Research Article
A User-Priority-Driven Multi-UAV Cooperative Reconnaissance Strategy

Zeyuan Liu, Cuntao Liu, Wendong Zhao, and Aijing Li
College of Communications Engineering, Army Engineering University of PLA, Nanjing 210042, China
Correspondence should be addressed to Wendong Zhao; zhaowendong123@aliyun.com
Received 17 June 2021; Revised 29 August 2021; Accepted 7 September 2021; Published 11 October 2021

1. Introduction

Unmanned aerial vehicles (UAVs) have been widely used as mobile carriers to store information in many fields, such as military reconnaissance and disaster relief, due to their mobility and flexibility [1–3]. Usually, users need to obtain information on distant targets in a short time for their decision. In this case, using UAVs to reconnoiter distant targets and deliver information back to users is an excellent strategy [4]. Due to the limited capability of a single UAV, multiple UAVs are usually used to conduct cooperative reconnaissance to deal with complex tasks [5], e.g., providing reconnaissance services simultaneously for multiple users with different requirements or conducting a reconnaissance process with dynamic and uncertain task flows.

Unlike reconnaissance scenarios where only a single user exists [6, 7], we consider the scenario where multiple users need different target information, and all expect to obtain this information in a short time. UAVs need to meet their differentiated demands as best as possible while considering service fairness for different users. In addition, users may have different statuses or perform tasks with different importance; thus, we need to consider that they have different priorities. The higher the user’s priority, the faster the information acquisition should be guaranteed. To accomplish this, effective task assignment and path planning strategies are essential. Current studies on multi-UAV cooperative reconnaissance with multiple users usually ignore the differences between users, i.e., users are supposed to obtain the same target’s information or have the same priorities [8–11]. The other work did not consider different information acquisition time of different priority users [12–20], or their information backhaul strategies did not balance the demand of different priority users well [21–25]. The impact of differences among users is not considered in task assignment and path planning. These above studies’ path planning...
methods and evaluation criteria do not work well in scenarios where users’ priorities and demands are different.

In this paper, we consider a reconnaissance scenario where multiple UAVs are used to provide services cooperatively for multiple users with different priorities. Taking users’ diverse priorities into account, a multiuser satisfaction model and a user-priority-driven reconnaissance strategy are proposed. Then, the cooperative path planning problem of multiple UAVs is considered and optimized with the goal of maximizing multiuser satisfaction. Our main contributions are summarized as follows:

(i) A priority-driven multiuser satisfaction model is proposed, where users’ satisfaction is described and quantified according to their different priorities, information acquisition demands, and the time they obtain their desired information. It comprehensively considers the information acquisition time of each user and achieves a good trade-off between the needs of high priority and low priority users.

(ii) A batchwise information backhaul strategy is proposed to ensure high priority users’ fast information acquisition while avoiding excessive delay in low priority users’ information acquisition. In each batch, by adopting a cooperative consultative and information sharing mechanism, one of the task execution UAVs is selected as a data ferry to carry the information collected by all UAVs back to users.

(iii) The above reconnaissance process with a batchwise information backhaul strategy is formulated as a cooperative path planning problem, where the optimization objective is maximizing users’ total satisfaction. To solve this problem effectively, a batchwise information transmission-based path planning algorithm (BITPP) is proposed. Simulation results show the effectiveness of our proposed BITPP algorithm in ensuring lower information acquisition time and higher user satisfaction for users with higher priorities.

2. Related Work

To date, much research has been done on the path planning problem in cooperative reconnaissance, which is beneficial for shortening the information acquisition time as well as reducing UAVs’ energy consumption [12–21]. Among them, to ensure users’ fast information acquisition, some studies have focused on minimizing the maximum reconnaissance time of UAVs [10–12]. However, in these studies, the difference in the information acquisition demands of users was not well considered, and users’ information acquisition time was not evaluated comprehensively regarding their different priorities. When considering tasks’ different priorities, several studies used the priorities as different weights to maximize the task metrics’ weighted results [13–15]. Several studies designed reward values and penalty values pertaining different tasks according to their priorities to maximize the final reward [16, 17]. However, these means cannot guarantee that higher priority tasks can be completed before low priority tasks. If an evaluation criterion was to be that UAVs completed their tasks sequentially according to the order of their priorities, it could cause too much delay for the completion of low priority tasks, such as [18], or the low priority tasks will be given up, such as [19], which is not fair to low priority users.

In addition, different information backhaul strategies have been proposed and adopted in several studies. Among them, Ref. [20–22] favored a one-time strategy of information backhaul, where UAVs transmitted information to users after finishing the reconnaissance process of all targets. Under this condition, higher priority users cannot obtain their desired information until the UAV assigned to provide service for them has reconnoitered all targets, which will significantly prolong the time for their obtaining information. To address this problem, Ref. [23] proposed a multi-round information backhaul strategy, where UAVs conducted multiple trips between the reconnaissance area and users to realize data unloading. During each trip, multiple UAVs cooperatively performed reconnaissance a group of targets and deliver corresponding information to the users who need it. Although higher priority users can obtain their desired information much more quickly, users with lower priorities will experience much more delay in their information acquisition, especially when users are far away from the reconnaissance area or too many round trips are needed because of several factors, e.g., the number of available UAVs is limited, the distribution of target points is overly discrete, or the number of users is large, and the information acquisition delay for lower priority users may be too long to be acceptable. Ref. [24] adopted an immediate information backhaul strategy by using UAVs as fixed relays to construct available communication links between the reconnaissance area and users. However, when facing situations where users are far away from the reconnaissance area or the number of users is large, too many UAVs are needed to maintain effective communication links, which will result in a huge reconnaissance cost.

