation. As the coalition formation is temporary, new tasks can be allocated to the members after the tasks completion. Forming a coalition to solving task allocation problems has been applied both in MAS [7] and multi-robot system (MRS) [8]. However the coalition formation algorithms put forward in the MAS cannot be used directly to multi-UAV system because of the failing resources transfer among UAVs. Also, high computation cost and larger amount of communication required in the MAS may not be guaranteed in multi-UAV system because of the fast move feature and inability to stop in the air of UAVs.

The main contribution of this paper is to develop mechanisms to form a sub-team of UAVs named the coalition to destroy the targets. The coalition formation problem is NP-hard. In a complete decentralized setup, a set of UAVs communication with each other to form a coalition to engage a target. In practice, the communication capabilities between the UAVs are restricted by limited communication ranges, delays and so on. Some of the UAVs may get disconnected from the network during the coalition formation process. If a disconnected member happens to be part of the determined coalition, then it cannot be informed about its requirement in the coalition. Hence, the coalition becomes invalid, and a new coalition has to be formed. Thus, determining a coalition in dynamic environments is very challenging.

In this paper, considering limited sensing and communication ranges, delays in communication, and limited consumable resources of UAVs, we propose new techniques to determine coalition quickly for multiple heterogeneous UAVs searching and attacking targets collaboratively. The rest contents of the paper are as shown

* This work was partially sponsored by the Fundamental Research Funds for the Central Universities under Grant (3102015ZY092).
below: In Sec. 2, the mission scenario together with the problem formulation is provided. In Sec. 3, a mechanism to find potential coalition members with communication constraints is presented. In Sec. 4, we propose a multistage sub-optimal coalition formation algorithm with low computational complexity. A distributed autonomous control strategy based on the Finite-State Machine is given in Sec. 5. The results of examples and Monte-Carlo simulation are demonstrated in Sec. 6. At last, the conclusions are given in Sec. 7.

2 Problem Formulation

![Diagram of UAVs and Targets](image)

Fig. 1. UAVs and Targets in a region where a search and prosecute need to be carried out.

As shown in Fig. 1, we consider a search and destroy mission using \( N \) heterogeneous UAVs. In this paper, the UAVs can be called agents. Each UAV is identified by its unique identity number \( \mathcal{A}_i \). The limited number of resources types carried by UAVs is denoted by \( n \). Some of these resources deplete while other ones like sensors are consumable. Resources of \( \mathcal{A}_i \) can be represented by a capability vector \( \mathbf{R}_i \) as following:

\[
\mathbf{R}_i^j = (R_{ij}^1, ..., R_{ij}^q)
\]

where \( R_{ij}^p, p = 1,...,n \) denotes the type-\( p \) resource number.

Assume that there are \( M \) targets whose resource requirements and their locations are unknown a priori. The agents have to execute the search task with limited sensor range, denoted by \( r_i^s \), to detect the targets. When \( \mathcal{A}_i \) detects a target \( \mathcal{T}_j \), the resources required to attack \( \mathcal{T}_j \) can be determined by \( \mathcal{A}_i \), which can be represented by \( \mathbf{R}_i^a \) as:

\[
\mathbf{R}_i^a = (R_{ia}^1, ..., R_{ia}^m)
\]

where \( R_{ia}^q, q = 1,...,m \) denotes the number of type-\( q \) resources required to prosecute the target \( \mathcal{T}_j \).

Each UAV has a communication range, denoted by \( r_i^c \), and can communicate with other UAVs that are within this range. Assume that the communication range of UAV \( \mathcal{A}_i \) is at least twice that of sensor range, that is \( r_i^c > 2r_i^s \). This assumption ensures that multiple UAVs within the communication ranges of each other do not form multiple coalitions for the same target.

The sequences of actions conducted during the coalition formation are described as follows. Once \( \mathcal{T}_j \) is located and identified by \( \mathcal{A}_i \), \( \mathcal{A}_i \) determines whether it has sufficient resources or not. If it has then it will prosecute \( \mathcal{T}_j \). Otherwise, \( \mathcal{A}_i \) becomes the coalition leader (CL), and has to form a coalition to prosecute \( \mathcal{T}_j \) by broadcasting the target location and resources requirement vector \( \mathbf{R}_j \) to other UAVs over the ad hoc network. This process is called Proposal.

