Proceedings of Machine Learning Research vol 245:1–10, 2024 2024 International Conference on Machine Learning and Intelligent Computing

Research on Multi-UAV Task Allocation Algorithm Considering Dynamic Priority Changes

Yulai He
School of Information Science and Electrical Engineering (School of Artificial Intelligence), ShanDong Jiao-Tong University, Jinan 250357, China

Hua Wu[*] WUHUA@SDJTU.EDU.CN
School of Information Science and Electrical Engineering (School of Artificial Intelligence), ShanDong Jiao-Tong University, Jinan 250357, China

Weizheng Zhang
School of Information Science and Electrical Engineering (School of Artificial Intelligence), ShanDong Jiao-Tong University, Jinan 250357, China

Editors: Nianyin Zeng and Ram Bilas Pachori

Abstract

In recent years, unmanned aerial vehicles (UAVs) have found increased application across various domains. Clusters of Multi-UAV can accomplish more complex tasks, optimizing overall efficiency through rational task allocation. In practical scenarios, they exhibit distinct advantages and characteristics. However, the allocation of tasks to Multi-UAV in special environments or emergency conditions poses a widely studied problem. Existing research or methods often impose fixed task priorities (task sequences) during Multi-UAV task allocation. Yet, real-world UAV operations may witness fluctuations in task priorities due to environmental changes or human factors. For instance, areas experiencing a drop in temperature may heighten the urgency of certain supplies, or sudden outbreaks of disease may increase the demand for medical supplies. In such cases, convention-al planning methods become inadequate. Hence, this paper addresses these scenarios by proposing a model for dynamic task priority changes in Multi-UAV task allocation within special environments. This thesis introduces an improved genetic algorithm, termed the improved partheno genetic-greedy combination algorithm. Through comparative experiments, the efficacy of the proposed algorithm in addressing dynamic priority changes in Multi-UAV collaborative task allocation problems is validated, enhancing problem-solving efficiency.

Keywords: Multi-UAV; Task Allocation, Task Priority; Genetic Algorithm

1. Introduction

In recent years, there is a significant surge in the utilization of Unmanned Aerial Vehicles (UAVs) across various domains, including domestic and military spheres. Such domains necessitate the implementation of task allocation methodologies within multi-UAV systems. (Skaltsis et al., 2023) Considering task priority is an essential aspect of ensuring effective task allocation in Multi-UAV systems. Luo et al. (2021) prioritized objectives based on their value and proximity or minimum execution time when planning tasks for Multi-UAV systems. Zhang et al. (2020) assign higher priority to targets of greater importance in their research on the allocation of objectives for coordinated UAV attacks against enemies. Chen et al. (2018) focus on giving priority to tasks closer to each UAV, proposing a cluster first strategy combined with a CBBA based algorithm for task allocation. Many studies typically fix task priorities (task sequences) without considering changes in task priorities.

This thesis establishes a model for Multi-UAV task allocation with dynamic task priority changes. This model uses the safety stock model as a reference to generate a demand state change function to generate differ-ent task priorities (A, B, C), thereby generating dynamic priorities for different receiving points. On the basis of considering the priority of the target task, maximizing the total reward value is the main goal of the task. This article improves the genetic algorithm, proposes an improved partheno genetic-greedy combination algorithm suitable for solving this problem, designs an encoding method that conforms to the problem model, adds corresponding constraints for drones, and designs adaptive mutation operators and adaptive crossover operators.

2. Problem formulation

2.1. Model Description

Given N UAVs: $U_N = \{u_1, u_2, \ldots, u_\mathrm{n}\}$ at a distribution center, all of which possess identical performance capabilities. These UAVs are tasked with delivering supplies to various receiving points in need of assistance, accomplishing this by shuttling between the receiving points and the distribution center. The primary objective of Multi-UAVs is to efficiently execute distribution tasks to all M receiving points within a defined number of rounds while satisfying various constraints. The main focus is to determine the optimal task allocation plan for multiple UAVs. As shown in Figure 1, the distribution task area is divided into z sub-regions: $G = \{G_1, G_2, \ldots, Gz\}$, where several receiving points exist in each sub-region.

