

Article

# Mission Planning of UAVs and UGV for Building Inspection in Rural Area

Xiao Chen <sup>1</sup>, Yu Wu <sup>1,\*</sup>  and Shuting Xu <sup>2</sup> <sup>1</sup> College of Aerospace Engineering, Chongqing University, Chongqing 400044, China; chenx\_up@163.com<sup>2</sup> College of Engineering, Beijing Forestry University, Beijing 100083, China; xushuting@buaa.edu.cn

\* Correspondence: cquwuyu@cqu.edu.cn

**Abstract:** Unmanned aerial vehicles (UAVs) have become increasingly popular in the civil field, and building inspection is one of the most promising applications. In a rural area, the UAVs are assigned to inspect the surface of buildings, and an unmanned ground vehicle (UGV) is introduced to carry the UAVs to reach the rural area and also serve as a charging station. In this paper, the mission planning problem for UAVs and UGV systems is focused on, and the goal is to realize an efficient inspection of buildings in a specific rural area. Firstly, the mission planning problem (MPP) involving UGVs and UAVs is described, and an optimization model is established with the objective of minimizing the total UAV operation time, fully considering the impact of UAV operation time and its cruising capability. Subsequently, the locations of parking points are determined based on the information about task points. Finally, a hybrid ant colony optimization-genetic algorithm (ACO-GA) is designed to solve the problem. The update mechanism of ACO is incorporated into the selection operation of GA. At the same time, the GA is improved and the defects that make GA easy to fall into local optimal and ACO have insufficient searching ability are solved. Simulation results demonstrate that the ACO-GA algorithm can obtain reasonable solutions for MPP, and the search capability of the algorithm is enhanced, presenting significant advantages over the original GA and ACO.

**Keywords:** building inspection; mission planning; UAVs and UGV; ACO-GA



**Citation:** Chen, X.; Wu, Y.; Xu, S. Mission Planning of UAVs and UGV for Building Inspection in Rural Area. *Algorithms* **2024**, *17*, 177. <https://doi.org/10.3390/a17050177>

Academic Editor: Takeshi Yamada

Received: 8 March 2024

Revised: 22 April 2024

Accepted: 22 April 2024

Published: 26 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, UAV technology has garnered significant attention from scholars and researchers. Its applications span across various domains, including the military, civilian, and commercial sectors, demonstrating increasing versatility [1]. UAVs are not only cost-effective, agile, and flexible in deployment, but they also offer the advantage of conducting missions without risking human lives and are reusable. Equipped with cameras and sensors mounted atop, they can effortlessly capture, record, and measure images and data in remote or complex terrains, hence finding extensive applications in aerial photography such as sports events, commercial activities, and inspections [2,3]. Inspection tasks enable the timely detection of hazards and defects in various areas and equipment, playing a crucial role in ensuring human safety. For instance, many aging rural buildings exhibit cracks, tilts, tile detachment, or excessive deformation, posing threats to the residents. Therefore, utilizing UAVs for inspection tasks emerges as a promising and vital technology.

Most inspection tasks for UAVs are focused on transmission towers [4] and ports [5], and there is little research on the application of UAVs in building inspection. The building inspection mission of a UAV primarily involves two key aspects: building image acquisition and mission planning. Zhang Z et al. have proposed a co-optimal coverage path planning (CCPP) method that jointly optimizes the image quality and the path of the UAV to realize the image detection of complex structures [6]. Zhang et al. proposed a method for automatically detecting structural damage and exterior wall cracks in building structures using UAV remote sensing information extraction, successfully extracting post-earthquake

damage information [7]. Ruiz et al. employed a field research experimental approach, emphasizing the feasibility and effectiveness of utilizing UAV technology for inspections [8]. Additionally, mission planning comprises task allocation and path planning. Typically, inspection mission planning employs a collaborative approach involving UGV and multiple UAVs. While UGV boasts high payload capacity and endurance, it faces challenges in rough terrain. Similarly, UAVs are constrained by factors such as limited payload capacity and flight duration, and a single UAV may even have defects such as a high risk of failure and long task time. In the face of a wide range of inspections and complex tasks, reliability is reduced, and it is often unable to ensure the smooth completion of the task. Hence, the collaborative technique of UGVs and UAVs offers distinct advantages. Radmanesh et al. successfully planned rational paths for medical or emergency supply delivery using UAV-UGV cooperation in disaster scenarios [9], while Wu et al. effectively employed UAV-UGV synergy for continuous urban surveillance [10].

The MPP of UGV-UAVs is a challenging and pivotal problem. MPP is a general term for task assignment issues and path planning problems in UAVs [11]. By assigning one or more ordered tasks to a group of UAVs, redundancy and conflicts between UAVs are avoided, optimizing overall task efficiency. Typically, the optimization objective aims to minimize the flight cost of multiple UAVs, such as by minimizing flight path length or overall cost. To obtain rational task allocation solutions, it is imperative to establish mathematical models for collaborative UAV planning and employ algorithms for solving. Common algorithms of MPP include market-based algorithms [12], clustering algorithms [13], and metaheuristic algorithms [14]. Currently, research on the algorithms of MPP primarily focuses on metaheuristic algorithms due to their ability to strike a balance between solution quality and computational time, producing satisfactory solutions within acceptable time frames. These algorithms have low computational complexity, fast execution, and robustness, making them more adaptable to complex task and resource variations compared to market-based and clustering algorithms.

