

Research Article

Cooperative Task Allocation Method of MCAV/UCAV Formation

Yun Zhong,¹ Peiyang Yao,¹ Yu Sun,¹ and Juan Yang²

¹Information and Navigation College, Air Force Engineering University, Xi'an 710077, China

²Unit 93010 of Chinese People's Liberation Army, Shenyang 110015, China

Correspondence should be addressed to Yun Zhong; 13227739751@189.cn

Received 5 August 2016; Revised 10 November 2016; Accepted 29 November 2016

Academic Editor: Eusebio Valero

Copyright © 2016 Yun Zhong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Unmanned Combat Aerial Vehicle (UCAV) cooperative task allocation under Manned Combat Aerial Vehicle's (MCAV) limited control is one of the important problems in UCAV research field. Hereto, we analyze the key technical and tactical indices influence task allocation problem and build an appropriate model to maximize the objective function values as well as reflecting various constraints. A novel improved multigroup ant colony algorithm (IMGACA) is proposed to solve the model; the algorithm mainly includes random sequence-based UCAV selection strategy, constraint-based candidate task generation strategy, objective function value-based state transition strategy, and crossover operator-based local search strategy. Simulation results show that the built model is reasonable and the proposed algorithm performs well in feasibility, timeliness, and stability.

1. Introduction

UCAVs being equipped with communication system have been widely used in battlefield under the fierce battlefield countermeasure and complex battlefield environment [1]. By means of forming task coalition, MCAVs cooperate with UCAVs to execute combat tasks, and in this way they can exert respective advantages [2]. By taking advantages of strong maneuver, outstanding stealth, and low cost, UCAVs get deep into the battlefield front with harsh battlefield environment, carrying out various combat tasks such as battlefield detection, interference, attack, and assessment, while MCAVs mainly provide command and control (C2) outside the combat zone and give full play to MCAV commanders' comprehensive judgment and rational decision-making ability.

MCAV/UCAV cooperative task allocation problem is one of the key technologies in MCAV/UCAV cooperative engagement, and it is a NP-hard problem [3]. Firstly, it takes task types, values, and locations into account and then establishes the allocation model which maximizes allocation efficiency function and meets all technical and tactical constraints, and finally it applies the appropriate algorithm to solve the model. Currently, the research on the problem focuses primarily on model building [4–6] and algorithm solving [7, 8].

In terms of task allocation models, researchers usually abstracted the task allocation problem into existing mature models such as multiple travelling salesman problem (MTSP), mixed integer linear programming (MILP), and vehicle routing problem (VRP). Besides, Air Force Research Laboratory (AFRL) raised a model called capacitated vehicle routing problem with time window (CVRPTW) and applied it to reconnaissance mission planning of Global Hawks and Predators [9].

As far as task allocation algorithms are concerned, they mainly include centralized algorithms and distributed algorithms. The former ones include integer programming algorithm, swarm intelligence heuristic algorithm and evolutionary algorithm, and the latter ones include contract net protocol and auction algorithm, which are based on market mechanism and realize dynamic task allocation through negotiation. In [10], a novel task allocation method based on genetic algorithm (GA) was proposed; it realized quick and effective allocation within a relatively short time, while, in [11], the balance of UCAVs' task load was considered, and the adaptive inertia weight improved particle swarm algorithm was introduced to solve the allocation model; in the end, the business contract mechanism was adopted to realize task coordination. Choi et al. [12] put forward two decentralized algorithms to coordinate a fleet of autonomous

vehicles: consensus-based auction algorithm (CBAA) and its generalization to the multiallocation problem, that is, consensus-based bundle algorithm (CBBA), both of which produced conflict-free feasible solutions that were robust to inconsistencies in the situational awareness and communication topology changes via the synchronous interaction of information status. Furthermore, to solve the information transmission chaos problem in asynchronous communication system in [12], an asynchronous consensus-based bundle algorithm (ACBBA) was proposed in [13]; however, it still could not realize cooperation between UCAVs.

Ant colony algorithm (ACA) was first presented by Italian scholar Dorigo and Gambardella [14] to solve travelling salesman problem (TSP), and it has been applied to solve various kinds of combinatorial optimization problems afterwards [15–17]. The principle of ACA is to simulate the real ant colony foraging mechanism in nature and introduce heuristic information into pheromone release/volatile mechanism, which will speed up the convergence rate of ACA.

The remainder of this paper is organized as follows. A task allocation optimization model of MCAV/UCAV formation, which overall considers various constraints, is built in Section 2, and in this section, mathematical complexity of the model is also analyzed. Based on ACA, we propose a novel improved multigroup ant colony algorithm (IMGACA) in Section 3, and the algorithm includes random sequence-based UCAV selection strategy, constraint-based candidate task generation strategy, objective function value-based state transition strategy, and crossover operator-based local search strategy. By adopting these strategies, the ability of algorithm to eliminate conflicts and escape from local optimum is enhanced. Three simulation cases are designed to verify that the algorithm is feasible, time-effective, and stable in Section 4. At last, in Section 5, a summarization is given, and the future work is prospected.

2. Model Building

Basic elements in MCAV/UCAV cooperative task allocation problem include MCAVs, UCAVs, and tasks. Under the centralized-distributed command and control structure of MCAV, UCAVs formation attacks enemy targets cooperatively. The set of UCAVs is $\mathbf{U} = \{U_1, U_2, \dots, U_{N_v}\}$, where N_v is the number of UCAVs in the set, and the attribute values of U_i ($i = 1, 2, \dots, N_v$) mainly include location, value, task load, and velocity, which are represented as LU_i , VU_i , TL_i , and Vel_i , respectively. The set of tasks is $\mathbf{T} = \{T_1, T_2, \dots, T_{N_t}\}$, where N_t is the number of tasks in the set, and the attribute values of T_j ($j = 1, 2, \dots, N_t$) mainly include location and value, which are represented as LT_j and VT_j , respectively. There are no no-fly zones and unexpected obstacles.