In the search region, the responded UAVs should possess at least one of the resources required to attack \( \mathcal{T}_j \). The information responded to \( \mathcal{A}_i \) includes the resources capabilities and the earliest time to arrive (ETA) at the target position. This process is called Bid. These agents that respond to CL are called as potential coalition members (PCMs).

Once the CL \( \mathcal{A}_i \) receives the bids from PCMs, its task is to determine a coalition. This process is called Formation. The UAVs in the coalition are called coalition members (CM). The coalition should satisfy some constraint conditions that include: (1) minimizing the attack time to targets. This ensures that the UAVs take less time to accomplish the attack tasks; (2) minimizing the size of coalition. This ensures that more UAVs carry out the search task results a quick detection for targets; (3) the coalition members should attack the target at the same time, inducing maximal damage to targets; (4) the total resources of coalition should satisfy resources requirement to attack the target. This ensures that the target could be destroyed.

The CL \( \mathcal{A}_i \) is to form a coalition \( \mathbf{C}_i \) for \( \mathcal{T}_j \) taking the constraints 1-4 into account. The total resources of coalition \( \mathbf{C}_i \) is \( \mathbf{R}_j^c = \sum_{\mathcal{A}_k \in \mathbf{C}_i} \mathbf{R}_k^a \). Let \( \mathcal{A} \) be the set of all PCMs including the CL bidding to be members of the coalition. Let \( \lambda_k \) be the ETA of the \( \mathcal{A}_k \in \mathcal{A} \) to arrive at \( \mathcal{T}_j \). To determine a coalition, the CL has to solve the objective function:

Objective: \[ \min_{\mathcal{A}} \max_{\lambda_k \in \mathcal{A}} \lambda_k \]  

Subject to: \[ \sum_{\mathcal{A}_k \in \mathcal{A}} R_{jq} \geq R_{jq}^c \text{ for all } q = 1,...,n \]  

where \( \mathcal{A} \subseteq \mathcal{A} \). In order to attack at the same time, the UAVs have to arrive at the target simultaneously. The latest arrival time determines the coalition earliest total strike time. The function (3) determines a minimum size of coalition with a minimum latest arrival time. The constraint 4 is described by (4).

3 Determining potential coalition members with communication Constraints

A CL, after detecting a target, broadcasts the proposal for coalition. The UAVs satisfying the requirements will respond to the CL with their resources capabilities and ETA. These responded UAVs become the PCMs. Then a feasible coalition can be selected from these members together with the corresponding broadcast of acceptance or rejection for the PCMs. Due to the communication constraints, such as communication range limitation, communication time delay, it could not ensure that every UAV in the dynamic network can receive the request for coalition formation from the CL.

To form coalition, we design a flexible and efficient mechanism to determine PCMs over a dynamic network within communication constraints.

In this network, intermediate UAVs called relay nodes will transfer information from a UAV to another. This communication method is referred as ‘flooding’. The messages float around in the network, so this protocol is very simple. However, it does not involve any collision detection, and will cause a lot of data packets contention for the bandwidth of the network, which called “broadcast
storm problem”. To avoid reducing the available network bandwidth resulted from the message floating indefinitely in the network, we refer the notion of time-to-live (TTL) for a message. The TTL of a message is a counter that determines the number of hops the packet can travel at most on its way from the source to the destination. When the packet reaches a node, the value of TTL is reduced by one then transferred to the neighboring nodes. This process ends when packet reaches the destination or the TTL counter reaches 0 where it is abandoned (not rebroadcast). In this paper, we specify maximum number of hops for TTL, denoted by \( H_{\text{max}} \), which determines the depth of the message traveling in the network. Each message has a current hop counter, \( H_i \), whose initial value is \( H_{\text{max}} \). Whenever a relay UAV rebroadcasts the message, \( H_i \) is reset to \( H_i - 1 \). If \( H_i = 0 \), then the message is not rebroadcast by a relay UAV. The decision process and the communication protocol during a coalition formation process are given in Fig. 2.

