



An efficient multi-objective ant colony optimization for task allocation of heterogeneous unmanned aerial vehicles

Lizhi Chen^a, Wei-Li Liu^{b,*}, Jinghui Zhong^{a,*}

^a School of Computer Science and Engineering, South China University of Technology, China

^b Guangdong Polytechnic Normal University, Guangzhou, China

ARTICLE INFO

Keywords:

Unmanned aerial vehicles
Cooperative task allocation
Multi-objective Optimization
Ant colony optimization

ABSTRACT

Unmanned aerial vehicles (UAVs) have become powerful tools in modern military combat. How to properly allocate the tasks of heterogeneous UAVs in a combat is a fundamental and challenging problem. In this paper, we formulate the cooperative task allocation of heterogeneous UAVs as a constrained multi-objective optimization problem. To efficiently resolve the formulated problem, we further propose a multi-objective ant colony optimization (MOACO) algorithm with a new pheromone updating mechanism and four newly defined heuristic information. Simulation results on test cases with different scales and characteristics have shown that the proposed methods can perform better than several recently published algorithms, in terms of convergence speed, solution quality and solution diversity.

1. Introduction

Unmanned aerial vehicle (UAV) has now become a common and powerful tool in modern military and civilian domains owing to its high mobility and low cost [1,2]. In the main combat mission of military UAVs, the Suppression of Enemy Air Defence (SEAD) usually uses heterogeneous UAVs [3] to work corporately rather than using a single type of UAV. Such a heterogeneous system has advantages of strong fault tolerance, high performance and good adaptability, but it meets challenges of cooperative task allocation and collective motion control [4,5]. Generally, the cooperative task allocation requires properly assigning different tasks to multiple UAVs with different kinematic characteristics and operational capabilities, so that certain optimization objectives can be optimized and the physical and logical constraints of UAVs can be satisfied [6].

Over the past decade, a number of efforts have been made to solve the cooperative task allocation for heterogeneous UAVs. By considering different features and constraints, the cooperative task allocation problem has been formulated as different kinds of NP-hard problems such as vehicle routing problem [7], multiple traveling salesman problem [8], and mixed integer linear programming problem [9]. Existing works to solve the cooperative task allocation problem generally can be classified into two categories. The first category utilizes deterministic algorithms such as branch and bound [10], and dynamic programming

[11], but these methods have difficulties in tackling multiple optimization objectives and constraints. They can hardly find a feasible solution as the numbers of UAVs and targets grow, because of the exponential increase of computational cost.

The second category utilizes computational intelligence (CI) algorithms such as genetic algorithm (GA) [12,13] and particle swarm optimization (PSO) [14,15]. For examples, Deng et al. [16] proposed an enhanced GA with multi-type genes to solve the problem. Wang et al. [17] developed an enhanced GA with opposition-based learning to solve the problem. In [15], the task allocation problem is solved by using an improved multi-objective quantum-behaved PSO. These CI-based algorithms have potential to find global or near global optimal solutions, but they ignore the heterogeneity of UAVs and targets during the evolutionary searching procedure, which limits their search efficiency and solution flexibility in practical applications. By using heuristic information to reflect the heterogeneity of UAVs, ACO [18] algorithm seems to be more suitable than GA and PSO to solve the task allocation for heterogeneous UAVs. In the literature [19–21], several preliminary ACOs have been proposed to solve the task allocation problem. These ACO algorithms either simply formulated the task allocation problem for UAVs as a single objective optimization problem, or simply converted the multiple objective optimization into the single objective optimization using a weighted-sum method. Therefore, a multi-objective ACO algorithm to solve the task allocation problem of UAV, which can

* Corresponding authors.

E-mail addresses: 8631649@qq.com (L. Chen), liuweili@gpnu.edu.cn (W.-L. Liu), jinghuizhong@scut.edu.cn (J. Zhong).

<https://doi.org/10.1016/j.jocs.2021.101545>

Received 10 August 2021; Received in revised form 15 December 2021; Accepted 21 December 2021

Available online 30 December 2021

1877-7503/© 2021 Elsevier B.V. All rights reserved.

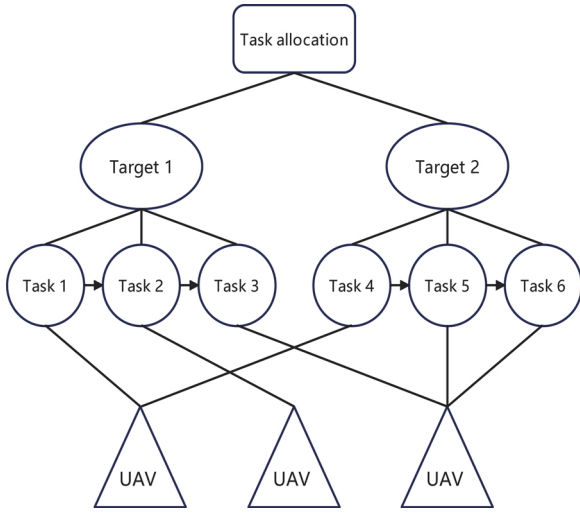


Fig. 1. Task allocation of heterogeneous UAVs.

provide decision-makers with a variety of compromise solutions and have higher flexibility in practical applications is urgently needed.

To overcome the above limitations, this paper proposes an effective multi-objective ACO to solve the cooperative task allocation for heterogeneous UAVs. The proposed methods does not require weight settings and can provide decision-makers with a set of alternative solutions which have trade-offs among multiple objectives. First, the cooperative task allocation for heterogeneous UAVs is formulated as a constrained multi-objective optimization problem, by considering different parts of interest, e.g., UAV, target, task, and path planning. The problem model contains three optimization objectives, namely, task benefit, UAV damage, and total range, under both physical and logical constraints. A new multi-colony strategy and a pheromone updating method are proposed in the MOACO to improve the convergence speed and search efficacy. The performance of the proposed MOACO is verified in comparison with several recently published methods, with performance indices of convergence speed, solution quality and solution diversity.

The rest of this paper is organized as follows. Section 2 formulates the problem model, by detailedly describing problem assumption. Section 3 provides details of the proposed MOACO. Section 4 conducts experiments on the MOACO algorithm. And finally Section 5 draws the conclusions.

2. Task allocation of heterogeneous UAVs

2.1. Problem assumptions

In the SEAD operation, three types of tasks are performed sequentially for each target [22]. The first task is reconnaissance, which aims to reconnoiter the target to improve the attack accuracy. The second task is attack, which attacks the target after reconnaissance. The third task is verification, which aims to verify whether the attacking purpose has been achieved or not. One task of a target could not be performed if its predecessor task has not been completed. Each task of a target is assigned to one and only one UAV, but one UAV can perform multiple tasks of different targets. Fig. 1 shows the schematic of the task allocation in the SEAD operation. The arrows indicate task order of the same target.

As done in [23,24], the following preconditions are adopted to model the problem.

1. The positions of targets are known in advance and kept fixed.
2. The velocity of each UAV is constant.
3. All UAVs take off and land on the same platform.