Figure 1 shows a typical task execution process of MCAV/UCAV task coalition. Tasks are mainly executed by UCAVs; only when asked for cooperation request, MCAV provides cooperative guidance support.

2.1. Assumptions. To simplify the problem, three assumptions are given.

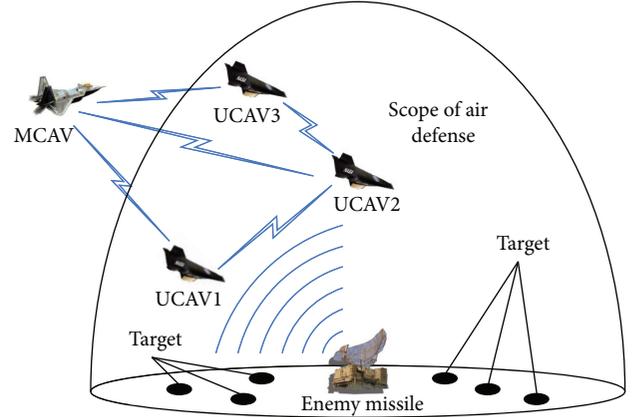


FIGURE 1: Typical task execution process.

Assumption 1. For MCAV/UCAV cooperative engagement system, all platforms are not limited to flight range.

Assumption 2. All platforms move in a straight line; that is, there are no platform kinematical constraints.

Assumption 3. The number of tasks N_t is less than the sum of task loads of all UCAVs: $\sum_{i=1}^{N_v} TL_i = N_t$; that is, all tasks can be executed.

2.2. Definition of Decision Variables and Constraint Conditions. To build the mathematical model for the MCAV/UCAV cooperative task allocation, at first, we define the related decision variables.

- (i) Allocation variable x_{ij} , which represents the element in task allocation matrix $X = (x_{ij})_{N_v \times N_t}$, $x_{ij} = 1$, denotes the idea that U_i executes T_j , while $x_{ij} = 0$ on behalf of the opposite meaning.
- (ii) Transfer variable y_{nji} , which represents the transfer status between tasks, $y_{nji} = 1$, denotes the idea that U_i is assigned to T_j after executing T_n , while $y_{nji} = 0$ on behalf the opposite meaning. In particular, if $n = j$ is true, then $y_{nji} = 0$.

To U_i and T_j , if $x_{ij} = 1$, then it contains two cases. The first case is that U_i is assigned to T_j after executing T_n , which means $y_{nji} = 1$. The second case is that U_i is assigned tasks for the first time; that is, T_n has no leading tasks, which means $y_{nji} = 0$. Assuming that there is a virtual task, T_0 should be assigned to all UCAVs at first, so the constraint relationship between x_{ij} and y_{nji} can be expressed as

$$\sum_{n=0}^{N_t} y_{nji} - x_{ij} = 0, \quad i = 1, 2, \dots, N_v; \quad j = 1, 2, \dots, N_t. \quad (1)$$

Meanwhile, after executing T_n , U_i can be assigned only one task; namely,

$$\sum_{j=1}^{N_t} y_{nji} \leq 1, \quad i = 1, 2, \dots, N_v; \quad n = 0, 1, \dots, N_t. \quad (2)$$

In addition, in consideration of task load constraint, task number constraint, and so on, we give task allocation constraints as follows:

$$\sum_{j=1}^{N_t} x_{ij} \leq TL_i, \quad i = 1, 2, \dots, N_v \quad (3)$$

$$\sum_{i=1}^{N_v} x_{ij} = 1, \quad j = 1, 2, \dots, N_t \quad (4)$$

$$\sum_{i=1}^{N_v} \sum_{j=1}^{N_t} x_{ij} \leq N_d = \min(N_t, N_v) = N_t, \quad (5)$$

where formula (3) represents the task load constraint, formula (4) denotes the idea that a task only can be executed by UCAV for once, and formula (5) represents the task number constraint.

2.3. MCAV Trust Degree. Due to the heterogeneous characteristics of UCAV, it is bound to have differences in performance and result when different UCAVs execute the same task. Hence, when decomposing a mission into several tasks, the MCAV commander has his subjective preference, which is defined as trust degree. The value of trust degree depends on task executed quality (TEQ) of UCAVs, drawing on the experience of [18]; TEQ_{ij} is defined as

$$\text{TEQ}_{ij} = \frac{1}{1 + \exp[6(1 - 2AT_{ij})]}, \quad (6)$$

where AT_{ij} ($i = 1, 2, \dots, N_v; j = 1, 2, \dots, N_t$) is U_i 's kill probability to T_j .

2.4. Attack Benefit Function. The definition of attack benefit is the benefit value gained during the execution of tasks, and it is related to the MCAV trust degree. This paper defines attack benefit function as

$$G = \sum_{i=1}^{N_v} \sum_{j=1}^{N_t} \text{TEQ}_{ij} \cdot x_{ij}. \quad (7)$$

2.5. Attack Cost Function. When U_i executes combat tasks, it has got to pay the corresponding costs such as threat cost, flight range cost, and ammunition cost; we will define the concept of these three costs in the following text.

If T_j 's damage probability to U_i is DA_{ij} , the value of U_i is VU_i , then the threat cost of T_j to U_i can be expressed as C_1 , and the calculating formula is

$$C_1 = \sum_{i=1}^{N_v} \left[\sum_{j=1}^{N_t} DA_{ij} \cdot x_{ij} \right] \cdot VU_i. \quad (8)$$

In addition, to make the flight range of all UCAVs in MCAV/UCAV cooperative engagement system as short as possible is also an important optimization objective; we define the flight range cost as

$$C_2 = \sum_{i=1}^{N_v} \left[\sum_{j=1}^{N_t} \frac{D_{ij}}{D_{\max}} \cdot x_{ij} \right], \quad (9)$$

where D_{ij} represents the distance from U_i to T_j , and D_{\max} is the maximum value within D_{ij} .