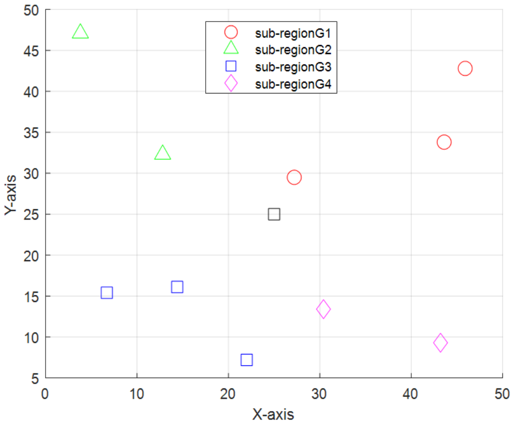


Figure 1: The receiving points in different sub-regions are indicated by symbols of different shapes, and black square indicates the delivery center.

During the execution of delivery tasks by UAVs, the demand and urgency for sup-plies in different sub-regions may fluctuate. In such cases, periodic functions are employed to represent the demand status of receiving points in sub-region Gz. To avoid overly complex function designs, this paper adopts simple yet representative functions to model the functions within the cycle. Drawing on recent research, the safety stock model, which has been widely used, is referenced (Gonçalves et al., 2020). The function describing the variation of demand status in sub-region Gz over time is derived from this model, with certain simplifications and modifications made based on the original literature.

$$\begin{cases} W_{G_Z}(t) = \mathrm{c}\sqrt{\xi_z t^2} \quad t \in [0, T] \\ W_{G_Z}(t+T) = W_{G_Z}(t) \end{cases} \tag{1}$$

The $W_{G_Z}(t)$ represents the demand status value of sub-region Gz at time tt, ξ_z denoting the rate of change of demand status in sub-region Gz, c denotes the level of material consumption coefficient, and the function operates within a period of T.

To incorporate a metric reflecting the urgency of task demands, a set of dynamic priority levels $P = \{A, B, C\}$ is established. Here, A, B, and C respectively denote different levels of task priority, where A signifies critical scarcity, B denotes moderate scarcity, and C indicates non-urgent status. When the periodic function values of the demand status in sub-region Gz increase and reach the boundary value ($W_{G_Z}(t_n) = W_0$)during each cycle, all receiving points in sub-region Gz transition to priority level A. For each receiving point in every sub-region, the default initial state is assumed to be B. Upon receiving supplies from a UAV, the state transitions to C. In the absence of supplies, at time t1, sub-region GZ transitions from state B to state A. After spending a certain duration in state C, sub-region GZ transitions back to state A at the next scheduled state transition moment. Upon receiving supplies from a UAV, state A transitions back to state C. This cycle between states A and C repeats thereafter.

2.2. Objective Function

In the process of constructing the objective function for the Multi-UAV task allocation, the parameters related to the UAVs are set as shown in Table 1.

Table 1: Parameter settings.

Parameter	Symbol
Each receiving point receives a single fixed load item	L_0
Unloading time of each UAV at the receiving point	T_{unload}
The time when the UAV returns to the delivery center for the loading and refueling process	T_{load}
The average speed of each UAV in its operational state	V_{AVG}
The distance between the target receiving point $\mathrm{M}_{\mathrm{Z1ml}}$ and the target task point $\mathrm{M}_{\mathrm{Z2m2}}$	D_{z2m2}^{z1m1}
Delivery center position	P
The distance between the receiving point M_{Zm} and the delivery center	D_{zm}^{P}
(0-1) Variable, if the UAV u_n moves from the target receiving point $\mathrm{M}_{\mathrm{Z1ml}}$ to the target mission point $\mathrm{M}_{\mathrm{Z2m2}}$	$x_{z2m2}^{z1m1} = 1$, other-wise$x_{z2m2}^{z1m1} = 0$
(0-1) Variable, if the UAV u_n moves from the delivery center to the receiving point M_{Zm}	$x_{zm}^{p} = 1$, otherwise $x_{zm}^{p} = 0$
Cost of transportation in unit distance	d

The reward value of the UAV at the receiving point M_{zm} in the subarea G_z is $P_{Zm}(t_i)$, which is obtained by equation (2).

