

# Multi-task Allocation of Multi-UAV Coalition Based on Improved Quantum Genetic Algorithm

Pengfei Liu<sup>1</sup>, Bing Wang<sup>2</sup>, Wenjie Liu<sup>1</sup>, Lan Zhang<sup>1</sup>

1. University of Science & Technology Beijing, Beijing 100083  
E-mail: zhanglan2013@ustb.edu.cn

2. Tianjin Aerospace Zhongwei Date System Technology Co.,LTD, Tianjin 300345, China  
E-mail: wangbing0001@126.com

**Abstract:** In the multi-UAV task allocation problem, one task may need several UAVs with specific capabilities to form a coalition to execute cooperatively. In this paper, considering the UAV coalition loss cost and time cost, an improved quantum genetic algorithm is proposed to solve the task allocation problem of multi-UAV coalition. On the basis of quantum genetic algorithm, the historical optimal solution retention mechanism and estimation of distribution algorithm are introduced to speed up the convergence speed and reduce the probability falling into the local optimum of the algorithm, respectively. Then, a gene repair strategy is used to improve the efficiency of the algorithm according to the problem background. Finally, the simulation results show that the proposed algorithm has more advantages than other comparison algorithms in terms of search ability and convergence speed in solving the task allocation problem of multi-UAV coalition.

**Key Words:** Unmanned aerial vehicle, Task allocation, Combinatorial optimization, Quantum genetic algorithm

## 1 Introduction

Unmanned aerial vehicles (UAVs) have found an increasingly wide utilization in military and civilian fields, such as target reconnaissance, disaster rescue and precision agriculture, etc [1]. Due to the limitation of single UAV, it is usually difficult to complete a complex task independently. Therefore, the task allocation system must integrate UAVs with different capabilities and resources according to the characteristics of task attributes to form UAV coalitions to perform tasks cooperatively [2]. To solve this task allocation problem, various optimization algorithms have been proposed, which are mainly divided into distributed task allocation and centralized task allocation [3].

Distributed task allocation means that there is no central controller to grasp the global information. Each UAV in the system, as an independent agent with certain intelligence, shares information and negotiates among agents, and then obtains the task allocation scheme. It has the advantages of good timeliness, flexibility and robustness. Common distributed task allocation methods include contract network protocol algorithms, distributed auction algorithms and game theory methods [4, 5]. However, the distributed task allocation methods fail to consider global factors, so they can not provide an optimal allocation scheme.

Centralized task allocation has a central processing unit to integrate the global information and formulate a reasonable assignment plan. Since it can be considered from a global perspective, it has the advantages of good globalization and high solution quality [6]. With the rapid developments of metaheuristic optimization algorithms, they have been widely applied to solve centralized task allocation problems. Li et al. [7] accomplished the allocation of multi-task to multi-agent through a novel quantum particle swarm optimization algorithm (QPSO) method. It is aimed at assigning the best possible agent coalition for each task. Guo et al. [8] proposed an efficient genetic algorithm with heuristic initialization and repair strategy (GAHIR), which can handle various task allocation models in a uniform algo-

rithm framework.

Quantum genetic algorithm (QGA) [9,10] is also a metaheuristic optimization algorithm, which has the characteristics of fast convergence speed and strong global search ability with small population size. With the advantages of the QGA, it has been widely used to solve combinatorial optimization problems such as task allocation and path planning. Mousavi et al. [11] used QGA in a large scale UAV network to allocate the optimal UAV coalitions to perform the detected tasks in an unknown environment. Gao et al. [12] designed QGA to solve the robot path planning problem. The algorithm worked in the discretized environment and approximated the optimal robot planning path in a highly computationally efficient fashion. Tian et al. [13] presented an improved QGA to solve the vehicle routing problem. It combined route selection with quantum chromosome coding and quantum updating and used the high parallel computing characteristics of a quantum computer to obtain the best solution. However, QGA only uses the current generation optimal solution to guide population evolution, which makes the population diversity to shrink rapidly in the later stage of the optimization process and the algorithm easy to fall into a local optimum.

Different from the evolution of QGA, estimation of distribution algorithm (EDA) adopts statistical learning methods to construct a macroscopic probability model describing the solution distribution, and then randomly samples the probability model to generate a new population [14, 15]. It is often used in combination with other algorithms to reduce the probability of the algorithm falling into the local optimum [16,17].

The main contributions of this paper are as follows: (1) QGA is used to solve task allocation problem of multi-UAV coalition by encoding UAV and task information in quantum chromosome. (2) EDA is introduced into the QGA to reduce the probability of the QGA falling into the local optimum. Then the convergence speed of the algorithm is further accelerated by using the historical optimal solution retention

mechanism. And finally, the gene repair strategy is used to modify the solutions to improve the efficiency of the algorithm.