Table 1
Attributes of targets and tasks.

Model	Attribute	Parameter
Target, T_i	Number of the targets	N_T
	Number of the task types	N_{type}
	Target location	$L_i^T = (X_i^T, Y_i^T)$
	Reward	V_i^T
	Current task type	M_i^T
	Task end time	E_i^T
	Threat radius	R_i^{threat}
Task, M_k	Threat level	β_i
	Number of the tasks	N_K
	Task type	M_{type}
	Target belong	T_{belong}^k

Table 2
Attributes of UAVs.

Model	Attribute	Parameter
UAV, U_j	Number of UAVs	N_U
	Location	$L_j^U = (X_j^U, Y_j^U)$
	Speed	$Speed_j$
	Cost	V_j^U
	Detection radius	R_j
	Ammunition	A_j
	Range limit	S_j
	Task capability	P_j^T
	Execution time	t_j
	Attack time	Δt_{attack}

4. UAVs cannot continuously perform tasks of the same target.
5. The turning radius of UAV is ignored because it is too small compared with the flying distance.
6. UAVs perform tasks in different altitudes without route intersection. Therefore, the spatial dimension of UAVs is identified as two-dimensional.

2.2. Problem definitions

Tables 1 and 2 list some parameter settings of the problem. The parameters are defined based on the works in [15,25]. Specifically, the number of targets is labelled as N_T , and each target contains three different tasks (i.e., $N_{type} = 3$). For the i th target, L_i^T denotes the location of the target. V_i^T represents the reward for completing all tasks of the target. $M_i^T = (1, 2, 3)$ is the current task type of the target and E_i^T is the end time of the previous task of the target. R_i^{threat} and β_i are threat radius and threat level, respectively. The threat level indicates probable damage caused by the target to UAV. M_k , $k = (1, 2, \dots, N_K)$ indicates the k th task. $M_{type} = (1, 2, 3)$ denotes the task type, where the numbers 1, 2 and 3 represent reconnaissance task, attack task and verification task, respectively. $T_{belong}^k = i$ denotes that task M_k belongs to target T_i .

Three types of UAVs constitute a heterogeneous UAV system for SEAD operation: reconnaissance UAV, attack UAV and utility UAV. Reconnaissance UAV owns features of small size, light weight and high airspeed, but it could not carry heavy load. Its high airspeed guarantees excellent performance in the reconnaissance and verification tasks. Attack UAV is equipped with weapon units and ammunition to perform attack task. Trading speed for endurance and ammunition storage makes it unsuitable for reconnaissance task. Expensive utility UAV carries a variety of equipments and can perform all tasks, but it has the lower airspeed than reconnaissance UAV, and smaller ammunition than attack UAV.

There are N_U UAVs in the heterogeneous UAV system. Attributes of UAV model are shown in Table 2. Specifically, L_j^U denotes the location of

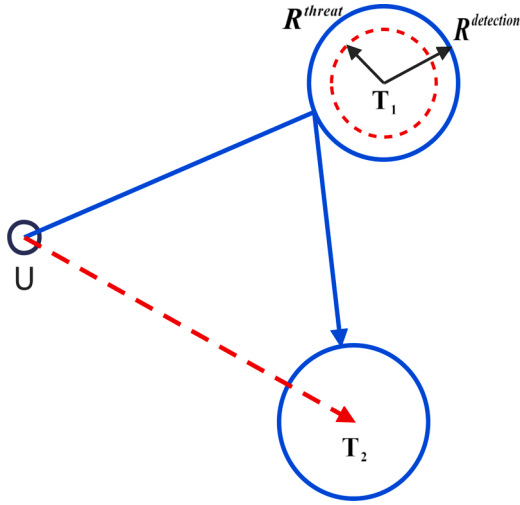


Fig. 2. Path planning of UAVs.

the j th UAV, $j = (1, 2, \dots, N_U)$. The take-off and landing locations of UAVs are set to the map origin by default. $Speed_j$ denotes the airspeed of the j th UAV. V_j^U represents the UAV cost. R_j represents the detection radius. Turning radius is not considered in the model. Load parameter A_j represents the ammunition of the UAV. By reducing A_j by 1 for each attack task, the j th UAV will be unable to perform attack task when $A_j = 0$. S_j is the range limit of UAV. UAV completes the task within the range limit and returns to the take-off base. $P_j^t \in [0, 1]$ represents the capability to perform tasks of type t , where 0 indicates that the UAV lacks the ability to perform this type of tasks. UAVs perform the same type of tasks with identical performance. t_j represents the time to complete the previous task, namely, execution time. To perform an attack task requires time Δt_{attack} .

When assigning tasks to UAVs, it is a cardinal problem to plan the route from UAV to task in advance. Since the requirements for different tasks are different, the trajectory of a UAV depends on the type of task assigned to the UAV. Reconnaissance and verification tasks require careful observation of the target to obtain enough information. Therefore, the UAV circles around the target with detection radius, when performing the above two tasks. In order to attack the target at close range, the UAV needs to reach the target position and hovers over the target for a while to complete the attack task. A UAV needs to return to the take-off base immediately once it completes its assigned tasks.

The path planning of a UAV is based on a two-dimensional plan, where the target position and UAV parameters are known. Fig. 2 shows a schematic diagram of the path planning. In the figure, U represents the take-off base; T_1 and T_2 represent two targets; $R^{detection}$ is the detection radius of UAVs; R^{threat} is the threat radius of targets; the solid lines and circles represent the flight trajectory of a UAV performing reconnaissance and verification tasks, and the dotted line represents the track of a UAV performing attack task. Because the turning radius of UAVs is ignored, the route of attack UAVs is the shortest path between the UAV and the target. The UAV should follow the shortest path to reach the circle and hang around the circle when performing reconnaissance and verification tasks.

S_j^k represents the distance of the path between the j th UAV and the k th task. The actual distance ΔS of attack task is the airline distance between UAV and target. The ΔS of other tasks are also calculated according to Fig. 2. However, S_j^k is not equal to the actual distance of the UAV to perform the task. When the previous task completion time E_i^T of target is later than the execution time t_j of the UAV, the UAV needs to hover a period of time t_h and embark on next task. Extra hover time Δt_{attack} is needed for the UAV to execute attack task. Since hovering consumes as much energy as flying, the S_j^k for the path is calculated by

flight time.

$$t_h = \begin{cases} E_i^T - t_j; & E_i^T > t_j \\ 0; & \text{else} \end{cases}$$

$$S_j^k = \begin{cases} \Delta S + t_h * Speed_j; & M_{type} = (1, 3) \\ \Delta S + (t_h + \Delta t_{attack}) * Speed_j; & M_{type} = 2 \end{cases} \quad (1)$$

2.3. Optimization objectives and constraints

Since several UAVs need to take off from the base and return to the base after completing their tasks, the task allocation problem can be modeled as a multiple travelling salesman problem (MTSP). The essence of the problem is to assign N_K tasks to N_U UAVs and specify the execution sequence. To describe the relationship between tasks and UAVs, a relation matrix R_K^U of size $N_K * N_U$ is introduced in the model. If task M_k is assigned to UAV U_j , $R_K^U(k, j) = 1$. Otherwise, $R_K^U(k, j) = 0$.