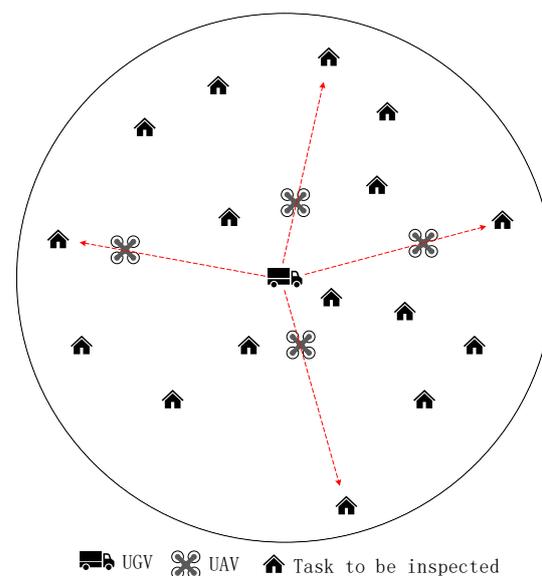
Metaheuristic algorithms draw inspiration from natural phenomena or biological behaviors [15]. Typical algorithms include Genetic Algorithm (GA) [16], Ant Colony Optimization (ACO) [17], Simulated Annealing (SA) [18], Particle Swarm Optimization (PSO) [19], and Tabu Search (TS) [20]. Local search algorithms such as TS and SA search for a single solution and prevent local optima using strategies like maintaining a Tabu list or probabilistic acceptance of inferior solutions. These algorithms are simple and easy to operate, but they are often inefficient for solving complex problems. GA, inspired by biological evolution processes of chromosome selection, crossover, and mutation, exhibits good optimization capabilities and population diversity. Zhang et al. improved GA convergence by incorporating a gravity search mechanism into the update process, solving collaborative reconnaissance task planning problems for multiple UAVs [21]. Zhu et al. addressed the task pre-allocation problem for multiple UAVs with different types of targets using a genetic algorithm with dual-chromosome encoding, achieving satisfactory results [22]. PSO, originating from the study of bird flock foraging, demonstrates good global search capabilities, fast convergence speed, and robustness [23]. Gou et al. proposed an inertia weight adaptive strategy to overcome PSO's susceptibility to local optima [24], though efficiency concerns remained unaddressed. ACO simulates the optimization of foraging routes by ants, exhibiting good convergence speed and optimal pathfinding. Ebadinezhad introduced a dynamic evaporation strategy to adapt ACO, improving its convergence speed and alleviating the tendency to fall into local optima [25]. While much research has focused on improving traditional algorithms and enhancing optimization outcomes, inherent flaws in singular algorithms persist. Consequently, scholars have attempted to integrate two algorithms to complement each other's strengths and weaknesses. Shang et al. proposed a genetic-ant colony algorithm for reconnaissance task planning [26]; meanwhile, Jia et al. combined genetic concepts to propose an improved particle swarm optimization algorithm for solving multi-UAV task planning problems [27]. And Jiang et al. proposed an ant colony-single parent genetic algorithm to solve large-scale multiple traveling salesman

problems, improving algorithm convergence speed, yet optimization outcomes warrant further improvement [28].

From the aforementioned studies, several issues in current research emerge: firstly, most studies of MPP focus solely on multiple UAVs without considering UGV and neglecting the issues caused by the low endurance of the UAVs. Secondly, in optimization processes, most research prioritizes the shortest paths or flight times, overlooking UAV operation processes or treating them homogeneously. However, the complexity of tasks at various points in real environments varies, leading to differences in operation times. Lastly, many studies reveal flaws in algorithms during the solving process, indicating room for improvement in overall optimization. Thus, addressing these issues, this paper calculates parking point positions before task planning, fully considering the positions of various task points and their impact on operation times. Subsequently, an ACO-GA is proposed, integrating the search process of ACO into GA's selection operation to enhance the initial solution quality. Furthermore, a series of improvements are made to the crossover and mutation operations to increase search capabilities and prevent falling into local optima. During optimization, considerations are given to the impact of UAV battery levels and operation time, aiming to solve the multi-UAV task problem effectively.

## 2. Establishment of Mathematical Model

During the inspection task in a specific area, UGV will park at a designated parking point, and the locations of task points are determined. Each task point is assigned to only one UAV, as illustrated in Figure 1 for the collaborative inspection of UAVs. Each rotary-wing UAV only needs to fly to a few task points for operation, with UAVs jointly completing all tasks in the area. Due to the scattered distribution of task points and the limitations of UAV positioning accuracy and safe return requirements, UGV needs to remain stationary and wait for UAVs to complete inspection tasks before departure. UAVs are influenced by their own battery levels; when the battery is low, they need to return to parking points to change batteries before continuing tasks. Simultaneously, considering the synchronicity of multiple UAVs completing tasks, the inspection task at the parking point is only considered to have been completed when the last UAV completes its task and lands on the UGV.



**Figure 1.** Single parking point inspection conceptual diagram.

To facilitate the quantitative study of this MPP, the following assumptions are made for the MPP model of UAVs from a single parking point:

1. Each UAV on the UGV possesses uniform characteristics, with identical cruising speeds during flight and equal battery capacities.

2. UAVs take off with full battery capacity, and when battery levels are low, they return to the UGV for battery replacement, disregarding the time for takeoff, landing, ascent, descent, and battery replacement.
3. Constrained by the distribution of task points and the precision requirements of UAV positioning, the UGV needs to remain stationary at the parking point until all UAVs complete their inspection tasks.
4. During the flight between the start and the destination, the time variations caused by altitude changes have been incorporated into the operational time of each task.

These assumptions and settings allow for the establishment of a mathematical model for task planning. For a region with  $n$  task points and  $m$  UAVs taking off from parking point  $Q_0$  to execute inspection tasks at task points in accordance with their respective planned sequences, where the inspection time at task point  $Q_i$  is denoted as  $t_i$ , from the above parameters, the optimization model for this task planning can be formulated. The objective is to minimize the total time spent by all UAVs to complete their respective inspection tasks. Constraints are on UAV battery capacity and the requirement that each task point be inspected by only one UAV, and all task points must be inspected. The decision variables represent the sequence of UAV flights to each task point. They can be expressed using the following formulas.

Decision Variables:

$$x_{k,i,j} = \begin{cases} 1, & \text{while } k = \{1, 2, \dots, m\}, i = \{0, 1, \dots, n\}, j = \{0, 1, \dots, n\} \text{ and } i \neq j \\ 0, & \text{others} \end{cases} \quad (1)$$

Constraints:

$$x_{k,i,0} = \begin{cases} 1, & \text{electrical value } w < 15\% \text{ or } \sum_{i=0, i \neq j}^n \sum_{j=1, j \neq i}^n x_{k,i,j} = m_k - 1, \text{ while } w = \frac{t_{\max} - t_{use}}{t_{\max}} \\ 0, & \text{others} \end{cases} \quad (2)$$

$$\sum_{k=1}^m \sum_{i=1, j \neq i}^{n_k} x_{k,i,j} = 1, j = 0, 1, 2 \dots, n_k \quad (3)$$

$$\sum_{k=1}^m \sum_{j=1, j \neq i}^{n_k} x_{k,i,j} = 1, i = 0, 1, 2 \dots, n_k \quad (4)$$

$$\sum_{k=1}^m \sum_{i=0}^n \sum_{j=1, j \neq i}^n x_{k,i,j} = m \quad (5)$$

Objective Function:

$$\begin{aligned} \min F &= \max \{F_1, F_2, \dots, F_m\} \\ \text{while } F_k &= \min \left\{ \sum_{i=0}^{m_k} \sum_{j=0}^{m_k} x_{k,i,j} S_{i,j} / v + \sum_{i=1}^{n_k} t_{k,i} \right\}, k = \{1, 2, \dots, m\} \end{aligned} \quad (6)$$