After a CL broadcasts the proposal for coalition formation, the bid process begin. Time for bid messages transferring from PCMs to CL may be different because of different communication routes taken in the network. The longest time for coalition leader should wait can be calculated as follows. To mimic a real-world wireless network, we assume a delay for coalition leader should wait can be calculated as follows.

\[
\text{ETA from the go-ahead location} = \text{H_{max} + \Delta c}
\]

4. Coalition formation algorithm

4.1 Multistage Sub-Optimal Coalition Formation Algorithm

To determine the CM, the CL has to solve the optimization problem according to formula (3), (4). The traditional solution is to search the complete solution space but this method requires exponentially increased computation when the number of UAVs and targets increases. In this paper, we divide the whole process into two-stages to reduce the algorithm complexity and then a multistage sub-optimal coalition formation algorithm (MSOCFA) that has low computational complexity is presented. In this section, we describe this algorithm.

\( A_i \) detected target \( T_j \) that requires \( R_{ij}^T \) resources. If \( R_{ij}^T \geq R_{ij}^\delta \), \( \forall p = 1, ..., m \), and \( A_i \) is not already a part of another coalition, then \( A_i \) would attack target \( T_j \) alone. If \( A_i \) cannot satisfy the resources requirement, then \( A_i \) assumes the CL role and broadcasts the target information to the other UAVs. The information includes the required sources and location of \( T_j \). The UAVs that have at least one of the required resources will respond to CL with their resource vector and ETA, and will be PCMs. Later, the CL checks all the responded information and determines whether a coalition can be formed or not. It will send a discard coalition broadcast when
failing to form a coalition. Otherwise, a coalition $C'$ will be formed and the last arrival time for attacking $T_j$ will also be broadcasted by the CL. After the CMs reach their respective go-ahead location, they will re-plan their paths to prosecute the target such that the ETA is equal to the latest arrival time. The rejected UAVs will continue their search tasks.

In above process, the MSOCFA is described by Algorithms 1 and 2, as shown in Figs. 3, 4. In the first stage, a set of UAVs satisfying the minimum time requirement is determined and then in the second stage, the minimum coalition size is achieved by pruning the set obtained in the first stage.

### Algorithm 1 First stage of MSOCFA
1: $C_i^j = []$ and $R_i^j = []$
2: $A = [A_1, A_2, \ldots, A_N]^T$
3: $ETAs = \{\eta_1, \eta_2, \ldots, \eta_N\}$
4: $A_j = [\eta_1, \eta_2, \ldots, \eta_N]^T$
5: $A_j = []$
6: $\text{Initialize: (begin)}$
7: $\text{sort} A_j$ by their $ETAs$ in ascending order
8: $[A_j, \eta_i] = \text{sorted}(ETAs)$; $\%$ $A_j \rightarrow$ corresponding UAV index of $A_j$
9: for $k = 1 \rightarrow N$
10: $A_j = [A_j, \eta_i]^T$
11: $C_j = \text{append} A_j$ to $C_j$
12: $R_j = R_j^c + R_j^v$
13: $A_j = \text{append} A_j(k)$
14: if $R_j \geq R_j^c$, for all $j$
15: return $(C_j, R_j^c, A_j)$
16: $\text{BREAK}$
17: end if
18: end for
19: $\text{end}$
20: $\text{return \{No Coalition\}}$

### Algorithm 2 Second stage of MSOCFA
1: $A = C_j$
2: for $k = 1 \rightarrow |C_j|$
3: $A_j = C_j(k)$
4: $R_j = R_j^c + R_j^v$
5: if $R_j \geq R_j^c$, for all $j$
6: $A = \text{remove} A_j$ from $A$
7: $R_j = R_j^v$
8: end if
9: end for
10: $C_j = [\eta_1, \eta_2, \ldots, \eta_N]^T$

The computational complexity of MSOCFA is $O(N \log N + 2m)$. So MSOCFA produces a sub-optimal solution that has polynomial time complexity. Therefore, it can be applied to real-time decision process.

### 4.2 Simultaneous Strike

Once a coalition is formed, the latest arrival time for the coalition $C_j$ is determined by the CL $A_j$ and later is provided to the CMs. The CMs adjust their radius $r$ (bounded below by the minimum turning radius) of the Dubins curve to make the ETA equal to the latest arrival time. Then they track the desired paths to attack the target at the same time, causing a maximal damage. The Dubins curves can be seen in [9], and the path tracking algorithm can be seen in [10].

### 5 Distributed cooperation approach

Every UAV, at its core is a behavior-based control system. At the higher level though, every UAV is capable of operating in one of the following 7 states: Search, Proposal, Bid, Approach, Formation, Dubins and Prosecute. The finite state machine as shown in Fig. 5 is responsible for deciding the operational state of a UAV.