In this study, three optimization objectives are considered in the task allocation model: task benefit, UAV damage, and total range.

Task benefit: Benefit is defined as the overall benefit of performing all tasks, and used to ensure that more valuable tasks can be assigned to UAVs with high performance. In order to minimize all optimization objectives conveniently, the benefit optimization function is converted into the residual value of targets, which is calculated by Eq. (2). In general, the lower the residual value, the better the completion of the tasks:

$$F_{benefit} = \frac{\sum_{i=1}^{N_T} V_i^T \left[1 - \prod_{j=1}^{N_U} \prod_{k=1}^{N_K} (P_j^k)^{R_K^U(k, j)} \right]}{\sum_{i=1}^{N_T} V_i^T} \quad (2)$$

UAV damage: The damage objective is defined to assess the total loss of UAVs in task execution. In the SEAD operation, target is the cardinal threat to UAVs, so the UAV damage is calculated based on the UAV value V_j^U and the threat level of the target β_i . ($i = T_{belong}^k$ for task M_k belongs to i th target) UAVs can perform tasks even though the detection radius is less than the threat radius of the target. But in this case, the damage to UAV is more severe. UAVs performing attack tasks will have a close range to the target, and thus they will subject to greater threats. $\omega(k, j)$ describes whether the UAV works within the target threat range. If the UAV works within R_i^{threat} , $\omega(k, j) = 1$. Otherwise, $\omega(k, j) = 1.5$. The damage objective can be calculated by Eq. (3).

$$F_{damage} = \frac{\sum_{j=1}^{N_U} \sum_{k=1}^{N_K} \beta_i V_j^U R_K^U(k, j) \omega(k, j)}{\sum_{j=1}^{N_U} V_j^U} \quad (3)$$

Total range: The total range objective is used to assess the resource consumption of UAVs, which is related to the total flying time of UAVs. The calculation method of distance is proposed in Eq. (1). The total range also includes the return distance of all UAVs to the base, and the time required for one UAV to return to the base is represented by t_{back} . The total range function is given by Eq. (4).

$$F_{range} = \frac{\sum_{j=1}^{N_U} (t_j + t_{back}) Speed_j}{\sum_{j=1}^{N_U} S_j^U} \quad (4)$$

To achieve the above goals, the UAVs should satisfy certain physical and logical constraints. Physical constraints are related to the limited performance and resources of UAV, e.g., capacity constraint, ammunition constraint and range constraint. Logical constraints are related to task requirement, e.g., sequence constraints and allocation principle. Specifically, if the k th task is assigned to the j th UAV, the following conditions must be satisfied.

1. Capacity constraint: The selected UAV must have the capacity to perform this kind of task.

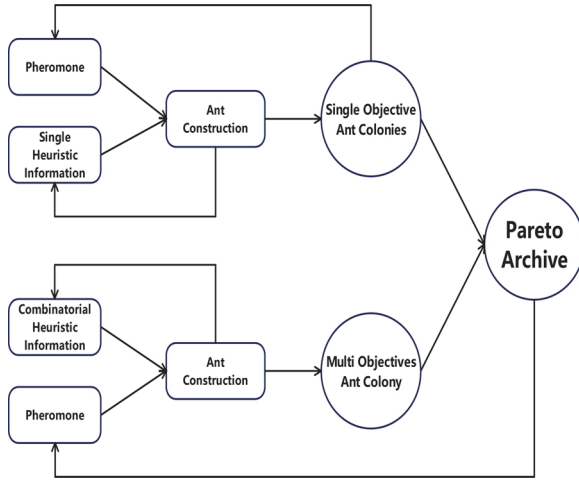


Fig. 3. The framework of the MOACO.

2. Ammunition constraint: The UAV has enough ammunition A_j left to carry out the attack task.
3. Range constraint: Due to the limited load, the total distance for UAV to perform the tasks and return to the take-off base should be less than the range limit. The return distance is calculated as $t_{back} * Speed_j$.
4. Sequence constraints: Tasks of the same target should be performed in sequence. Therefore, the task cannot be executed if the type M_{type} is inconsistent with the current task type M_i^T of the target.
5. Allocation principle: A task can only be assigned to one UAV.

Based on the above definitions, the cooperative task allocation in heterogeneous UAV system can be described by Eq. (5).

$$\min(F_{benefit}, F_{damage}, F_{range}) \quad (5)$$

s.t.

$$\begin{cases} P_j^K > 0; \\ A_j > 0; \\ M_{type} == M_i^T; \\ (t_j + t_{back}) * Speed_j \leq S_j; \end{cases} \quad \text{when } M_{type} == 2$$

3. Proposed MOACO Algorithm

In this section, the general framework of the proposed MOACO is presented, followed by the implementation details of the important components.

3.1. Algorithm framework

Fig. 3 illustrates the overall framework of the proposed MOACO algorithm. Generally, the proposed MOACO contains two kinds of ant colonies to search for the solution set and an archive to keep the non-dominated solutions found during the search procedure. The first kind of ant colonies is called single-objective ant colony, each of which focuses on optimizing one objective. For this kind of ant colony, the heuristic information is defined based on the specific objective assigned to the colony, and the pheromone is updated based on the newly constructed solutions of the corresponding colony. The second kind of ant colony is called multi-objective ant colony, which is utilized to search for trade-off solutions. For this kind of ant colony, the heuristic information is the aggregation of multiple optimization objectives, and the pheromone is updated based on the solutions in the archive.

The pseudocode of the proposed MOACO is shown in Algorithm 1, which contains the following seven steps:

- Step 1:** Set algorithm parameters and initialize the heuristic information.
- Step 2:** Construct ant colonies and corresponding pheromone matrix.
- Step 3:** Each ant constructs a new solution and the three objective values of the constructed solution are calculated.
- Step 4:** Update the pheromone matrices of the three single objective ant colonies based on the solutions newly constructed.
- Step 5:** Update the Pareto archive based on the newly constructed solutions. When the archive size exceeds the limit, remove the redundant solutions according to the crowding degree.
- Step 6:** Update the pheromone matrix of the multi-objective ant colony based on the solutions in Pareto archive.
- Step 7:** If the termination condition is not met, return to **Step 3**. Otherwise, output the non-dominated solutions as the final results.

In the following parts, the major steps related to the above mentioned procedures are described.