The rest of the paper is organized as follows: Section 2 formulates mathematical model for task allocation problem of multi-UAV coalition. Section 3 describes the proposed distributed estimation quantum genetic algorithm (DE-QGA) in detail. Section 4 presents and discusses the simulation results. Finally, Section 5 concludes the paper and notes the directions of future work.

## 2 Problem description and formulation

### 2.1 Problem description

As shown in Fig. 1, the problem in this paper is to allocate an optimal coalition of several UAVs for each task. The UAV, task and UAV coalition are described firstly.

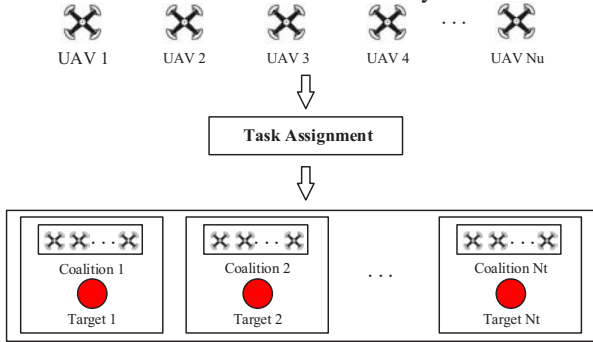


Fig. 1: Task allocation diagram

**UAV:**  $U = \{U_1, U_2, \dots, U_{N_u}\}$  is the set of  $N_u$  UAVs. The information of  $U_i$  can be represented by a vector  $A_{U_i} = \{V_{U_i}, FT_{max_{U_i}}, v_{U_i}, P_{U_i}\}$ , which is made up of the value, maximum flight time, flight speed and coordinates of  $U_i$ .  $R_{U_i} = [r_{U_i}^1, r_{U_i}^2, \dots, r_{U_i}^{N_r}]$  denotes the vector of resources carried by  $U_i$ , where  $r_{U_i}^k$  is the amount of  $k^{th}$  resource carried by  $U_i$  and  $N_r$  is the number of resource types.

**Task:** There exists  $N_t$  tasks  $T = \{T_1, T_2, \dots, T_{N_t}\}$ . The information vector of  $T_j$  is  $A_{T_j} = \{Q_{T_j}, ExcT_{T_j}, P_{T_j}\}$ , including the danger level, execution time and coordinates of  $T_j$ . The resource vector required by  $T_j$  is defined as  $R_{T_j} = [r_{T_j}^1, r_{T_j}^2, \dots, r_{T_j}^{N_r}]$ , where  $r_{T_j}^k$  represents the number of  $k^{th}$  resources required by task  $T_j$ . Finally, there are sequential logic relationships between tasks. As shown in Fig. 2,  $T_1$  is the synchronization task of  $T_2$ , defined as  $T_1 = Syn(T_2)$ , which means that the two tasks must be executed synchronously;  $T_1$  is the predecessor task of  $T_3$ , defined as  $T_1 = Pre(T_3)$ , which means that  $T_3$  must be executed after  $T_1$ .

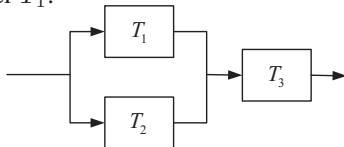


Fig. 2: The sequential logic relationship between tasks

**UAV Coalition:** The UAV coalition set is  $C = \{C_1, C_2, \dots, C_{N_t}\}$ , and each coalition corresponds to a task. The resource vector carried by  $C_j$  is defined as  $R_{C_j} = [r_{C_j}^1, r_{C_j}^2, \dots, r_{C_j}^{N_r}]$ , where  $r_{C_j}^k = \sum r_{U_i}^k, U_i \in C_j$ . The same UAV can participate in multiple coalitions at different times, but only can participate in one coalition at a time.

### 2.2 Problem Formulation

Task allocation of multi-UAV coalition can be formulated as a complex combinatorial optimization problem.

$$\min J = \alpha_1 \cdot f_1 + \alpha_2 \cdot f_2 \quad (1)$$

s.t.

$$r_{C_j}^k \geq r_{T_j}^k (\forall k = 1, 2, \dots, N_r, j = 1, 2, \dots, N_t) \quad (2)$$

$$FT_{U_i} \leq FT_{max_{U_i}} (\forall i = 1, 2, \dots, N_u) \quad (3)$$

$$C_j \cap Syn(C_j) = \emptyset (\forall j = 1, 2, \dots, N_t) \quad (4)$$

$$St(T_j) \geq Et(Pre(T_j)) (\forall j = 1, 2, \dots, N_t) \quad (5)$$

$$St(T_j) = St(Syn(T_j)) (\forall j = 1, 2, \dots, N_t) \quad (6)$$

$$St(T_j) \geq Et(T_{ji}) + \frac{|P_{T_j}, P_{T_{ji}}|}{v_{U_i}} (\forall j = 1, 2, \dots, N_t, \forall U_i \in C_j) \quad (7)$$

