Cooperative Task Allocation for Unmanned Combat Aerial Vehicles Using Improved Ant Colony Algorithm

Jun Tao and Yantao Tian
College of Communication Engineering
Jilin University
Changchun 130025, China
taojun_saia@163.com, tianyt@mail.jlu.edu.cn

Xiangheng Meng
Department of Aviation Control Engineering
Aviation University of Air Force
Changchun 130022, China
mengheng2008@126.com

***Abstract*—Task allocation plays an important role in Unmanned Combat Aerial Vehicles' (UCAVs) cooperative control. In order to solve the problem of multiple UCAVs' cooperative task allocation, an improved ant colony algorithm (ACA) is proposed. On the basis of modeling cooperative multiple task assignment problem, the application of improved ACA is discussed. Cooperative task allocation for UCAVs shows a property of dynamic multiple phased decision problems and a task tree is used to represent that case. In the improved ACA, pheromone change is very different from other classic improved ACA. Especially when pop-up targets appear, with the help of changed pheromone matrix which is gained from former iterations, it becomes easier and quicker to find good solutions.**

***Keywords*—ant colony algorithm, task allocation, UCAV, cooperative control**

I. INTRODUCTION

Following with the development of unmanned combat aerial vehicles' (UCAV) technology, cooperating unmanned aerial vehicles control will receive a great attention in the future. For many military missions, it is unthinkable to be taken on a single UCAV .And more instances show that it is more complex and expensive to exploit a single UCAV than to create a multiple UCAVs' system .So in the future battlefield cooperating using of UCAVs will be the main feature. As the key technique of cooperating mission planning, task allocation has attracted many scholars' great attention [1]-[2].

In that context, an intensive research effort has been conducted in recent years on the development of task allocation algorithms. The approaches that have been reported for solving the task allocation problem can be classified into the following categories: market based approach [3], mixed integer linear approach [4], genetic algorithm [5], and ant colony algorithm [6]. The ant colony algorithm is a kind of metaheuristic that simulates the behavior of ant's foraging in nature. It is an iterative algorithm that maintains a pool of feasible solutions for each iteration. Its variants are widely employed to solve optimization problems and have demonstrated satisfactory performances. Generally speaking, ant colony algorithm has two important features. One is the positive feedback in the process searching for the best solutions. And it increases the pheromone values associated with good and promising solutions. The other feature is the distributed paralleled computing for multi-agents which improves its search efficiency in nature.

The reminder of this paper is organized as follows: Section Ⅱ describes the establishment of cooperating task assignment's model. The implementation of ant colony optimization necessary to support a cooperating task allocation is presented in section Ⅲ.Section Ⅳ examines the effectiveness of the algorithm presented in the former section. Conclusions are drawn in section Ⅴ.

II. COOPERATIVE TASK ALLOCATION MODEL

Now we assume that some territory of the battlefield has been searched and N_t targets have been found in all. $T=\{1,2,3,\ldots,N_t\}$ represents the set of targets. $V=\{1,2,3,\ldots,N_v\}$ represents the set of vehicles. The missions that needed to be executed on every target are represented by $M=\{Classify, Attack, Verify\}$ and N_m is used to count the total number of missions. Next we will discuss the model in detail in two ways: the restriction requirement and the performance requirement.

A. Restriction Requirement

Restriction requirement of multi-objective optimization problems is difficult to solve. And the restriction requirement is various according to the problem to be settled. In this paper, it is necessary to satisfy three qualifications to find a feasible solution: task precedence, multiple UCAVs cooperation and timing constraint.

Task precedence states the fact that tasks must be executed in special order. For example, the vehicle can attack a target only when the target has been classified. So it is easy for us to know that the task of Verify should be arranged after the task of Attack.

Timing constraint states that some task should be completed in an appointed time window. These constraints are extremely important for the target of ground-to-air missile because of its tremendous threat to the UCAV.