Algorithm 1. MOACO algorithm

Algorithm 1: MOACO algorithm

- 1 Set parameters: N (number of all ants), I (number of iteration), N_{sub} , N_P , Q , ρ , α , β ;
- 2 Input the details of UAVs, targets and tasks;
- 3 Initialize heuristic information and pheromone matrices;
- 4 Construct three single objective ant colonies and multi objectives ant colony C_{all} ;
- 5 **for** $i=1:I$ **do**
- 6 **for** $j=1:N$ **do**
- 7 Calculate the distance matrix between UAVs and tasks;
- 8 **for** $k=1:N_K$ **do**
- 9 Calculate selection probability;
- 10 Select next task and UAV by roulette;
- 11 Update UAV state and distance matrix;
- 12 Add the selected task into tabu list;
- 13 Calculate the objective optimization functions value;
- 14 Reset tabu list, UAVs and targets state;
- 15 Update Pareto archive P ;
- 16 **while** The number of the individuals in P exceeds N_P **do**
- 17 Calculate the crowding degree of each individual;
- 18 Remove the individual with the minimum crowding degree;
- 19 Update pheromone matrices of single objective ant colonies;
- 20 Update pheromone matrix of the C_{all} ant colony;
- 21 Pheromone evaporation;

3.2. Solution construction

To solve a task allocation problem, N_K tasks need to be assigned to N_U UAVs. Each ant represents a solution to the task assignment problem, which needs to select N_K possible edges between UAVs and tasks. The selection probability $P(j, k)$ of the edge connecting the j th UAV and the k th task is calculated by heuristic information $H(j, k)$ and pheromone $\tau_{j,k}$ as in Eq. (6).

$$P(j, k) = \frac{[\tau_{j,k}]^\alpha [H(j, k)]^\beta}{\sum_{j=1}^{N_U} \sum_{k=1}^{N_K} R_K^U(k, j) [\tau_{j,k}]^\alpha [H(j, k)]^\beta} \quad (6)$$

Parameters α and β indicate the importance of pheromone and heuristic weight, respectively. After each task is assigned, the status of UAV and target is updated. The change of UAV position leads to the dynamic update of S_j^k and selected probability for other tasks.

There are $N_U * N_K$ edges between UAVs and tasks in the mathematical sense. However, some edges may not be selected due to constraints. To solve this problem, the probability $P(j, k)$ of selecting a forbidden edge is set to 0. An edge is called forbidden edge if it satisfies at least one of the following six conditions.

1. The task has been completed.
2. The predecessor task is performed by the same UAV.
3. The UAV is short of ammunition to perform attack task.
4. The UAV does not have the capacity to perform this type of task.
5. The UAV cannot complete the task and return to the base within the range limit.
6. The tasks of each target need to be performed in sequence. This task cannot be executed if the predecessor task is not completed.

The schematic diagram of ant path selection process is illustrated in Fig. 4. UAV1, UAV2, UAV3 represent the reconnaissance, attack and utility UAVs, respectively. The solid line indicates that the task has been assigned to the UAV. There are five tasks assigned and it is going to assign the sixth task. The serial numbers of assigned tasks represent the execution sequence. The dotted line indicates that the edge between the task and the UAV is available for next selection. There is no line between task3 and UAV2 because the UAV2 can only perform attack tasks. UAV3 may lack ammunition so it cannot perform task8. Task9 is not connected with any UAV for the predecessor task8 is not completed. Select the next task to execute from all possible edges and update the relationship between remaining tasks and all UAVs. The above procedure is repeated

until all tasks have been assigned. If none of the remaining edges are selectable, the current solution is discarded. The ant will reconstruct a solution.

3.3. Pareto archive

The Pareto archive is introduced in the MOACO algorithm as an elite strategy for storing non-dominated solutions in each generation. In each iteration, the newly constructed solutions in all ant colonies are inserted into the Pareto archive, and the non-dominated sorting algorithm is performed to rank the solutions in the archive. To maintain the archive, dominated solutions will be removed, and all non-dominated solutions are kept in the archive.

If the number of non-dominated solutions is larger than a predefined size N_p , more solutions will be removed according to their crowding distances in the archive, so as to keep the archive size to be N_p . The crowding distance is calculated as in [26]. It should be noted that the non-dominated solutions in the archive are provided as the final outputs for decision makers to select trade-off solutions according to their preferences.

3.4. Multiple heuristics for multi-objective optimization

In the proposed algorithm, there are multiple matrices storing heuristic information which are defined based on the optimization objectives. The heuristic information of each edge is weighted and aggregated by multiple matrices. Follows are two alternative strategies for setting the weights:

1. Dynamic setting strategy [27]: In this strategy, different weights are assigned to the ants dynamically in each iteration.
2. Fixed setting strategy [28]: In this strategy, every objective has the same importance, and thus all ants are assigned with the same weight during the entire algorithm.

Generally, the dynamical setting strategy can introduce more search diversity and will slow down the convergence rate, while the fixed setting strategy will reduce the search diversity. To balance the exploring and exploiting ability of the algorithm, a new multiple ant colony algorithm with hybridized setting strategy is proposed.

For a combinatorial optimization problem with N optimization objectives, the whole ant colony is divided into $N + 1$ sub ant colonies. Each optimization objective corresponds to one sub ant colony with a

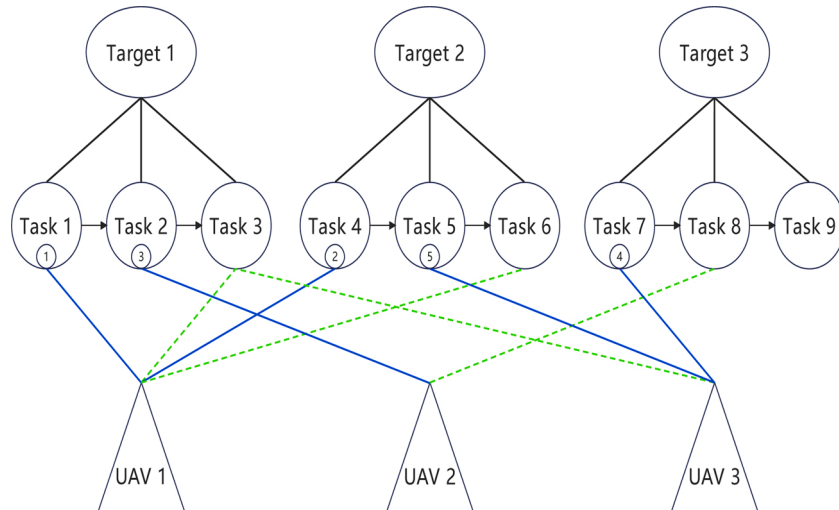


Fig. 4. Construction process of an ant solution.

size of N_{sub} . The last sub ant colony optimizes all the objectives at the same time, whose size is larger than the sum of other colonies. Since the UAV task allocation problem we study has three optimization objectives (i.e., benefit, damage and range), the corresponding MOACO algorithm contains four sub ant colonies: $C_{benefit}$, C_{damage} , C_{range} and C_{all} .