$$P_{Zm}(t_i) = \begin{cases} P_1 \times \left[1 - \frac{2(t_i - t_{\text{Astart}})}{T}\right] & (t_i \in T_A) \\ P_2 \times \left[1 - \frac{2(t_i - t_{\text{Bstart}})}{T}\right] & (t_i \in T_B) \end{cases} \tag{2}$$

In equation (2), $P_{Zm}(t_i)$ represents the moment when UAV arrives at receiving point M_{Zm}, where $t_i \in (0, T_{\max}]$, and $T_{\max}$ denotes the latest arrival time. P_1 and P_2 represent the initial reward values for delivery tasks in critical scarcity (state A) and moderate scarcity (state B) respectively. The initial reward value for critical scarcity state A should exceed that for moderate scarcity state B, thus requiring $P_1 > P_2$. T_A and T_B denote the periods of critical scarcity and moderate scarcity respectively. t_{Astart} and t_{Bstart} represent the moment before when the most recent transition to state A or state B occurred.The objective function is defined as the final profit value obtained by subtracting the costs of all UAVs from the total reward values obtained by all UAVs within the time constraint $T_{\max}$. The objective function is represented as equation (3).

$$MaxP = [\sum_{1}^{i_{max}} \sum_{1}^{n} P_{Zm}(t_i)] - F_{cost} - T_{cost} \tag{3}$$

The $i_{\max}$ represents the most number of times a single UAV performs the same receiving point mission. The fixed cost is denoted by F_{cost}, which encompasses expenses such as UAV maintenance, repair costs, and administrative insurance fees. The transportation cost is represented as T_{cost}, calculated by multiplying the transportation cost per unit distance by the total transportation distance.

3. Task Allocation Algorithms

3.1. Greedy Algorithm

The Greedy Algorithm is a heuristic method that makes locally optimal choices at each step, aiming to achieve an overall optimal solution in terms of a certain metric (Kim and Feron, 2017). Although the greedy approach is a general strategy, its specific application to new problems requires refinement based on specific conditions. In this thesis, a Greedy Algorithm designed for solving the dynamic priority change Multi-UAV task allocation problem is outlined as follows:

Step 1: Initialize UAV and receiving point positions, and set up the scenario.

Step 2: Define the total number of rounds of tasks as k.

Step 3: Iterate over all unassigned receiving points. Calculate the expected reward value of assigning the current task to the UAV that is expected to arrive at this receiving point next, considering the current positions of UAVs and the time required to reach this point after completing previously assigned tasks.

Step 4: Sort all reward values, identify the maximum reward value, and record the combination of UAV and task with the highest reward value.

Step 5: Update UAV positions and assigned tasks. If all tasks are assigned and executed, proceed to Step 6; otherwise, return to Step 3.

Step 6: Return all UAVs to the distribution center and calculate the total reward value for this round.

Step 7: Accumulate the reward values obtained in each round. Determine whether the total number of rounds has been reached. If k rounds have been completed, output the total reward value; otherwise, return to Step 3.

The time complexity of the Greedy Algorithm applied to this problem as equation (4), m_{un} is the number of unassigned receiving points.

$$O(T) = k \bullet n \bullet O((m - m_{un}) + (m - m_{un} + 1) + ... + (m - 1)) \tag{4}$$

3.2. Partheno Genetic Algorithm

The Partheno Genetic Algorithm(PGA) is a specialized genetic algorithm (Yang et al., 2015). This algorithm deviates from traditional genetic algorithms in solving combinatorial optimization problems by abandoning the specialized crossover operators found in traditional genetic algorithms.

Instead, PGA achieves genetic recombination solely by performing operations such as gene inversion and gene shifting on a single chromosome. The core of the PGA algorithm lies in single-parent mutation, where individuals undergo mutations based on their parents. The mutation operations combined with the concept of random insertion to form the following four single-parent mutation operations:

(1) Gene flip-insert operation: Two random exchange points are generated, and the genes between these two points are exchanged. Then, the segment between these two exchange points is cut as a whole and inserted between two adjacent genes randomly outside the segment.

