



Article

# Multigene and Improved Anti-Collision RRT\* Algorithms for Unmanned Aerial Vehicle Task Allocation and Route Planning in an Urban Air Mobility Scenario

Qiang Zhou <sup>\*</sup>, Houze Feng and Yueyang Liu 

School of Electronic and Information Engineering, Beihang University, 37 XueYuan Road, Haidian District, Beijing 100191, China; sy2102512@buaa.edu.cn (H.F.); sy2202315@buaa.edu.cn (Y.L.)

\* Correspondence: zhouqiang\_ee@buaa.edu.cn

**Abstract:** Compared to terrestrial transportation systems, the expansion of urban traffic into airspace can not only mitigate traffic congestion, but also foster establish eco-friendly transportation networks. Additionally, unmanned aerial vehicle (UAV) task allocation and trajectory planning are essential research topics for an Urban Air Mobility (UAM) scenario. However, heterogeneous tasks, temporary flight restriction zones, physical buildings, and environment prerequisites put forward challenges for the research. In this paper, multigene and improved anti-collision RRT\* (IAC-RRT\*) algorithms are proposed to address the challenge of task allocation and path planning problems in UAM scenarios by tailoring the chance of crossover and mutation. It is proved that multigene and IAC-RRT\* algorithms can effectively minimize energy consumption and tasks' completion duration of UAVs. Simulation results demonstrate that the strategy of this work surpasses traditional optimization algorithms, i.e., RRT algorithm and gene algorithm, in terms of numerical stability and convergence speed.

**Keywords:** UAVs; UAM scenario; task allocation; path planning; multigene algorithm; RRT\* algorithm



**Citation:** Zhou, Q.; Feng, H.; Liu, Y. Multigene and Improved Anti-Collision RRT\* Algorithms for Unmanned Aerial Vehicle Task Allocation and Route Planning in an Urban Air Mobility Scenario. *Biomimetics* **2024**, *9*, 125. <https://doi.org/10.3390/biomimetics9030125>

Academic Editor: Heming Jia

Received: 9 January 2024

Revised: 14 February 2024

Accepted: 19 February 2024

Published: 21 February 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

Unmanned aerial vehicles (UAVs), also known as drones, have attracted lots of attention from the industry and academia, owing to their versatility, flexibility, and coordinated swarming capabilities [1]. UAVs are primarily categorized into two fixed-wing and rotary-wing UAVs [2]. Unlike fixed-wing UAVs, the latter can hover at fixed sites and vertically takeoff or land without runways or launchers. Therefore, the deployment and collaboration of rotary-wing UAVs offers a practical solution to alleviate congestion in ground transportation and delivery backlogs in urban environments [3]. In this context, Urban Air Mobility (UAM) scenario, demonstrated by Uber in [4], uses a type of rechargeable rotary-wing UAV called electric Verticals Take-Off and Landing (eVTOL) to balance the delivery of goods and passenger transport.

The industrial landscape of applying UAVs in the logistics and transportation sectors is also in progress. For instance, Walmart employed automated UAVs to deliver goods to customers in rural areas. DHL, one of Europe's biggest mailing companies, tested a new automation UAV-based platform for providing services. Regarding aeronautical manufacturing enterprises, e.g., NASA, Uber, and Hyundai Motor have focused their attention on the architectural and technical challenges of integrating automated UAVs into UAM transportation systems [2]. A key technique challenge in UAM is the strategic deployment of UAVs for provision services. The procedures of UAV deployment can be mainly classified into route planning and task allocation [5].

In terms of task-allocation problems, collaboration among UAVs is crucial to reduce conflicts and enhance efficiency. Hao-Xiang et al. put forward a modified organism search algorithm to solve task assignment problems with multiple UAVs [6]. The work [7]

proposed an adaptive genetic algorithm to assign multiple heterogeneous UAVs performing military tasks within a limited time. Randal et al. studied cooperative target assignments under enemy threats.

Mathematical models of previous works mainly include Mixed Integer Linear Programming (MILP), Multiple Traveling Salesman Programming (MTSP), Multiple Vehicle Routing Programming (MVRP) and Cooperative Multiple Task Assignment Programming (CMTAP) [8–11]. The MILP model is widely used in task assignment problems owing to its extensibility. However, the simple processes of MILP models are more suitable for small-scale problems. Additionally, the MTSP model does not consider the heterogeneity of tasks. Additionally, the MVRP model suits well time-constrained tasks, but does not refer to the flight dynamic performance of UAVs [12–14]. On the contrary, the CMTAP model is suitable for heterogeneous task assignments with multiple UAVs. Therefore, the CMTAP model of this work is built and studied for task assignment problems with multiple UAVs in a UAM scenario.

As for addressing the CMTAP model of task assignment problems, genetic algorithms are commonly used for their scalability, simplicity, and extensibility. However, the failure of a single UAV can cause deadlock in the genetic algorithm. Moreover, different task constraints for heterogeneous tasks can lead to the local existence of the solution. In this regard, the work improves and proposes the efficient and innovative multigene algorithm by designing customized selection and multipoint crossover for heterogeneous tasks. Specific details will be expanded in the following sections [15,16].

The main goal of the trajectory design of multiple UAVs is to improve energy efficiency and enhance the timeliness of the system. Huimin et al. optimized UAVs' trajectory to minimize time-related performance [17]. They presented a novel framework to solve the realistic non-convex problem. The work [18] adopted the genetic algorithm and the particle swarm algorithm in optimizing the trajectories of UAVs. The simulation proves that the genetic algorithm performs better than the particle swarm algorithm [19].

According to route planning problems of multiple UAVs in previous works, an efficient path planning algorithm can improve the execution efficiency and robustness of the system. Therefore, reducing the obstacle avoidance system's discovery time and path length is worthy of research. Hassan et al. proposed a novel hybrid form of the Dubins-simulated annealing (HDSA) optimization framework for emergency landing, which can plan the most suitable and admissible trajectory to the landing site in emergency flight conditions [20]. Dolgov et al. proposed the A\* algorithm, a graph traversal algorithm, to find the optimal path in an obstacle avoidance system [21]. However, the computation load of each step increases fast with the growth of the complexity of the environment [22,23]. The rapidly exploring random tree (RRT) algorithm has the integrity of spatial search ability to fast and randomly sample in the map without pre-modeling [24]. However, the convergence speed of the RRT algorithm is low, and the smoothness of the generated path is poor due to the randomness of the search direction [25–29].

Previous works have improved and optimized the RRT algorithm to address its shortcomings in the efficiency of path planning. The RRT\* algorithm proposed by Karaman et al. adopts two strategies, reselecting parent nodes and rewiring, to plan a near-optimal path, which may increase time cost [30]. Luo et al. proposed an informed RRT\* algorithm based on Karaman's work. The informed RRT\* algorithm limits the sampling area to an ellipse to improve the speed of obtaining the asymptotic optimal path [31]. Kuffner et al. proposed the RRT Connect algorithm, which adds one more random tree with the target point as the root node [32,33]. Two random trees of the RRT Connect algorithm grow towards each other simultaneously [34].

Although scholars have performed a lot of previous work on improving the RRT\* algorithm to solve route planning problems, there still exists poor adaptability to the UAM scenario, slow convergence speed in finding the target, and poor planning path quality [35,36]. To address the above shortcomings, this work proposes an improved anti-collision RRT\* (IAC-RRT\*) algorithm. First, the IAC-RRT\* algorithm offers the map

complexity evaluation strategy to calculate the corresponding map's suitable step size and bias probability. The time switching factor is also used to adapt the UAM scenario. Moreover, the normal line between the source and the goal is introduced to ensure the random tree explores positively to the target point. Simulation results have demonstrated the IAC-RRT\* algorithm can improve the effectiveness and quality of sampling points, leading to better planning path quality.

Overall, the main contributions of this paper are listed as follows:

- To the best of our knowledge, we first study to minimize energy consumption in the context of heterogeneous task assignments for multiple UAVs in urban environments. This research not only tackles the task assignment problem but also considers charging requirements and diverse task types. Consequently, the task assignment problem constitutes a CMTAP model. To address this model, we propose a multigene algorithm by designing customized selection and multipoint crossover mechanisms to handle the heterogeneity of tasks.
- This paper also proposes the IAC-RRT\* algorithm to compute the route planning problems in the UAM scenario. A time-switching factor is proposed to adapt the airspace management regulations of the UAM scenario. Moreover, the normal line between the source and the goal is introduced to ensure the random tree explores positively to the target point. Compared to the RRT\* algorithm, the novel route planning method works well in urban environments.
- Simulation results indicate that the strategy of this work exhibits superior adaptability and efficiency than the traditional optimization algorithm, i.e., RRT\* algorithm, and gene algorithm. This advancement marks a significant stride in optimizing UAV schedules within UAM scenarios.