And lastly, if the different types of ammunition loaded on UCAVs have the similar attack effect to the same task, then it should be possible to choose the ammunition that has lower cost. Suppose the cost of ammunition Am_i loaded on U_i is VA_i ; the ammunition cost in task allocation is

$$C_3 = \sum_{i=1}^{N_v} \left[\sum_{j=1}^{N_t} VA_i \cdot (1 - AT_{ij}) \cdot x_{ij} \right]. \quad (10)$$

So the total cost is

$$C = C_1 + C_2 + C_3. \quad (11)$$

2.6. Allocation Model. MCAV commanders always want task completion time to be as short as possible, in MCAV/UCAV cooperative engagement system; it is necessary to introduce the influence of task completion time into the model. Suppose the expected task completion time of MCAV commanders is t_{exp} ; then the definition of task completion time as well as aging factor is expressed as

$$t_{\text{cpl}} = \max_{j \in \{1, 2, \dots, N_t\}} t_j \quad (12)$$

$$AF_{\text{time}} = \lambda^{t_{\text{cpl}}/t_{\text{exp}}},$$

where t_j is the completion time of task T_j , and λ is regulatory parameter and satisfied with $0 < \lambda < 1$.

From the above analysis, it can be seen that the MCAV/UCAV cooperative task allocation is not a single-objective optimization problem but a multiobjective optimization problem. We adopt linear weight sum method to translate it to a single-objective optimization one; therefore, the objective function is defined as

$$\max f = (\omega_1 G - \omega_2 C) \cdot AF_{\text{time}}, \quad (13)$$

where ω_1 and ω_2 are weighting coefficient of G and C , respectively; we usually take them to be 1.

The task allocation model can be expressed as

$$\begin{aligned}
\max \quad & f \\
& \sum_{n=0}^{N_t} y_{nji} - x_{ij} = 0 \\
& \sum_{j=1}^{N_t} y_{nji} \leq 1 \\
& \sum_{j=1}^{N_t} x_{ij} \leq TL_i \\
& \sum_{i=1}^{N_v} x_{ij} = 1 \\
& \sum_{i=1}^{N_v} \sum_{j=1}^{N_t} x_{ij} \leq N_d = \min(N_t, N_i) = N_t \\
& x_{ij}, y_{nji} \in \{0, 1\} \\
& i = 1, 2, \dots, N_v; \quad n = 0, 1, \dots, N_t; \\
& j = 1, 2, \dots, N_t.
\end{aligned} \tag{14}$$

2.7. Mathematical Complexity of the Model. Analyzing the problem size, the number of feasible solutions is provided below.

Lemma 4. *If $N = N_1 + N_2 + \dots + N_n$ (N, N_1, N_2, \dots, N_n are all positive integers), then $N! \geq N_1! \times N_2! \times \dots \times N_n!$ is always to be true.*

Proof. Firstly, take a simple case; if $N = N_1 + N_2$, then $N_1! \times N_2! = N_1! \times (N - N_1)! = 1 \times 2 \times \dots \times N_1 \times 1 \times 2 \times \dots \times (N - N_1) \leq 1 \times 2 \times \dots \times N_1 \times (N_1 + 1) \times (N_1 + 2) \times \dots \times N = N!$ is always to be true. Then, the conclusion just drawn above can be extended to general case; that is, if $N = N_1 + N_2 + \dots + N_n$, then $N_1! \times N_2! \times \dots \times N_n! \leq N_{12}! \times N_3! \times N_4! \times \dots \times N_n! \leq N_{123}! \times N_4! \times N_5! \times \dots \times N_n! \leq \dots \leq N_{123\dots n}! = N!$, where $N_{12} = N_1 + N_2, N_{123} = N_1 + N_2 + N_3, N_{123\dots n} = N_1 + N_2 + \dots + N_n$. \square

Theorem 5. *When task loads of UCAVs are unlimited, the upper bound of feasible solution number is $N_s = 2P(N_t, N_t)$.*

Proof. According to Assumption 3, $N_t \leq N_l$, so it can be seen that N_t tasks are allocated to N_l task loads independently, and the permutation of allocation plan is $C(N_l, N_t)$. So if N_t tasks are allocated among N_v UCAVs, there are two cases: (i) the most extreme case is that all tasks are allocated to one UCAV, and, in this case, the permutation of N_t tasks is $N_t!$; (ii) tasks are allocated to more than two UCAVs; based on Lemma 4, in this case, the permutation of N_t tasks is less than $N!$. So the total feasible solution number is less than $C(N_l, N_t) \times (N! + N!) = 2C(N_l, N_t) \times N! = 2P(N_l, N_t)$. \square

3. IMGACA

3.1. Establishment of Job-Division Ant Colony. There are obvious differences between cooperative task allocation problem and TSP, and the latter one can be solved effectively by traditional ACA. The main differences are listed as follows.

- (i) Task execution has parallel characteristic; that is, tasks are likely to be executed by UCAVs simultaneously, and it is difficult for traditional ACA to reflect the influence of such parallel characteristic.
- (ii) Platform has collaborative characteristic, and there exists task and resource collaboration among different UCAVs; traditional ACA is difficult to handle this kind of collaborative relationship with mutual coupling.
- (iii) Task execution has aftereffect characteristic; that is, due to the presence of various constraints, current task execution order will affect the subsequent task, and traditional ACA is difficult to solve this kind of aftereffect characteristic.