3. System Model and Problem Formulation

3.1. Scenario Description. We consider a scenario in which UAVs need to reconnoiter targets and distribute information to users in the user area. The set of users and targets is denoted as \( \mathcal{M} = \{1, \cdots M\} \), \( m \in \mathcal{M} \), and \( \mathcal{K} = \{1, \cdots K\} \), \( k \in \mathcal{K} \). Each user is interested in multiple target points, and the information requirement relationship of users with targets is denoted as a binary variable \( r^k_m \), where \( r^k_m = 1 \) represents that \( m \) needs the information of \( k \), and \( r^k_m = 0 \) represents others. The number of UAVs is \( N \), and this set is denoted as \( \mathcal{N} = \{1, \cdots N\} \), \( n \in \mathcal{N} \). We assumed that UAVs need to be near the user when they transmit information. In addition, it is assumed that the UAVs involved in reconnaissance have sufficient flight time to complete the assigned targets’ reconnaissance.

All users need to obtain information as soon as possible, and each user gives a degree of satisfaction for the UAV reconnaissance service according to their information needs.
acquisition time. The faster a user obtains the required information, the more satisfied it is with the UAVs’ reconnaissance service. Each user has a priority value to measure the importance of the user and determine the order in which the information is obtained. Users with higher priority need to obtain information faster than users with lower priority. \( \lambda_m \) represents the priority value of \( m \). Although users all expect a short information acquisition time, UAVs cannot ensure that each user obtains the same high service quality under a limited number. Thus, we pursue maximizing the overall user satisfaction and propose a batchwise information backhaul strategy to improve multiuser satisfaction.

3.2. Priority-Based Multiuser Satisfaction Model. We propose a priority-based multiuser satisfaction model to measure the information acquisition time of users with different priorities and measure the order in which the user obtains the information. In this model, \( s_m \) represents the satisfaction of user \( m \), and \( t_{m,\text{end}} \) represents the time when \( m \) obtains all information it needs. The \( s_m \) will change with the size of time \( t_{m,\text{end}} \). We assumed that each user has a uniform expected time for obtaining the needed information and is denoted as \( T_{\text{expect}} \). We denoted a satisfaction constant \( \gamma \) to represent one user obtaining the required information at time \( T_{\text{expect}} \). When \( t_{m,\text{end}} \) is earlier than \( T_{\text{expect}} \), \( s_m \) will equal \( \gamma \), and when \( t_{m,\text{end}} \) is later than \( T_{\text{expect}} \), \( s_m \) will decrease over time. In addition, each user has a maximum tolerance time \( T_{\text{deadline}} \) and user satisfaction becomes 0 when \( t_{m,\text{end}} \) is later than \( T_{\text{deadline}} \). Thus, \( s_m \) is calculated by equation (1).

\[
\begin{align*}
\delta_m & = \begin{cases} 
\gamma & \text{if } t_{m,\text{end}} < T_{\text{expect}} \\
\frac{\gamma(T_{\text{deadline}} - t_{m,\text{end}})}{T_{\text{deadline}} - T_{\text{expect}}} & \text{if } T_{\text{expect}} < t_{m,\text{end}} < T_{\text{deadline}} \\
0 & \text{if } t_{m,\text{end}} \geq T_{\text{deadline}}
\end{cases} \\
& = \frac{T_{\text{expect}} - t_{m,\text{end}}}{T_{\text{deadline}} - t_{m,\text{end}}} 
\end{align*}
\]

Multiuser satisfaction is represented by \( \xi \), which is determined following two principles. The first is that a user must obtain information earlier than a user with lower priority. The second is that the impact of higher priority user satisfaction on \( \xi \) must be greater than the impact of lower priority user satisfaction. We denoted variable \( \delta \) to measure whether users obtain information following the order of priority. When the user obtains information later than the user with a lower priority than it, \( \delta \) takes the value of negative infinity to represent the penalty for violating the user priority order, which is calculated as follows:

\[
\delta = \begin{cases} 
-\infty & \text{if } (t_{m,\text{end}} - t_{j,\text{end}})(\lambda_m - \lambda_j) > 0 \\
0 & \text{other}
\end{cases}
\]

Subsequently, each user’s priority values are normalized, and then the processing results are used as weighting coefficients to weigh the satisfaction of each user. The equation of \( \xi \) is as follows:

\[
\xi = \delta + \sum_{m=1}^{M} \left( \frac{\lambda_m s_m}{\sum_{m=1}^{M} \lambda_m} \right). \tag{3}
\]

3.3. Batchwise Information Backhaul Strategy. This section proposes a batchwise information backhaul strategy; that is, UAVs leave the reconnaissance area in batches to provide information for users. We divide the entire reconnaissance process into \( L \) batches. In each batch, the UAV serves one or more users. They first come in the reconnaissance phase and reconnoiter the targets in which these users are interested. \( \mathcal{K}_l(\mathcal{K}_l \subset \mathcal{K}) \) represents the target set that UAVs need to reconnoiter in the \( l \)th batch. \( \mathcal{K}_l \) represents the target set assigned to \( n \) in the \( l \)th batch. \( \mathcal{M}_l \) represents the user set that UAVs need to serve in the \( l \)th batch. \( \lambda_{\min}[\mathcal{M}_l] \) represents the priority value of the user with the lowest priority in \( \mathcal{M}_l \), and \( \lambda_{\max}[\mathcal{M}_l] \) represents the priority value of the user with the lowest priority in \( \mathcal{M}_l \). The priority of users served in the previous batch is generally higher than that of users served in the latter batch, with the constraint expressed as (4).