where  $x_{k,i,j}$  represents the decision variable. When  $x_{k,i,j} = 1$ , it means that the  $k$ th UAV flies from the task point  $i$  to the task point  $j$ ; otherwise, it does not fly from the task point  $i$  to the task point  $j$ ;  $t_{k,i}$  represents the time spent by the  $k$ th UAV inspecting the task point  $i$ ;  $t_{\max}$  is the maximum flight time that the UAV can operate under full charge.  $t_{use}$  is the time that has been flown,  $n_k$  is the number of task points assigned to the  $k$ th UAV,  $m_k$  is the total number of flight sections required by the  $k$ th UAV during the execution of the task,  $F$  is the value of the objective function, indicating the maximum time consumed by each UAV to complete the task,  $F_k$  is the time consumed by the  $k$ th UAV to complete the task.  $S_{i,j}$  represents the distance from mission point  $i$  to mission point  $j$ , and  $v$  represents the flying speed of the drone. Equation (2) indicates the constraint that the UAV needs to return to the parking point after completing the task or at low power. The reference battery level is

set at 15% based on the fact that the power consumption for a single task point will not exceed 10%. This threshold ensures both the maximization of task efficiency and the safe return of the UAV. Equations (3) and (4) ensure that only one UAV is required to perform the inspection task for each task point  $i$  or  $j$ . Equation (5) ensures that all task points have been inspected. Equation (6) shows how the value of the objective function is calculated.

### 3. Design of Planning Algorithm for Cooperative Inspection

The mathematical optimization model for MPP requires the design of corresponding algorithms for resolution. This section will present the method for determining the coordinates of the parking point, followed by the design of the ACO-GA to solve the mission planning problem.

#### 3.1. Determination of the Parking Point Location

Based on the mathematical optimization model established in Section 2, it is known that the objective function is the maximum value of operating time for each UAV, comprising two components: task operating time and round-trip time for task points. Therefore, to minimize the total task operating time, the selection method for parking points needs to consider these two aspects. As a classical location selection method, the barycentric method is often used to calculate the centroid coordinates of multiple homogeneous discrete points. Since this study needs to consider UAV operating time at task points, weights are assigned to each point during calculation, with task points having longer operating times assigned greater weights. The calculation method is as follows:

$$\bar{X} = \frac{\sum_{i=1}^m t_i X_i}{\sum_{i=1}^m t_i}, \bar{Y} = \frac{\sum_{i=1}^m t_i Y_i}{\sum_{i=1}^m t_i} \quad (7)$$

In Equation (7),  $\bar{X}$  and  $\bar{Y}$  are the horizontal and vertical coordinates of the calculated parking point, respectively,  $X_i$  and  $Y_i$  are the coordinates of task point  $i$ , and  $t_i$  is the operating time of the UAV at task point  $i$ .

#### 3.2. Mission Planning Algorithm Based on ACO-GA

The mission planning problem addressed in this study is a large-scale NP-Hard optimization problem involving multiple vehicles and objectives, making it challenging to solve using traditional mathematical methods. In recent years, swarm intelligence algorithms based on natural laws or biological behaviors have received considerable attention and research. These algorithms have shown excellent performance in solving large-scale combinatorial optimization problems, offering advantages such as fast computation, high precision, and strong applicability.

Both ACO and GA have demonstrated good performance in solving such problems, but they also have their limitations. ACO can easily fall into local optima and stagnate during the solution process due to the influence of transition probabilities and pheromones. GA, on the other hand, may suffer from slow convergence and low efficiency as individuals with high fitness values are heavily replicated during the selection process. Therefore, this study utilizes the method by which each ant in ACO obtains its path to replace the selection operation in GA. The obtained results are then subjected to crossover and mutation operations. Additionally, appropriate improvements are made to GA to enhance its global search capabilities. The specific process is outlined as follows:

##### 3.2.1. Encoding and Decoding of Individual Solutions

The solution to the MPP studied involves assigning  $m$  UAVs to execute tasks at  $n$  task points. Therefore, the final result of the solution is the sequence in which different UAVs

inspect each task point. Consequently, the encoding method for the solution is illustrated in Figure 2.

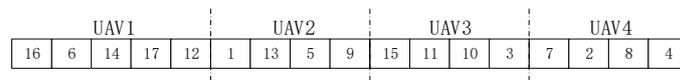


Figure 2. Encoding method for individual solutions.

The task points are evenly distributed among the UAVs as much as possible, and there are no duplicate task points within the encoding of the solution, ensuring that each task point is assigned to only one UAV for operation. During decoding, each UAV departs from the parking point and sequentially flies from left to right towards the corresponding task points for operation. For example, UAV1 in Figure 2 needs to sequentially visit task nodes 16, 6, 14, 17, and 12 to complete the tasks before returning to the parking point. At each task point, its battery level needs to be calculated according to Equation (2). If the battery level is low, the UAV needs to fly back to the parking point to replace the battery before continuing to the next task point; otherwise, it can fly directly to the next task point until all tasks assigned to the UAV are completed and then return to the parking point.

### 3.2.2. Selection Method Based on ACO

Traditional genetic algorithm selection operations effectively implement the survival of the fittest in the population, preserving individuals with higher fitness values for crossover and mutation operations in the next generation. Common selection methods include roulette wheel selection, probability selection, and tournament selection, all of which involve transmitting some individuals unchanged to the next generation. However, individuals with high fitness values may be heavily replicated, reducing solution efficiency and potentially causing the algorithm to converge to local optima. In the ACO, each ant determines its next task point based on the transfer probability of its task points. This transfer probability is calculated based on the residual pheromones and heuristic information along the path, which accelerates convergence. Each ant operates independently and communicates through pheromones, providing certain advantages for increasing solution diversity.

For the ant  $k$ , the probability of its transfer from task point  $i$  to  $j$  at time  $t$  is calculated using:

$$p_{i,j}^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}(t)]^\beta}{\sum_{s \in J_k(i)} [\tau_{i,s}(t)]^\alpha \cdot [\eta_{i,s}(t)]^\beta}, & j \in J_k(i) \\ 0, & \text{others} \end{cases} \quad (8)$$

where  $\tau_{i,j}(t)$  is the size of the pheromone from task point  $i$  to  $j$ , its initial value is usually a small constant, and its value is updated after all ants have obtained the path.  $\eta_{i,j}(t)$  is a heuristic factor that represents the desired degree from task points  $i$  to  $j$ , the value of which is usually the reciprocal of the distance between the two task points. The size of  $\alpha$  and  $\beta$  determines the importance of the above two parameters.

Therefore, two strategies were adopted in the selection operation: half of the solutions were generated using ACO, and then the tournament selection method was used until the number of solutions reached  $NP$ .