So, based on the analysis to MCAV/UCAV cooperative task allocation, UCAVs are mapped to ant subgroups in ant colony, ants belonging to different subgroups with the same serial number make up ant subclusters, and each cluster realizes implicit communication via pheromone interaction to generate the allocation plan.

Suppose the number of subgroup of artificial ant colony is $N_{SG} = N_v$; for U_i , the number of ants in subgroup is m . The set of U_i 's ant subgroup is $\mathbf{SG}_i = \{\text{ant}_{ij}, j = 1, 2, \dots, m\}$; all ant subgroups form ant colony system $\mathbf{G} = \{\mathbf{SG}_i, i = 1, 2, \dots, N_{SG}\}$. Note that ant_{ij} and ant_{kj} belonging to different ant subgroups \mathbf{SG}_i and \mathbf{SG}_k , respectively, will influence each other while generating their own task allocation plan; to avoid this situation, we establish m local tabu table $\mathbf{Tabu} = \{\text{tabu}_1, \text{tabu}_2, \dots, \text{tabu}_m\}$, and each tabu table is written in the task index that has been executed by other ants and in this way can solve the repeated execution problem for tasks.

Above all, job-division ant colony can be shown as ant matrix in Figure 2, the matrix row represents all ant subgroups, and the matrix column represents all ant subclusters $\mathbf{SG}_j = \{\text{ant}_{ij}, i = 1, 2, \dots, N_v\}$.

3.2. Random Sequence-Based UCAV Selection Strategy. When selecting specific UCAV to execute task, deterministic selection sequence will lead the search algorithm to trap in local optimum, so, to guarantee the solution diversity in the algorithm solving process, random sequence is adopted to select UCAV. Moreover, to avoid conflict, it is necessary to exclude that the UCAV has reached maximum task load.

3.3. Constraint-Based Candidate Task Generation Strategy. By formulas (3), (4), and (5), in the MCAV/UCAV cooperative task allocation problem, constraints mainly include load constraint of UCAVs and number constraint of tasks. To avoid the generation of infeasible solutions brought by blindness or randomness of search algorithm and further result in performance degrading of algorithm, when ant_{ij}

Input: $\mathbf{TA}(\text{ant}_{i-j})$, $\mathbf{allow_T}(\text{ant}_{ij}) = \emptyset$, \mathbf{T} , \mathbf{TL}_i ; Output: $\mathbf{allow_T}(\text{ant}_{ij})$

- (1) **procedure** Select ant_{ij} and task set \mathbf{T}
- (2) **if** $|\mathbf{TA}(\text{ant}_{ij})| < \mathbf{TL}_i$ **then**
- (3) **if** $T_j \notin \mathbf{TA}(\text{ant}_{i-j})$ **then**
- (4) $\mathbf{allow_T}(\text{ant}_{ij}) = \mathbf{allow_T}(\text{ant}_{ij}) \cup \{T_j\}$, $\forall j \in [1, N_i]$
- (5) **end if**
- (6) **end if**
- (7) **end procedure**

ALGORITHM 1: CTSGA for ant_{ij} based on constraint condition. Where $|\cdot|$ denotes the cardinality of the vector, $\mathbf{TA}(\text{ant}_{ij})$ and $\mathbf{TA}(\text{ant}_{i-j})$ denote the task set whose task element has been executed by ant_{ij} and other ants in \mathbf{SC}_j except for ant_{ij} , respectively.

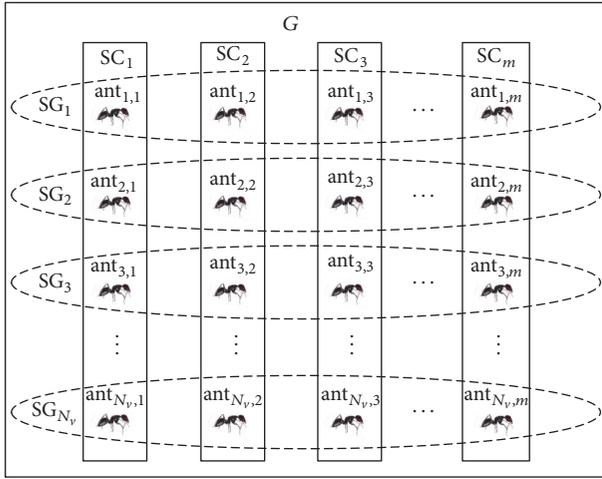


FIGURE 2: Ant colony based on the job-division mechanism.

chooses candidate task in the set $\mathbf{allow_T}(\text{ant}_{ij})$, it should screen out those tasks that do not meet constraints; only in this way can computational scale and complexity be reduced.

Based on the above analysis, a candidate task set generating algorithm (CTSGA) is proposed; the details are shown in Algorithm 1.

3.4. Objective Function Value-Based State Transition Strategy.

When ant_{ij} selects a task in $\mathbf{allow_T}(\text{ant}_{ij})$, which one to be selected and how to be selected are significant issues. In this paper, pseudo-random proportional rule is considered to be a reasonable method and it can be expressed as

$$y = \begin{cases} \arg \max_{s \in \mathbf{allow_T}(\text{ant}_{ij})} \{ [\tau_{rs}(k)]^\alpha \cdot [\eta_{rs}(k)]^\beta \} & \text{if } q \leq q_0 \\ Y & \text{else,} \end{cases} \quad (15)$$

where $\tau_{rs}(k)$ is pheromone value from r to s in k th iteration and $\eta_{rs}(k)$ is heuristic information value from r to s in k th iteration, while α and β represent pheromone factor and heuristic information factor, respectively, reflecting the relative importance of pheromones and heuristic information in solving process to get feasible solutions. And q_0 is a

constant value that lies between 0 and 1, while q is a random number that obeys uniform distribution in $[0, 1]$.