## 2. Models and Methods

### 2.1. System Model

We consider a UAV-enabled task scheduling system in UAM scenarios with UAVs,  $u \in \mathcal{U} = \{1, 2, \dots, Nu\}$ , and tasks,  $s \in \mathcal{S} = \{1, 2, \dots, Ns\}$ , as shown in Figure 1, where  $\mathcal{U}$  and  $\mathcal{S}$  are sets of UAVs and tasks, respectively. Additionally, the number of UAVs and tasks are denoted as  $Nu$  and  $Ns$ . Each UAV has communication and navigation devices, allowing persistent real-time communication with the ground control center. Heterogeneous tasks, randomly generated and distributed in urban space, are divided into different classes according to the traffic demand, which will be discussed in the following subsection.  $Nu$  UAVs cooperatively visit  $Ns$  known task sites in clusters, and the IAC-RRT\* algorithm is employed to optimize the trajectory of each UAV, thereby guaranteeing the minimization of flight paths while adhering to safety protocols.

The location of the  $s$ -th task point is denoted by coordinates  $(x_s, y_s, h_s)$ , where  $s \in \mathcal{S}$ . Similarly, the position of the  $u$ -th UAV at a given time  $t$  is denoted by  $(x_u^t, y_u^t, h_u^t)$ . Although tasks are heterogeneous, the mechanical energy consumption of UAVs during the task execution is not incorporated in the total energy consumption in subsequent problem formulations.

#### 2.1.1. Heterogeneous Task Execution Model

As for UAV tasks, the  $k$ -th task generated at time  $t$  is denoted as

$$s_l^t(k) = \{F_l(k), D_l(k), (x(k), y(k), h(k))\}, \quad (1)$$

where  $l \in 1, 2, 3$  represents different task types, and  $F_l(k)$  denotes the total number of UAVs are required to finish the  $k$ -th task, which is defined as

$$F_l(k) = \begin{cases} 1, & \text{single - UAV mission,} \\ \{n | n \geq 2, n \in \mathcal{N}^+\}, & \text{multi - UAVs mission,} \end{cases} \quad (2)$$

where  $D_l(k)$  represents the duration of completing the task, which remains consistent for tasks of identical type. And  $(x(k), y(k), h(k))$  is the position of a specific task  $s_l^i(k)$ . In this paper, UAV missions are divided into three typical types, each tailored to meet the demands of the UAM scenario.

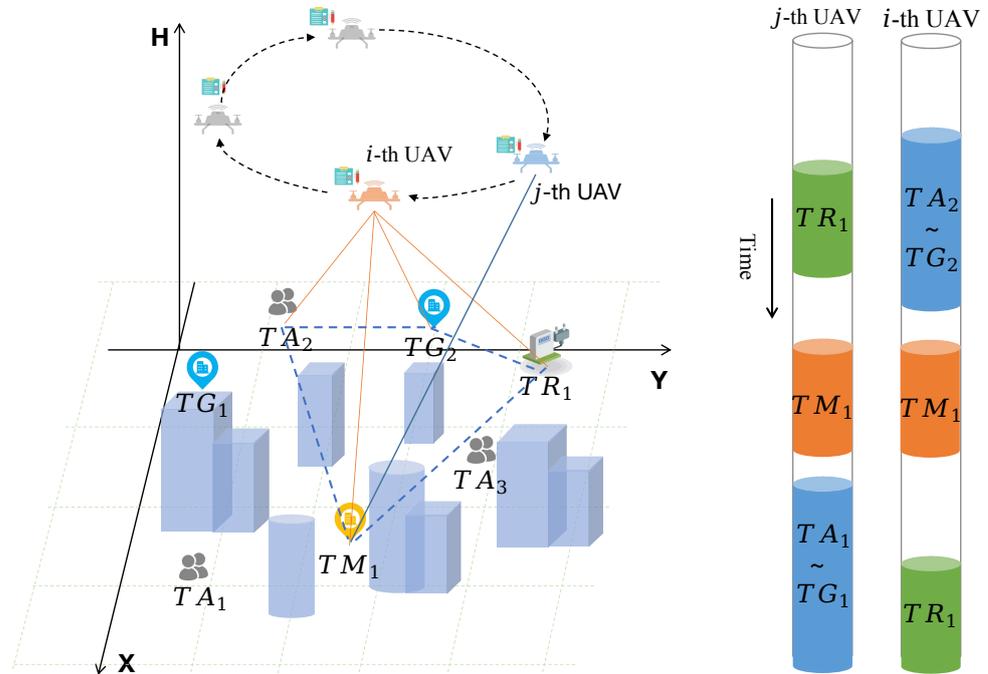


Figure 1. System model of UAV-enabled task scheduling system in UAM scenario.

**Air taxi task I:** This task type is denoted by  $l = 1$ . As shown in Figure 1, there are  $N_a$  air taxi tasks in the UAM scenario, e.g., UAV  $i$  transports loads in demand from one designated site  $TA_2$  to another  $TG_2$ . The subscripts of  $TA$  and  $TG$  are numbered sequentially. Additionally, this is the single-UAV mission type and  $F_l = 1$ . The air taxi task is marked in blue from the timeline of the left part of Figure 1.

**Hover and monitor task II:** This task type is denoted by  $l = 2$ . Multi-cooperated UAVs can be used as effective tools to hover at fixed locations and monitor some specific applications in UAM scenarios, i.e., precision agriculture, smart traffic monitor. To mathematically measure the problem of this type, we define a general task type with multi-cooperated UAVs performing tasks, and thus  $F_l \geq 2$ . Moreover, assigned UAVs cooperate and hover at task sites for a given length of time under unified control. Additionally, there are  $N_h$  hover and monitor tasks in the task set  $\mathcal{T}$ .

**Recharge task III:** This task type is denoted by  $l = 3$ , and  $F_l = 1$ . Considering the battery capacity and the flight endurance of rotary-wing UAVs, charging stations are established in specific urban areas. Drones with low battery status can autonomously navigate to these stations for recharging. It is worth noting that each UAV's battery capacity is defined as  $C$ , with three operational modes to ensure drone safety.

The battery level restricts task types that UAVs can accept. Firstly, the  $i$ -th UAV can not take air taxi task I if its current battery level  $C_i(t)$  is less than  $0.4C$ . Then, if the  $i$ -th UAV is in the middle of some task and its current battery level is less than  $0.3C$ , it will flee back to the charge station once it finishes the task. Lastly, the  $i$ -th UAV is forced to fly back to the charge station when its battery level is less than  $0.2C$ , regardless of whether it is in the middle of the task. However, if the UAV leaves its current task, this task will be judged

as a failure. Therefore, task types taken by the  $i$ -th UAV with different battery levels are given as

$$\begin{cases} s_1(k), s_2(k), & 0.4C \leq C_i(t) \leq C, \\ s_2(k), & 0.3C \leq C_i(t) \leq 0.4C, \\ s_2(k), s_3(k), & 0.2C \leq C_i(t) \leq 0.3C, \\ s_3(k) & C_i(t) \leq 0.2C. \end{cases} \quad (3)$$

In (3),  $s_1(k)$ ,  $s_2(k)$ ,  $s_3(k)$  are used to represent air taxi task I, hover and monitor task II and recharge task III.

### 2.1.2. Energy Consumption Model

This subsection illustrates the energy model for a UAV during a cruise. Within the specific scenario as shown in Figure 2, the analysis of energy consumption during UAV movements is divided into four distinct phases, contingent upon the UAV's state of motion. We assume  $G_V$  as the standard gravity of the UAV. The rotary-UAV's thrust can be calculated as  $F_V = \sqrt{(G_V - (a_1(V_{hor} \cos(\alpha))^2 + a_2F_V)^2 + (a_3V_{hor}^2)^2}$ .  $V_{hor}$  and  $V_{hor}$  are horizontal speed and vertical speed, respectively.  $\alpha$  is the directional angle between the next target location and its orientation to the next waypoint. Although the flight dynamics of UAVs are intricate, any flight maneuver executed by a UAV can be decomposed into horizontal and vertical components. When calculating the energy consumption of complex flight movements, it can be effectively represented by summing the energy expenditures of both horizontal and vertical flight motions.