### 3.2.3. Crossover Method

The crossover operation of the genetic algorithm can lead to a qualitative leap in solution quality. Typically, a pair of chromosomes are selected, and with a certain probability, some of their genetic information is exchanged to create new individuals. However, due to the randomness of crossover, it is not conducive to algorithm convergence. Inspired by the convergence factor in whale optimization algorithms, a convergence factor  $A$  was introduced into the algorithm. Its calculation formula is:

$$A = 2a \cdot r - a \tag{9}$$

In Equation (9),  $r$  is a random number between  $[0, 1]$ , and  $a$  is the weight factor, which will gradually decrease with the progress of iteration. Its specific value is as follows:

$$a = 2 - 2t_{iter}/t_{iter,max} \tag{10}$$

In Equation (10),  $t_{iter}$  is the current number of iterations and  $t_{iter,max}$  is the maximum number of iterations.

Firstly, the solution with the maximum fitness value in the previous iteration is set as the parent chromosome  $R_0$ . Next, the value of  $A$  calculated by Equation (9) is judged. If  $|A| \leq 1$ , the chromosome  $R_1$  is crossed with the parent chromosome  $R_0$  to obtain a new chromosome  $R_1^*$  and reserve it; if  $|A| > 1$ , another chromosome  $R_2$  is randomly selected to cross with  $R_1$  and reserve  $R_1^*$ . With this setting, as the iteration proceeds, the value of  $|A|$  approaches 0, which greatly improves the probability of crossing with the parent chromosome, thus accelerating the convergence of the algorithm.

During the crossover operation, two random positions are selected along the length of the chromosomes, as illustrated in Figure 3. The genes at these positions on chromosome  $R_2$  are found and cleared on chromosome  $R_1$ . Then, the genes at these two positions on chromosome  $R_2$  are exchanged with those on chromosome  $R_1$ . Finally, the genes originally located at these positions on chromosome  $R_1$  are inserted into the blank positions. The specific process of chromosome  $R_1$  during this crossover operation is depicted in Figure 4.

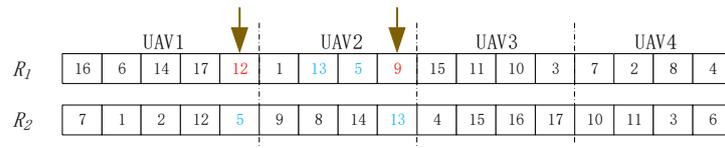


Figure 3. Select the positions for crossover.

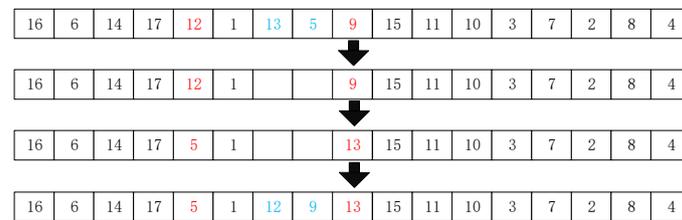


Figure 4. Detailed process of crossover.

### 3.2.4. Mutation Method

Mutation is a crucial technique in GA to prevent falling into local optima, thereby enhancing the diversity of the population. Typically, this operation is to modify local genes. However, in this study, the solutions are encoded as integers, and no repetition of numbers is allowed. Hence, a form of gene exchange is chosen. For the chromosome to be mutated, two positions of data are randomly selected for exchange, as shown in Figure 5, thereby generating a new chromosome.

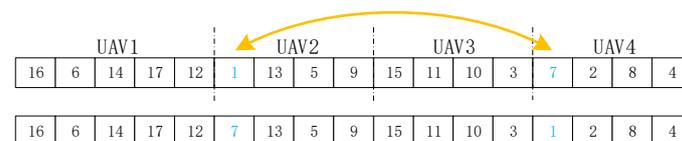


Figure 5. Chromosomal changes before and after mutation.

In traditional GA, the probabilities of crossover and mutation are usually constant. At the beginning of iterations, the changes in chromosomes caused by crossover are evidently

more significant than mutation, which is more conducive to solving large-scale problems. However, as iterations progress, the solutions in the population tend to converge. The more similar the chromosomes subjected to crossover are, the less beneficial it is for the diversity of the population. Therefore, it is advisable to decrease the probability of crossover and increase the likelihood of mutation. Hence, the probabilities of both during iterations can be adjusted as follows:

$$\begin{cases} P_m^{iter} = \frac{1}{2}P_m \cdot a \\ P_e^{iter} = \frac{1}{2}P_e(2 - a) \end{cases} \quad (11)$$

In Equation (11),  $P_m$  and  $P_e$  are the probability of crossover and mutation, respectively. In this setting, the probability of crossover and mutation will decrease and increase with the progress of iteration, respectively, which will have a more favorable impact on the search ability of chromosomes and the diversity of solutions.

### 3.2.5. Computing Fitness Values and Updating Pheromones

During each iteration, the aforementioned operations are performed. Subsequently, the fitness values of the newly generated population are computed. The best individual of each round is recorded, and its fitness value is output. Finally, the pheromones of the ant colony are updated according to the following formula:

$$\tau_{i,j}(t + 1) = (1 - \rho) \cdot \tau_{i,j}(t) + \Delta\tau_{i,j} \quad (12)$$

In Equation (12),  $\rho$  is the evaporation coefficient of the pheromone, and  $\Delta\tau$  represents the pheromone increment from task points  $i$  to  $j$  in this iteration, which is calculated as follows:

$$\begin{aligned} \Delta\tau &= \sum_{k=1}^{n_{NP}} \Delta\tau_{i,j}^k \\ \Delta\tau_{i,j}^k &= \begin{cases} \frac{Q}{S_k}, & \text{while ant } k \text{ passes } i, j \text{ in this iteration} \\ 0, & \text{others} \end{cases} \end{aligned} \quad (13)$$

In Equation (13),  $\Delta\tau_{i,j}^k$  refers to the amount of pheromone left by the ant  $k$  from task points  $i$  to  $j$  in this round of iteration,  $n_{NP}$  represents the population size,  $Q$  is the pheromone enhancement coefficient, and  $S_k$  is the fitness value of the ant  $k$  in this iteration.

The detailed procedure of the complete ACO-GA algorithm for solving the mission planning problem is shown in Figure 6.