The corresponding author: tianyt@mail.jlu.edu.cn
This work is sponsored by the National Science Foundation of China (Grant No: 60675057)

Cooperation constraints. Every task will be executed only once, except that the task is predetermined to be executed more times when evaluation result shows that the task was not completed successfully. We assume $x_{ij} \in \{0,1\}$ is the decision variable, and is satisfies the following equation:

$$x_{ij} = \begin{cases} 1 & \text{if UAV}_i \text{ is allocated to excute task}_j \\ 0 & \text{else} \end{cases} \quad (1)$$

So the cooperation constraints can be represented as follows:

$$\sum_{i=1}^{N_v} x_{ij} = 1, j = 1, 2, \cdots N_m \quad (2)$$

B. Goals of Cooperative Task Allocation

Ratio of task covering. It reflects how many tasks that all the UCAVs have completed. Generally speaking, the more tasks have been completed, the better task allocation plan it will be. Because of complex coupling constraints among multiple targets and multiple missions, not every task will be executed successfully, especially in the phase of algorithm initialization. There are N_v UCAVs to be allocated to targets' $N_t \bullet M$ tasks. The ratio of task covering is:

$$J_1 = \frac{\sum_{i=1}^{N_v} N_v \bullet N_{m_i}}{N_t \bullet M} \quad (3)$$

Cumulative distances traveled by all vehicles to perform all required tasks .There are constraints that every vehicle can only take a limited amount of fuel. So the distance that a vehicle can travel should be controlled under the maximal distance. In addition, a shorter distance means less probability of being detected as flying in a threaten environment. This performance criterion can be represented by:

$$J_2 = \sum_{i=1}^{N} \sum_{k=1}^{Nm} Route_k^{(i)} = \sum_{i=1}^{N_v} R_i \quad (4)$$

Where R_i is the distance traveled by vehicle $i \in V$ to perform its task plan .The objective is to minimize the total distance traveled by all vehicles. And the minimization of the objective function means the use of UCAV's resource is optimized. Another performance requirement is to minimize the time that all vehicles complete their task plan. Assuming constant equal speed for all UCAVs, performance requirement can be represented as follows:

$$J_3 = \max_{i \in v} R_i \quad (5)$$

Gaining maximal total value. The value that a vehicle get when complete a task is different. The higher a task's value is, the more preferential it should be assigned to vehicles. The objective function can be represented as follows:

$$J_4 = \sum_{i=1}^{N_v} \sum_{j=1}^{x} Value_i^x \quad (6)$$

III. IMPROVED ANT COLONY ALGORITHM

A. Task Tree Representation Methodology

Cooperative task allocation for UCAVs can be expressed by the task tree. The task tree not only covers the decision space, but also includes various constraints through nodes and edges. The tree is established based on at some time $UAV_i \in V$ is assigned to $Target_j \in T$ to execute task $k \in M$. The start-point and end-point separately represent the UCAV's starting and final location.

A feasible task allocation set can be represented by a path from the start-point to the end-point. For briefly explaining the process of establishing a feasible solution, we take two vehicles in a scenario of three targets for example. All constraints between every two tasks can be achieved through if there is access or no. In the following discussion, we put our emphasis on the requirement of task precedence and cooperation among UCAVs.

The complexity of the task tree varies as the missions that vehicles can execute change. $V = \{V_1, V_2\}$ is the set of UCAVs. And UCAVs can be competent for missions of $Vkind(V_i) = \{Classify, Attack\}$ .The set of target is $T = \{T_1, T_2, T_3\}$. $T_{mi} = \{Classify, Attack, Verify\}$ is the set of tasks that need to be executed by vehicles. The process of UCAV's executing tasks can be expressed as Fig. 1 (CL denotes Classify, AT denotes Attack, VE denotes Verify).

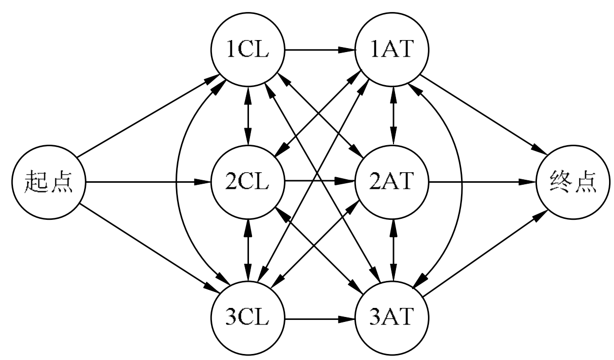

Figure 1. Task tree representation

For a target, the task of Attack must be assigned after the task of Classify because of the need of task precedence constraint. So along with the process of task assignment the task tree will be dynamically changed. For example, if V_1 and V_2 are separately assigned to execute the task of Classify in T_1 and T_2 , the task tree will be dynamically adjusted as Fig. 2. And the final task allocation result in a certain condition shows in Fig. 3.