#### 5.1 Search state

All agents start out in the search state. In this state, an agent flies around the entire search region to look targets [11]. During the search, when an agent detects a target and if the agent is equipped with required capabilities ($I_1$), the transition from the search state to the Dubins state happens for the agent. After an agent detects a target, if it has insufficient resources ($I_l$), the agent immediately switches to the proposal state and becomes the CL. If an agent receives message of proposal from a coalition leader during the search ($I_9$), the search agent becomes a PCM and switch to bid state. If an agent deplanes its all carried resource, or all targets have been destroyed ($I_4$), it keeps in the search state.

#### 5.2 Proposal state

In the proposal state, the CL broadcasts the message of proposal for coalition formation to the other UAVs. In addition to broadcasting the proposal message, the CL needs to estimate its goal position, and switches to the approach state ($I_s$). The CL flies toward its estimated goal position in the approach state. As approaching its go-ahead location, the CL has been waiting for the message of bids from all PCMs. When the CL receives all the responses ($I_4$), it immediately switches to the formation state.

#### 5.3 Bid state

In the bid state, the PCM calculates the ETA from its estimated goal position to the target location according to the proposal message. Then the PCM will send the ETA $\eta_i$ and its resource vector $R_i^d$ back to the CL by broadcasting the message of bid for coalition formation. Then, the PCM immediately switches to the approach state ($I_s$), waits for the message of result for coalition formation from the CL as approaching its go-ahead location.

#### 5.4 Formation state

In the formation state, the CL will send a discard coalition broadcast when failing to form a coalition. Otherwise, it informs acceptance or rejection decisions to all the PCMs by broadcasting the message of result for coalition formation. If the coalition can be formed successful and the CL becomes the CM but has not arrived at its estimated go-ahead location ($I_6$), the transition from the formation state to the approach state takes place, and then the CL will continue to fly toward its estimated go-ahead location. If the coalition formation is failed or the CM is excluded from the formed coalition ($I_9$), then it switches back to the search state.

#### 5.5 Approach state

If a coalition can be formed successful and the PCM is accepted to become the CM but it has not arrived at its estimated goal position ($I_6$), it keeps in the approach state, and keeps on flying toward its estimated goal position. If the coalition formation is failed or the PCM is not included in the formed coalition ($I_9$), then it switches back to the search state.
After the CMs reach their respective goal position ($I_{12}$), they switch to the Dubins state.

### 5.6 Dubins state

In Dubins state, the selected CMs will re-plan their flight paths applying the developed simultaneous strike mechanism to prosecute the target. After their paths be re-planned successful ($I_{13}$), the CMs switch to the prosecute state.

#### 5.7 Prosecute state

In the prosecute state, the coalition members fly toward the target location by tracking their Dubins paths. When they attack a target, they deploy their resources. If the target is destroyed ($I_{14}$), the coalition members switch back to the search state.

---

**Fig. 5 State Diagram for UAVs in search and prosecute mission**

**Table 1 The initial settings of 4 UAVs in the comparison simulations**

<table>
<thead>
<tr>
<th>Position ($x$, $y$)</th>
<th>Heading $V_t$</th>
<th>Capability Vector $R^d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$ (500,400)</td>
<td>180</td>
<td>(1,0,3)</td>
</tr>
<tr>
<td>$A_2$ (-950,-350)</td>
<td>0</td>
<td>(1,1,1)</td>
</tr>
<tr>
<td>$A_3$ (900,-100)</td>
<td>170</td>
<td>(1,2,1)</td>
</tr>
<tr>
<td>$A_4$ (600,-600)</td>
<td>190</td>
<td>(1,2,0)</td>
</tr>
</tbody>
</table>

**Table 2 The initial settings of 1 target in the comparison simulations**

<table>
<thead>
<tr>
<th>Position ($x$, $y$) $R^d$</th>
<th>Requirement Vector $R^d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0)</td>
<td>(2,3,2)</td>
</tr>
</tbody>
</table>