\[
\lambda_{\min}[\mathcal{M}_{l-1}] > \lambda_{\max}[\mathcal{M}_l]. \tag{4}
\]

After completing the reconnaissance of the assigned target, the UAV enters the information sharing phase. A ferry UAV per batch is selected to share information with other UAVs to obtain the information of \( \mathcal{K}_l \). exit, represents the ferry UAV in the \( l \)th batch. We assumed that other UAVs only share information with the ferry UAV of the current batch and that ferry UAVs share information with only one UAV at the same time. We denoted \( s_{\text{exit},n} \) as the information sharing point of \( n \) and exit, and they share information near it. Thus, in a batch, the ferry UAV has multiple information sharing points, and other UAVs have one information sharing point. These UAVs that share information with exit, move on to the next batch’s reconnaissance phase immediately after sharing information, while exit, proceeds to the information distribution phase. During the information distribution phase, exit, travels to the user area and distributes information to users in order of priority. Note that the UAV finished information distribution returns to the take-off position and is no longer involved in subsequent reconnaissance activity. The status of \( n \) in the batch \( l \) is denoted as \( b_{l,n} \), where \( b_{l,n} = 1 \) represents that \( n \) has entered the information distribution phase before entering the \( l \)th batch. \( b_{l,n} = 0 \) indicates that \( n \) continues to reconnoiter targets in the \( l \)th batch. In addition, exit, can only be selected among the UAVs involved in the reconnaissance activity within the \( l \)th batch, with the constraint expressed as (5).

\[
b_{l,\text{exit}} = 0 \& b_{l+1,\text{exit}} = 1. \tag{5}
\]

To simplify the model, we assumed LN; thus, only one UAV per batch is selected to transmit information for the user. The UAVs that are not selected at the last batch return directly to the take-off position after finishing the information sharing phase.
In this section, we model the reconnaissance process as a path planning problem to maximize multiuser satisfaction. The two-dimensional trajectory of the UAV is denoted as $p_n(t)$ in the $i$th batch. In the first batch, UAV $n_1$ reconnoiters target 1, and $n_2$ reconnoiters target 2 during the reconnaissance phase. Then, $n_1$ and $n_2$ went to the information sharing position after finishing the reconnaissance of the assigned target. $n_1$ obtains the information of target 2 by sharing information with $n_2$. Subsequently, $n_1$ comes to the information distribution phase and transmits information for users $m_1$ and $m_2$, while $n_2$ comes to the second batch and reconnoiter targets 3. Finally, $n_2$ transmits information for $m_3$.

3.4. Path Planning Problems. In this section, we model the reconnaissance process as a path planning problem to maximize multiuser satisfaction. The two-dimensional trajectory of the UAV $n$ is denoted as $p_n(t)$. In the $i$th batch, UAV first reconnoiter targets of $N_{in_i}$. The relationship between UAV and target reconnaissance is represented by binary function $f^k_n(t)$, where $f^k_n(t) = 1$ means that the $n$ reconnoiter for the target $k$ at time $t$. $f^k_n(t) = 0$ for other cases. UAVs must stay or hover in the area over the target for $T_{target}$ time to obtain information, and the constraint is expressed by Equations (6) and (7).

$$f^k_n(t)p_n(t) - c(k) = 0, \quad (6)$$

$$\int_0^{T_{max}} f^k_n(t) dt = T_{target}, \quad (7)$$

where $c(k)$ represents the target’s two-dimensional coordinates, and $T_{max}$ represents the UAV’s maximum flight time. The UAV needs to return the take-off position before reaching the maximum flight time. The constraint is expressed as

$$p_n(T_{max}) = p_n(0). \quad (8)$$

UAV $n$ goes to the predetermined information sharing point after finishing the reconnaissance phase. The information sharing relationship between UAVs is represented as a binary function $g_{n,i}^k(t)$, where $g_{n,i}^k(t) = 1$ means that $i$ provides information about target $k$ for $n$ at time $t$, and $g_{n,i}^k(t) = 0$ represents the other cases. In addition, the remaining UAVs that share information with exit can only provide the target information that they have obtained, with the constraint shown in (9).

$$g^k_{exit,i}(t) \left| p_{exit}(t) - p_i(t) \right| = 0. \quad (9)$$

Because the two UAVs are close to each other, the information transmission speed is faster, and the shared information time can be ignored compared with the flight time. We assume that the two UAVs can complete the information sharing in an instant. $t_{exit}$ is used to indicate the time when exit ends the information sharing phase and enters the information distribution phase in the $i$th batch. After that, exit will not participate in the subsequent reconnaissance activity. The constraints are shown in (11).

$$b_{exit,i} \int_{t_{exit}}^{t_{max}} g^k_{exit,i}(t) dt = 0. \quad (11)$$

In the information distribution phase, the information transmission relationship between the UAV and user is denoted as binary function $s_n^m$, where $s_n^m = 1$ indicates that UAV $n$ transmits information to user $m$, and $s_n^m = 0$ indicates that $n$ does not transmit information to $m$. We assumed that the UAV is close to the user when it transmits information for this user, and it finishes information transmission instantly. The constraint is shown in (12).

$$s_{exit} \left| p_{exit}(t_{m,end}) - c(m) \right| = 0. \quad (12)$$

This equation indicates that the UAV must be at the user’s location when its transmission pertains to the user, where $c(m)$ represents the two-dimensional coordinate of $m$, and $t_{m,end}$ represents the time when $m$ obtains the needed
at a constant speed \( v \) sharing before transmitting them. The constraint is represented as follows.