In other words, ants will select the optimal path for state transition with probability q_0 and will explore a new path in the roulette way with probability $1 - q_0$. Y in formula (15) is a selected task in random proportional rule shown as follows:

$$p_k(r, s) = \begin{cases} \frac{[\tau_{rs}(k)]^\alpha \cdot [\eta_{rs}(k)]^\beta}{\sum_u [\tau_{ru}(k)]^\alpha \cdot [\eta_{ru}(k)]^\beta}, & u \in \mathbf{allow_T}(\text{ant}_{ij}) \\ 0, & u \notin \mathbf{allow_T}(\text{ant}_{ij}). \end{cases} \quad (16)$$

ACA has the positive feedback, self-organization and distributed characteristics; the positive feedback mechanism especially will lead search algorithm to trap in local optimum in later iterations [19]. Aiming at this problem, an adaptive adjustment strategy for task selection is proposed: at the earlier stage of iteration, to improve the computational convergence, we set smaller candidate tasks size, while at the later stage of iteration, to ensure the diversity of solutions, we set larger candidate tasks size.

The concept of task selection window, which can be utilized to shift between intensification and diversification [20], is applied to set the size of candidate tasks dynamically. When ant_{ij} selects a specific path at a certain probability, it does not mean that all candidate tasks in $\mathbf{allow_T}(\text{ant}_{ij})$ will have the chance to be selected as the next executed task; specific steps are as follows.

Step 1. Sort all tasks in $\mathbf{allow_T}(\text{ant}_{ij})$ on the basis of aggregation operator of pheromone and heuristic information value.

Step 2. Judge if current number of iteration has reached half of the total number of iterations; if so, set $\sigma_1 = 0.5$; otherwise, set $\sigma_2 = 0.9$.

Step 3. On the basis of formula (17), it generates updated size of candidate task set, where σ is filtration factor, the value of σ is assigned σ_1 or σ_2 depending on the iterative process, and $\lceil \cdot \rceil$ is rounded-up operation. Note that if $\mathbf{allow_T}(\text{ant}_{ij})$ contains only one task, then it will be selected immediately.

$$|\mathbf{allow_T}(\text{ant}_{ij})| \leftarrow \lceil \sigma \cdot |\mathbf{allow_T}(\text{ant}_{ij})| \rceil. \quad (17)$$

3.5. Crossover Operator-Based Local Search Strategy. ACA has strong global search ability, but search efficiency is low; therefore, local optimizer is introduced to raise its solution precision and convergence rate.

After all ant subclusters finish a round of searching, the feasible solution space Θ_k ($k = 1, 2, \dots, K$) is formed, and we utilize neighborhood heuristic knowledge to direct its local search; the steps are as follows.

Step 1. Sort all objective function values in $\mathbf{U} - \mathbf{T}$ allocation matrix where $x_{ij} = 1$, and seek out the $U_i - T_j$ pair which has the minimum objective function value.

Step 2. Find out UCAVs whose remaining task load is greater than 0 in \mathbf{U} except for U_i , then put these UCAVs into the set \mathbf{FY} , and, at last, calculate and record the element number fy of \mathbf{FY} .

Step 3. Allocate T_j to UCAV in \mathbf{FY} randomly; if it satisfies the idea that the variation of objective function value $\Delta^{\text{swap}}(U_i, U_{i'}, T_j) > 0$, then change T_j 's execution platform from U_i to $U_{i'}$; otherwise, stay the same.

Step 4. Update \mathbf{FY} , and repeat Steps 1 – 3 for N times; N is a positive integer, and its usual value is set to be 3.

3.6. Pheromone Update Strategy. In multigroup ant colony search system, only iteration-best ant can update global pheromone; the goal is to make the search process more targeted and guide ants to search near the optimal solution in the current iteration. So when IMGACA updates global pheromone, it mainly adopts ant-density system update strategy, and the updating formula is defined as

$$\tau(r, s) = (1 - \rho_{\text{glo}}) \cdot \tau(r, s) + \rho_{\text{glo}} \cdot Q, \quad (18)$$

where Q is a constant value and ρ_{glo} is a global update factor, and it is satisfied with $0 < \rho_{\text{glo}} < 1$.

When updating local pheromone, in order to guide ants to search near the solution where task completion time is shorter, it mainly adopts ant-quantity system update strategy, and the principle is to release more pheromone in such search path.

$$\tau(r, s) = (1 - \rho_{\text{loc}}) \cdot \tau(r, s) + \rho_{\text{loc}} \cdot \frac{Q}{E(t)} \quad (19)$$

$$E(t) = \frac{t_{\text{SC}_k}}{\min t_{\text{SC}_k}},$$

where ρ_{loc} is a local update factor, and it is satisfied with $0 < \rho_{\text{loc}} < 1$, and t_{SC_k} is task completion time of the k th ant subcluster SC_k .

4. Simulation Experiment

4.1. Parameter Setting. For purpose of verifying the superiority of the model and IMGACA, we run the IMGACA on a PC with Pentium (R) Dual-Core CPU E6600 3.06 GHz, and it is based on MATLAB R2010a software environment. Parameter

TABLE 1: Task information.

Task	LT	VT
T_1	(568, 508)	22.3
T_2	(259, 113)	32.6
T_3	(307, 365)	54.2
T_4	(467, 621)	34.6
T_5	(134, 311)	35.4
T_6	(514, 236)	33.1
T_7	(384, 469)	32.6
T_8	(156, 543)	43.4
T_9	(469, 182)	31.9
T_{10}	(281, 503)	44.1
T_{11}	(456, 247)	62.5
T_{12}	(89, 254)	33.6
T_{13}	(313, 256)	32.9
T_{14}	(617, 92)	23.2
T_{15}	(418, 507)	44.3
T_{16}	(383, 116)	23.0
T_{17}	(164, 229)	67.4
T_{18}	(605, 539)	33.4

TABLE 2: UCAV information.