The objective function (1) to minimize the weighted cost includes two main components: coalition loss cost and time cost, which will be described below. Constraint (2) ensures that the resource vector component required by each task should be less than that carried by corresponding UAV coalition. Constraint (3) ensures that the total time for each UAV to perform its tasks shall not exceed its maximum flight time. Where  $FT_{U_i}$  is the total time taken by  $U_i$  to perform tasks. Constraint (4) ensures that the same UAV can only participate in one coalition at a time. Where  $Syn(C_j)$  is the UAV coalition corresponding to synchronization task  $Syn(T_j)$  of task  $T_j$ . Constraint (5) ensures that task  $T_j$  can not start to be executed until all the predecessor tasks have been executed. Where  $St(T_j)$  and  $Et(T_j)$  are the start time and end time of task  $T_j$ , respectively. Constraint (6) ensures that task  $T_j$  must start to be executed at the same time as its synchronization task. Constraint (7) means that if the UAV  $U_i \in C_j$  performing task  $T_j$  participates in task  $T_{ji}$  before, the start time of task  $T_j$  is greater than the end time of task  $T_{ji}$  plus the transfer time of UAV  $U_i$  at two tasks. Where  $|P_{T_j}, P_{T_{ji}}|$  is the distance between task  $T_j$  and  $T_{ji}$ .

The detailed description of the objective function (1) is as follows:

$$\min J = \alpha_1 \cdot f_1 + \alpha_2 \cdot f_2 \quad (8)$$

$$f_2 = \sum_{j=1}^{N_t} Q_{T_j} \cdot \sum_{U_i \in C_j} V_{U_i} / \delta_1 \quad (9)$$

$$\delta_1 = V_{U_{max}} \quad (10)$$

$$f_2 = (Et(T_{N_t}) - St(T_1)) / \delta_2 \quad (11)$$

$$\delta_2 = ExcT_{max} \quad (12)$$

Formula (9) is the coalition loss cost function which refers to the expected total cost of the members of coalitions being damaged when the coalitions are executing tasks. Formula (11) is the time cost function that can be calculated after the task allocation plan is determined. In order to unify the order of magnitude of the two parts of the objective function, the UAV value and task execution time are normalized in Formula (10) and Formula (12). Where  $V_{U_{max}}$  represents the maximum value of single UAV and  $ExcT_{max}$  represents the maximum execution time of single task.

### 3 Proposed algorithm

The flowchart of the DE-QGA is presented in Fig. 3, and the key steps of the algorithm will be described below.

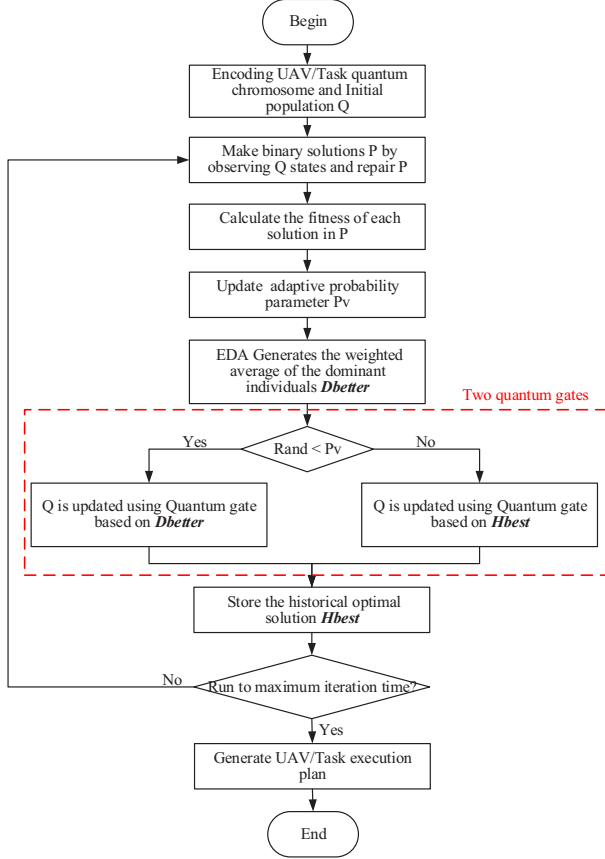


Fig. 3: The flowchart of the DE-QGA

#### 3.1 Quantum genetic algorithm

##### 3.1.1 Quantum encoding

QGA maintains a population of quantum chromosomes in the iterative process,  $Q = \{q_1, q_2, \dots, q_N\}$ , where  $N$  is population size, and quantum chromosome of the task allocation problem of multi-UAV coalition  $q_l$  is coded as:

$$q_l = \begin{bmatrix} \begin{bmatrix} \alpha_{11} \\ \beta_{11} \\ \alpha_{21} \\ \beta_{21} \\ \dots \\ \alpha_{N_u1} \\ \beta_{N_u1} \end{bmatrix} & \begin{bmatrix} \alpha_{12} \\ \beta_{12} \\ \alpha_{22} \\ \beta_{22} \\ \dots \\ \alpha_{N_u2} \\ \beta_{N_u2} \end{bmatrix} & \dots & \begin{bmatrix} \alpha_{1N_t} \\ \beta_{1N_t} \\ \alpha_{2N_t} \\ \beta_{2N_t} \\ \dots \\ \alpha_{N_uN_t} \\ \beta_{N_uN_t} \end{bmatrix} \end{bmatrix} \quad (13)$$

where  $[\alpha_{ij}, \beta_{ij}]^T$  is a qubit.  $\alpha_{ij}^2$  and  $\beta_{ij}^2$  respectively represent the probability of  $U_i$  not joining or joining  $C_j$ , and satisfy the constraint:

$$\alpha_{ij}^2 + \beta_{ij}^2 = 1 \quad (14)$$

In (15),  $p_l \in P$  is a binary solution, which can be obtained by observing  $q_l$  [11]. Where  $b_{ij} \in \{0, 1\}$ . If  $b_{ij}=1$ , then  $U_i$  will join coalition  $C_j$ , otherwise it will not join.

$$p_l = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1N_t} \\ b_{21} & b_{22} & \dots & b_{2N_t} \\ \dots & \dots & \dots & \dots \\ b_{N_u1} & b_{N_u2} & \dots & b_{N_uN_t} \end{bmatrix} \quad (15)$$

##### 3.1.2 Quantum gate

In QGA, quantum gate is used to act on qubit to change the distribution of the probability amplitude. Taking the first quadrant as an example, the quantum gate update is shown as Fig. 4.

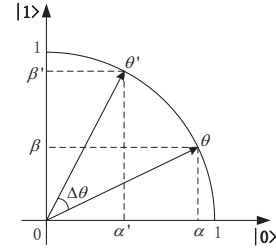


Fig. 4: Schematic diagram of quantum gate update

The update operation of quantum gate is as follows:

$$\begin{bmatrix} \alpha'_{ij} \\ \beta'_{ij} \end{bmatrix} = U(\Delta\theta_{ij}) \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix} \quad (16)$$

$$U(\Delta\theta_{ij}) = \begin{bmatrix} \cos \Delta\theta_{ij} & -\sin \Delta\theta_{ij} \\ \sin \Delta\theta_{ij} & \cos \Delta\theta_{ij} \end{bmatrix} \quad (17)$$

$$\Delta\theta_{ij} = |\Delta\theta_{ij}| s(\alpha_{ij}, \beta_{ij}) \quad (18)$$

where  $U(\Delta\theta_{ij})$  is quantum gate,  $\Delta\theta_{ij}$  is rotation angle.  $|\Delta\theta_{ij}|$  and  $s(\alpha_{ij}, \beta_{ij})$  are the size and sign of  $\Delta\theta_{ij}$ .

#### 3.2 Improvement of QGA

Constraint (19) is used to confine the qubit to the first quadrant as the basis of the subsequent improved method.

$$\begin{cases} \alpha_{ij}^2 + \beta_{ij}^2 = 1 \\ 0 \leq \alpha_{ij} \leq 1 \\ 0 \leq \beta_{ij} \leq 1 \end{cases} \quad (19)$$

##### 3.2.1 Historical optimal solution retention mechanism

In DE-QGA, the population uses the historical optimal solution  $Hbest$  to update the quantum gate instead of the current generation optimal solution, so as to improve the probability of population updating to the optimal direction and accelerate the convergence speed of the algorithm. The rotation angle based on  $Hbest$  table is shown in Table 1.

Table 1: Rotation Angle Based on  $Hbest$

$b_{ij}$	$Hbest_{ij}$	$J(Hbest) < J(p_l)$	$ \Delta\theta_{ij} $	$s(\alpha_{ij}, \beta_{ij})$
0	0	True	$0.005\pi$	-1
0	0	False	$0.005\pi$	+1
0	1	True	$0.025\pi$	+1
0	1	False	$0.025\pi$	-1
1	0	True	$0.025\pi$	-1
1	0	False	$0.025\pi$	+1
1	1	True	$0.005\pi$	+1
1	1	False	$0.005\pi$	-1

Where  $J(\cdot)$  is the objective function, and  $b_{ij}$  and  $Hbest_{ij}$  are the  $ij^{th}$  bits of  $p_l$  and  $Hbest$ , respectively.