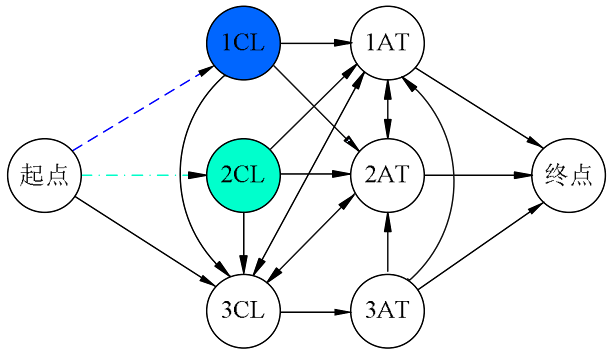

Figure 2. Dynamic adjusting of the task tree

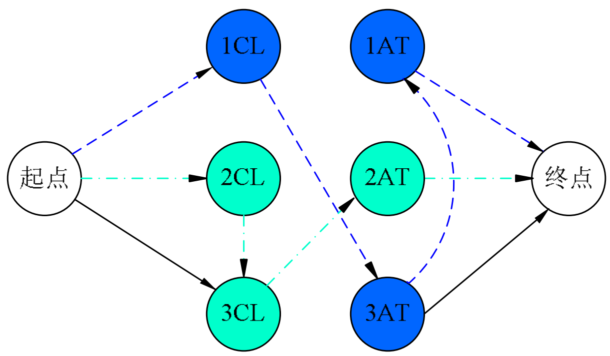

Figure 3. Final result of task allocation

B. *Design of Improved Ant Colony Algorithm*

There are many similarities between task allocation and ant's behavior of foraging and particular correspondences between them are shown in table Ⅰ.

TABLE I. SIMILARITY BETWEEN TASK ALLOCATION AND ANT'S FORAGING

Task allocation	Ant's foraging
Start-point	Nest
End-point	Food source
Task of the target	City
Edge between tasks	Path between cities
Difference among tasks	Difference among paths
Optimal task allocation	Shortest path

Every ant corresponds to a vehicle and they can complete different tasks. When the ant get a whole path (from start-point to end-point), there will be a feasible solution to the task allocation problem .Every time the ant get or complete a task it will communicate with other ants which make them change their space of choice. When all ants has cycled once the evaluation of the feasible solution will be made. Then both the global pheromone matrix and local pheromone matrix will be updated and the next circulation happens. At last comparing every feasible solutions and output the best one.

As discussed in the former section, we assume that there are N_v vehicles, N_t targets and N_m tasks. The improved ant colony algorithm is mainly analyzed and designed from the following four aspects: data structure, data initialization, state transfer rule and mechanism of pheromone update [7].

Due to the complex correspondence between multi-agent and multi-task, the data structure of the improved ant colony algorithm is more complicated than the traditional algorithm .We should firstly define them to store interrelated data matrices used in the task allocation problem. The distance matrix between every two targets $D=[d_{ij}]_{N_t\times N_t}$ is a matrix with N_t dimensions.

Access matrix between every two tasks. $T=\left[t_{ij}\right]_{(N_m+2)\times(N_m+2)}$ represents the access matrix with the dimension of N_m+2 which means N_m tasks plus start-point and end-point. Not every task can be executed after certain task because many constraints should be satisfied. And this is also the main difference between task allocation problem and traveling salesman problem. Ants that belong to different styles will have a different initial access matrix. And along with the process, the access matrix will change regularly.