(2) Gene swap-insert operation: Two random inversion points are generated, and the genes between these two points are rearranged in reverse order. Then, the new segment is cut and inserted between two adjacent genes randomly outside the segment.

(3) Gene left-shift operation: Two random left shift points are generated, and each gene within these two left shift points is moved one point to the left. Then, the new gene segment is cut and inserted between two adjacent genes randomly outside the segment.

(4) Gene right-shift operation: Two random right shift points are generated, and each gene within these two right shift points is moved one point to the right. Then, the new gene segment is cut and inserted between two adjacent genes randomly outside the segment.

3.3. Improved Partheno Genetic-Greedy Combination Algorithm

For the task allocation of Multi-UAV, it is necessary to cut the chromosomes of the task sequence into several segments to represent the route scheme for each UAV to access the receiving point. IPGA-G algorithm uses two-segment chromosome coding, which consists of two parts. (Zhou et al., 2018) As shown in Figure 2, the first part is the random arrangement of n material receiving points, and the second part represents the coding segment of discontinuous points, and the first part is divided into m segments. The starting point delivery center of the UAV is 0, the first UAV‘s current mission access receiving point route is (0-4-6-0). The first part has n!kinds of arrangement, the second part has C_{n-1}^{m-1} kinds of arrangement, and the solution space of the two-segment chromosome is $n! * C_{n-1}^{m-1}$.

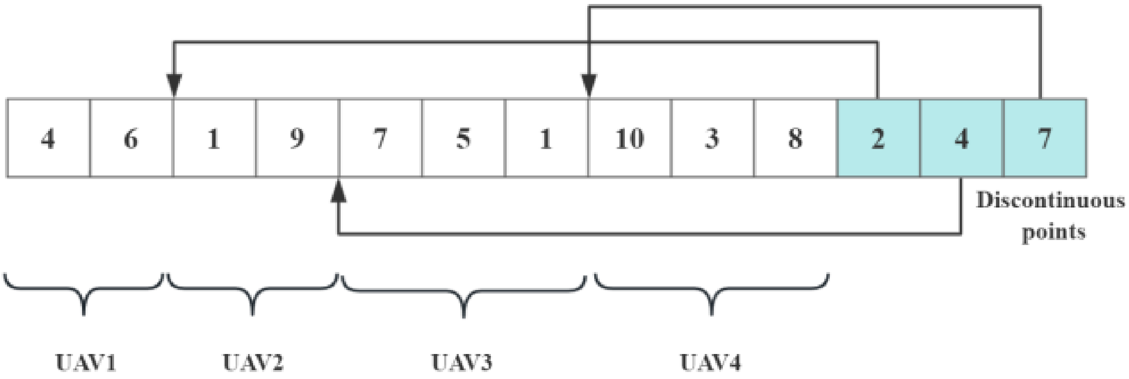


Figure 2: Example of a two-segment chromosome encoding.

In order to make the offspring population more diversified and expand the optimal range of the offspring population, the IPGA-G algorithm modifies the variation mode of the IPGA algorithm, and expands the genetic recombination to ten genetic recombination methods:

(1) Do not do anything to the first individual.

(2) The second individual performs the gene flip-insert operation.

(3) The third individual performs the gene swap-insert operation.

(4) The fourth individual performs the gene left-shift operation.

(5) The fifth individual performs the gene right-shift operation.

(6) The sixth individual performs the modification of discontinuous points.

(7) The seventh individual performs a gene flip-insert operation and a modification of discontinuous points.

(8) The eighth individual performs a gene swap-insert operation and a modification of discontinuous points.

(9) The ninth individual performs a gene left-shift operation and a modification of discontinuous points.

(10) The tenth individual performs the right-shift operation and the modification of discontinuous points.