##### 3.2.2 Estimation of distribution algorithm

The EDA is introduced to predict the evolution direction of the population by using the weighted average of the dominant individuals of the population, so that the population can

be updated according to the evolution direction of the population with a certain probability and reduce the probability of the algorithm falling into the local optimum.

As can be seen from Fig. 4, each qubit corresponds to an angle [18], so each angle chromosome  $\varphi_l$  can be defined as:

$$\varphi_l = \begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1N_t} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2N_t} \\ \cdots & \cdots & \cdots & \cdots \\ \theta_{N_u1} & \theta_{N_u1} & \cdots & \theta_{N_uN_t} \end{bmatrix} \quad (20)$$

$$\theta_{ij} = \arccos \alpha_{ij} \quad (21)$$

The weighted average of the dominant individuals of the population  $Dbetter$  is calculated as the following:

$$Dbetter = \sum_{l=1}^m \omega_l \cdot \varphi_l \quad (22)$$

$$\omega_l = (\ln(m+1) - \ln(l)) / \sum_{l=1}^m (\ln(m+1) - \ln(l)) \quad (23)$$

The half of population with lower objective function value is chosen as the dominant population. In (22),  $m = N/2$ , and  $\varphi_1, \varphi_2, \dots, \varphi_m$  are the  $m$  dominant individuals with the objective function values of the corresponding binary solutions ranked from low to high. Formula (23) means that the higher the ranking, the greater the weight coefficient. The rotation angle based on  $Dbetter$  table is shown in Table 2.

Table 2: Rotation Angle Based on  $Dbetter$

	$ \Delta\theta_{ij} $	$s(\alpha_{ij}, \beta_{ij})$
$\theta_{ij} > Dbetter_{ij}$	$0.025\pi$	-1
$\theta_{ij} < Dbetter_{ij}$	$0.025\pi$	+1
$\theta_{ij} = Dbetter_{ij}$	$0.005\pi$	$\pm 1$

Where  $\theta_{ij}$  and  $Dbetter_{ij}$  are the  $ij^{th}$  bits of  $\varphi_l$  and  $Dbetter$ , respectively.

### 3.2.3 Adaptive probability parameter

In Fig. 3,  $Pv = \{pv_1, pv_2, \dots, pv_N\}$  is adaptive probability parameter, which is adopted to choose the quantum gate update strategies. After each iteration,  $Pv$  is adaptively updated based on evolution information. If quantum gate based on  $Hbest$  is selected to generate the offspring and the objective function value is reduced, or quantum gate based on  $Dbetter$  is selected to generate the offspring and the objective function value is increased,  $pv_l$  is updated as follows:

$$pv'_l = pv_l - 0.1 \cdot pv_l \cdot (1 - t/t_{\max}) \quad (24)$$

If not,  $pv_l$  is updated as follows:

$$pv'_l = pv_l + 0.1 \cdot (1 - pv_l) \cdot t/t_{\max} \quad (25)$$

where  $t$  is current iteration time,  $t_{\max}$  is maximum iteration time.

### 3.2.4 Gene repair strategy

In order to make  $p_l$  as feasible as possible,  $p_l$  is repaired according to the constraint (4). Firstly,  $p_l$  is checked row by row. If the same UAV participates in  $n$  synchronous tasks at

a time, the  $n - 1$  genes with a value of 1 in the synchronous task genes are randomly selected to become 0.

Fig. 5 depicts an example of gene repair process. The two yellow genes represent two tasks in synchronous relationship. And they are assigned to the same UAV, so the UAV abandons one of the synchronization tasks at random.

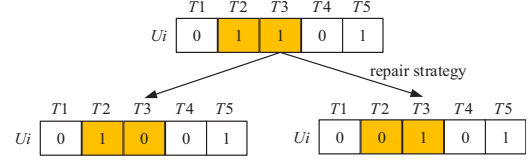


Fig. 5: Gene repair strategy

## 4 Simulation results and analysis

To evaluate the performance of DE-QGA algorithm, three scenarios with different number of UAVs and tasks were simulated, namely 4 tasks allocated to 8 UAVs, 8 tasks allocated to 16 UAVs, and 16 tasks allocated to 32 UAVs. It is assumed that the task sequence has been planned in advance.