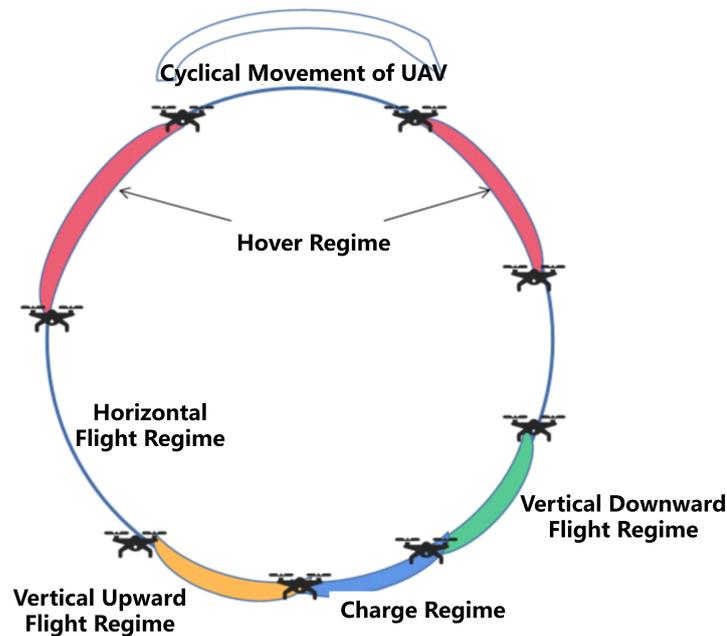


Figure 2. Cyclical movement of UAV.

(1) UAV horizontally flying regime: In this state, UAVs horizontally fly during its transit. Suppose the horizontal flight velocity remains constant at  $V_t$ , and the consumption power  $P_I(V_t)$  for horizontal flight is expressed as [37]

$$P_I(V_t) = \left( \frac{k_1}{k_2} + a_4 \right) F_V^{3/2} + a_3 V_t^3. \quad (4)$$

(2) UAV hovering regime: Similarly, the power consumption during UAV hovering  $P_H$  is represented as [37]

$$P_H = \left( \frac{k_1}{k_2} + a_4 \right) G_v^{3/2}, \tag{5}$$

where charging stations are located at ground level, necessitating UAVs to ascend or descend for recharging or task execution.

(3) UAV vertically flying upward regime: By neglecting the acceleration and deceleration process, UAVs ascend or descend at a constant velocity  $V_u$ . The ascending related power denoted by  $P_A$  is calculated by [37]

$$P_A(V_u) = k_1 G_v \left[ \frac{V_u}{2} + \sqrt{\left( \frac{V_u}{2} \right)^2 + \frac{G_v}{k_2^2}} \right] + a_4 G_v^{3/2}. \tag{6}$$

(4) UAV vertically flying downward regime: the descending related power denoted by  $P_D$  with  $V_u$  is formulated as [37]

$$P_D(V_u) = k_1 G_v \left[ \frac{-V_u}{2} + \sqrt{\left( \frac{V_u}{2} \right)^2 + \frac{G_v}{k_2^2}} \right] + a_4 G_v^{3/2}. \tag{7}$$

In (4)–(7),  $k_1, k_2, a_1, a_2, a_3$ , and  $a_4$  are technical parameters of vehicles, independent of the UAV’s motion or environment factors. Therefore, the energy consumption analysis for the  $k$ -th flight of UAV  $i$  can be segmented into four components corresponding to the movement states. The total energy consumption of UAV  $i$ ’s  $k$ -th flight,  $W_i^k$ , can be given as

$$W_i^k = \int_0^{\tau_i^k} P_i^k(t) dt = \underbrace{\sum_{k=1}^{N_j+1} P_I(V_t) \frac{d_{j,i}^k}{V_t}}_{\text{Horizontal flight}} + \underbrace{\sum_{n=1}^{N_j} P_H \tau_{i,j}^{k,H}}_{\text{Hover}} + \underbrace{P_A(V_a) \frac{H_k}{V_a}}_{\text{Ascent}} + \underbrace{P_D(V_d) \frac{H_k}{V_d}}_{\text{Descent}}, \tag{8}$$

where  $N_j$  is the total tasks that UAV  $i$  should finish in its  $k$ -th flight.  $d_{j,i}^k$  is the distance between the site of task  $j$  and its previous site.  $\tau_i^k$  is the  $k$ -th total flight duration of UAV  $i$ .  $\tau_{i,j}^{k,H}$  is the hovering duration at site of task  $j$ .  $H_k$  is the sum of elevated height in its  $k$ -th flight. Additionally,  $V_a$  and  $V_d$  are the velocities of upward and downward flight, respectively.

## 2.2. Problem Formulation

### 2.2.1. Task Allocation Problems

The goal of task allocation is to achieve the optimal assignment of  $N_s$  tasks to  $N_u$  UAVs to maximize the overall reward. Additionally, the task list comprises  $N_a$  air taxi tasks,  $N_h$  hover and monitor tasks, which satisfies

$$N_s = N_a + N_h, \tag{9}$$

and each task is exclusively assigned to a specific vehicle during a single flight. Furthermore, each UAV can be assigned a maximum of  $L_t$  tasks, subject to the condition

$$N_s < N_u \cdot L_t. \tag{10}$$

The global reward function is the summation of local rewards for each vehicle, which is determined by a function of the tasks assigned to each UAV. Optimization goals of time and energy consumption are studied separately. The time-oriented problem focuses on minimizing the time required to complete all tasks in an urban scenario, which can be mathematically represented as

$$\begin{aligned}
 \mathbf{P}_1 : \quad & \min \sum_{i=1}^{Nu} t_i \\
 \text{s.t.} \quad & \\
 t_i = & \underbrace{\sum_p \sum_j S_{jp}^i \times t_{jp}^i + \sum_p S_{pa}^i \times t_{pa}^i}_{\text{flight time between task points}} + \underbrace{\sum_{j=1}^{Ns} D_j^i \times t_j^d}_{\text{duration of task } j} \\
 & + \underbrace{\sum_a RE_a^i \times t^c}_{\text{charging duration}} \tag{11} \\
 D_j^i = & \begin{cases} 1, & \text{if UAV } i \text{ handles task } j, \\ 0, & \text{otherwise} \end{cases} \\
 S_{ak}^i = & \begin{cases} 1, & \text{if UAV } i \text{ handles tasks } a, j, \\ 0, & \text{otherwise,} \end{cases} \\
 RE_a^i = & \begin{cases} 1, & \text{if UAV } i \text{ charges at site } a, \\ 0, & \text{otherwise} \end{cases}
 \end{aligned}$$

where  $t_j^d$  is the operation time of task  $j$ , which is uniform for all UAVs.  $t^c$  denotes the fixed charging duration,  $t_{pa}^i$  signifies the flight duration between site  $p$  and  $a$ , and the charging station  $a$ . Constraints indicate that the operation time of a single UAV depends on whether the task is undertaken and which charging station is chosen.

The minimal power consumption problem is independent of the minimal time problem, as the power consumption of each vehicle determines the power consumption of completing all tasks in the system. According to Equations (4)–(8), the power consumption optimization problem of  $k$ -th flight can be represented by

$$\begin{aligned}
 \mathbf{P}_2 : \quad & \min \sum_{i=1}^{Nu} W_i^k \\
 \text{s.t.} \quad & \\
 & i \in \mathcal{U}, \quad i \leq Nu, \\
 & k \in \mathcal{N}^*,
 \end{aligned} \tag{12}$$

where  $k$  belongs to the set of positive integers  $\mathcal{N}^*$ , and represents any flight of UAV  $i$ .

### 2.2.2. Route Planning Problems

The goal of route planning in this paper is to find the path with the shortest distance from the initial task site to the final task site within a limited time while avoiding obstacles in the UAM Operating Environment (UOE). The UOE, a flexible airspace area, can be redefined and modified over time. For example, if the traffic patterns at a nearby airport change, the available UOE may be adjusted to avoid potential traffic conflicts. Changes in the available UOE are likely to occur several times daily and can be promptly detected by the ground control center. Taking into account the dynamic nature of the UOE, the cost function for distance,  $C_{length}$ , can be represented by the node  $i$  [38],

$$\begin{aligned}
 \mathbf{P}_3 : \quad & \min \sum_{i=1}^{N_u} C_{length} \\
 \text{s.t.} \quad & \mathcal{D}_{points} = \{i_d \mid i_d \in \text{Discovered Points}\}, \\
 & i_d \in Q_n + \gamma_{st} Q_{st}, \\
 & \gamma_{st} = \begin{cases} 1, & \text{if normal traffic of UAM scenario,} \\ 0, & \text{otherwise,} \end{cases}
 \end{aligned} \tag{13}$$

where  $\gamma_{st}$  represents the time transformation factor, which can be utilized to indicate the available UOE over time. The depreciation of (13) aims to identify the minimal  $C_{length}$  encompassing the points  $i_d$  discovered by the proposed algorithm within the available UOE.  $Q_n$ ,  $Q_{st}$ , and  $Q_{ob}$  are the general area, “flexible” area and obstacle area, respectively, where the “flexible” area can be controlled to switch by  $\gamma_{st}$ .