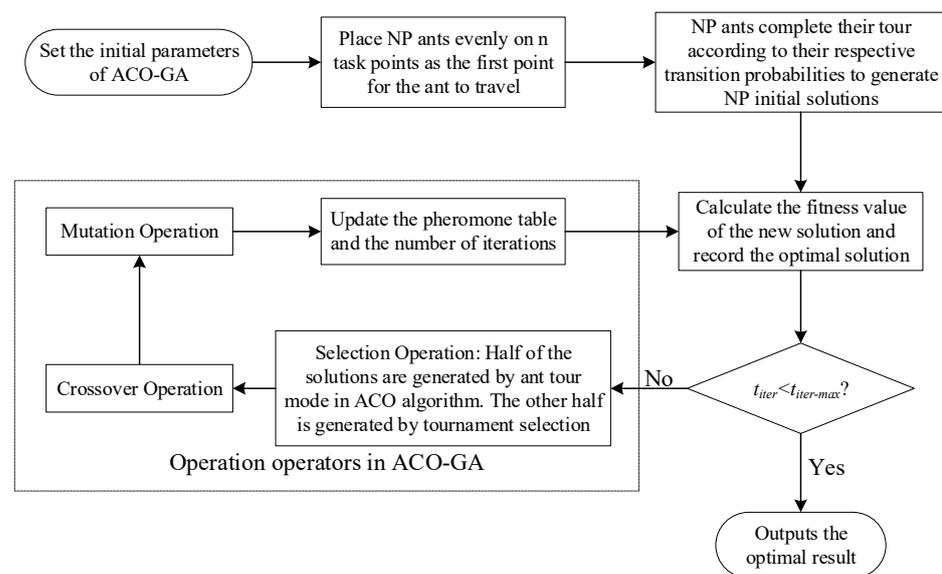


Figure 6. The flow chart of the computation of ACO-GA.

#### 4. Simulation Studies

To validate the superiority and practicality of the proposed method, corresponding scenarios were designed in MATLAB for analysis and computation. This section consists of several parts. Firstly, the results of ACO-GA proposed in this paper are demonstrated. Secondly, the results obtained in this study are compared with standard GA, ACO, and improved GA. Finally, the time difference in task execution is compared for different numbers of deployed UAVs to determine the optimal number of UAVs. The relevant parameter settings used in the simulation process are shown in Tables 1 and 2.

**Table 1.** Parameter settings of building inspection tasks.

Parameter	Parameter Values
Numbers of UAVs $m$	5
Maximum time of flight $t_{max}$	900 s
Flight speed $v$	15 m/s
Number of task points $n$	46

**Table 2.** Parameters settings for ACO-GA.

Parameter	Value	Parameter	Value
Number of populations $NP$	200	Mutation probability $P_e$	0.5
Maximum number of iterations $G$	5000	Enhancement factor $\alpha, \beta$	1.5
Crossover probability $P_m$	0.9	Pheromone enhancement coefficient $Q$	100

##### 4.1. Optimization Results of ACO-GA

The locations of 46 task points are randomly generated as objects for building inspection in a certain area. The coordinates of these task points, along with the required operation time for each point, are listed in Table 3. The centroid method was employed to compute the location of the parking point. The initial location of the parking point for UGV, calculated using Equation (1), is determined as: (1206.5, 1103.9).

**Table 3.** Coordinates and Operating Time of Task Points and the Parking Point.

Task Point	Coordinates	Operating Time	Task Point	Coordinates	Operating Time	Task Point	Coordinates	Operating Time
1	(6.3, 928.1)	75 s	17	(1155.5, 790.9)	90 s	33	(1487.8, 1049.2)	46 s
2	(72.5, 708.3)	61 s	18	(1171.9, 1166.5)	35 s	34	(1500.9, 1092.8)	39 s
3	(242.7, 92.0)	45 s	19	(1183.9, 1216.3)	57 s	35	(1493.0, 1415.2)	38 s
4	(335.6, 1380.9)	76 s	20	(1209.0, 1270.7)	36 s	36	(1526.8, 1235.1)	83 s
5	(489.2, 989.3)	87 s	21	(1218.8, 817.6)	88 s	37	(1603.3, 2174.9)	65 s
6	(533.2, 529.4)	40 s	22	(1245.3, 805.9)	30 s	38	(1688.6, 1677.0)	63 s
7	(569.1, 779.7)	62 s	23	(1233.7, 939.8)	77 s	39	(1753.9, 1035.0)	39 s
8	(606.0, 1532.8)	66 s	24	(1278.9, 902.9)	79 s	40	(1862.3, 1479.5)	82 s
9	(958.4, 1128.3)	75 s	25	(1284.2, 1665.1)	83 s	41	(1877.0, 1186.6)	68 s
10	(985.8, 1949.2)	57 s	26	(1315.5, 963.6)	35 s	42	(1941.7, 621.0)	51 s
11	(1063.1, 990.7)	35 s	27	(1343.1, 993.25)	54 s	43	(2008.3, 615.7)	61 s
12	(1067.0, 1048.7)	44 s	28	(1332.4, 1800.0)	46 s	44	(2005.1, 925.1)	54 s
13	(1063.6, 1048.7)	85 s	29	(1369.2, 996.8)	78 s	45	(2017.6, 621.4)	35 s
14	(1104.9, 977.8)	39 s	30	(1381.7, 854.5)	56 s	46	(2011.5, 891.1)	45 s
15	(1103.1, 1053.4)	80 s	31	(1425.0, 1195.1)	85 s		Parking position	
16	(1104.9, 1090.6)	63 s	32	(1440.1, 1024.0)	41 s		(1206.5, 1103.9)	

Based on the coordinates of the task points and the parking point provided above, the mission planning scheme is solved using ACO-GA proposed in this study. The distribution map of the UAV task planning results obtained is shown in Figure 7. In each iteration, the UAV with the longest time consumption is selected as the fitness value for that iteration.

The fitness change curve is depicted in Figure 8. The optimization rate has improved from 895.61 s to 785.88 s, resulting in a 12.25% improvement. From Figure 7, it can be observed that the route map optimized using the ACO-GA proposed in this study is highly reasonable. Each task point is assigned to a corresponding UAV for operation, demonstrating a clear division of labor and high efficiency. Additionally, each UAV does not need to return to the parking point for recharging but can directly return after completing the task. From the change curve shown in Figure 8, it can be seen that the optimization rate does not seem to be very high. This is because the operation time of UAVs at each task point is fixed, and this part of the time consumption cannot be optimized. Only the flight time of UAVs in the air can be optimized, resulting in an improvement of over one hundred seconds in flight time. Considering only the flight time, the optimization degree is quite significant.