\[
\int_0^{t_{\text{end}}} f(t) + \sum_{i=1}^{N} g_{\text{exit}_i}(t) dt = \xi_{\text{exit}_i}^l \tag{13}
\]

In addition, constraint (14) indicates that the time the user obtains information is below the UAV’s maximum flight time, and constraint (15) indicates that each UAV flies at a constant speed \( v \).

\[
t_{\text{end}} T_{\text{max}} \tag{14}
\]

\[
\frac{d}{dt} \rho_n(t) = v. \tag{15}
\]

The UAV path planning optimization objective is to maximize multiuser satisfaction, and the UAV path planning problem is formalized as follows.

\[
\max \xi_{\text{exit}_i}^l \rho_n(t) f_n(t) g_n(t) \tag{16}
\]

s.t. (6) – (15)

4. Algorithm Description

The difficulty in solving the above problems is mainly due to two aspects. First, the three functions \( f, g \) and \( s \) are tightly coupled with many optimization variables. Second, the solution to the UAV path planning function \( p \) is known as an NP problem, which is challenging to find the optimal solution or cannot be solved in a short time [25]. The four functions are coupled with each other, which makes the problem more difficult to solve. To this end, we proposed the batchwise information transmission path planning algorithm (BITPP) to solve the above problems.

Three subproblems need to be solved sequentially for planning the UAV cooperative reconnaissance paths. The first is to determine which users are served by UAVs in each batch. The second and third are to determine UAVs’ paths of the reconnaissance phase and the information sharing phase during each batch. Thus, the BITPP algorithm is proposed, which includes three subalgorithms A1, A2, and A3 to solve the above subproblems. The algorithm inputs are \( \mathcal{N}, \mathcal{M}, \) and \( \mathcal{K} \), and the output is \( \mathcal{P}f, \mathcal{P}i \subseteq \mathcal{P} \) represents the path set of the UAVs in set \( \mathcal{N} \) and is constantly updated in the running of the BITPP algorithm. \( \mathcal{N}_f \) represents the UAV set that continues to reconnoiter the targets in the \( l \)th batch, and \( \mathcal{P}f \) represents the path set of the UAVs in set \( \mathcal{N}_f \), where \( \mathcal{P}f \subseteq \mathcal{P} \). BITPP first establishes an empty path array for each UAV (steps 1-4). Then, it uses subalgorithm A1 to determine the set of served users in each batch (step 5). Subsequently, \( L \) cycles are performed, and each cycle determines the path of each UAV in a batch. For example, in the \( l \)th cycle, step 9 uses algorithm A2 to determine the path of the UAVs in \( \mathcal{N}_f \) when they complete the reconnaissance phase in the \( l \)th batch, and step 10 determines UAVs’ path when they complete the information sharing phase by algorithm A3. In addition, step 10 also determines the ferry UAV exit and the set \( \mathcal{N}_f^l \). Then, the information distribution paths for the ferry UAVs in this batch are determined (steps 11-13), where the function rank() represents ranking the UAVs in order of priority from large to small. Step 15 updates set \( \mathcal{P}_f \) and obtains the final path of each UAV.

4.1. Determining the Users Served in Each Batch. Planning a path for UAVs starts with determining that the users are
served in each batch. Subalgorithm A1 arranges the served users following the principle that higher priority users are served preferentially. A1 inputs are \( \mathcal{N}, \mathcal{M}, \) and \( L \), and outputs are \( \mathcal{M}_1 \cdots \mathcal{M}_l \). \( \mathcal{M}_l \) represents the user set that is served by UAVs in the \( l \)th batch. Step 1 obtains set \( \mathcal{M}_{\text{sort}} \) by sorting the users in set \( \mathcal{M} \) in descending order of priority. The user priority value sorted before in \( \mathcal{M}_{\text{sort}} \) is greater than the user sorted after. In steps 3-6, \( \lambda_{\text{total}} \) is the sum of all users’ priority values in the current \( \mathcal{M}_{\text{sort}} \). In steps 7 and 8, res is the number of remaining batches, and \( \lambda_{\text{average}} \) is the average priority value of each batch and is calculated by equation \( \lambda_{\text{total}}/\text{res} \). Step 9 creates empty set \( \mathcal{M}_p \), and step 10 uses \( \lambda_{\text{pro}} \) to represent the sum of all users’ priority values in the current \( \mathcal{M}_l \). Steps 11-18 adds users from \( \mathcal{M}_{\text{sort}} \) to \( \mathcal{M}_l \) in descending order of priority until \( \lambda_{\text{pro}} \) is greater than \( \lambda_{\text{average}} \); then, it stops adding users and proceeds to the next loop. For example, users 1-5 have priority values of 50, 40, 30, 20, and 10, and the UAVs perform reconnaissance tasks in three batches. When searching for the users to be served in the first batch, we judge res as 3 and calculate \( \lambda_{\text{total}} \) as 150, where \( \lambda_{\text{average}} \) is 50. Thus, user 1 is served in the first batch. Subsequently, res is 2, and \( \lambda_{\text{total}} \) is 100. Then, \( \lambda_{\text{average}} \) is 50, and users 2 and 3 are served in the second batch. Following this process, the third batch serves users 4 and 5.

4.2. Planning Path in the Reconnaissance Phase. The set of targets \( \mathcal{K}_l \) can be determined based on the users served in the \( l \)th batch and the relationship between the user’s demand for targets. UAVs need to plan paths that can traverse all the targets in \( \mathcal{K}_l \) during the \( l \)th batch’s reconnaissance phase, which is similar to the multiple traveling salesman problem [26]. To solve this problem, people usually use graph theory to construct targets as an undirected graph and find multiple routes that can traverse all the vertices in the graph with the least total cost [27]. In this paper, the target in \( \mathcal{K}_l \) and the starting point of the UAVs entering the reconnaissance phase are used as vertices to construct an undirected graph \( G(V_t, E_t, W_t) \).