Based on the algorithm proposed by Srinivas and Patnaik (1994), the IPGA-G algorithm has designed an adaptive crossover operator that can improve computational speed. Set different crossover probabilities based on different fitness values, and use the adaptive crossover operator P_c to represent the probability of executing operation (2-5) during gene recombination. The calculation of the adaptive crossover operator in the IPGA-G algorithm is shown in equation (5).

$$P_c = \begin{cases} k_1 - \frac{(k_1-k_2)\times|F-F_{\text{avg}}|}{F_{\max}-F_{\text{avg}}} F \geq F_{\text{avg}} \\ k_1 F < F_{\text{avg}} \end{cases} \tag{5}$$

The $F_{\max}$ is the current maximum fitness value; F_{avg} is the average fitness value of this generation, and k_1 and k_2 are the crossover operator coefficients, satisfying the condition that $k_1 > k_2$. When performing mutation operations, the mutation operator is used to represent the probability of performing modification discontinuous points operation during gene recombination. k_3 and k_4 are the mutation operator coefficients and $k_3 > k_4$. The adaptive mutation operator calculation is shown in equation (6).

$$P_m = \begin{cases} k_3 - \frac{(k_3-k_4)\times|F-F_{\text{avg}}|}{F_{\max}-F_{\text{avg}}} F \geq F_{\text{avg}} \\ k_3 F < F_{\text{avg}} \end{cases} \tag{6}$$

The IPGA-G algorithm can improve the formula for calculating genetic operators and the search efficiency to some extent by incorporating feasible solutions obtained from greedy algorithms into the initial population. By genetic recombination of feasible solutions obtained through greedy algorithms, the mutated solution is highly likely to have higher fitness values than randomly generated chromosomes, and is more likely to be selected to continue with the next round of reproduction.

There are five main steps in the IPGA-G algorithm process:

(1)Initialize the population and add the solution obtained by the greedy algorithm to the population.

(2)Fitness value calculation.

(3)Perform the sorting and selection operation.

(4)Calculate the cross operator and the mutation operator, and perform gene re-combination.

(5)Determine whether the maximum number of iterations is reached, and if so, output the results; if not, return to step 2.

4. Simulation Results

To validate the effectiveness of the IPGA-G algorithm for dynamic priority Multi-UAV task allocation problem, a comparative experiment was conducted between the Greedy Algorithm and the IPGA algorithm. The simulation experiments were conducted on hardware with an Intel(R) Core i5 processor running at 2.50GHz. The simulation environment was implemented using Matlab software.

The experimental simulation was set in a 50*50 square kilometer area, with coordinates representing various locations. The coordinates of the distribution center were set as (25,25). The experimental area was divided into sub-regions G_1 ($x > 25$, $y > 25$), G_2 ($x > 25$, $y < 25$), G_3 ($x < 25$, $y < 25$), and G_4 ($x < 25$,$y > 25$).

Each region was assigned a fixed rate of demand status change. The data are shown in Table 2.

Table 2: Rate of demand state change in different sub-regions.

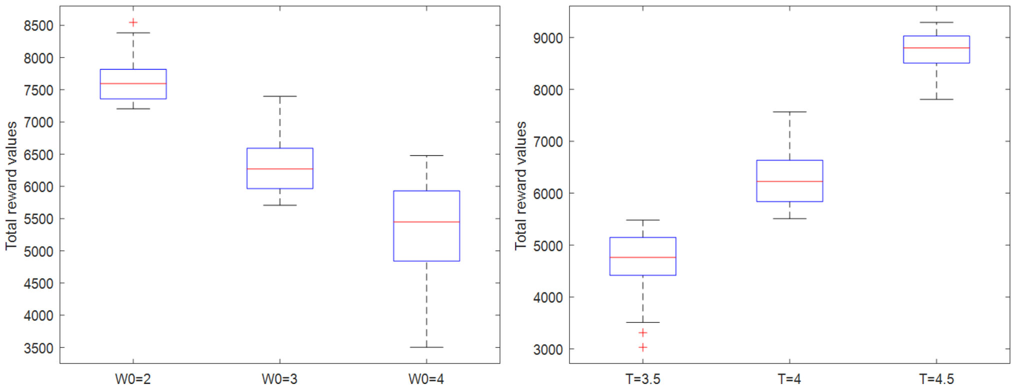
ξ_1	ξ_2	ξ_3	ξ_4
0.4	0.8	1.3	1.6

In this section, all experiments were set with a total of 4 rounds of task allocation for UAVs. Four UAVs were deployed from the distribution center. The average flight speed was set to $V_{AVG} =$ 11 km/h, with loading time $T_{\text{load}} = 0.5$ hours and unloading time $T_{\text{unload}} = 0.1$ hours. Each delivery carried a fixed weight of material $L_0 = 2.5$ kg. The coefficient of material consumption was set to $c = 2$. The parameters for the task reward function were $P_1 = 200$ and $P_2 = 180$. The transportation cost per unit distance was . The fixed cost for each UAV was $F_n = 80$.