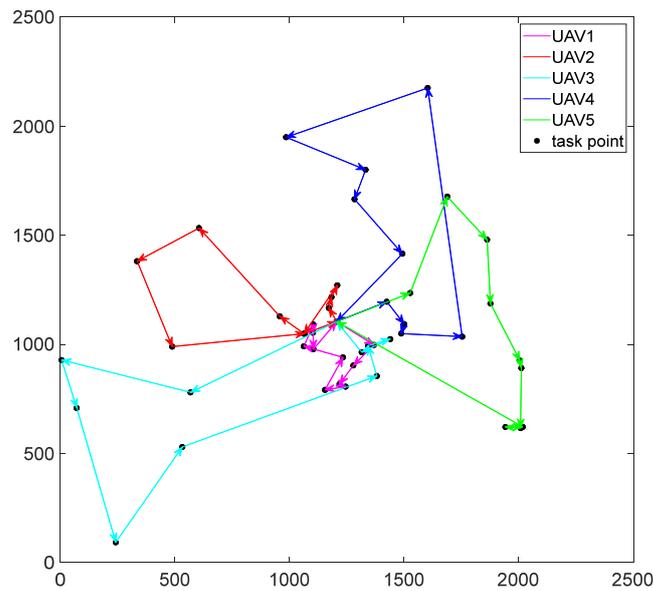


Figure 7. Distribution of UAVs in inspection task.

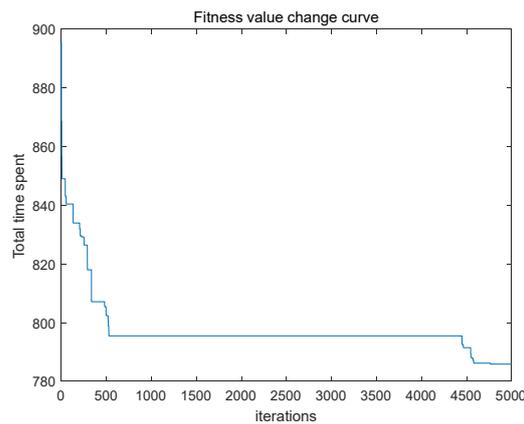


Figure 8. Fitness change curve of ACO-GA.

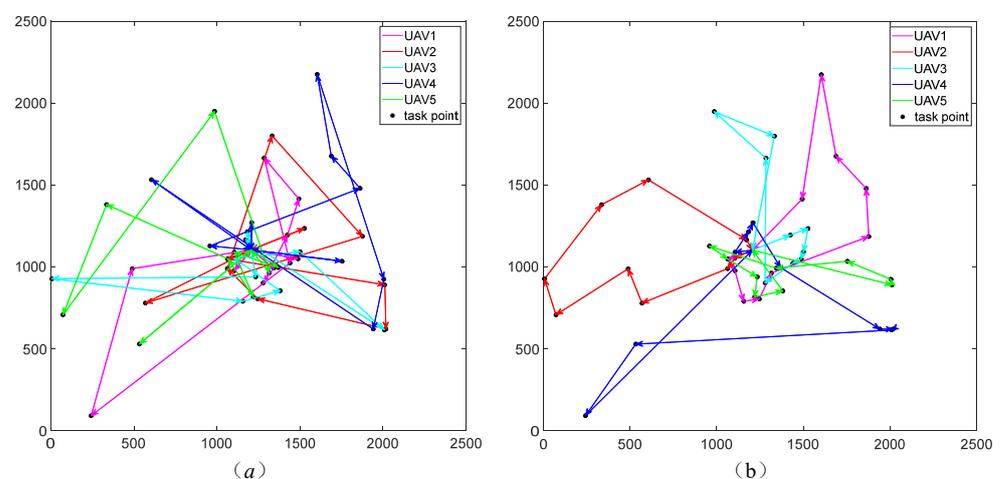
The operation time, flight time, and total time consumed by each UAV are presented in Table 4. From the data in Table 4, it can be observed that the proportion of flight time to the total time consumption of UAVs is relatively small. Moreover, the total time consumption of each UAV is relatively uniform. If a UAV is assigned a longer operation time at task points, its flight time is shorter, thereby achieving a relatively balanced workload distribution among UAVs.

**Table 4.** The time spent by each UAV.

Serial Number	UAV1	UAV2	UAV3	UAV4	UAV5
Operating time (s)	708	577	504	586	545
Flight time (s)	75.88	209.02	277.46	194.49	221.11
Total time (s)	783.88	786.02	781.46	780.49	766.11

#### 4.2. Comparison with Other Algorithm Optimization Results

The optimization results obtained using traditional GA and the improved GA are compared in Figure 9, with both algorithms starting from the same initial solution. In the optimization process, the traditional GA utilizes a roulette wheel selection operation, while the improved GA employs tournament selection. Additionally, improved GA incorporates a weighting factor “ $a$ ” in the crossover and mutation operations as well.

**Figure 9.** Optimization results of GA and Improved GA: (a) GA; (b) Improved GA.

From Figure 9a, it can be observed that the paths generated by the traditional GA for UAVs are highly disorganized. The UAVs need to return to the parking point for battery replacement to continue performing the task. In contrast, the results obtained from the improved GA exhibit significant improvement, as shown in Figure 9b. It is not necessary for UAVs to return to the parking point for recharging. However, it can be noted from Figure 9b that some paths are still suboptimal, such as the partially intersected paths of UAV3.

The optimization results obtained by ACO are shown in Figure 10. Overall, the paths observed are reasonable, and the results are superior, with fewer instances of UAVs deviating from the optimal paths. Analyzing the distribution of the formed paths, it can be observed that the quality of solutions obtained from ACO is higher compared to the aforementioned GA. This is attributed to the influence of heuristic factors and pheromones in ACO. The initial value of the heuristic factor is inversely proportional to the distance between task points, and the pheromone gradually stabilizes with iterations. Both factors jointly affect the probability of selecting a task point, making the search for solutions more informed. Consequently, the algorithm considers the impact of distance to a certain extent, making the search for solutions more efficient.

To validate the superiority and generality of the proposed algorithm, ten independent runs were conducted for each algorithm. The statistical results of the relevant data are summarized in Table 5. Representative fitness curves for each algorithm are selected to illustrate their variation, as shown in Figure 11. From Table 5 and Figure 11, it can be observed that the ACO-GA proposed in this paper initially exhibits significantly higher quality of initial solutions compared to GA. As the iterations progress, its search capability surpasses that of ACO, resulting in smaller average optimization values compared to other algorithms.

Moreover, the proposed algorithm demonstrates a certain level of stability, as indicated by the relatively small standard deviation. While ACO requires longer computation time due to the calculation of the selection probability for each task point based on its pheromone and heuristic factor during the iterative process, the ACO-GA algorithm still outperforms the ACO algorithm in terms of computation time since only half of the solutions are generated using the ant colony algorithm during the selection operation. Based on these observations, the algorithm designed in this paper significantly improves the quality of initial solutions and overcomes the shortcomings of both ACO and GA in terms of search capability and premature convergence, thus demonstrating reliable superiority and stability.