- $H_{\text{max}} = 1$

Fig. 6 shows the result of coalition formation and trajectories of the UAVs with the maximum number of hops $H_{\text{max}} = 1$. At time $t=37.4$s, agent $A_1$ detects target $T_1$. The required resources of $T_1$ is $R_t^T = (2,3,2)$, but the available resources of $A_3$ is $R_t^d = (1,2,1)$, so $A_3$ has insufficient resources, it becomes the coalition leader and broadcasts proposal message to form coalition $C_1$ for target $T_1$. The positions of all UAVs and communication topology when $T_1$ is located by $A_3$ are shown in Fig. 7. In this figure we can see that since $H_{\text{max}} = 1$, only the neighbors $A_1$ and $A_3$ can communicate with $A_3$. At time $t=38.4$s, $A_1$ and $A_3$ receive the proposal message from the coalition leader $A_3$ due to the communication delay with each message hop $\delta = 1$s. The resource vector of $A_1$ is $R_{t}^d = (1,0,3)$, and its ETA to arrive at $T_1$ is $\lambda = 66.6$s. The resource vector of $A_3$ is $R_{t}^d = (1,2,0)$, and its ETA to arrive at $T_1$ is $\lambda = 86.2$s. The potential coalition members $A_1$ and $A_3$ respond $A_3$ by sending their $R_{t}^d$ and $R_{t}^d$, respectively. The coalition leader $A_3$ considers all the responses combining with its own resource vector $R_{t}^d = (1,2,1)$ and ETA $\lambda = 63.1$s, then determines the coalition $C_1 = \{A_1, A_3, A_4\}$. The total resources vector of coalition $C_1$ is $R^{T} = (3,4,4)$ and the latest arrival time is 86.2s. The go-ahead locations of $A_1$, $A_3$ and $A_4$ are $G_1 = (-104.5, 400.0)$m, $G_3 = (306.7, 4.7)$m, $G_4 = (47, -705.0)$m respectively. The symbol “+” represents the go-ahead locations of the UAVs in Fig. 6.
The agent $A_2$ does not receive the proposal from the coalition leader $A_3$, so it performs the search task at all times.

- $H_{\text{max}}=2$

Fig. 8 shows the result of coalition formation and trajectories of the UAVs with the maximum number of hops $H_{\text{max}}=2$. At time $t=37.4s$, agent $A_3$ detects target $T_1$, $A_1$ becomes the coalition leader and broadcasts proposal message to form coalition $C_1^1$ for target $T_1$. The positions of all UAVs and communication topology when $T_1$ is located by $A_3$ are shown in Fig. 9. In this figure we can see that since $H_{\text{max}}=2$, the neighbors $A_1$ and $A_3$ can communicate with $A_1$ directly, $A_2$ can also receive information from $A_3$ as $A_4$ to be a relaying agent. So the selected potential coalition members include $A_1$, $A_2$ and $A_4$. Due to the communication delay with each message hop $\delta t=1s$, at time $t=38.4s$, $A_1$ and $A_4$ receive the proposal message from $A_3$. The resource vector of $A_1$ is $R_t^1=(1,0,3)$, and its ETA to arrive at $T_1$ is $\lambda_t=69.7s$. The resource vector of $A_4$ is $R_t^4=(2,3,2)$, and its ETA to arrive at $T_1$ is $\lambda_t=83.5s$. The coalition $C_1^1$ with $A_1$ and $A_4$ was formed, its total resources vector is $(2,3,2)$ and the latest arrival time is $83.5s$. The estimated goal positions of agents $A_1$, $A_2$, $A_3$ and $A_4$ are $G_1^1=\langle-149.5, 400.0, 0\rangle m$, $G_2^1=\langle-300.5, -500.0, 0\rangle m$, $G_3^1=\langle261.8, 12.5\rangle m$, $G_4^1=\langle-39.6, -712.8\rangle m$ respectively.

The comparison of results for coalition formation is shown in Table 3. From the comparison, we can see that the maximum number of allowed hops $H_{\text{max}}$ plays a key role in the coalition formation process. As the $H_{\text{max}}$ could restrict the depths of finding coalition partners from the network, and further, could affect the results of coalition formation.

6.2 Effect of Hop Delay and Max-Hops

We applied Monte-Carlo simulations to analyze the effect of the maximum number of allowed hops ($H_{\text{max}}$) and hop delay ($\delta$) on the mission performance. The effectiveness is measured by the average time taken to destroy all the targets. The experiments were carried out on 2000m×2000m area with 5 targets and 10 UAVs, when the delays $\delta$ are 1s, 2s and 3s, while $H_{\text{max}}$ was increased from 1, 2 and 3. So the Monte-Carlo simulations can be divided into 9 groups of experiments. The number of experiments was 100 in each group. We need record the mission completion time for every experiment, and calculate the Average Mission Completion Time (AMCT) of each group of experiments.