UCAV	LU	VU	Vel	VA
U_1	(156, 443)	4.5	200	1.4
U_2	(195, 487)	6.0	150	2.3
U_3	(312, 584)	5.5	250	1.6
U_4	(237, 285)	3.0	250	1.6
U_5	(565, 591)	3.5	200	2.5
U_6	(652, 368)	5.0	300	2.3

setting in the IMGACA is listed as follows: $t_{\text{exp}} = 10$, $\alpha = 3$, $\beta = 8$, $\lambda = 0.85$, $m = 50$, $q_0 = 0.75$, $Q = 3$, $\rho_{\text{glo}} = 0.05$, and $\rho_{\text{loc}} = 0.02$.

Table 1 lists the task information including location and value.

Table 2 lists the UCAV information including location, value, velocity, and value of ammunition loaded on UCAVs.

4.2. Feasibility Analysis. Set the number of UCAVs as 6 and that of tasks as 18; besides, the matrices representing the kill probability of UCAVs to tasks and the damage probability of tasks to UCAVs are \mathbf{AT} and \mathbf{DA} ; they are, respectively, set to be as follows.

We make a comparative simulation between simulation scenarios with limited (task load vector is $\mathbf{TL} = [2, 2, 2, 2, 2, 8]$) or unlimited task load for all UCAVs; the iteration number is 600, and, as shown in Table 3, is the simulation results in the two cases.

It is observed that when task load of UCAVs is limited, the task allocation solution is not an optimal one, and when

TABLE 3: Task allocation plan with limited or unlimited task load.

	UCAV	Flight range	Attack benefit	Attack cost	Allocation plan
Limited	U_1	380.47	61.10	3.96	$T_{10} \rightarrow T_5$
	U_2	573.81	64.85	3.61	$T_{18} \rightarrow T_4$
	U_3	182.01	65.78	5.64	$T_{15} \rightarrow T_7$
	U_4	498.24	68.71	3.99	$T_9 \rightarrow T_3$
	U_5	504.14	78.20	5.33	$T_{11} \rightarrow T_{13}$
	U_6	2433.92	246.59	21.56	$T_{17} \rightarrow T_{12} \rightarrow T_8 \rightarrow T_6 \rightarrow T_{16} \rightarrow T_2 \rightarrow T_{14} \rightarrow T_1$
Unlimited	U_1	927.81	82.11	7.96	$T_5 \rightarrow T_{16} \rightarrow T_{18}$
	U_2	755.31	93.52	4.95	$T_{17} \rightarrow T_4$
	U_3	500.71	62.54	5.42	$T_7 \rightarrow T_{12}$
	U_4	839.52	132.23	7.74	$T_3 \rightarrow T_8 \rightarrow T_{13} \rightarrow T_9$
	U_5	1020.19	144.63	9.13	$T_{10} \rightarrow T_{11} \rightarrow T_{15} \rightarrow T_1$
	U_6	726.63	78.54	6.70	$T_6 \rightarrow T_{14} \rightarrow T_2$

task load of UCAVs is unlimited, the task allocation solution is more balanced and better feasible one.

$$\begin{aligned}
 \mathbf{AT} &= \begin{bmatrix} 0.4 & 0.1 & 0.7 & 0.5 & 0.6 & 0.8 & 0.1 & 0.3 \\ 0.9 & 0.4 & 0.9 & 0.7 & 0.8 & 0.1 & 0.7 & 0.5 \\ 0.8 & 0.3 & 0.3 & 0.2 & 0.5 & 0.5 & 0.8 & 0.4 \\ 0.4 & 0.1 & 0.8 & 0.6 & 0.2 & 0.2 & 0.8 & 0.6 \\ 0.9 & 0.3 & 0.1 & 0.5 & 0.1 & 0.1 & 0.9 & 0.2 \\ 0.7 & 0.6 & 0.5 & 0.6 & 0.3 & 0.8 & 0.2 & 0.6 \end{bmatrix} \\
 \mathbf{DA} &= \begin{bmatrix} 0.3 & 0.1 & 0.5 & 0.4 & 0.4 & 0.5 \\ 0.1 & 0.4 & 0.2 & 0.2 & 0.5 & 0.3 \\ 0.3 & 0.5 & 0.2 & 0.4 & 0.1 & 0.3 \\ 0.3 & 0.2 & 0.4 & 0.5 & 0.4 & 0.3 \\ 0.3 & 0.2 & 0.1 & 0.4 & 0.1 & 0.5 \\ 0.1 & 0.5 & 0.3 & 0.3 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.4 & 0.4 & 0.5 & 0.1 \\ 0.3 & 0.4 & 0.2 & 0.5 & 0.1 & 0.3 \end{bmatrix}.
 \end{aligned} \tag{20}$$

Figures 3–5 give, respectively, comparison results of the objective function value, task completion time, and flight range of all UCAVs between the two cases. It can be seen from Figure 3 that when task load of UCAVs is unlimited, the final objective function value is superior to that of another one; it is because the feasible solution space of the former case is larger, and it is more likely to find out a better feasible solution, while the convergence rate is slower. From Figures 4 and 5, we can draw the conclusion that the task completion time of the case with unlimited task load is much shorter, while the flight range is longer. This is because, with limited task load, the task allocation is unbalanced, tasks are mainly executed by U_6 , and the task completion time is relatively long.