### 4.1 Scenario I

In an area of 50km \* 50km, there are 8 UAVs which need to perform 4 tasks. The task sequence is shown as Fig. 6. And scenario I provides detailed tasks and UAVs information, which are shown in Table 3 and Table 4, respectively.

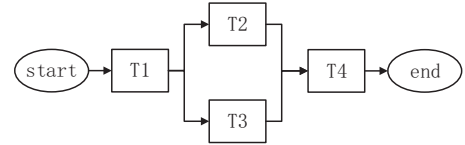


Fig. 6: Scenario I task sequence diagram

Table 3: Tasks Information Table

Task	$P_{T_j}$	$ExcT_{T_j}$	$Q_{T_j}$	$R_{T_j}$
$T_1$	(22,27)km	0.20h	0.3	[5,0,2,0,4]
$T_2$	(32,25)km	0.25h	0.2	[0,5,0,6,2]
$T_3$	(30,37)km	0.25h	0.2	[6,0,6,0,2]
$T_4$	(27,42)km	0.20h	0.3	[0,5,0,4,2]

Table 4: UAVs Information Table

UAV	$P_{U_i}$	$FTmax_{U_i}$	$v_{U_i}$	$V_{U_i}$	$R_{U_i}$
$U_1$	(5,5)km	2.0h	200km/h	0.20	[2,3,0,3,0]
$U_2$	(5,5)km	2.5h	250km/h	0.30	[3,0,3,0,2]
$U_3$	(5,5)km	2.5h	250km/h	0.30	[3,0,3,2,0]
$U_4$	(5,5)km	2.0h	250km/h	0.25	[0,2,0,3,2]
$U_5$	(5,5)km	2.0h	250km/h	0.25	[0,3,2,2,0]
$U_6$	(5,5)km	2.5h	250km/h	0.30	[2,0,2,0,4]
$U_7$	(5,5)km	2.0h	200km/h	0.20	[1,2,0,3,2]
$U_8$	(5,5)km	2.0h	250km/h	0.25	[0,0,3,2,3]

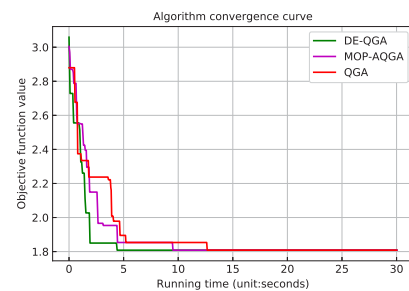


Fig. 7: Algorithm convergence comparison graph



Fig. 7 displays the convergence curves of the objective function values of the three algorithms with the same iteration time, which are original QGA [9], Multiple Operators Based Adaptive Parallel Quantum Genetic Algorithm (MOP-AQGA) [19] and proposed DE-QGA, respectively. All the three algorithms converge to the same objective function value of 1.808, but the convergence time of the three algorithms are 12.66s, 9.52s and 4.41s, which demonstrates that the proposed algorithm has a faster convergence speed when dealing with scenario I.

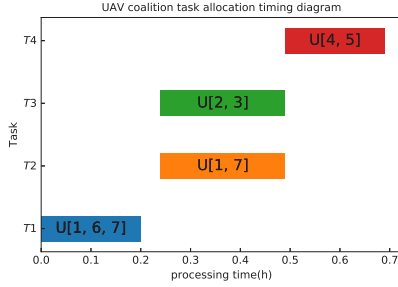


Fig. 8: UAV task and allocation timing diagram

Fig. 8 shows the task assignment results and execution time of multi-UAV coalition obtained by the DE-QGA algorithm in Scenario I. Different colors represent different tasks, and the numbers in the figure represent the serial number of UAVs. As seen in the figure, each task is completed by a coalition of several UAVs, and the resources of the coalition are greater than or equal to the resources required by the corresponding task to ensure that each task is successfully executed. The result indicates the feasibility of the task allocation scheme obtained by the proposed algorithm.

#### 4.2 Scenario II and Scenario III

In order to further verify the impact of problem scale expansion on the proposed algorithm, scenario II and scenario III are set for simulation experiments, which have higher dimension and wider search range.

Scenario II: In an area of 100km \* 100km, there are 16 UAVs which need to perform 8 tasks. The task sequence is shown as Fig. 9(a).

Scenario III: In an area of 200km \* 200km, there are 32 UAVs which need to perform 16 tasks. The task sequence is shown as Fig. 9(b).