Each experiment was carried out for 1000 seconds. We consider the completion time as 1000 seconds in the case that the targets are partially destroyed. The initial values of targets position, UAVs position and UAVs heading angles...
are generated at random. The target resources were randomly generated between 0 and 3, while the resources for each UAV were generated between 2 and 4.

- The effect of varying $H_{\text{max}}$ for a given $\delta$
  
  When the delay is 1s, the AMCT decreases with increase in $H_{\text{max}}$, that is, the agents take less time to accomplish the task. As in this case, the agents can find coalition partners from the network with larger depths, and hence will be able to determine feasible coalitions that can destroy the target quicker.

  However, when the delay is increased, the AMCT does not increase with increase in $H_{\text{max}}$. This is because, the value of delay is large and the accumulative delay of reaching $H_{\text{max}}$ depth in the network increases the ETA significantly further and this impact the performance. This effect can be shown in the performance curves with $\delta = 2s$ and $\delta = 3s$.

  These results indicate that in the cases of a low delay, finding PCMs deep in the network is effective, however, under a significant high delay, selecting coalition members from the immediate neighbor agents is sufficient.

- The effect of varying $\delta$ for a given $H_{\text{max}}$
  
  In addition, we can see obvious phenomena, where the performance degrades correspondingly with increase in $\delta$.

  From Monte-Carlo simulations, it can be concluded that the each message hop delay in the network and the maximum number of allowed hops are important for the mission performance in term of AMCT. If the delay is large, then a coalition formed from the immediate neighbors is sufficient for a good performance. Under smaller delays, including neighbors up to a few hops will increase performance, and any additional increase in hop count will degrade performance.

6.3 Difference in solutions obtained using MSOCFA and PSO

When the targets resources and their locations are known a priori, we can achieve the combinatorial coalition formation using PSO [12]. In order to evaluate the performances of MSOCFA, we applied Monte-Carlo simulations to analyze the effect in terms of the average percentage mission completed (APMC), the average mission completion time (AMCT), and the average CPU time spent on the coalition formation in both the MSOCFA algorithm and the PSO algorithm. The experiments were carried out on 2000m $\times$ 2000m area with 10 targets, while changing the number of UAVs to 5, 10, 15 and 20. Each experiment was carried out for 1000 seconds. The performance curves are shown in Figs. 11, 12, 13 and 14.
From Fig. 11, it can be seen that in the case of 5 UAVs, the PSO algorithm gains a lower APMC than the MSOCFA algorithm. The reason is that with insufficient resources to attack all the targets, the optimal solution cannot be obtained in the PSO algorithm, resulting in a 0% mission completion. But, for MSOCFA algorithm, the targets are destroyed once detected, contributing to a higher APMC. In the cases of 10, 15, 20 UAVs, the APMC is 100% because of the sufficient resource for attacking all the targets.

Fig. 12 shows the comparison of average mission completion time. The AMCT experienced a downward trend with the increasing number of UAVs under a given number of targets. Besides, the mission can be completed in a shorter period of time in the PSO algorithm compared with the MSOCFA algorithm as the target positions are known a priori.

From Figs. 13 and 14, we can see that the average CPU time spent on the coalition formation in the PSO algorithm is much higher than that in MSOCFA algorithm. What’s more, in case of PSO solution, the time taken to produce a solution increases exponentially as we increase the number of agents and targets.

From these studies, it can be concluded that the MSOCFA can provide a similar solution to that provided by the global optimal solution, however, requires far less computing resources.

7 Conclusion

In this paper, a coalition formation algorithm for UAVs having limited communication ranges with associated communication delays is presented. A novel mechanism is developed to determine PCMs in a dynamic network formed by UAVs. The effect of communication delay and the number of maximum allowed hops for a message in the network to determine PCMs for a coalition is studied by Monte-Carlo simulations. The results indicate that, in a small communication delay, finding PCMs deep in the network can enhance the performances, but under a significant communication delay, it is more efficient to select coalition members from immediate neighbors. Comparison of simulation results shows that the MSOCFA algorithm has the similar mission performances to the PSO algorithm, while the MSOCFA algorithm requires far less computational resources.

References