### 2.3. Proposed Algorithms

#### 2.3.1. Multigene Algorithm for Task Allocation Problem

The initial step of the multigene algorithm involves the design of an encoding scheme that establishes connections between task sets, UAV sets, execution sequences, and assignment rules. Therefore, the proposed algorithm encodes the solution for (11) and (12) into a matrix form, also referred to as a chromosome, as shown in Figure 3. Each “gene” corresponds to a column with the matrix, representing a specific task assignment. In the first row of Figure 3, there are 10 tasks, comprising seven air taxi tasks labeled as 1–7, and three hover and monitor tasks labeled as 8–13. Additionally, No.8 and No.9 are associated with the same multi-vehicle task, necessitating the participation of two distinct vehicles. Rows two through six represent task type, execution order, UAV order number, execution time, and energy consumption.

<b>Task Number</b>	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Task Type</b>	I	I	I	I	I	I	I	II	II	II	II	II	II
<b>Execution Order</b>	3	4	5	6	7	11	12	1	2	9	13	8	10
<b>UAV Order Number</b>	U2	U1	U3	U1	U1	U2	U2	U3	U2	U1	U1	U2	U3
<b>Execution Time</b>	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$
<b>Energy Consumption</b>	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$	$W_{13}$

Figure 3. Figure of multigene chromosome coding method.

The second step involves chromosome initialization, randomly generating  $N_{chrom}$  encoding schemes. These  $N_{chrom}$  initial chromosomes constitute the first generation of task

allocation schemes. Subsequently, the initial set of chromosomes undergoes the iteration of crossover, mutation, and reinsertion until either the maximum number of iterations is reached or the optimal task allocation scheme is found. The processes mentioned above are outlined as follows.

Phase 1, crossover process: By employing roulette wheel selection, the selected chromosome randomly modifies segments of the gene sequence within its vicinity. The crossover probability  $P_c$  varies depending on the task type and is defined as follows:

$$P_c = \begin{cases} P_c(l = 1), & l = 1, \\ \frac{Na}{2Nh} P_c(l = 1), & l = 2, \end{cases} \quad (14)$$

where  $P_c(l = 1)$  is set as a constant according to the simulation needs,  $Na$  and  $Nh$  represent the numbers of air taxi tasks and hover and monitor tasks, respectively. Given that the hover and monitor task requires two different UAVs, the crossover probability  $P_c$  matching with the quantitative proportion of two task types can improve the search efficiency of the multigene algorithm. Furthermore, a partially matched crossover method ensures that each gene occurs only once, as shown in Figure 4.

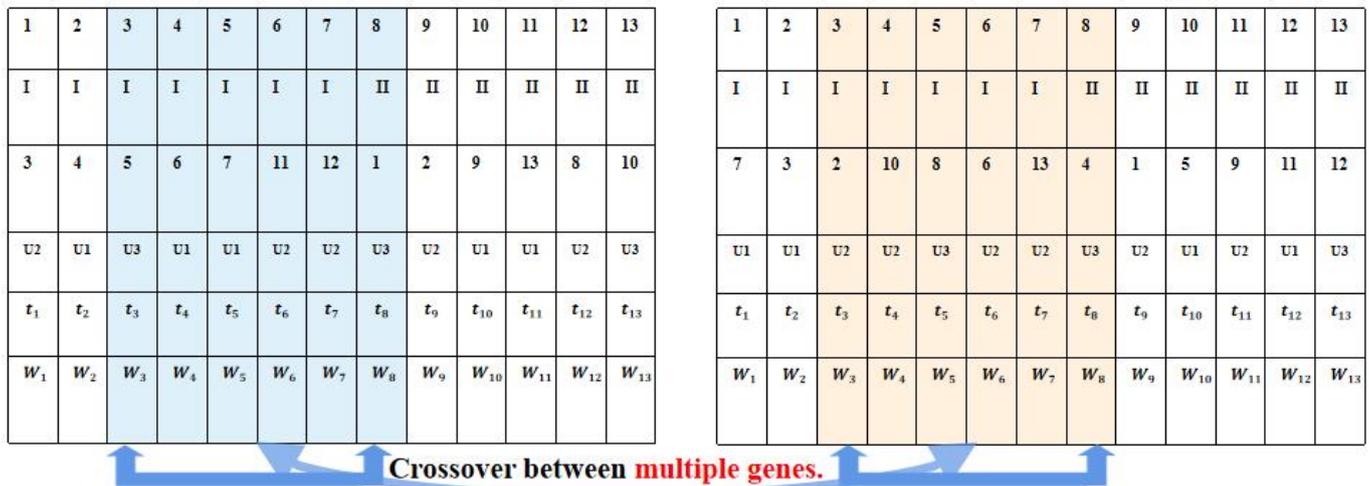


Figure 4. Figure of chromosome crossover.

Phase 2, mutation process: In Figure 5, chromosome variation encompasses single-point mutation and hybrid mutation. The single-point mutation alters the UAV order number, while hybrid-point mutation modifies execution order. If the probability of single-point mutation and hybrid-point mutation set as a constant,  $P_m$ , the risk of losing excellent individuals increases. Therefore, the probabilities of single-point mutation and hybrid-points mutation are customized for each scheme based on its fitness value,  $F$ , in comparison to the average fitness value,  $\bar{F}$ , which are defined as

$$F = \frac{1}{\sum_{i=1}^{N_{U}} t_i}, \quad (15)$$

$$\bar{F} = \frac{1}{\sum_{i=1}^{N_{chrom}} F}.$$

<b>Task Number</b>	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Task Type</b>	I	I	I	I	I	I	I	II	II	II	II	II	II
<b>Execution Order</b>	3	4	5	6	7	11	12	1	2	9	13	8	10
<b>UAV Order Number</b>	U2	U1	U3	U1	U1	U2	U2	U3	U2	U1	U1	U2	U3
<b>Execution Time</b>	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$
<b>Energy Consumption</b>	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$	$W_7$	$W_8$	$W_9$	$W_{10}$	$W_{11}$	$W_{12}$	$W_{13}$

**Hybrid mutation changes execution order.**

Figure 5. Figure of chromosome variation.

To accelerate the retention of superior schemes and the elimination of inferior schemes, the mutation probability of the current scheme is calculated as

$$P_m = \begin{cases} (P_{m1} - P_{m2}) \cdot \ln(A \cdot F + B) + P_{m1}, & F \geq \bar{F}, \\ P_{m1}, & F < \bar{F}, \end{cases} \quad (16)$$

where  $P_m$ ,  $P_{m1}$ , and  $P_{m2}$  represent the current, minimum, and maximum values of mutation probability.  $A$  and  $B$  are functions designed to balance the relationship between mutation probability and fitness function, which are defined as

$$A = \frac{e - 1}{e(\bar{F} - F_{max})}, \quad (17)$$

$$B = \frac{\bar{F} - eF_{max}}{e(\bar{F} - F_{max})}, \quad (18)$$

where  $F$ ,  $F_{max}$ , and  $\bar{F}$  are the fitness function's current, maximum, and average values.  $e$  is Euler constant. Specifically, when calculating the fitness function value  $F$  for the current assignment, if  $F \geq \bar{F}$ , mutation probability should be set as the first case of (16). Otherwise, the mutation probability of the current scheme should be set to the maximum value of the mutation probability.

Phase 3, reinsertion process: The reinsertion process selects the chromosome with the best performance and inserts it into the next iteration. It ensures the subsequent generation maintains or improves upon the quality of the solution to the problem.

Overall, the process of the multigene algorithm has been organized and shown in Algorithm 1.