Every edge has a pheromone concentration τ_{ij} in TSP. In this problem there are two matrices will be established to represent pheromone concentration: $P_g=\left[p_{ij}\right]_{(N_m+2)\times(N_m+2)}$ denotes the global pheromone matrix and $P_l=\left[p_{ij}\right]_{(N_m+2)\times(N_m+2)}$ denotes the local pheromone matrix.

Combination matrix of pheromone and heuristic information. The transition probability from city i to city j for the k_{th} ant is in direct ratio with $[\tau_{ij}]^{\alpha}\cdot[\eta_{ij}]^{\beta}$. In the task allocation problem the definition of heuristic information is different from TSP .But put all these elements in a matrix will save a lot of computation time.

On account of many performance requirements for task allocation problem, the heuristic information is more complex than that in traditional TSP. It mainly determined by two factors: the distance between task i and task j and the value of task j. So the equation of heuristics information can be denoted as:

$$\eta_{ij}=\begin{cases}\dfrac{Value(j)}{d_{ij}}, & t_{ij}=1\\ 0, & t_{ij}=0\end{cases} \tag{7}$$

Then we use a choice mechanism similar to roulette wheel in evolutionary algorithm to select the city to be visited. And we define the transition probability from city i to j for the k_{th} ant as:

$$p_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\bullet[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\in allowed_k}[\tau_{is}(t)]^{\alpha}\bullet[\eta_{is}(t)]^{\beta}}, & \text{if } j\in allowed_k \\ 0 & \text{otherwise}\end{cases} \tag{8}$$

Where $allowed_k=\{N-tabu_k\}$ and α and β are parameters that control the relative importance of trail versus visibility .Therefore the transition probability is a trade off between visibility and intensity.

Pheromone update rules are the soul of ant colony algorithm and are also the basic difference from other evolutionary algorithm. In the course of circulation, if ant has passed through the edge, the local pheromone matrix is updated as follows:

$$\tau_{ij}=\rho\bullet\tau_{ij} \tag{9}$$

where ρ is the decline of pheromone concentration on edge (i,j). So the transition probability to the visited nodes will be reduced which will be good for finding better solution globally.

The global pheromone update is the function of $J_i(i=1,2,3,4)$. It is in direct ration with J_1, J_4 and is in inverse ratio with J_2, J_3. It represents the quality of all feasible solutions got from all ants in one cycle .The equation is:

$$\Delta\tau_{ij}=\begin{cases}Q\bullet J_1\bullet J_4 \big/ J_2\bullet J_3, & \text{if ant pass edge}(i,j) \\ 0, & \text{otherwise}\end{cases} \tag{10}$$

where Q is a constant.

Formally the improved ant colony algorithm is:

Step 1. Initialization;

Step 2. Set the number of cycle $N_c=N_c+1$;

Step 3. Set the number of transition $N_t=N_t+1$;

Step 4. Set the number of ant $k=k+1$;

Step 5. Choose the next city based on (8) and add the city to its taboo list. Update pheromone matrices according to (9). Update the access matrices of all ants;

Step 6. If all ants reach end-point turns to step (3), else to step (4);

Step 7. If all ants have transferred once, turns to step (3), else to step (4). If all ants reach end-point, turns to step (2), and update global pheromone matrix according to (10), else to step (3);

Step 8. If $N_c\geq N_{c\max}$, end circulation and output the best solution

C. Computational coplexity of the improved ACA

Because all the algorithms should be implemented on the computer, computational complexity is the decisive factor for the speed of gaining solutions. So in this part of the paper we will discuss the computational complexity of the improved ACA. To accurately calculate an algorithm's computational complexity we not only need to analyze the procedure step by step, but also need to consider the system's compiling ability which is very difficult to compare. So it is unnecessary to accurately calculate the computational complexity of an algorithm. And we only illuminate the magnitude of every step to found a common standard for comparing different methods' efficiency.

The computational complexity of the initialization is $O((N_m+2)^2+N_v)$ .The step of update local and global pheromone matrices are $O(N_v\bullet(N_m+2)^2)$ and $O((N_m+2)^2)$ .In the process of all ants looking for the feasible solutions, the computational complexity is $O(N_v\bullet(N_m+2)^2)$. The judgment of the so far best solution is $O(N_v)$ .At last, the output of the best solution is $O(1)$. So the total computational complexity of the improved ACA can be briefly represented by $O(N_{c\max}\bullet N_v\bullet(N_m+2)^2)$.