To calculate the heuristic information of each sub ant colony, we firstly define the fitness value for each optimization objective. For the edge connects j th UAV and k th task, the fitness value of benefit $B(j, k)$ is defined as the product of task capability P_j and target reward V_i^T ($i = T_{belong}^k$). The fitness value of damage $D(j, k)$ is calculated by threat level β_i and UAV cost V_j^U . The fitness value of range $S(j, k)$ is the reciprocal of the distance. The specific calculations are given by Eq. (7).

$$\begin{cases} B(j, k) = P_j * V_i^T \\ D(j, k) = 1 / (\beta_i * V_j^U) \\ S(j, k) = 1 / S_j^k \end{cases} \quad (7)$$

In the above equations, the cost and the task capacity of UAV, the threat level and the reward of target are constants. Therefore, damage and benefit matrices are stationary and shared by all sub ant colonies during the entire algorithm run. However, the distance of the edge S_j^k is dynamic. When an edge is successfully selected by the ant, the selected UAV will move to the selected location, and the distance between the UAV and remaining tasks will be updated. Hence, the initial distance matrix will change during the construction process.

The heuristic information matrices of the three single objective sub ant colonies, namely, $C_{benefit}$, C_{damage} and C_{range} , are defined as the corresponding edge fitness value matrices in Eq. (8), Eq. (9), and Eq. (10), respectively.

$$H_{benefit} = B \quad (8)$$

$$H_{damage} = D \quad (9)$$

$$H_{range} = S \quad (10)$$

Since the last sub ant colony C_{all} aims to optimize all objectives at the same time, its heuristic information matrix is the aggregation of the above three matrices, which is defined by Eq. (11).

$$H_{all} = \frac{B}{\sum(B)} + \frac{D}{\sum(D)} + \frac{S}{\sum(S)} \quad (11)$$

3.5. Pheromone update

In the proposed algorithm, each sub ant colony has its own pheromone matrix τ . As in the ant colony system (ACS) [29], there are two pheromone update operations. The first operation is global pheromone update operation, which is performed at the end of each iteration. In the global pheromone update operation, ants leave fixed amount of pheromones Q on the edges they traversed and the pheromone on each edge will partially evaporate, which can be expressed by Eq. (12).

$$\tau_{j,k}(t) = \tau_{j,k}(t-1) * (1 - \rho) + Q * n \quad (12)$$

where ρ denotes global pheromone evaporation rate, n denotes the number of ants traverse the edge between the j th UAV and k th task and t represents the current iterations. The second method is local pheromone update.

The pheromone of each single objective optimization ant colony, C_{damage} , $C_{benefit}$ and C_{range} , updates according to the solutions in each colony. As for the last sub ant colony C_{all} , the solutions in the approximation Pareto set are selected to update pheromone. Specifically, at the end of each iteration, the pheromone is updated according to the non-dominated solutions in the Pareto archive instead of the individuals in the colony. In this way, the C_{all} can maintain a relatively high search diversity, which is good for finding more trade-off solutions.

4. Simulation studies

4.1. Simulation settings

In order to verify the performance of the proposed MOACO algorithm, a test scenario based on SEAD operation is firstly described in this section. The heterogeneity of both UAVs and targets, kinematic characteristics, and the constraints are considered in the scenario setting.

The targets are deployed in a $100 \times 100 \text{ km}^2$ rectangle area, and all UAVs are also flying within the rectangle area. Six targets and ten UAVs are involved in the simulation. Table 3 shows the details of the targets.

In the simulation, ten UAVs are constructed to perform reconnaissance, attack, and verification tasks. U_j ($j = 1, 2, 3$) are reconnaissance UAVs (Type 1). U_j ($j = 4, 5, 6$) are attack UAVs (Type 2). U_j ($j = 7, 8, 9, 10$) are utility UAVs (Type 3). The parameter settings of the above three types of UAVs are given in Table 4. The performance, ammunition, and range of UAVs are limited and diverse.

Fig. 5 illustrates the initial locations of the UAVs and targets. The six targets are marked by the diamonds, and the take-off base of UAVs is marked by the triangle at origin.

In order to further verify the performance of the MOACO algorithm, this paper studies the algorithm performance under different scale problems. In addition to the above test scenario 1, we set up another five test scenarios. The number configurations of targets and UAVs are shown in Table 5. In Tests 2 and 3, the number of targets and UAVs decreased. In Test 4, we mainly change the location of targets and related parameters of UAVs. In Tests 5 and 6, we studied the change of algorithm performance when the problem scale becomes larger.

Besides the proposed algorithm, the greedy algorithm and three multi-objective optimization algorithms, standard multi-objective particle swarm optimization (SMOPSO), standard multi-objective ant colony optimization (SMOACO) and the variant (MOACO*) of the multi-objective multiple ant colony optimization, are used to solve the task allocation problem in this paper and the performance comparisons are carried out.

There is a greedy algorithm for each optimization objective, which obtains only one solution with the best fitness value. The SMOPSO algorithm is based on the multi-layer encoding strategy and the constraint

Table 3
Attributes of targets.

Target no.	Position/km	Reward	Threat level	Threat radius	Tasks
1	(30,20)	200	0.1	1.2	1–3
2	(20,80)	250	0.15	1.4	4–6
3	(40,70)	350	0.25	1.2	7–9
4	(75,50)	500	0.3	1.6	10–12
5	(85,90)	400	0.4	1.8	13–15
6	(90,25)	300	0.2	1.4	16–18

Table 4
Attributes of UAVs.

Parameter	UAV 1–3	UAV 4–6	UAV 7–10
Type	1	2	3
Position/km	(0,0)	(0,0)	(0,0)
Cost	100	120	170
Speed/(km h ⁻¹)	180	140	150
Range limit	800	1000	900
Detection radius	1.6	0	2.0
Ammunition	0	2	1
Attack time/h	0	0.1	0.1
Reconnaissance capacity	0.92	0	0.91
Attack capacity	0	0.95	0.94
Verification capacity	0.93	0	0.95

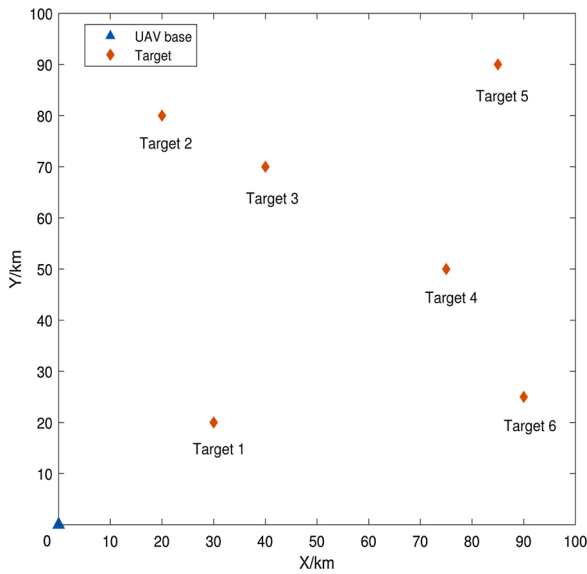


Fig. 5. Locations of UAV base and targets.

Table 5
The scale of test scenarios.

Test no.	Target	UAV
1	6	10
2	3	5
3	5	6
4	6	9
5	8	12
6	10	15

Table 6
Related parameters of algorithms.