---

**Algorithm 1** Multigene algorithm for solving task allocation problem

---

**Input:**

$$U = \{1, 2, \dots, Nu\}, S = \{1, 2, \dots, Ns\}, Na, Nh;$$

**Output:**

- task allocation scheme with the highest  $F$ ;
  - 1: Initialize  $N_{chrom}$  schemes in initial chromosome;
  - 2: Calculate  $F$  of each scheme of the initial chromosome;
  - 3: Set  $F_{max}$  and calculate  $\bar{F}$  of the initial chromosome;
  - 4:  $G_e$  = number of iterations;
  - 5: Set  $P_c, P_m, P_{m1}, P_{m2}$ ;
  - 6: **while**  $i \leq G_e$  **do**
  - 7:   Crossover process in Phase 1;
  - 8:   Mutation Process in Phase 2;
  - 9:   Reinsertion Process in Phase 3;
  - 10:   Select the scheme with the highest  $F$  in this generation;
  - 11:    $i = i + 1$ ;
  - 12: **end while**
  - 13: Output the scheme with the highest  $F$  in the last generation.
- 

2.3.2. Improved Anti-Collision RRT\* (IAC-RRT\*) Algorithm for Route Planning Problem

In this subsection, we proposed the IAC-RRT\* algorithm, a robust and efficient algorithm for route planning designed to address the challenge of identifying feasible pathways within the UAM operation environment. The principles and concrete steps are as follows.

The first step involves map initialization and updating of task sites. The urban airspace area  $Q$  is divided into available area,  $Q_n$ , “flexible” area,  $Q_{st}$ , obstacle area,  $Q_{ob}$  in Section 2.2.2. And the “flexible” area is controlled to switch by  $\gamma_{st}$ ; the relationship is given as

$$Q = Q_n + Q_{st} + Q_{ob}, \tag{19}$$

then the start-point,  $q_{start}$  and the end-point,  $q_{end}$  are positioned within the domains  $Q_n$  or  $Q_{st}$ . The task is deemed unsuccessful if these points fall outside the accessible regions, represented as

$$q_{start}, q_{end} \in Q_n \cup Q_{st}. \tag{20}$$

Subsequently, a direct line connecting  $q_{start}$  and  $q_{end}$  is established, designated as the normal line  $R_1$ , which aids in assessing exploration node efficiency.

The second step entails the construction of the tree structure. The algorithm explores the surrounding area with the radius  $R_e$  from the root node  $q_{start}$  and creates the newly found node  $q_{new}$ . The search and explore principle can be given that the smaller the included angle,  $\theta_{ne}(i)$ , between the normal direction and explore direction, the higher the chance of the node,  $p_{ne}(i)$  to be chosen.

$$p_{ne}(i) = \frac{1}{\pi/\theta_{ne}(i)}. \tag{21}$$

The algorithm determines whether the newly generated node  $q_{new}$  belongs to  $Q_n \cup Q_{st}$ . If  $q_{new} \in Q_{ob}$ , the node will be discarded at once. Otherwise, the time transformation factor  $\gamma_{st}$  will be used to determine whether the “feasible” area  $Q_{st}$  would allow flight action.

Then, IAC-RRT\* algorithm reselects neighbor node  $q_{near}$  of  $q_{new}$  within the radius of  $R$ , which is defined as

$$q_{near} \in \{dis(q_{near}, q_{new}) < R\}, \tag{22}$$

By comparing the distance from  $q_{near}$  to  $q_{start}$  and  $q_{new}$ , the IAC-RRT\* algorithm decides whether to change the parent node of  $q_{new}$ .

$$\begin{aligned} \min D_{new} &= dis(q_{start}, q_{near}) + dis(q_{near}, q_{new}), \\ s.t. q_{near} &\in \{dis(q_{near}, q_{new}) < R\}, \end{aligned} \quad (23)$$

the optimal reselected neighbor, denoted as  $q_{near}^*$ , becomes the new parent node of  $q_{new}$ . The rewiring process aims to reduce the path length across the global tree nodes, i.e., calculating the distance  $D_{near}$  between each neighbor node and  $q_{new}$ , and selecting the minimal distance to  $q_{start}$ . The mathematical representation of the rewiring process is as follows:

$$\begin{aligned} \min D_{near} &= dis(q_{start}, q_{near}), \\ s.t. q_{near} &\in \{dis(q_{near}, q_{new}) < R\}, \\ q_{new} &\in \{parent\ nodes\ of\ q_{near}\}, \end{aligned} \quad (24)$$

These steps are reiterated until the distance between the newly generated node  $q_{new}$  and  $q_{end}$  is less than  $R$ . IAC-RRT\* algorithm is described in Algorithm 2.

---

#### Algorithm 2 IAC-RRT\* algorithm for solving route planning problem

---

**Input:**

$Q_n, Q_{st}, Q_{ob}, q_{start}, q_{end}, n$  iteration times;

**Output:**

Route from  $q_{start}$  to  $q_{end}$ ;

- 1: Search from  $q_{start}$ ;
- 2: **while** Iterate  $n$  times **do**
- 3:   **while**  $dis(q_{new}, q_{end}) > R$  **do**
- 4:     Generate  $q_{new}$  within the radius  $R$ ;
- 5:     **if**  $q_{new} \in Q_{ob}$  **do**
- 6:       discard  $q_{new}$ ;
- 7:     **else**  $q_{new} \in Q_n \cup \gamma_{st} Q_{st}$  **do**
- 8:       reselect parent node for  $q_{new}$ ;
- 9:        $\min D_{new} = dis(q_{start}, q_{near}) + dis(q_{near}, q_{new})$ ;
- 10:       rewire with  $q_{near}$  as the parent node;
- 11:        $\min D_{near} = dis(q_{start}, q_{near})$ ;
- 12:     **end if**
- 13:   **end while**
- 14:   Backtrack along  $q_{new}$  from  $q_{end}$  to  $q_{start}$ , i.e., find the route;
- 15:   Select the route with minimal distance from each iteration;
- 16: **end while**
- 17: Output the selected route from  $q_{start}$  to  $q_{end}$ .

---

### 3. Results

To evaluate the performance of the multigene algorithm and IAC-RRT\* algorithm, we conduct simulations and compare the results with a few traditional task allocation and route planning algorithms. For all simulations without particular illustration, the parameters are fixed: we set the number of UAVs to 4, the number of tasks to 10, the number of air taxi tasks to 7, and the number of hover and monitor tasks to 3. The details of the simulations are presented as follows.

#### 3.1. Simulation Results of Multigene Algorithm

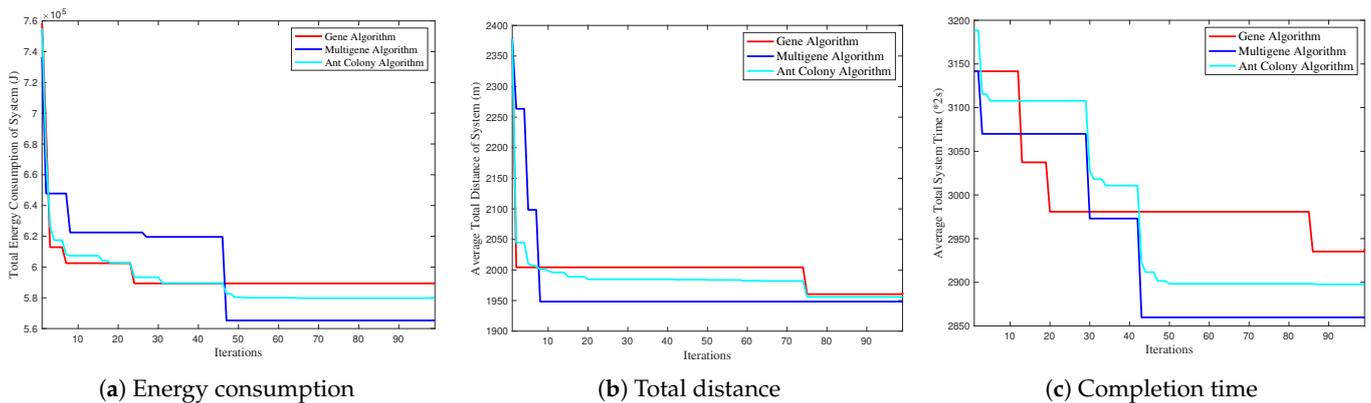
In this subsection, we employ the gene and ant colony algorithms as comparative benchmarks. To assess the convergence probability, a series of experiments are conducted to search the most energy-efficient and quickest task allocation schemes.