In the first batch, the UAV’s starting point in the reconnaissance phase is its take-off position. In the other batches, the starting point is the endpoint of the UAV’s information sharing phase during the previous batch. \( V_l \) is the set of vertices, and the vertex \( v_{ij} \in V_l \) represents a starting point or targets. \( E_l \) is the set of edges, \( e_{ij} \in E_l \) represents the edges from \( v_{ij} \) to \( v_{ij} \), and UAVs need to fly along the edges. \( W_t \) is the weight set, \( w(v_{ij}) \in W_t \) is the point weight and represents the time of UAV reconnoiter \( v_{ij} \), and \( w(e_{ij}) \in W_t \) is the edge weight, representing the time spent by the UAV flying along edge \( e_{ij} \). \( w(v_{ij}, v_{ij}) \) represents the cost time that the UAV departed from \( v_{ij} \) to complete the reconnaissance of \( v_{ij} \). It includes the time spent by the UAV flying along edge \( e_{ij} \) and the time spent on reconnoiter \( v_{ij} \). The calculation formula is as follows: \( w(v_{ij}, v_{ij}) = w(e_{ij}) + w(v_{ij}) \). Assuming that the UAVs fly at the same speed and have the same reconnaissance time at each target, they share the same weight set.

| Algorithm 3: A2 |
|-----------------|
| **Input:** \( \mathcal{M}_l, \mathcal{P}_{\mathcal{A}_1}, \mathcal{N}_l, \mathcal{K}_l \) |
| **Output:** \( \mathcal{P}_{\mathcal{A}_1} \) |
| 1 Get \( \mathcal{K}_l \) according to \( \mathcal{M}_l \) and \( \mathcal{K}_l \); |
| 2 Get undirected graph \( G_l(V_t, E_l) \) according to \( \mathcal{K}_l \); |
| 3 while \( V_l \notin \varnothing \) do |
| 4 for each \( n \in \mathcal{N}_l \) do |
| 5 \( v_{in,\text{end}} = \text{end}(p_n) \); |
| 6 \( v_{next} = \text{chose}(v_{in,\text{end}}, V_l) \); |
| 7 \( p_n = \text{add}(p_n, v_{next}) \); |
| 8 \( V_l = \text{remove}(V_l, v_{next}) \); |
| 9 if \( V_l = \varnothing \) then |
| 10 break; |
| 11 end |
| 12 end |
| 13 end |
| 14 while true do |
| 15 \( (\text{time}_{\text{max}}, \max, \text{p}_{\text{max}}) = \max \text{time}(\mathcal{P}_{\mathcal{A}_1}) \); |
| 16 \( \text{p}_{\text{pro}} = \text{p}_{\text{max}} \); |
| 17 \( \mathcal{N}_{\text{pro}} = \text{remove}(\mathcal{N}_l, \max) \); |
| 18 for \( n \in \mathcal{N}_{\text{pro}} \) do |
| 19 \( p_n = \text{add}(p_n, v_{\text{max,end}}) \); |
| 20 \( \text{p}_{\text{max}} = \text{remove}(\text{p}_{\text{max}}, v_{\text{max,end}}) \); |
| 21 if \( t(p_n) < \text{time}_{\text{max}} \) then |
| 22 \( \mathcal{N}_{\text{pro}} = \text{remove}(\mathcal{N}_{\text{pro}}, n) \); |
| 23 \( p_n = \text{remove}(p_n, v_{\text{max,end}}) \); |
| 24 \( \text{p}_{\text{max}} = \text{add}(\text{p}_{\text{max}}, v_{\text{max,end}}) \); |
| 25 end |
| 26 if \( \text{p}_{\text{max}} = \text{p}_{\text{max}} \) then |
| 27 break; |
| 28 end |
| 29 end |
| 30 update(\( \mathcal{P}_{\mathcal{A}_1} \)); |
| 31 if \( \text{p}_{\text{pro}} = \text{p}_{\text{max}} \) then |
| 32 break; |
| 33 end |
| 34 end |

The UAV’s path in the reconnaissance phase can be represented as a combination of a series of vertices. We obtain it by using subalgorithm A2, which is divided into two parts: forming the initial path (steps 1-13) and path adjustment (steps 14-34). When running A2 in the \( l \)th cycle, the algorithm inputs are \( \mathcal{N}_l, \mathcal{M}_l \), and \( \mathcal{P}_{\mathcal{A}_1} \), and the outputs are \( \mathcal{P}_{\mathcal{A}_2}, \mathcal{N}_l \) represents the UAV set that continues to reconnoiter targets in the \( l \)th batch. \( \mathcal{P}_{\mathcal{A}_2} \) represents the path set of the UAVs in \( \mathcal{N}_l \). Moreover, \( \mathcal{N}_l = \mathcal{N}_i \) when \( l = 1 \). The first part is to search UAVs’ initial paths based on a greedy strategy. Each UAV takes turns to find the next vertex, which costs the least when the UAV departs from its current position to complete this vertex reconnaissance. The initial flight path of UAVs is obtained when all the vertices are traversed. Step 5 uses function \( \text{end}(p_n) \) to find the endpoint \( v_{n,\text{end}} \) of the current \( p_n \), and step 6 uses function \( \text{chose}(v_{n,\text{end}}, V_l) \) to find the next visit vertex \( v_{next} \) when UAV \( n \) completes the reconnaissance of vertex \( v_{n,\text{end}} \). Step 7 uses function \( \text{add}(p_n, v_{next}) \)
to add $v_{next}$ to the end of $p_n$. Then, $v_{next}$ is removed from $V_l$ in step 8. When $V_l$ becomes an empty set, the loop is terminated, and $\mathcal{P}_{X_l}$ is updated.