Algorithm	Parameters
SMOPSO [15]	$\omega = 1.2, C_1 = C_2 = 2$
SMOACO [30]	$\alpha = \beta = 1, \rho = 0.2, Q = 5$
MOACO*	$\alpha = 1, \beta = 2, \rho = 0.2, Q = 10, N_{sub} = 10$
MOACO	$\alpha = 1, \beta = 2, \rho = 0.2, Q = 10, N_{sub} = 10$

Table 7
Optimal objective values for the six test scenarios.

Algorithm		Greedy	SMOPSO	SMOACO	MOACO*	MOACO
Objective	Test					
Benefit	1	0.1800	0.1817	0.1819	0.1810	0.1800
	2	0.1800	0.1800	0.1800	0.1800	0.1800
	3	0.1811	0.1812	0.1826	0.1815	0.1811
	4	0.2100	0.2117	0.2130	0.2110	0.2100
	5	0.1800	0.1832	0.1849	0.1821	0.1800
	6	0.1800	0.1867	0.1922	0.1862	0.1843
Damage	1	0.4145	0.4198	0.4245	0.4209	0.4175
	2	0.3114	0.3115	0.3115	0.3115	0.3115
	3	0.5015	0.5345	0.5687	0.5326	0.5045
	4	0.4250	0.4263	0.4293	0.4266	0.4250
	5	0.4397	0.4426	0.4496	0.4423	0.4397
	6	0.4611	0.4641	0.4687	0.4678	0.4632
Range	1	0.1776	0.1880	0.1972	0.1825	0.1735
	2	0.1870	0.1863	0.1868	0.1861	0.1852
	3	0.2412	0.2383	0.2404	0.2368	0.2290
	4	0.2340	0.2303	0.2327	0.2299	0.2196
	5	0.1700	0.1782	0.1809	0.1748	0.1646
	6	0.1597	0.1596	0.1645	0.1527	0.1513

scheduling method introduced in [15], but uses the Pareto archive in the MOACO algorithm as the solution evaluation method. The comparison with SMOPSO is adopted to prove the performance superiority of ACO algorithm in solving task allocation problem. The SMOACO algorithm utilizes the aggregation of three optimization objective values with the fixed weights as the heuristic information, and globally updates pheromone. The MOACO* algorithm is based on multiple ant colonies strategy in MOACO with normal pheromone update. Both ACO algorithms use a Pareto archive as the solution evaluation method. The comparisons with them are used to verify the validity of the multiple ant colonies strategy and the improved pheromone update method.

The detailed parameters of these algorithms are listed in Table 6. ω represents the inertia weight of SMOPSO algorithm. These algorithms have the same size of population ($N = 100$), iteration ($I = 500$) and maximum size of Pareto archive ($N_p = 80$). To improve the accuracy, all multi-objective algorithms run for 50 times independently.

Two performance evaluation indices, inversion generation distance (IGD) and hyper volume (HV) are proposed to compare the comprehensive performance of multi-objective algorithms. Reference solution set S^* to calculate IGD is constructed by dominance relation. HV is calculated by reference point $Z = (1, 1, 1)$. The solution set with the lower IGD and the larger HV value has better convergence and uniformity. The simulation environment is an Intel (R) Core(TM) i7-9750H CPU with 8 GB RAM. The algorithms are coded in MATLAB R2019a.

4.2. Algorithms performance comparison

Table 7 shows the optimal values for each objective generated by multiple algorithms, with the best comparison values marked in bold. For multi-objective algorithms, the optimal objective fitness values are the average of 50 runs.

The greedy algorithm obtains the solution with the minimum benefit and damage fitness values at the cost of other objective values and it only returns a specific solution. The MOACO algorithm obtains the solution with the smaller range fitness value than greedy algorithm, since the distance information is changing in the searching. And it obtains the minimum benefit and damage fitness value in all multi-objective algorithms. Preserving the ability to search the boundary solutions, the SMOPSO and MOACO* algorithms get the smaller single objective fitness values than SMOACO algorithm. For Test 2, the small number of UAVs and targets leads to few possible solutions. Hence, almost all algorithms can get the optimal objective fitness values.

Fig. 6 compares the IGD standard deviation, IGD mean and HV mean

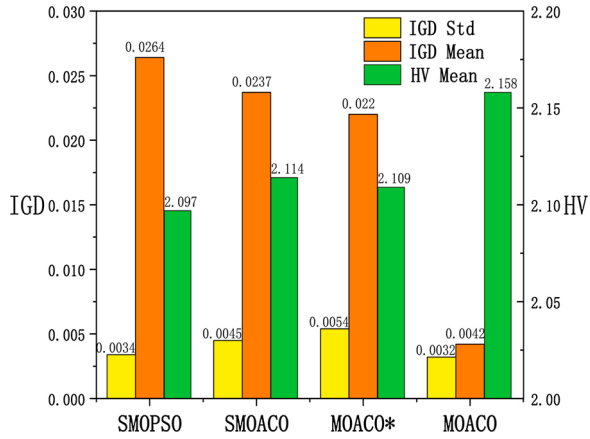


Fig. 6. IGD standard deviation, mean and HV mean.

of the final Pareto solution sets. The IGD mean and standard deviation (Std) values of the MOACO algorithm are both lowest in all algorithms, which improves its high convergence and stability. The highest hyper volume also indicates MOACO a better solution set.

Fig. 7 illustrates the IGD trend of multi-objective algorithms for six test scenarios. In all test scenarios, the MOACO algorithm converges faster than others, and the MOACO* algorithm always converges faster than SMOACO algorithm. Fig. 7(a)–(f) represents the IGD trend in Test 1

to Test 6 in turn. In Fig. 7(a), the IGD values of all algorithms decreased rapidly before generation 200, and then all tended to converge. The IGD convergence value of MOACO algorithm is obviously smaller than those of other algorithms, and its IGD value declines fastest, indicating the MOACO algorithm has the highest convergence rate. The IGD convergence values of other algorithms are similar. Compared to SMOACO, MOACO* converges faster, which proves that the effectiveness of multiple ant colonies strategy. The comparison between MOACO and MOACO* indicates the improved pheromone update method enhances the convergence rate. There is no clear relationship between the IGD convergence rates of SMOACO and SMOPSO algorithms.

In Fig. 7(a) and (b), the IGD of the algorithms almost converge together, owing to the small scale of problem and the limited solution space. With the expansion of the scenarios scale, the gap of IGD convergence value between MOACO and other algorithms is gradually increasing. Generally, these tests show that the MOACO algorithm deals with the cooperative task allocation for heterogeneous UAVs effectively.

4.3. Allocation plans analysis

Figs. 8, 9, 10 describe three representational task allocation plans of test scenario 1 generated by MOACO algorithm. In these Gantt charts, the A in A-B represents the target number, and the B is the task number.

Solution 1 has the lowest total range objective value. In this solution, the good task execution sequence reduces the extra range cost caused by the unfinished front-end tasks. But high-value integrated UAVs (7–10) perform more tasks in this plan results in greater UAV cost.