Figure 6 limits the maximum iterations to 100 to ensure consistent experimental conditions, thereby evaluating the search performance of the multigene algorithm. Figure 6a–c

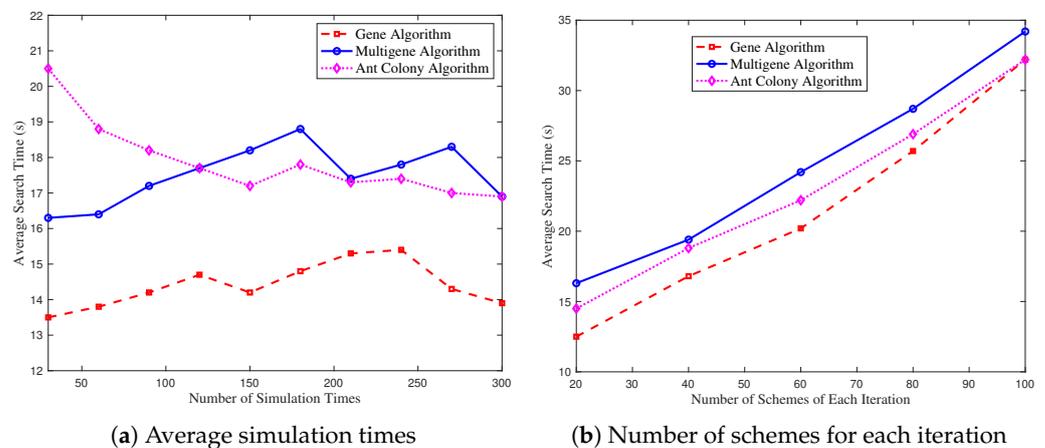
demonstrate that our method significantly surpasses the gene and ant colony algorithms regarding search speed and optimal search outcomes, focusing on energy consumption, distance, and completion time. This superiority is attributed to the multigene algorithm’s targeted crossover and mutation, unlike the random approach of the other algorithms. Thus, we yield that heterogeneity-oriented crossover and mutation enhance the chance of discovering the optimal solution.

Figure 7a depicts the relationship between the average system search time and the average number of simulation times. We can yield that with the increase in an average number of simulation times, the trend of average search time reveals no direct correlation. Additionally, the ant colony algorithm exhibits marginally inferior performance at lower simulation times. Figure 7b examines the association between average search time and the number of schemes of each iteration. A consistent trend is observed from Figure 7b. Given its superior search capabilities and outcomes, the extra search time invested in the multigene algorithm is deemed cost-effective.

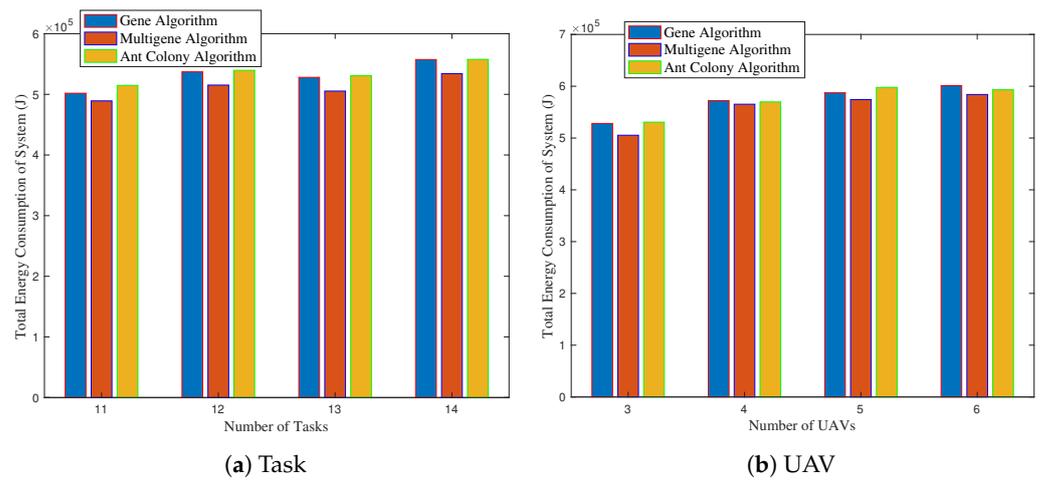
Figure 8a,b present the system’s total energy consumption by varying tasks and UAV numbers. As depicted in Figure 8a, the number of UAVs is set to 10. Additionally, we also set 11 tasks with 7 single-UAV tasks and 2 multi-UAV tasks, 12 tasks with 8 single-UAV tasks and 2 multi-UAV tasks, 13 tasks with 7 single-UAV tasks and 3 multi-UAV tasks, and 14 tasks with 8 single-UAV tasks and 3 multi-UAV tasks. We can conclude that total energy consumption increases with the number of tasks. Compared to the single-UAV tasks, the multi-UAV tasks will cause more significant growth in energy consumption. It can be explained by inherent complexity of heterogeneous tasks.



**Figure 6.** System performance of multigene compared with iterations. (a) Total energy consumption with iterations. (b) Average distance with iterations. (c) Average completion time with iterations.



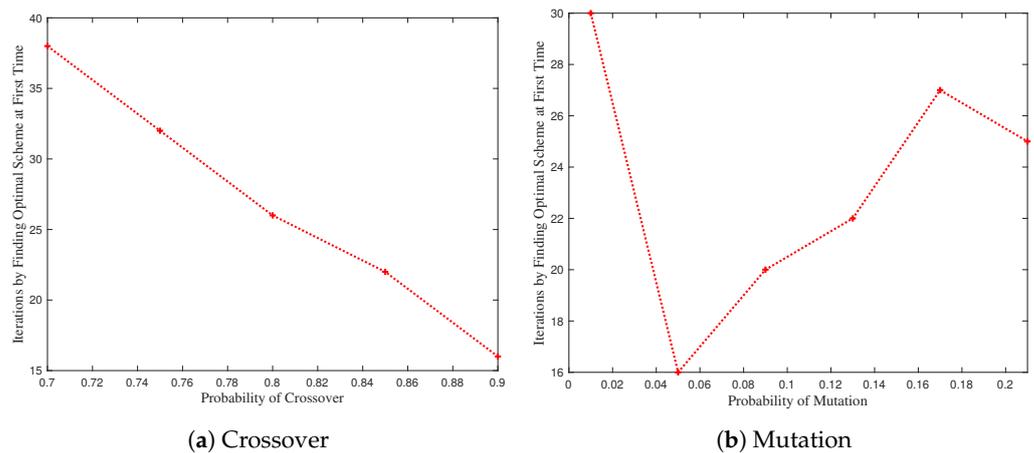
**Figure 7.** System performance comparison of three algorithms. (a) Average search time by varying average simulation times. (b) Average search time by varying number of schemes for each iteration.



**Figure 8.** Energy consumption by varying number of tasks and UAVs. (a) Total energy consumption by varying number of tasks. (b) Total energy consumption by varying number of UAVs.

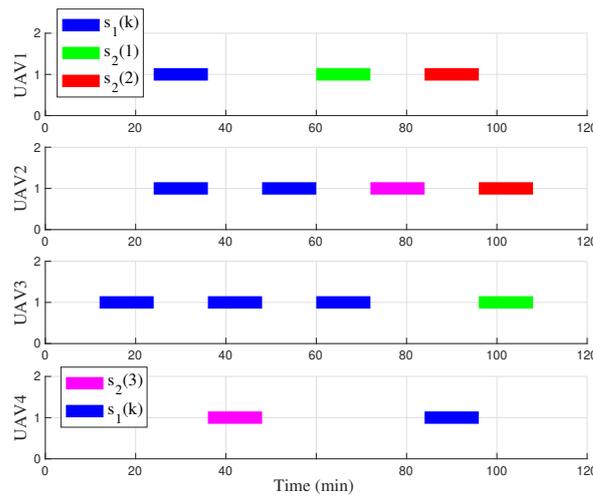
Figure 8b shows the system’s energy consumption variation by changing the number of UAVs. We can see that increasing the number of UAVs can not permanently reduce energy consumption. Notably, excessive UAVs can lead to inefficiencies and increased energy use. An inadequate number of UAVs can cause extreme round trips from charging stations to task sites.

Figure 9 investigates the impact of crossover and mutation probabilities on the iterations required to identify the optimal scheme. We set schemes with system time under 6000s as the optimal schemes. Figure 9a reveals an inverse relationship between crossover probability and required iterations, as a higher crossover probability enhances the chance of identifying more suitable task assignments. On the contrary, the iterations to find the optimal scheme vary with the likelihood of mutation Figure 9b. An increase in mutation probability risks losing excellent schemes, and an excessively low mutation probability may fail to generate sufficient options for optimal selection.