4.3. *Timeliness Analysis.* Select the general situation that task load is limited (task load vector is $\mathbf{TL} = [4, 5, 4, 3, 4, 3]$), set the simulation time to be 50 s, and take an average for the

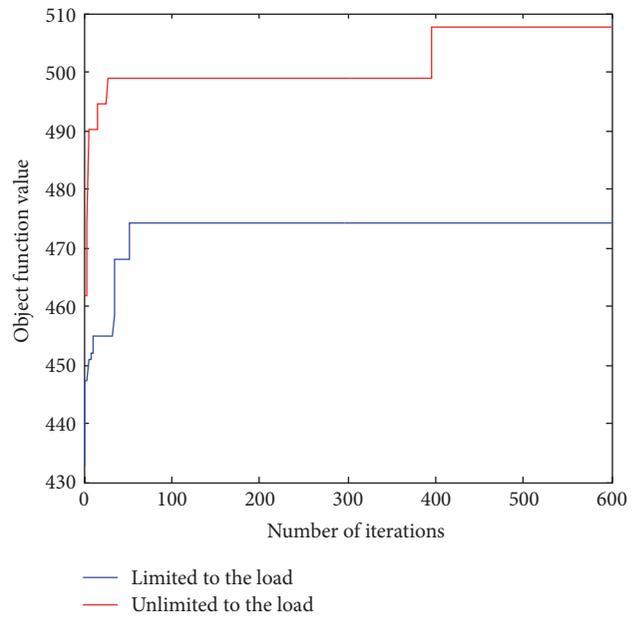


FIGURE 3: Comparison figure on objective function value.

TABLE 4: Comparison result between IMGACA and GA in 50 s.

	Iterations	Initial value	Final value
IMGACA	230.2	445.42	492.61
GA	842.3	317.58	426.28

data to compare IMGACA with GA by running 10 times, respectively.

From Table 4, we can see that when simulation time is 50 s, IMGACA can iterate for 230.2 times on average, while GA is 842.3 times; this is due to GA's algorithm design which is simple, so it can iterate for more times in unit time, because IMGACA adopts better strategy to carry out state transition, while GA adopts random generation method, so the initial value of IMGACA is superior to that of GA. When it comes to the final objective function value, IMGACA can reach 492.61

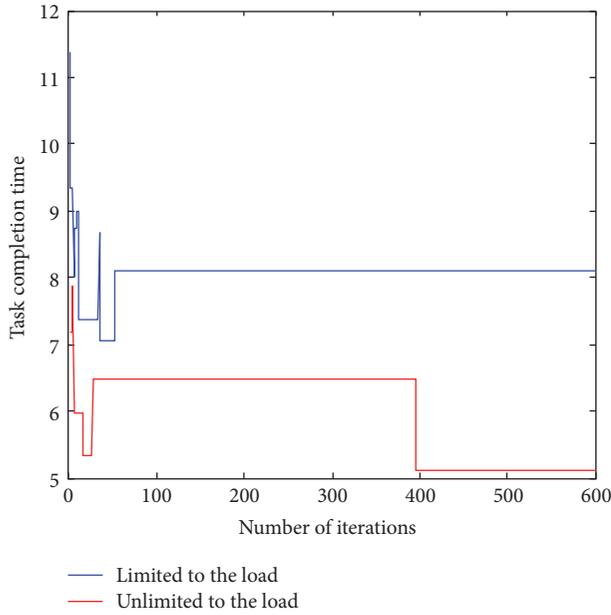


FIGURE 4: Comparison figure on task completion time.

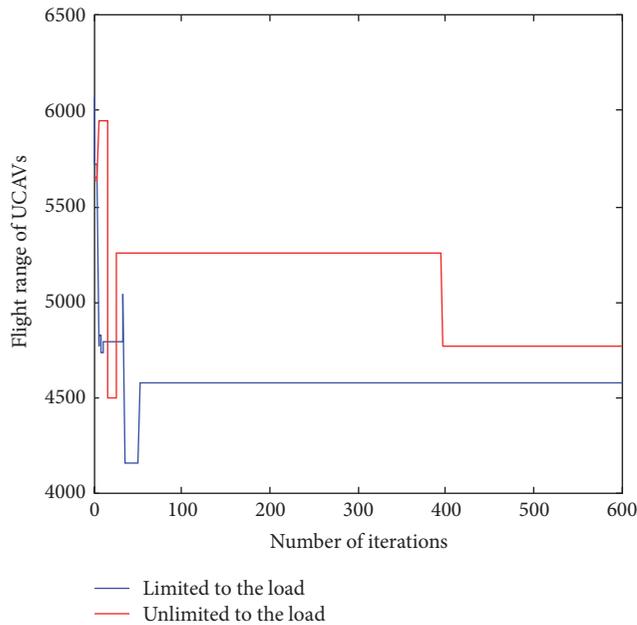


FIGURE 5: Comparison figure on flight range of all UCAVs.

and GA is 426.28, so, at the same time constraint, IMGACA outperforms GA with better find-best ability and efficiency.

4.4. Stability Analysis. According to Theorem 5, mathematical complexity of the model is high, so IMGACA and GA are difficult to converge to the global optimal solution. To prove the stability of IMGACA, we design two cases to illustrate: (i) in small problem scale, IMGACA can converge rapidly to a fixed value (it is very likely to be global optimal solution); (ii) in large problem scale, the variance of all final objective function values of IMGACA is small.