IV. SIMULATION AND ANALYSIS

In the scenario, there are 5 target and 4 UCAVs in the battle field. Their initial locations distribute as shown in Fig. 4 and the task requirements for every target are denoted in table II.

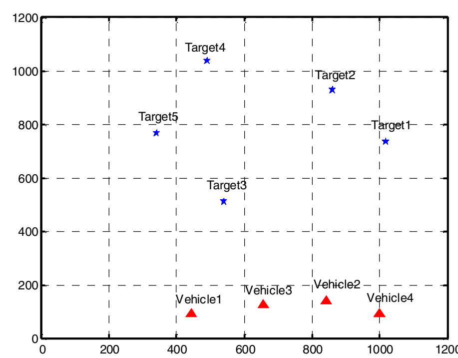

Figure 4. Initial setting of the simulation

TABLE II. TAGET'S MISSION REQIURMENT

Targets	Location	Classify	Attack	Verify
Target-1	(341.1, 768.0)	√	√	√
Target-2	(489.7, 103.8)	√	√	√
Target-3	(541.9, 509.2)	√	√	√
Target-4	(860.2, 927.1)	√	√	√
Target-5	(101.8, 734.0)	√	√	√

We assume that except for the task precedence applied on every target itself there are no other precedence requirements. In the simulation, target 3 has timing constraints. That is to say as to target 3 the task of Attack should be executed after task of Classify within [t, t+50]. The constant equal speed of cruise is 30 and the timing constraint can be transformed to distance constraint in this case.

We suppose that the task's value matrix is constant. And its value matrix is denoted as (11):

$$\begin{array}{cccc} & m1 & m2 & m3 \\ T1 & 2 & 4 & 1 \\ T2 & 3 & 6 & 8 \\ T3 & 4 & 2 & 9 \\ T4 & 1 & 6 & 7 \\ T5 & 5 & 7 & 5 \end{array} \qquad (11)$$

The task style that a UAV can execute is listed in table III.

TABLE III. STYLE OF UCAVS

UCAV	Location	Classify	Attack	Verify
Vehicle -1	(444.5, 88.9)	√		√
Vehicle -2	(843.2, 138.9)		√	
Vehicle -3	(655.8, 124.2)	√		√
Vehicle -4	(1000，88.9)		√	

A. *Experimental Study 1*

Based on these conditions stated above, we set main parameters of improved ant colony algorithms as follows: $\alpha = 1$, $\beta = 3$, $m = 3$, $Ncmax = 100$, $Q = 10$. As a result, we get the best task allocation plan as table IV shows The total distance that all the UCAVs traveled is 5632.3 and the number of completed tasks is 15 which means all tasks have been completed.

TABLE IV. OPTIMAL TASK ALLOCATION

UCAV	Task allocation Plan
Vehicle-1	（Target-03，CL）→（Target-05，CL）→（Target-03，VE）→（Target-01，VE）→（Target-02，VE）
Vehicle-2	（Target-03，AT）→（Target-05，AT）→（Target-04，AT）
Vehicle-3	（Target-01，CL）→（Target-02，CL）→（Target-04，CL）→（Target-05，VE）→（Target-04，VE）
Vehicle-4	（Target-01，AT）→（Target-02，AT）

From table IV we can know that when V_1 has completed the task of Classify of target 3, V_2 started to execute the task of Attack of target 3. The distance between T_2 and T_3 is 477.4. Through computation we know that V_1 can execute the task of Attack within the timing constraint.

B. *Experimental Study 2*

In this simulation we mainly discuss in the case of pop-up threat the quality of the algorithm to find the optimal solutions. The global pheromone matrix gets from experimental study 1. And we compare the result with the simply restarting the algorithm to find the best solutions. Target 6 in Fig. 5 is the pop up target and Fig. 6 shows the difference between the two methods. The dotted line represents the change of integration function using changed pheromone gained from experimental study 1 [8].