**Figure 9.** The number of iterations to find the optimal scheme by varying probabilities of crossover and mutation. (a) The number of iterations to find the optimal scheme by varying probability of crossover. (b) The number of iterations to find the optimal scheme by varying probability of mutation.

Figure 10 displays the UAV task schedule for an optimal task assignment. There are seven single-UAV tasks  $s_1(k)$  and three multi-UAV tasks, color-coded for differentiation,  $s_2(1)$ ,  $s_2(2)$ ,  $s_2(3)$ . It can be observed that the optimal scheme balances fairness and effectiveness. Each UAV is assigned a similar number of tasks. However, the UAV to finish particularly challenging or isolated tasks, i.e., UAV4, may undertake fewer additional tasks.



**Figure 10.** Task time schedule of four UAVs.

### 3.2. Simulation Results of IAC-RRT\* Algorithm

This subsection explores the performance of the IAC-RRT\* algorithm. We simulated 4 UAV-assisted route planning scenarios in an urban environment. The environmental map is modeled as a  $2000 \times 2000 \times 100 \text{ m}^3$  cube, which is divided into available areas,  $Q_n$ , “flexible” areas,  $Q_{st}$ , obstacle areas,  $Q_{ob}$ . Flexible areas  $Q_{st}$ , modeled as green box regions, are controlled by a time-switching factor  $\gamma_{st}$ . Obstacles  $Q_{ob}$  are modeled as randomly located hemispherical regions. All UAVs start and end at  $(0, 0, 5)$ . Task sites which have seven single-UAV tasks and three multi-UAV tasks are randomly distributed in  $Q_n$ . The radius of exploration sets as 5 m. The RRT algorithm serves as a benchmark for comparison.

Figure 11 represents the searching time performance of three algorithms over different building densities. To test the obstacle avoidance performance of the IAC-RRT\* algorithm, RRT and RRT\* algorithms work as comparison groups [25]. Accordingly, the gap is not distinct at low building density because random search methods may not easily bump into obstacles. However, searching methods along the normal direction have better effects at high building densities. Additionally, simulation results prove that the IAC-RRT\* algorithm performs better than the RRT algorithm and RRT\* algorithm in a complex obstacle environment. Figure 12 portrays the IAC-RRT\* algorithm’s route search process. Figure 12a depicts the initial single route of the IAC-RRT\* algorithm by finishing three tasks. Consequently, Figure 12b defines the search process of the IAC-RRT\* algorithm. It can be obtained that exploration detection of the search process prioritizes the normal line  $R1$  to increase the target discovery chance. Figure 12c depicts the final planned route for all vehicles of the IAC-RRT\* algorithm. We can affirm the IAC-RRT\* algorithm’s proficiency in task completion.

Figure 13 compares the system performance of the IAC-RRT\* algorithm and RRT algorithm. Figure 13a shows the IAC-RRT\* algorithm’s faster convergence time by 13% compared to the RRT algorithm. Then, Figure 13b depicts the search distance performance of the IAC-RRT\* algorithm. The variation in the number of average simulations between 5 and 15 owes to too few experiments. It can also be obtained that converge speed of the IAC-RRT\* algorithm is quicker 12% than the RRT algorithm. Figure 13c demonstrates times passing through  $Q_{st}$  of the IAC-RRT\* algorithm and RRT algorithm. The time-switching factor  $\gamma_{st}$  reduces 85% per cent passengers through  $Q_{st}$ . Remaining points occurs in  $Q_{st}$  of IAC-RRT\* algorithm when  $\gamma_{st} = 1$ .

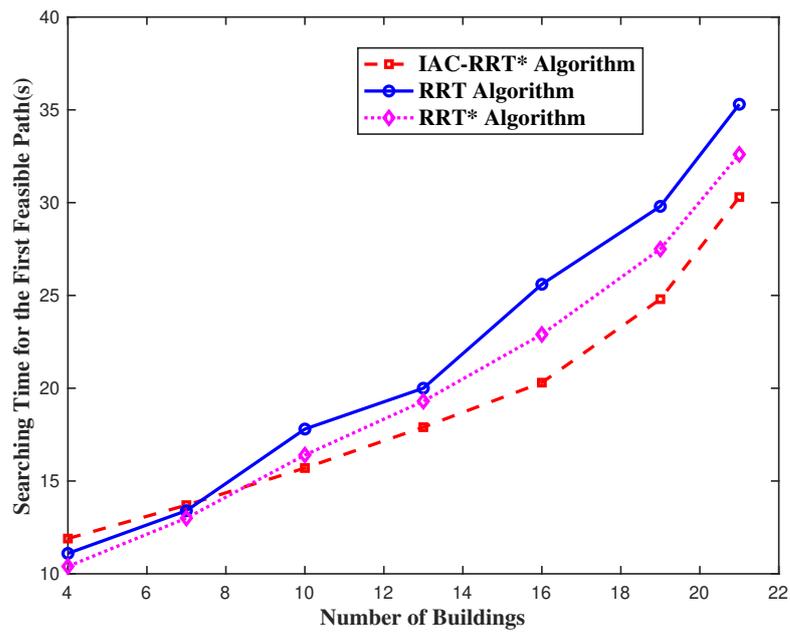


Figure 11. Searching time performance over different number of buildings.

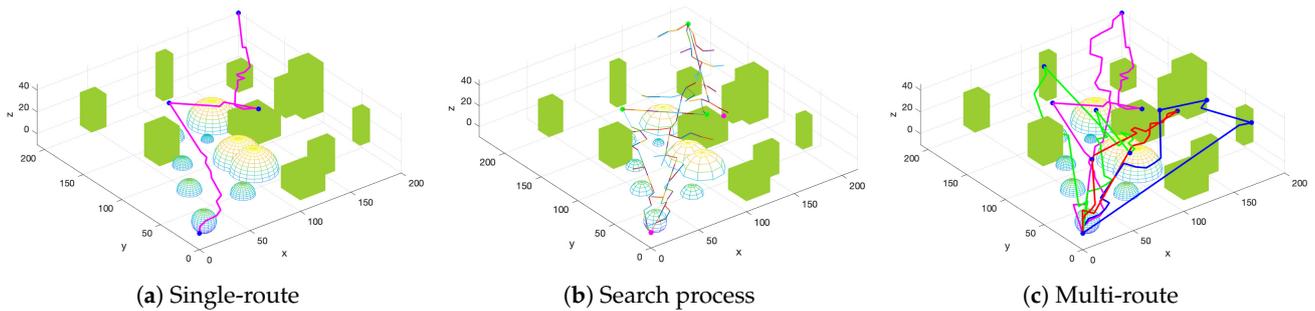


Figure 12. Routing planning map of IAC-RRT\* algorithm in urban environment. (a) Single route found by IAC-RRT\* algorithm. (b) Search process of IAC-RRT\* algorithm. (c) Final planning route for all vehicles of IAC-RRT\* algorithm.

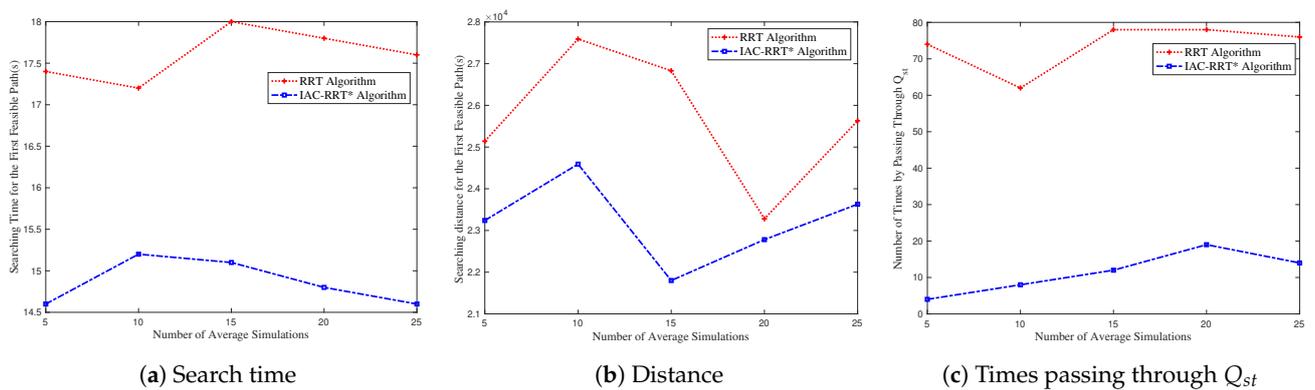


Figure 13. System performance of IAC-RRT\* algorithm compared with RRT algorithm. (a) Searching time performance of IAC-RRT\* algorithm compared with RRT algorithm. (b) The optimal feasible distance of IAC-RRT\* algorithm compared with RRT algorithm. (c) Times passing through  $Q_{st}$  of IAC-RRT\* algorithm compared with RRT algorithm.