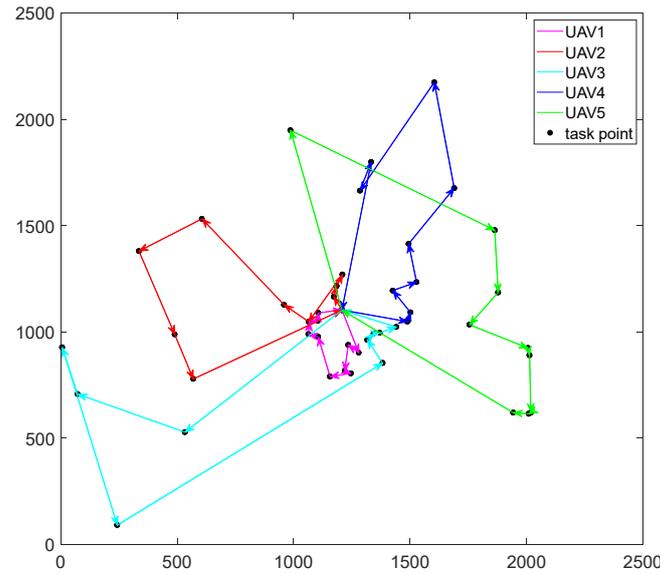


Figure 10. Optimization results of ACO.

Table 5. Comparison of optimization results after ten independent runs.

Algorithm	ACO-GA	GA	Improved GA	ACO
Average (s)	787.69	1110.42	825.40	825.51
Maximum (s)	790.04	1141.97	843.05	829.17
Minimum (s)	785.88	1048.86	805.13	819.98
Standard deviation (s)	2.07	32.39	13.20	4.47
Mean computation time (s)	316.39	253.11	290.61	456.99

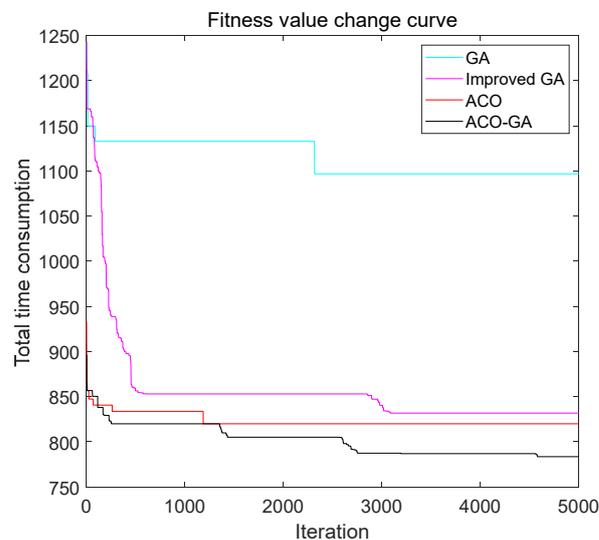


Figure 11. The variation of the fitness value of each algorithm.

#### 4.3. Influence of the Number of UAVs on the Task Allocation Scheme

During inspection tasks, the more UAVs there are, the less time is required to complete the tasks. However, in practical inspection tasks, not only time but also costs need to be considered. Based on market prices, the rental cost of a UAV is ¥150 yuan, the cost of replacing a battery for a UAV is ¥70 yuan, and the cost per second of each UAV flight is ¥0.1 yuan. By varying the number of UAVs, ten independent optimization runs were conducted, and the average operation time and costs are summarized in Table 6.

**Table 6.** Time consumption and costs under different numbers of UAVs.

Number of UAVs	3	4	5	6	7
Average time consuming (s)	1686.17	1060.20	787.69	698.63	614.55
Average cost (¥)	1165.85	1304.08	1143.85	1319.18	1480.19

Based on the data in Table 6, it can be observed that when the number of UAVs is 5, the inspection cost is the minimum. On this basis, increasing the number of UAVs will lead to higher rental costs, which have a greater impact on the total cost, resulting in an increase in overall costs. When the number of UAVs decreases, the cost does not decrease because each UAV requires battery replacement, incurring additional expenses. Therefore, the cost of completing the inspection task can be minimized when the number of UAVs is kept as low as possible, just enough to complete the inspection with the original battery.

## 5. Conclusions

This paper addresses MPP for building inspection by establishing a task planning model for coordinated inspection using UGV and UAVs. Firstly, the MPP problem is described, which involves dispatching UGV to designated location and utilizing multiple UAVs to inspect all buildings within a certain area. Secondly, a mathematical optimization model for the MPP is established, aiming to minimize the mission time. The constraints include ensuring coverage of all buildings and maintaining sufficient battery power for the UAVs. Additionally, the model takes into full account the impact of varying operational time requirements for different mission points. Subsequently, the location of parking spots is determined based on the positions of the mission points and the operational time required. Finally, to address the issues of weak search capabilities in ACO and the tendency of GA to get trapped in local optima, the paper proposes an ACO-GA hybrid approach.

The simulation results and comparisons with other algorithms demonstrate the superiority and effectiveness of the proposed ACO-GA, effectively addressing the task planning problem for building inspections. While this study focuses only on the task planning problem with a single parking point, the process of using UAVs for building inspections at individual task points has not been thoroughly studied, and real-world applications often involve inspection over large areas, requiring task planning with multiple parking points. In addition, the research findings of this paper primarily remain in the stages of algorithm design and simulation verification. Its applicability in the real world remains to be validated. Therefore, future research will focus on how UAVs can conduct inspections within individual buildings. And it is necessary to continue to explore solutions to task planning issues in larger areas and utilize actual data for verification and experimental validation of the effectiveness of the methods presented in this paper.