In the second part, $p_{max}$ and max, respectively, represent the most time-consuming path in $\mathcal{P}_{X_l}$ and the UAV corresponding to this path. $time_{max}$ represents the time of UAV max fly along $p_{max}$. We need to adjust each UAV's path by iterating. In each iteration, we first judge the $p_{max}$ of the current set $\mathcal{P}_{X_l}$, and then the endpoint of $p_{max}$ is transferred to the end of the other path in $\mathcal{P}_{X_l}$. If the algorithm can find the UAV path in which $time_{max}$ in $\mathcal{P}_{X_l}$ decreases after it receives the endpoint of $p_{max}$, the algorithm continues to adjust the path in set $\mathcal{P}_{X_l}$ in the next iteration. Otherwise, this adjustment operation is canceled, and the iteration is terminated. In detail, step 15 searches $p_{max}$ and $time_{max}$ in $\mathcal{P}_{X_l}$ by function $maxtime(p_{max})$. Function $t(p_n)$ calculates the time of $n$ fly along path $p_n$. Steps 18-30 adjust $p_{max}$ and update the $\mathcal{P}_{X_l}$. Steps 31-34 determine whether the $p_{max}$ has changed. If there is no change, the path adjustment operation cannot be performed, and the iteration is terminated.

For example, we obtain three paths by running subalgorithm A2, represented as $p_1 = (1, 4, 5), p_2 = (2, 6, 7), \text{and } p_3 = (3, 8, 9)$, and the relationship of these path's cost time is represented as $t(p_1) > t(p_2) > t(p_3)$. In the second step, we first judge $time_{max} = t(p_1)$ and $p_{max} = p_1$. Then, the endpoint of $p_1$ is judged as vertex 5. If $t(p_1) + w(7, 5) < time_{max}$, we remove vertex 5 from $p_1$ to the end of $p_5$. The adjusted paths are $p_1 = (1, 4), p_2 = (2, 6, 7, 5), \text{and } p_3 = (3, 8, 9)$. Then, the next iteration of adjustment is entered. If $t(p_2) + w(7, 5) > time_{max}$ and $t(p_3) + w(7, 5) > time_{max}$, no path can receive the endpoint of $p_i$; so, path adjustment can no longer be performed, and the current UAV path is output as the final result.

4.3. Planning Path in the Information Sharing Phase: UAVs' flight paths in the information sharing phase can be translated into a series of visited sequences of information sharing points. The ferry UAV has multiple information sharing points in each batch, while the remaining UAVs have only one information sharing point. We obtain the path of the UAVs' information sharing phase by subalgorithm A3. In the lth batch, A3's inputs are $N_l$ and $\mathcal{P}_{X_l}$, and the outputs are $N_{l+1}, exit_l$, and $\mathcal{P}_{X_{l+1}}$. A3 includes two parts. The first part is to determine the ferry UAV of the current batch, and the main idea is to select the UAV that ends the reconnaissance phase earliest as the ferry UAV exit. In the second part, UAV exit searches other UAVs and corresponding information sharing points for sharing information based on the earliest encounter principle. The behavior of two UAVs heading to the information sharing point in this principle is regarded as making a relative motion from the departure point.

The departure point is the position where the UAV changes from the current flight direction to the direction of the next information sharing point. The next UAV shared with the ferry UAV encountered the earliest with the ferry UAV when they made the relative motion. Their encounter position is regarded as their information sharing point. $sr_{exit,n}$ represents the ith information point of UAV n in the lth batch. Assuming n is the second UAV that shares information with $exit_{l,i}$ in l, their information sharing point $sr_{exit,n}$ is the first information sharing point of n and the second information sharing point of $exit_{l,i}$. That is, $sr_{n,1} = sr_{exit,n,1} = sr_{exit,n,2}$. Ferry UAV exit has multiple departure points. When it searches the first information sharing point, its departure point is the reconnaissance phase's endpoint in the current batch. When it searches the lth information sharing point, its departure point is the previous information sharing point $sr_{exit,I,j-1}$. Other UAVs have one departure point at which the reconnaissance phase endpoint is in the current batch.

When running subalgorithm 3 in the lth cycle of BITPP, steps 1-7 find the UAV with the fastest ending reconnaissance phase in batch l as the ferry UAV exit. Step 8 removes exit from set $N_l$ and obtain sets $N_{pro}$ and $N_{l+1}$, where $N_{pro}$ represents the UAV that currently needs to share information with exit_l. Step 9 determines the number of UAVs that need to share information with exit_l. Step 11 establishes $t_{pro}$ to record the current fastest encounter time between exit_l and other UAVs. Steps 12-20 find the UAV in set $N_{pro}$ that encounters exit_l, the earliest and use next to represent the UAV that can make the earliest encounter exit_l in the current iteration. In step 13, function $meet_time()$ calculates the relative motion time between exit_l and n, taking their current path end positions as the departure point.