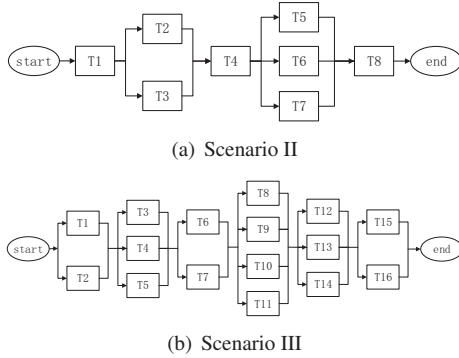
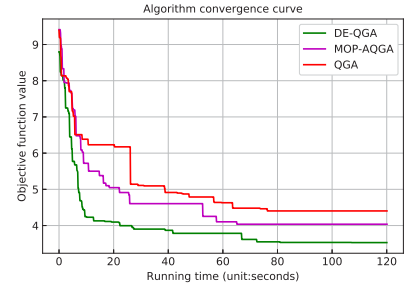
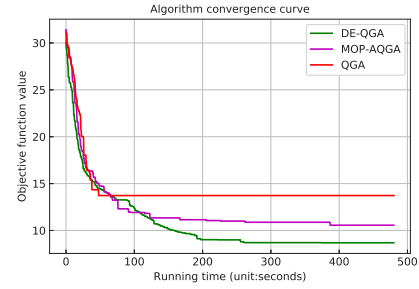


Fig. 9: Task sequence diagram

Fig. 10 displays the objective function value convergence curves of the three algorithms in two scenarios. As the problem scale expands, the three algorithms no longer converge to the same objective function value, and the convergence time also increases. However, the DE-QGA obtains a



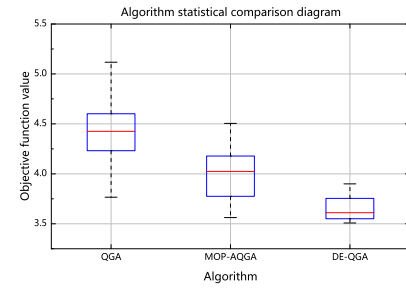
(a) Scenario II



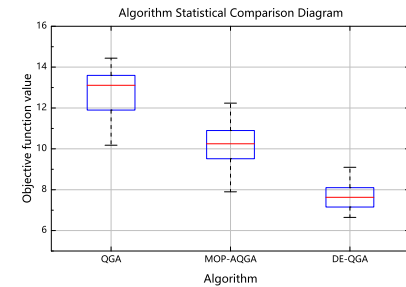
(b) Scenario III

Fig. 10: Algorithm convergence comparison graph

lower objective function value than the other two algorithms in both scenarios, and the convergence speed is also faster, which is consistent with the theoretical analysis.



(a) Scenario II

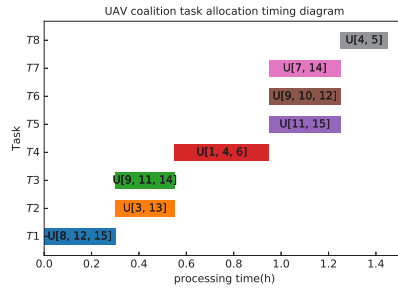


(b) Scenario III

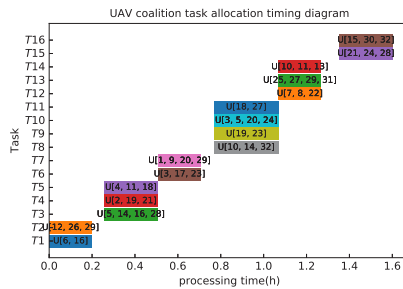
Fig. 11: Algorithm statistical comparison diagram

In scenario II, the three algorithms are run independently for 50 times with each iteration lasting for 2 minutes, and the statistical data box diagram obtained is shown in Fig. 11(a). Similarly, in scenario III, the three algorithms are run independently for 30 times with each iteration lasting for 8 minutes, and the statistical data box diagram obtained is shown in Fig. 11(b). It can be observed that not only DE-QGA box body is evidently lower than QGA and MOP-AQGA algorithms, but also the results of DE-QGA are more concen-

trated, which prove that DE-QGA has better solving ability and stability no matter which scenario to deal with. Moreover, the larger the problem dimension is, the more obvious the advantage of DE-QGA has.



(a) Scenario II



(b) Scenario III

Fig. 12: UAV task and allocation timing diagram

Fig. 12 shows the allocation result of DE-QGA algorithm in two scenarios. Each task is assigned a corresponding UAV coalition, which proves that DE-QGA algorithm can complete task allocation in complex scenarios. And there is no intersection between the coalitions executing synchronous tasks, which indicates that the proposed algorithm obtains the global optimal solution under the constraint conditions. In addition, there are intersections between the coalitions executing sequential tasks, because after the coalition completes the current task, if the UAVs in the coalition have any remaining resources, they can continue to join the subsequent coalition.