**Author Contributions:** Conceptualization, X.C. and Y.W.; methodology, X.C.; software, X.C.; validation, X.C., Y.W. and S.X.; formal analysis, Y.W.; investigation, S.X.; resources, X.C.; data curation, X.C.; writing—original draft preparation, X.C.; writing—review and editing, Y.W.; visualization, S.X.; supervision, Y.W.; project administration, Y.W.; funding acquisition, S.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China, grant number 52102453, and the APC was fund by State Key Laboratory of Efficient Production of Forest Resources, Key Laboratory of National Forestry and Grassland Administration on Forestry Equipment and Automation, and Fundamental Research Funds for the Central Universities, grant number BLX202224.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ye, F.; Chen, J.; Sun, Q.; Tian, Y.; Jiang, T. Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints. *J. Supercomput.* **2021**, *77*, 111–132. [\[CrossRef\]](#)
2. Beloev, I.H. A review on current and emerging application possibilities for unmanned aerial vehicles. *Acta Technol. Agric.* **2016**, *19*, 70–76. [\[CrossRef\]](#)
3. Yan, J.; Zhang, X.; Shen, S.; He, X.; Xia, X.; Li, N.; Wang, S.; Yang, Y.; Ding, N. A Real-Time Strand Breakage Detection Method for Power Line Inspection with UAVs. *Drones* **2023**, *7*, 574. [\[CrossRef\]](#)
4. Wu, M.; Chen, W.; Tian, X. Optimal Energy Consumption Path Planning for Quadrotor UAV Transmission Tower Inspection Based on Simulated Annealing Algorithm. *Energies* **2022**, *15*, 8036. [\[CrossRef\]](#)
5. Tang, G.; Wang, C.; Zhang, Z.; Men, S. UAV Path Planning for Container Terminal Yard Inspection in a Port Environment. *J. Mar. Sci. Eng.* **2024**, *12*, 128. [\[CrossRef\]](#)
6. Shang, Z.; Bradley, J.; Shen, Z. A co-optimal coverage path planning method for aerial scanning of complex structures. *Expert Syst. Appl.* **2020**, *158*, 113535. [\[CrossRef\]](#)
7. Zhang, R.; Li, H.; Duan, K.; You, S.; Liu, K.; Wang, F.; Hu, Y. Automatic Detection of Earthquake-Damaged Buildings by Integrating UAV Oblique Photography and Infrared Thermal Imaging. *Remote Sens.* **2020**, *12*, 2621. [\[CrossRef\]](#)
8. Ruiz, R.D.B.; Lordsleem, A.C., Jr.; Rocha, J.H.A.; Irizarry, J. Unmanned aerial vehicles (UAV) as a tool for visual inspection of building facades in AEC+ FM industry. *Constr. Innov.* **2022**, *22*, 1155–1170. [\[CrossRef\]](#)
9. Radmanesh, M.; Sharma, B.; Kumar, M.; French, D. PDE solution to UAV/UGV trajectory planning problem by spatio-temporal estimation during wildfires. *Chin. J. Aeronaut.* **2021**, *34*, 601–616. [\[CrossRef\]](#)
10. Wu, Y.; Wu, S.; Hu, X. Cooperative path planning of UAVs & UGVs for a persistent surveillance task in urban environments. *IEEE Internet Things J.* **2020**, *8*, 4906–4919.
11. Song, J.; Zhao, K.; Liu, Y. Survey on Mission Planning of Multiple Unmanned Aerial Vehicles. *Aerospace* **2023**, *10*, 208. [\[CrossRef\]](#)
12. Wang, Z.; Li, M.; Li, J.; Cao, J.; Wang, H. A task allocation algorithm based on market mechanism for multiple robot systems. In Proceedings of the 2016 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Angkor Wat, Cambodia, 6–10 June 2016; pp. 150–155.
13. Peng, Q.; Wu, H.; Xue, R. Review of dynamic task allocation methods for UAV swarms oriented to ground targets. *Complex Syst. Model. Simul.* **2021**, *1*, 163–175. [\[CrossRef\]](#)
14. Wu, Y.; Liang, T.; Gou, J.; Tao, C.; Wang, H. Heterogeneous Mission Planning for Multiple UAV Formations via Metaheuristic Algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 3924–3940. [\[CrossRef\]](#)
15. Wu, Y. A survey on population-based meta-heuristic algorithms for motion planning of aircraft. *Swarm Evol. Comput.* **2021**, *62*, 100844. [\[CrossRef\]](#)
16. Guangtong, X.; Li, L.; Long, T.; Wang, Z.; Cai, M. Cooperative multiple task assignment considering precedence constraints using multi-chromosome encoded genetic algorithm. In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018; p. 1859.
17. Chen, J.; Ling, F.; Zhang, Y.; You, T.; Liu, Y.; Du, X. Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. *Swarm Evol. Comput.* **2022**, *69*, 101005. [\[CrossRef\]](#)
18. Huo, L.; Zhu, J.; Wu, G.; Li, Z. A Novel Simulated Annealing Based Strategy for Balanced UAV Task Assignment and Path Planning. *Sensors* **2020**, *20*, 4769. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Zhang, Y.-Z.; Li, J.-W.; Hu, B.; Zhang, J.-D. An improved PSO algorithm for solving multi-UAV cooperative reconnaissance task decision-making problem. In Proceedings of the 2016 IEEE International Conference on Aircraft Utility Systems (AUS), Beijing, China, 10–12 October 2016; pp. 434–437.
20. Lee, M.-T.; Chen, B.-Y.; Lai, Y.-C. A Hybrid Tabu Search and 2-opt Path Programming for Mission Route Planning of Multiple Robots under Range Limitations. *Electronics* **2020**, *9*, 534. [\[CrossRef\]](#)
21. Zhang, Y.-Z.; Hu, B.; Li, J.-W.; Zhang, J.-D. Heterogeneous multi-UAVs cooperative task assignment based on GSA-GA. In Proceedings of the 2016 IEEE International Conference on Aircraft Utility Systems (AUS), Beijing, China, 10–12 October 2016; pp. 423–426.
22. Zhu, W.; Li, L.; Teng, L.; Yonglu, W. Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding. *Chin. J. Aeronaut.* **2018**, *31*, 339–350.
23. Han, H.; Bai, X.; Han, H.; Hou, Y.; Qiao, J. Self-adjusting multitask particle swarm optimization. *IEEE Trans. Evol. Comput.* **2021**, *26*, 145–158. [\[CrossRef\]](#)

24. Gou, Q.; Li, Q. Task assignment based on PSO algorithm based on logistic function inertia weight adaptive adjustment. In Proceedings of the 2020 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 27–28 November 2020; pp. 825–829.
25. Ebadinezhad, S. DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem. *Eng. Appl. Artif. Intell.* **2020**, *92*, 103649. [[CrossRef](#)]
26. Shang, K.; Karungaru, S.; Feng, Z.; Ke, L.; Terada, K. A GA-ACO hybrid algorithm for the multi-UAV mission planning problem. In Proceedings of the 2014 14th International Symposium on Communications and Information Technologies (ISCIT), Incheon, Republic of Korea, 24–26 September 2014; pp. 243–248.
27. Jia, Z.; Xiao, B.; Qian, H. Improved Mixed Discrete Particle Swarms based Multi-task Assignment for UAVs. In Proceedings of the 2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS), Xiangtan, China, 12–14 May 2023; pp. 442–448.
28. Jiang, C.; Wan, Z.; Peng, Z. A new efficient hybrid algorithm for large scale multiple traveling salesman problems. *Expert Syst. Appl.* **2020**, *139*, 112867. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.