Solution 2 is outstanding in benefit objective, indicating that the

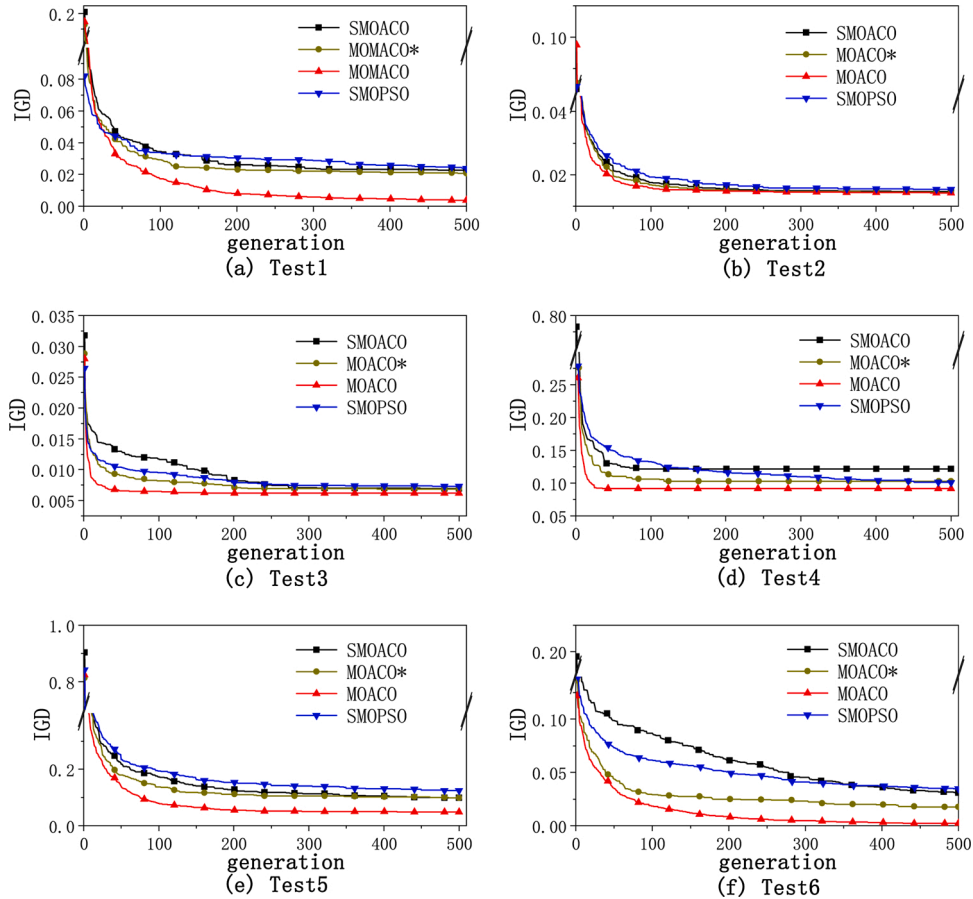


Fig. 7. IGD trends of multi-objective algorithms for six test scenarios.

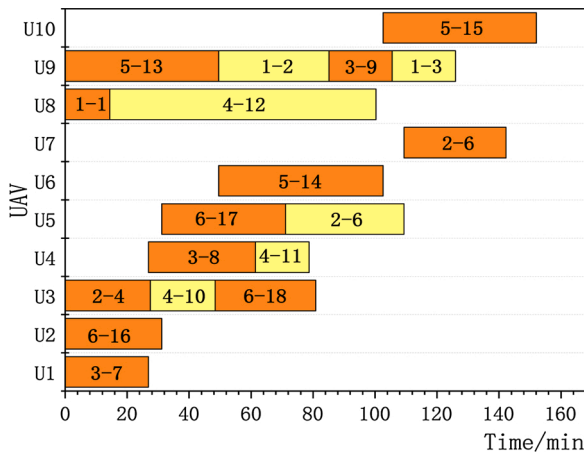


Fig. 8. Gantt chart for solution 1.

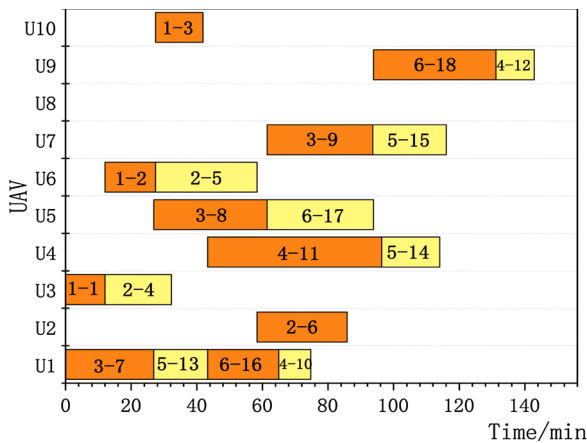


Fig. 9. Gantt chart for solution 2.

allocation plan has a high mission success rate. In this plan, all the attack tasks are performed by attack UAV (4–6). And the reconnaissance tasks are performed by the UAV (1–3) with high reconnaissance capacity. Efforts are made to let the efficient UAV to perform corresponding task, which will also lead to the increase of cost and distance.

Solution 3 performs well in cost objective. The targets with a high threat level are allocated to attack UAV and reconnaissance UAV to achieve the lower cost value. High-value UAVs have fewer dispatches.

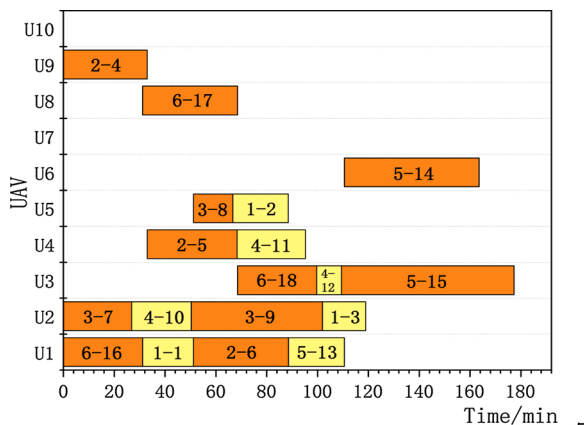


Fig. 10. Gantt chart for solution 3.

5. Conclusions

This paper formulates the cooperative task allocation problem of heterogeneous UAVs as a constrained multi-objective optimization problem which contains three optimization objectives. An efficient ant colonies optimization algorithm with multiple ant colonies (named MOACO) is proposed to solve the formulated problem. In the proposed MOACO, a new pheromone updating mechanism and four new heuristic information are specifically designed to improve the search efficiency and solution diversity. The proposed MOACO is tested on six scenarios with different scales and the simulation results have shown that the proposed algorithm performed better than several recently published algorithms, in terms of solutions quality and diversity.

Authors' contribution

Lizhi Chen: methodology, software, writing – original draft preparation. Wei-Li Liu: methodology, writing – reviewing and editing, project administration. Jinghui Zhong: conceptualization, supervision.