In addition, because the time when UAVs arrive at their respective departure point is different, it is necessary to calculate the time interval between exit_l and n arriving at their respective departure point. This time interval is expressed by $\Delta t_{exit,n,l}$, and the formula is as follows:

$$\Delta t_{exit,n,l} = |t(p_n) - t(p_{exit})|.$$ (17)

$meet_{exit,n}$ represents the encounter time between exit_l and n, and the formula is as follows:

$$meet_{exit,n} = \left(\frac{d(\text{end}(p_{exit}), \text{end}(p_n)) - \Delta t_{exit,n,l} \cdot v}{2v}\right).$$ (18)

where $d(\text{end}(p_{exit}), \text{end}(p_n))$ represents the distance between the endpoints of paths $p_{exit}$ and $p_n$. In step 17, the function $meet_point(exit_l, n, p_{exit}, p_n)$ calculates the encounter coordinates of exit_l and n. Steps 21-22 add this encounter point to the end of the paths $p_{exit}$ and $p_{next}$. In addition, it is used as the information sharing point of exit_l, and next. Step 23 updates $\mathcal{P}_{X_{l+1}}$. Step 24 removes next from the set $N_{pro}$ and then enters the next cycle.

An example of UAVs searching for information sharing points is shown in Figure 2. There are four UAVs in the picture. Assuming UAV 1 ends the reconnaissance phase earliest, it is regarded as the ferry UAV of the current batch. In
After A2 is run $L$ times, its complexity is $O(K)$. The complexity of A3 is related to the number of UAVs in $\mathcal{M}$. It includes two parts: finding the ferry UAV and the planning path of the information sharing phase. When determining the ferry UAV, it needs to traverse the $\mathcal{M}$ set once, and its complexity is not higher than $O(N)$. Thus, the complexity that runs this part $L$ times is $O(LN)$. When planning the information sharing path, it is necessary to traverse the number of UAVs in set $\mathcal{M}$ minus one squared at most. Thus, the complexity of running the second part $L$ times is $O(L(N-1)^2)$. In conclusion, the complexity of the BITPP algorithm is $O(M+K+L(N^2-N+1))$.

5. Simulation Results and Analysis

5.1. Simulation Setup. In this section, we discuss the performance and applicability of the BITPP algorithm by analyzing the simulation results. Assume that the UAV, user, and reconnaissance targets are in a square area with a side length of 3 km, the default parameters for the number of UAVs and users are 6, the number of batches is 6, and the two-dimensional coordinates of the UAV take-off position are $(0, 0)$. The user’s priority value is uniformly distributed in the value range of 10 to 100. That is, the priority value of users 1-6 is 100, 82, 64, 46, 28, and 10 in turn. Users are randomly distributed within 300 meters from the starting point of the drone. That is, the vertical and horizontal coordinates of users are randomly generated within 0-0.3 km. By default, each user is interested in 10 targets, and there is no duplication of user needs. In addition, this paper considers the situation in which the reconnaissance area is far from the user. Therefore, it is assumed that the reconnaissance target’s vertical and horizontal coordinates are randomly generated in the range of 1 km to 3 km. The specific parameters of the satisfaction model and path planning are shown in Table 1.

To prove the proposed multiuser satisfaction model’s ability to reflect the user’s information acquisition time and verify the performance of the proposed batchwise information backhaul strategy and the BITPP algorithm, we compare with the following four algorithms.

GRMRT: UAVs make multiple trips between the user and the reconnaissance area. On each trip, UAVs cooperate to reconnoiter the targets of one user interested and then all fly to the user area to transmit reconnaissance information. UAVs’ flight path is planned based on the greedy algorithm.

GAMRT: UAVs make multiple trips between the user and the reconnaissance area. On each trip, UAVs cooperate to reconnoiter the targets of one user interested and then all fly to the user area to transmit reconnaissance information. UAVs’ flight path is planned based on the greedy algorithm.

GROT: UAVs make one trip between the user and the reconnaissance area. On this trip, UAVs cooperate to reconnoiter the targets of all users interested and then all fly to the user area to transmit reconnaissance information. UAVs’ flight path is planned based on the greedy algorithm.

GRORT: UAVs make one trip between the user and the reconnaissance area. On this trip, UAVs cooperate to reconnoiter the targets of all users interested and then all fly to the

\[
\begin{align*}
\text{Input:} & \quad \mathcal{M}, \mathcal{P}, \text{exit}\_i, \text{next}\_i, \text{pro}\_i, \text{pro}\_p, \text{pro}\_n, \text{pro}\_l, \text{pro}\_m, \text{pro}\_s, \text{pro}\_t, \text{pro}\_p, \text{pro}\_n, \text{pro}\_l, \text{pro}\_m, \text{pro}\_s, \text{pro}\_t, \\
\text{Output:} & \quad \mathcal{M}_i, \mathcal{P}_i, \text{exit}\_i, \text{next}\_i, \mathcal{P}_i, \text{pro}\_i, \text{pro}\_p, \text{pro}\_n, \text{pro}\_l, \text{pro}\_m, \text{pro}\_s, \text{pro}\_t, \text{pro}\_p, \text{pro}\_n, \text{pro}\_l, \text{pro}\_m, \text{pro}\_s, \text{pro}\_t, \\
1 & \quad t_{\text{pro}} = \infty; \\
2 & \quad \text{for each } n \in \mathcal{M} \text{ do} \\
3 & \quad \text{if } t(p_n)_{\text{pro}} \text{ then} \\
4 & \quad \quad \text{exit}_i = n; \\
5 & \quad \quad t_{\text{pro}} = t(p_n); \\
6 & \quad \text{end} \\
7 & \quad \text{end} \\
8 & \quad \mathcal{M}_i = \{ \text{pro}\_i = \text{remove}(\mathcal{M}_i, \text{exit}_i) \}; \\
9 & \quad \text{number} = |\mathcal{M}_i|; \\
10 & \quad \text{for } i = 1 \text{ to number do} \\
11 & \quad t_{\text{pro}} = \infty; \\
12 & \quad \text{for each } n \in \mathcal{M}_i \text{ do} \\
13 & \quad \quad \text{meet}_i = \text{time}(\text{exit}_i, n, p_n, \text{pro}_i); \\
14 & \quad \quad \text{if } \text{meet}_i > t_{\text{pro}} \text{ then} \\
15 & \quad \quad \quad \text{next}_i = n; \\
16 & \quad \quad \quad t_{\text{pro}} = \text{meet}_i; \\
17 & \quad \quad \quad \text{start}_i = \text{point}(\text{exit}_i, n, p_n, \text{pro}_i); \\
18 & \quad \quad \quad \text{start}_i = \text{start}_i; \\
19 & \quad \quad \text{end} \\
20 & \quad \text{end} \\
21 & \quad \text{if } \text{meet}_i > t_{\text{pro}} \text{ then} \\
22 & \quad \quad \text{add}(\text{path}_i, \text{start}_i); \\
23 & \quad \quad \text{update}(\mathcal{P}_i); \\
24 & \quad \quad \text{remove}(\mathcal{M}_i, \text{next}_i); \\
25 & \quad \text{end} \\
\end{align*}
\]
user area to transmit reconnaissance information. UAVs' flight path is planned based on the genetic algorithm.