## 5 Conclusion

In this paper, the multi-task allocation of multi-UAV coalition is solved based on the DE-QGA. In the solution process, historical optimal solution retention mechanism, EDA and gene repair strategy are introduced to improve the performance of the algorithm. The simulation results show that the proposed algorithm has good convergence and optimization under different scale scenarios.

However, as the proposed algorithm is time-consuming, it is not suitable for handling dynamic situations during task execution. In the future, we will introduce the distributed control methods to study the dynamic adjustment strategy of task assignment scheme in response to uncertain emergencies.

## References

[1] G. Tian, L. Zhang, X. Bai, et al. Real-time Dynamic Track Planning of Multi-UAV Formation Based on Improved Artificial

Bee Colony Algorithm, in *2018 37th Chinese Control Conference(CCC)*, 2018: 10055-10060.

[2] H. Luan, Y. Xu, D. Liu, et al. Energy Efficient Task Cooperation for Multi-UAV Networks: A Coalition Formation Game Approach, *IEEE Access*, 8: 149372-149384, 2020.

[3] G. Oh, Y. Kim, J. Ahn, et al. Market-Based Task Assignment for Cooperative Timing Missions in Dynamic Environments, *Journal of Intelligent and Robotic Systems*, 87(1): 97-123, 2017.

[4] B. Xie, S. Chen, J. Chen, et al. A mutual-selecting market-based mechanism for dynamic coalition formation, *International Journal of Advanced Robotic Systems*, 15(1), 2018.

[5] X. Fu, J. Zhang, L. Zhang, et al. Coalition formation among unmanned aerial vehicles for uncertain task allocation, *Wireless Networks*, 25(1): 367-377, 2017.

[6] Z. Wang, B. Wang, Y. Wei, et al. Cooperative Multi-task Assignment of Multiple UAVs with Improved Genetic Algorithm Based on Beetle Antennae Search, in *2020 39th Chinese Control Conference(CCC)*, 2020: 1605-1610.

[7] M. Li, C. Liu, K. Li, et al. Multi-task allocation with an optimized quantum particle swarm method, *Applied Soft Computing*, 96, 2020.

[8] M. Guo, B. Xin, J. Chen, et al. Multi-agent coalition formation by an efficient genetic algorithm with heuristic initialization and repair strategy, *Swarm and Evolutionary Computation*, 55, 2020.

[9] KH. Han, JH. Kim, Genetic quantum algorithm and its application to combinatorial optimization problem, in *2000 Congress on Evolutionary Computation(CEC)*, 2000: 1354-1360.

[10] Y. Huang, X. Chen, K. Yuan, et al. Directional Modulation Based on a Quantum Genetic Algorithm for a Multiple-Reflection Model, *Computers, Materials & Continua*, 64(3): 1771-1783, 2020.

[11] S. Mousavi, F. Afghah, JD. Ashdown, et al. Use of a quantum genetic algorithm for coalition formation in large-scale UAV networks, *Ad Hoc Networks*, 87: 26-36, 2019.

[12] L. Gao, R. Liu, F. Wang, et al. An Advanced Quantum Optimization Algorithm for Robot Path Planning, *Journal of Circuits, Systems and Computers*, 29(08), 2019.

[13] Y. Tian, W. Hu, B. Du, et al. IQGA: A route selection method based on quantum genetic algorithm- toward urban traffic management under big data environment, *World Wide Web*, 22(5): 2129-2151, 2018.

[14] B. Doerr, W. Zheng, Sharp Bounds for Genetic Drift in Estimation of Distribution Algorithm, *IEEE Transactions on Evolutionary Computation*, 24(6): 1140-1149, 2020.

[15] B. Doerr, MS. Krejca, Significance-Based Estimation-of-Distribution Algorithms, *IEEE Transactions on Evolutionary Computation*, 24(6): 1025-1034, 2020.

[16] Y. Li, T. Han, H. Zhao, et al. An Adaptive Whale Optimization Algorithm Using Gaussian Distribution Strategies and Its Application in Heterogeneous UAVs Task Allocation, *IEEE Access*, 7: 110138-110158, 2019.

[17] X. Wang, H. Zhao, T. Han, et al. A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem, *Applied Soft Computing*, 78: 240-260, 2019.

[18] J. Yu, Y. Shi, D. Tang, et al. Optimizing sequential diagnostic strategy for large-scale engineering systems using a quantum-inspired genetic algorithm: A comparative study, *Applied Soft Computing*, 85, 2019.

[19] Z. Qu, Y. Chen, P. Li, et al. Cooperative Evolution of Multiple Operators Based Adaptive Parallel Quantum Genetic Algorithm, *ACTA ELECTRONICA SINICA*, 47(02): 266-273, 2019.