4.1. Simulation results of key parameter analysis

The comparative tests were conducted on the most crucial parameters determining the timing of demand status changes: the period T and the threshold value W_0. The experimental testing was conducted with different parameter values for the period. For $T = 4$, experiments were carried out with $W_0 = 2$, $W_0 = 3$, and $W_0 = 4$, totaling 100 groups of experiments. For $W_0 = 3$, experiments were conducted with $T = 3.5$, $T = 4$, and $T = 4.5$. In each group of experiments, receiving points were randomly generated. The results are shown in Figure 3.

From the experimental results, it can be observed that the results are relatively balanced with both upper and lower limits when $W_0 = 3$ and $T = 4$. The distribution of results is relatively moderate. Therefore, setting $W_0 = 3$ and $T = 4$ as fixed parameters for subsequent experiments would be appropriate, the schedule of demand status change in different sub-regions is shown as table 3.



(a) (b)

Figure 3: (a) The box-plot of the results of experiments for different threshold values W_0. (b) The box-plot of the results of experiments for periods T.

Table 3: Schedule of demand status change in different sub-regions.

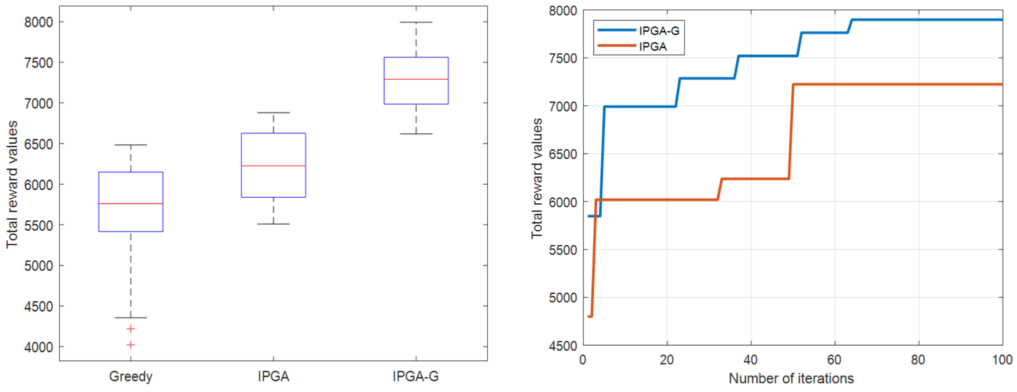
Sub-region	Round 1	Round 2	Round 3	Round 4
G_1	3.16	7.16	11.16	15.16
G_2	2.24	6.24	10.24	14.24
G_3	1.75	5.75	9.75	13.75
G_4	1.58	5.58	9.58	13.58

4.2. Simulation results of different algorithms

Testing the optimal solution of each algorithm under different scenarios, 50 groups of experiments were set up. In each group, 10 receiving points were randomly generated, and these receiving points were used to test the three algorithms. For both the IPGA and IPGA-G algorithms, the initial population size was set to 100, and the number of iterations was set to 100. The parameters for the IPGA-G algorithm were set as fol-lows: $k_1 = 0.7$, $k_2 = 0.4$, $k_3 = 0.1$, and $k_4 = 0.05$. The experimental results are shown in Figure 4(a). The experiments corresponding to the median total reward values obtained by the IPGA-G and IPGA algorithms were selected as references. From these experiments, the convergence curves of the two algorithms were compared, as shown in Figure 4(b).

Subsequently, an analysis of the computational time for both algorithms was conducted. The time taken for each iteration process of each algorithm to achieve or exceed the reward value of 6904.57 (the worst result of the IPGA algorithm) in each of the 50 rounds was recorded. The average time for the 50 rounds of experiments was calculated. The average time for the IPGA was 0.69 seconds, while the average time for the IPGA-G was 0.37 seconds.