#### 4. Conclusions

In this paper, we propose the multigene algorithm and IAC-RRT\* algorithm to solve UAV task allocation and route planning problems in the UAM scenario. Also, we studied the UAV power model and task model, and derived expressions of task allocation problems. Simulation results indicate that our algorithms have better numerical stability, which results in better solutions to the problem. The convergence speed of the proposed multigene algorithm is 20 percent faster than the traditional gene algorithm. Additionally, the IAC-RRT\* algorithm can handle airspace control issues with robust performance. It can converge quicker 12% than the RRT algorithm.

**Author Contributions:** Conceptualization and methodology, Q.Z. and H.F.; software and validation, H.F.; formal analysis and investigation, Q.Z. and H.F.; writing, H.F.; visualization, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

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

#### References

1. Hamid, M.; Ismail, G.; Kemal, A.; Selcuk U.; Abdullah, K.; Adem, T. UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Commun. Mag.* **2017**, *5*, 22–28.
2. Cheng, L.; Wen, Q.; Yan, L.; Long, H.; Peng, W. Overview of traffic management of urban air mobility (UAM) with eVTOL aircraft. *J. Traffic Transp. Eng.* **2020**, *5*, 35–54.
3. Andrey, G.; Young, I. Intelligent UAV in smart cities using IoT. In Proceedings of 2016 16th International Conference on Control, Automation and Systems (ICCAS), Gyeongju, Republic of Korea, 16–19 October 2016; pp. 207–210.
4. Jeff, H.; Nikhil, G. *Fast-Forwarding to a Future of on-Demand Urban Air Transportation*; Uber Elevator: San Francisco, CA, USA, 2016; pp. 23–26.
5. Wan, Z.; Qing, M.; Paul, C. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. *IEEE Trans. Cybern.* **2015**, *4*, 902–915.
6. Hao, C.; Ying, N.; Yi, Y. Multi-UAV reconnaissance task assignment for heterogeneous targets based on modified symbiotic organisms search algorithm. *Sensors* **2019**, *19*, 734.
7. Fang, Y.; Jie, C.; Yuan, T.; Tao, J. Cooperative task assignment of a heterogeneous multi-UAV system using an adaptive genetic algorithm. *Electronics* **2020**, *9*, 687.
8. Oh, H.; Kim, S.; Tsourdos, A. Cooperative road-network search planning of multiple UAVs using Dubins paths. In Proceedings of the AIAA, Long Beach, CA, USA, 27–29 September 2011; p. 6386.
9. Wang, C.; Mu, D.; Zhao, F. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup—Delivery and time window. *Comput. Ind. Eng.* **2015**, *5*, 111–122. [[CrossRef](#)]
10. Arsie, A.; Savla, K.; Frazzoli, E. Efficient routing algorithms for multiple vehicles with no explicit communications. *IEEE Trans. Autom. Control* **2009**, *10*, 2302–2317. [[CrossRef](#)]
11. Shima, T.; Rasmussen, S. J.; Sparks, A. G. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Comput. Oper. Res.* **2006**, *11*, 3252–3269. [[CrossRef](#)]
12. Edison, E.; Shima, T. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Comput. Oper. Res.* **2011**, *1*, 340–356. [[CrossRef](#)]
13. Jia, Z.; Yu, J.; Ai, X. Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm. *Aerosp. Sci. Technol.* **2018**, *6*, 112–125. [[CrossRef](#)]
14. Deng, Q.; Yu, J.; Wang, N. Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes. *Chin. J. Aeronaut.* **2013**, *5*, 1238–1250. [[CrossRef](#)]
15. Deng, Q.; Yu, J.; Mei, Y. Deadlock-free consecutive task assignment of multiple heterogeneous unmanned aerial vehicles. *J. Aircr.* **2014**, *2*, 596–605. [[CrossRef](#)]
16. Upadhyay, S. Ratnoo, A. Smooth path planning for unmanned aerial vehicles with airspace restrictions. *J. Guid. Control Dyn.* **2017**, *7*, 1596–1612. [[CrossRef](#)]
17. Hui, H.; Ke, X.; Gang, Q.; Qiang, N.; Ping, F.; Khaled, L. AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks. *IEEE Internet Things* **2021**, *1*, 1211–1223. [[CrossRef](#)]
18. Vincent, R.; Mohammed T.; Gilles, L. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Inform.* **2013**, *2*, 132–141.

19. Han, C.; Luc, B.; Jonathan, H. Path planning for autonomous vehicles in unknown semistructured environments. *IEEE Trans. Robot.* **2009**, *8*, 912–926.
20. Haghghi, H.; Delahaye, D.; Asadi, D. Performance-based emergency landing trajectory planning applying meta-heuristic and Dubins paths. *Appl. Soft Comput.* **2022**, *117*, 108453. [[CrossRef](#)]
21. Dolgov, D.; Thrun, S.; Montemerlo, M. Consensus-based decentralized auctions for robust task allocation. *Int. J. Robot. Res.* **2010**, *5*, 485–501. [[CrossRef](#)]
22. Yang, M.L.; Li, N. Study on mobile robot path planning based on improved A\* algorithm. *Mech. Sci. Technol. Aerosp. Eng.* **2022**, *5*, 795–800.
23. Wang, D. Indoor mobile-robot path planning based on an improved A\* algorithm. *J. Tsinghua Univ.* **2012**, *8*, 1085–1089.
24. laValle, S.; Kuffner, J. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2001**, *5*, 378–400. [[CrossRef](#)]
25. Chen, Z.; Li, M.; Shao, X. Obstacle avoidance path planning of bridge crane based on improved RRT algorithm. *J. Syst. Simul.* **2021**, *8*, 1832–1838.
26. Hsu, D.; Latombe, J.; Kurniawati, H. On the probabilistic foundations of probabilistic roadmap planning. *Int. J. Robot. Res.* **2006**, *7*, 83–97. [[CrossRef](#)]
27. Liu, Y.; Zhao, H.; Liu, X. An improved RRT based obstacle avoidance path planning algorithm for industrial robot. *Inf. Control* **2021**, *2*, 235–246.
28. Ruan, X.; Zhou, J.; Zhang, J. Robot goal guide RRT path planning based on sub-target search. *Control Decis.* **2020**, *10*, 2543–2548.
29. Yu, M.; Luo, J.; Wang, M. Coordinated path planning by integrating improved RRT\* and quartic spline. *Chin. J. Theor. Appl. Mech.* **2020**, *4*, 1024–1034.
30. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *7*, 846–894. [[CrossRef](#)]
31. Luo, S.; Liu, S.; Zhang, B. Path planning algorithm based on Gb informed RRT\* with heuristic bias. In Proceedings of 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 137–141.
32. Zhang, Y.; Zuo, Y.; Wu, G.; Research on path planning based on improved Informed-RRT\* algorithm. *Modul. Mach. Tool Autom. Manuf. Tech.* **2020**, *7*, 21–25.
33. Kuffner, J.; LaValle, S. RRT-Connect: An efficient approach to single-query path planning. In Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000; pp. 71–75.
34. Han, K.; Cheng, W. Path planning of robot arm based on improved RRT algorithm. *Comput. Appl. Softw.* **2022**, *3*, 260–265.
35. Zhang, D.; Zhen, Z.; Chen, Y. Collaborative path planning based on improved RRT-Connect algorithm. *Electron. Opt. Control* **2021**, *9*, 25–29.
36. Wang, K.; Huang, B.; Zeng, G. Faster path planning based on improved RRT-Connect algorithm. *J. Wuhan Univ.* **2019**, *3*, 283–289.
37. Liu, Z.; Sengupta, R.; Kurzhanskiy, A. A power consumption model for multi-rotor small unmanned aircraft systems. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 4–7 June 2017; pp. 310–315.
38. Haghghi, H.; Asadi, D.; Delahaye, D. Multi-Objective Cooperated Path Planning of Multiple Unmanned Aerial Vehicles Based on Revisit Time. *J. Aerosp. Inf. Syst.* **2021**, *18*, 919–932. [[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.