Conflict of interest

None declared.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62076098 and 62176067), the Guangdong Natural Science Foundation Research Team (Grant Nos. 2018B030312003 and 2020B1515120095), the innovation and entrepreneurship training program for college students of South China University of Technology (Grant No. 202110561063), and the research start-up funds of Guangdong Polytechnic Normal University (Grant No. 99166030417).

References

- [1] Y. Zhouyi, Z. Rui, C. Zongji, A mission planning algorithm for autonomous control system of unmanned air vehicle, *Instrumentation and Control Technology* (2003) 572–576.
- [2] L.J. Mangewa, P.A. Ndakidemi, L.K. Munishi, Integrating UAV technology in an ecological monitoring system for community wildlife management areas in Tanzania, *Sustainability* 11 (21) (2019) 1–17.
- [3] Z.A. Ali, Z. Han, R.J. Masood, Collective motion and self-organization of a swarm of UAVs: a cluster-based architecture, *Sensors* 21 (11) (2021).
- [4] S. Yi, Z. Long, J. Lin, Task assignment of heterogeneous UAV for anti-radar mission using CTAP models, 2019 IEEE International Conference on Mechatronics and Automation (ICMA) (2019) 1980–1985.
- [5] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles, *Phys. Rev. Lett.* 75 (1995) 1226–1229.
- [6] F. Ye, J. Chen, Y. Tian, T. Jiang, Cooperative task assignment of a heterogeneous multi-UAV system using an adaptive genetic algorithm, *Electronics* 9 (4) (2020) 687.
- [7] K.P. O'Rourke, W.B. Carlton, T.G. Bailey, R.R. Hill, Dynamic routing of unmanned aerial vehicles using reactive Tabu search, *Milit. Oper. Res.* 6 (1) (2001) 5–30.
- [8] P. Junjie, W. Dingwei, An ant colony optimization algorithm for multiple travelling salesman problem, *First International Conference on Innovative Computing, Information and Control – vol. I (ICIC'06)* (2006) 210–213.
- [9] M. Alighanbari, Y. Kuwata, J.P. How, Coordination and control of multiple UAVs with timing constraints and loitering, *Proceedings of the 2003 American Control Conference*, 2003, vol. 6 (2003) 5311–5316.
- [10] T. Vincent, F. Seipp, S. Ruzika, A. Przybylski, X. Gandibleux, Multiple objective branch and bound for mixed 0-1 linear programming: corrections and improvements for the biobjective case, *Comput. Oper. Res.* 40 (1) (2013) 498–509.
- [11] M. Alighanbari, J.P. How, Cooperative task assignment of unmanned aerial vehicles in adversarial environments, *Proceedings of the 2005, American Control Conference*, 2005, vol. 7 (2005) 4661–4666.
- [12] M. Darrah, W. Niland, B. Stolarik, L. Walp, UAV cooperative task assignments for a sea mission using genetic algorithms, *AIAA Guidance, Navigation, and Control Conference*, vol. 5 (2006) 3257–3266.

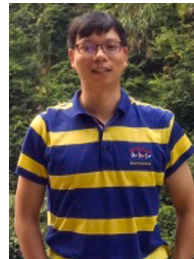
- [13] T. Shima, S.J. Rasmussen, A.G. Sparks, Uav cooperative multiple task assignments using genetic algorithms, in: 2005 American Control Conference (ACC), vol. 5, Portland, Oregon, USA, 2005, pp. 2989–2994.
- [14] Y. Gao, Y. Zhang, S. Zhu, Y. Sun, Multi-uav task allocation based on improved algorithm of multi-objective particle swarm optimization, 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (2018) 443–450.
- [15] J.F. Wang, G.W. Jia, J.C. Lin, Z.X. Hou, Cooperative task allocation for heterogeneous multi-UAV using multi-objective optimization algorithm, J. Central South Univ. 27 (2) (2020) 432–448.
- [16] N. Wang, Q. Deng, J. Yu, Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes, Chin. J. Aeronaut. 26 (5) (2013) 1238–1250.
- [17] Z. Wang, L. Liu, T. Long, Y. Wen, Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding, Chin. J. Aeronaut. 31 (2) (2018) 339–350.
- [18] M. Dorigo, G. Di Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artif. Life 5 (2) (1999) 137–172.
- [19] S. Gao, J. Wu, J. Ai, Multi-uav reconnaissance task allocation for heterogeneous targets using grouping ant colony optimization algorithm, Soft Comput. 25 (5) (2021) 7155–7167.
- [20] V.M.M.O. Ompusunggu, M.K.D. Hardhienata, K. Priandana, Application of ant colony optimization for the selection of multi-UAV coalition in agriculture, 2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA) (2020) 1–8.
- [21] W. Zhenhua, Z. Weiguo, L. Guangwen, UAVs task allocation using multiple colonies of ants, 2009 IEEE International Conference on Automation and Logistics (2009) 371–374.
- [22] L. Jin, Research on distributed task allocation algorithm for unmanned aerial vehicles based on consensus theory, 2016 Chinese Control and Decision Conference (CCDC) (2016) 4892–4897.
- [23] H.X. Chen, Y. Nan, Y. Yang, Multi-UAV reconnaissance task assignment for heterogeneous targets based on modified symbiotic organisms search algorithm, Sensors 19 (3) (2019).
- [24] H. Qingtian, An application of improved pso algorithm in cooperative search task allocation, 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA) (2021) 580–583.
- [25] D. Jing, C. Jian, S. Min, Cooperative task assignment for heterogeneous multi-uavs based on differential evolution algorithm, 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, vol. 2 (2009) 163–167.
- [26] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
- [27] Y.G. Zhong, B. Ai, A modified ant colony optimization algorithm for multi-objective assembly line balancing, Soft Comput. 21 (22) (2017) 6881–6894.
- [28] Y. Shan, Study on submarine path planning based on modified ant colony optimization algorithm, 2018 IEEE International Conference on Mechatronics and Automation (ICMA) (2018) 288–292.
- [29] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1) (1997) 53–66.
- [30] X. Li, Z. Liu, Y. Zhang, A novel improved ant colony algorithm for multi-robot task allocation, 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC) (2018) 1629–1633.



Lizhi Chen is pursuing the bachelor's degree in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include evolutionary computation and multi-objective optimization.



Wei-Li Liu received the Ph.D. degree from South China University of Technology, Guangzhou, China, in 2020. She received the B.S. and M.S. degrees from Sun Yat-Sen University, Guangzhou, China, in 2008 and 2010, respectively. Currently, she is a lecturer with the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China. Her current research interests include evolutionary computation, swarm intelligence, and their applications.



Jinghui Zhong received the Ph.D. degree from the School of Information Science and Technology, Sun Yat-Sen University, China, in 2012. He is currently an Associate Professor in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. During 2013 to 2016, he worked as a Postdoctoral Research Fellow at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include evolutionary computation, machine learning, and agent-based modeling.