Based on the results obtained from the simulation experiments, it can be concluded that the IPGA-G algorithm consistently achieves significantly higher average total reward values for optimal solutions across different scenarios compared to both the IPGA algorithm and the Greedy Algorithm. In summary, the proposed IPGA-G algorithm effectively addresses the task allocation



(*a*) (*b*)

Figure 4: (a) The box-plot of the results of experiments for different threshold values W_0 (b)The box-plot of the results of experiments for periods $T4$.

problem for multiple UAVs with dynamic priority changes. It exhibits significant advantages in terms of convergence speed, global search capability, and computational stability.

5. Conclusion

This thesis uses safety stock model as a reference to construct a problem model of Multi-UAV task allocation under dynamic priority changes, and the periodic function is used to represent the demand status of receiving points in sub-regions. The coding method is used to encode the greedy algorithm into the process of population initialization. Experimental results show that the improved partheno genetic-greedy combination algorithm has excellent solution search ability and search efficiency, and has high adaptability in dealing with the problem.

In the research process of this paper, in order to simplify the problem size and reduce the difficulty of calculation, many factors are ignored. Based on the existing results, future work can be further studied from the following aspects:

(1)In the process of realizing the collaborative task allocation of multiple UAVs, the communication limitations and computing power limitations of UAVs in the mode of complete centralized control are not considered, and these two points are factors that must not be ignored in the process of Multi-UAV collaborative operation planning in the future.

(2)In the process of constructing a Multi-UAV task allocation model in a dynamic environment, there may be situations where dynamic priorities affect constraints. In the real world, there are many more constraints than described in this article, and they may change as dynamic priorities changes.

References

Xinye Chen, Ping Zhang, Fang Li, and Guanglong Du. A cluster first strategy for distributed multi-robot task allocation problem with time constraints. In *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pages 102–107, 2018. doi: 10.1109/WRC-SARA.2018.8584210.

João N.C. Gonçalves, M. Sameiro Carvalho, and Paulo Cortez. Operations research models and methods for safety stock determination: A review. *Operations Research Perspectives*, 7:100164, 2020. doi: 10.1016/j.orp.2020.100164.

Sang Hyun Kim and Eric Feron. Robust gate assignment against gate conflicts. *Journal of Air Transportation*, 25(3):87–94, 2017. doi: 10.2514/1.D0067.

Qinqi Luo, Zhiqiang Li, Wentao Du, and Xuancen Liu. Uav task allocation based on behavioral mechanisms of wolf-pack hunting strategies. In *2021 7th International Conference on Control, Automation and Robotics (ICCAR)*, pages 179–183, 2021. doi: 10.1109/ICCAR52225.2021.9463442.

George Marios Skaltsis, Hyo-Sang Shin, and Antonios Tsourdos. A review of task allocation methods for uavs. *Journal of Intelligent & Robotic Systems*, 109(4):76, 2023. ISSN 1573-0409. doi: 10.1007/s10846-023-02011-0.

M. Srinivas and L.M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667, 1994. doi: 10.1109/21.286385.

Hongming Yang, Songping Yang, Yan Xu, Erbao Cao, Mingyong Lai, and Zhaoyang Dong. Electric vehicle route optimization considering time-of-use electricity price by learnable partheno-genetic algorithm. *IEEE Transactions on Smart Grid*, 6(2):657–666, 2015. doi: 10.1109/TSG.2014.2382684.

Yong Zhang, Pengfei Wang, Liuqing Yang, Yanbin Liu, Yuping Lu, and Xiaokang Zhu. Novel swarm intelligence algorithm for global optimization and multi-uavs cooperative path planning: Anas platyrhynchos optimizer. *Applied Sciences*, 10(14), 2020. doi: 10.3390/app10144821.

Honglu Zhou, Mingli Song, and Witold Pedrycz. A comparative study of improved ga and pso in solving multiple traveling salesmen problem. *Applied Soft Computing*, 64:564–580, 2018. doi: 10.1016/j.asoc.2017.12.031.