Although restarting the algorithm can get better solutions, it uses longer times. In the state of urgency we can accept the plan of the real line.

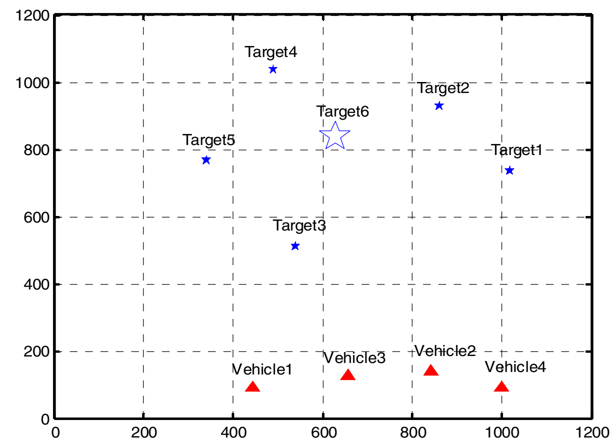

Figure 5. Scenario of pop-up threat

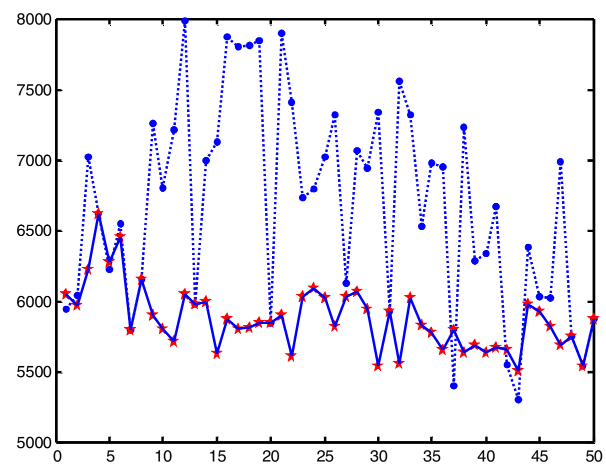

Figure 6. Comparision between different replanning methods

CONCLUSIONS

As a new heuristic algorithm, ant colony algorithm has used widely to solve combinatorial problems. This paper introduces its application in multiple UCAV's task allocation. And the simulation results show the effectiveness of this method. The initial comparison between different replanning methods is discussed and established stable base for future research.

REFERENCES

[1] PR. Chandler, M. Pachter, D. Swaroop, M. Foeler, K. Howlett, S. Rasmussen et al, “Complexity in UAV cooperative control”, Proceedings of American control conference, Anchorage, AK, 2002, pp.1831-1837.

[2] Sai-Ming Li, Jovan D.Boskovic, Sanjeev Seereeram, Randal W. Beard, Ravi Prasanth,Jayesh, Amin et al, “Autonomous hierachical control of of multiple unmanned combat air vehicles”,American Control Conference, Anchorage, AK,2002, pp.274-280.

[3] Robert Zlot and Anthony Stentz,“Market-based complex task alloction for multirobot teams”, The robotics institute,Carnegie Mellou University Pittsburgh,PA, ADA432898, pp.1-9.

[4] Richards A, Bellingham J, Tillerson M. and How JP, “Coordination and control of multiple UAVs”, In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, 2002, pp.1-11.

[5] Tal Shima, J. Rasmussen, G. Sparks and M. Passino “Multiple task assignment for cooperating uninhabited aerial vehicles using genetic algorithms”, Computers & Operations Research 33 (2006), pp. 3252-3269.

[6] Dorigo, M., Maniezzo, V. and Colorini, A., “Ant system: optimization by a colony of cooperating agents”, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1), pp.29-41.

[7] Dorigo, M.,Birattari,M. and Stützle,T., “Ant colony optimization”, IEEE Transactions on Computational Inteligence Magazine, November, 2006, pp.28-40.

[8] Genshe Chen and Jose B Cruz, “Genetic algorithm for task allocation in UAV cooperative control”, AIAA Guidance, Navigation, and Control Conference and Exhibit, Augest 2003, Austin, Texas.