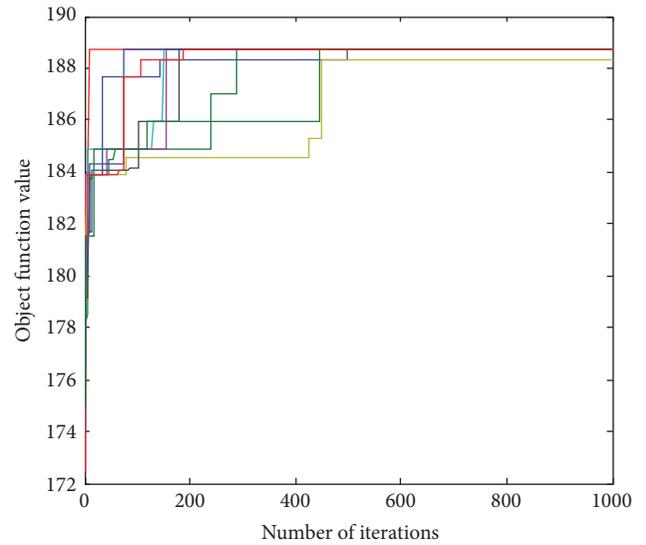


FIGURE 6: Convergence process in small problem scale.

As shown in Figure 6, the convergence process is in small problem scale for 10 times. From Figure 6, it can be seen that in small problem scale, for example, 3 UCAVs, 8 tasks (select U_1-U_3 , T_1-T_8 in Tables 1 and 2), IMGACA can always converge to 188.74, and we can draw a conclusion that IMGACA has excellent convergence ability.

However, in large problem scale, we use the simulation scenario in Section 4.3 (6 UCAVs, 18 tasks). After calculation, the variance of 10 final objective function values of IMGACA is 17.56, and that of GA is 135.40. Hence, compared to GA, IMGACA exhibits more stability in convergence.

5. Conclusions

Aiming at the UCAV cooperative task allocation problem under MCAV's limited control, we establish the task allocation model considering attack benefit, attack cost, and task completion time comprehensively and propose a novel algorithm called IMGACA to solve the model. The simulations reveal that IMGACA performs quite well in some aspects such as feasibility, timeliness, and stability.

The future work is to study the dynamic adjustment problem of task allocation plan existing parameter and event uncertainties.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Project Grant no. 61573017 and the Doctoral Foundation of Information and Navigation College, Air Force Engineering University, under Project Grant no. KGD08101604.

References

- [1] P. Delima and D. Pack, "Toward developing an optimal cooperative search algorithm for multiple unmanned aerial vehicles," in *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS '08)*, pp. 506–512, IEEE, Piscataway, NJ, USA, May 2008.
- [2] R. D. Garcia, L. Barnes, and M. Fields, "Unmanned aircraft systems as wingmen," *Journal of Defense Modeling and Simulation*, vol. 9, no. 1, pp. 5–15, 2012.
- [3] L. Johnson, *Decentralized Task Allocation for Dynamic Environments*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2012.
- [4] D. W. Casbeer and R. W. Holsapple, "Column generation for a UAV assignment problem with precedence constraints," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 12, pp. 1421–1433, 2011.
- [5] M. A. Darrah, W. M. Niland, and B. M. Stolarik, "Multiple UAV dynamic task allocation using mixed integer linear programming in a SEAD mission," in *Proceedings of the InfoTech at Aerospace: Advancing Contemporary Aerospace Technologies and Their Integration*, pp. 2324–2334, AIAA, Arlington, Va, USA, September 2005.
- [6] M. Alighanbari and J. P. How, "Decentralized task assignment for unmanned aerial vehicles," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 5668–5673, IEEE, Seville, Spain, December 2005.
- [7] E. Edison and T. Shima, "Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research*, vol. 38, no. 1, pp. 340–356, 2011.
- [8] P. Sujit, J. George, and B. Randy, "Multiple UAV task assignment using particle swarm optimization," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, pp. 18–21, Honolulu, Hawaii, USA, 2008.
- [9] R. Harder, *A Java Universal Vehicle Router in Support of Routing Unmanned Aerial Vehicles Air Missions*, Air Force Institute of Technology, Wright-Patterson, Ohio, USA, 2000.
- [10] M. A. Darrah, W. M. Niland, B. M. Stolarik, and L. E. Walp, "UAV cooperative task assignments for a SEAD mission using genetic algorithms," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 3257–3266, Keystone, Colo, USA, August 2006.
- [11] J. Y. Du, F. M. Zhang, J. Yang et al., "Cooperative task assignment for multiple UCAV using particle swarm optimization," *Control and Decision*, vol. 27, no. 11, pp. 1751–1755, 2012 (Chinese).
- [12] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.
- [13] B. Luke, S. Sameera, H. L. Choi et al., "Asynchronous decentralized task allocation for dynamic environment," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pp. 1441–1453, Minneapolis, Minn, USA, 2011.
- [14] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [15] J. Cai, Q. Li, L. Li, H. Peng, and Y. Yang, "A fuzzy adaptive chaotic ant swarm optimization for economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 34, no. 1, pp. 154–160, 2012.
- [16] E. Arcaute, K. Christensen, A. Sendova-Franks, T. Dahl, A. Espinosa, and H. J. Jensen, "Division of labour in ant colonies in terms of attractive fields," *Ecological Complexity*, vol. 6, no. 4, pp. 396–402, 2009.
- [17] Y. Li, R. Wang, and M. Xu, "Rescheduling of observing spacecraft using fuzzy neural network and ant colony algorithm," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 678–687, 2014.
- [18] L.-J. Wan, P.-Y. Yao, and P. Sun, "Distributed task allocation method of manned/unmanned combat Agents," *Systems Engineering and Electronics*, vol. 35, no. 2, pp. 310–316, 2013 (Chinese).
- [19] Y. Zhou, "Runtime analysis of an ant colony optimization algorithm for TSP instances," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1083–1092, 2009.
- [20] X. Hu, H. Ma, Q. Ye, and H. Luo, "Hierarchical method of task assignment for multiple cooperating UAV teams," *Journal of Systems Engineering and Electronics*, vol. 26, no. 5, Article ID 7347860, pp. 1000–1009, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