5.2. Simulation Analysis

5.2.1. UAV Trajectory in Different Batch. We reduce the number of UAVs and users to 3 for a clearer view of the UAV trajectory. Then, we show the UAV trajectory in batches 1 and 2. The green circle represents the information sharing point, the blue circle represents the user, the yellow circle represents the target point, the hollow circle represents the state where it has not been accessed by the drone, and the solid circle represents the state where it has been accessed by the drone. As shown in Figure 3, UAV 2 is selected as the ferry UAV in the first batch. After finishing the reconnaissance for assigned targets, it shares information with UAVs 1 and 3 in turn and finally reaches the corresponding user. As shown in Figure 4, UAV 2 does not involve the reconnaissance activity in the second batch. UAV 1 is selected as the ferry UAV in the second batch and shares information with UAV 3. UAV 3 will come in the third batch and reconnoiter the remaining targets not visited.

5.2.2. Information Acquisition Time of Different Users. According to the default parameter settings for simulation, Figures 5–7 are obtained. Figure 3 shows the comparison of the BITPP algorithm with GAMRT and GRMRT, and Figure 6 shows the comparison of the BITPP algorithm with GAORT and GRORT. Both figures reflect the information acquisition time of each user under different algorithms. Figure 7 reflects each user’s satisfaction under the different algorithms. The X-axis of these three figures represents the user series number sorted in descending order of priority. In Figures 5 and 6, the corresponding user information acquisition time increases with the user series number. In Figure 7, the corresponding user satisfaction decreases with increasing user series number. All five algorithms determine the order that transmits information to the user according to the priority level, and thus higher priority users obtain information earlier than lower priority users.

In Figure 5, the difference in each user’s information acquisition under BITPP is the smallest compared with the GAMRT and GRMRT algorithms. In addition, only the user with serial number 1 acquires information under the BITPP algorithm in a slightly higher time than the results using the

| Table 1: Simulation parameters. | Description | Symbol | Value |
|--------------------------------|-------------|--------|-------|
| Expected time for information acquisition | $T_{\text{expect}}$ | 200 s |
| Maximum tolerance time for information acquisition | $T_{\text{deadline}}$ | 1000 s |
| Initial satisfaction | $\gamma$ | 100 |
| Single target reconnaissance time | $T_{\text{target}}$ | 5 s |
| UAV maximum flight time | $T_{\text{max}}$ | 40 min |
| UAV flight speed | $v$ | 20 m/s |
other two algorithms. The rest of the users’ information acquisition time is lower than the results using the comparison algorithm. This is because all UAVs fly to the user area for information distribution after finishing reconnaissance for targets of one user interested under the GAMRT and GRMRT algorithms. Thus, the distance traveled by UAVs between the target area and the user area dramatically extends the time for the following users to obtain information. When the target is farther away from the user, the difference in information acquisition time between higher and lower priority users is more significant. This is likewise why lower priority user satisfaction decreases severely when the GAMRT and GRMRT algorithms are used, as shown in Figure 7. In contrast, only the ferry UAV in each batch transmits information to the user under the BITPP algorithm. When the ferry UAV flies to the user area, other UAVs can continue to reconnoiter the targets of the next user interested, and the next user does not need to wait according to the amount of time that UAVs travel between the user and the target area. Therefore, in Figures 5 and 7, the differences in information acquisition time and satisfaction between different users using the BITPP algorithm are minor compared to the GAMRT and GRMRT algorithms.

As seen in Figure 6, the difference in information acquisition time between different users under the GAORT and GRORT algorithms is slight. This reason is that both algorithms require UAVs traveling to the user area after reconnoitering the targets of interest to all users. However, it also causes the higher priority users’ information acquisition under the two algorithms to be much longer than using the BITPP algorithm. In contrast, the BITPP algorithm can make UAVs provide information to higher priority users in advance, thus ensuring that most higher priority users can obtain information faster. In addition, as seen in Figures 6 and 7, user 6 obtains information under the BITPP algorithm later than using the GAORT algorithm, and the satisfaction is also lower. In the BITPP algorithm, UAVs in each batch need to spend a small amount of time in the information sharing phase. As the batches increase, this time will continue to accumulate, and the information acquisition time for low priority users will be extended accordingly. However, compared with the GAORT and GRORT
algorithms, the BITPP algorithm still ensures a lower information acquisition time for most users. Although the use of the BITPP algorithm leads to a higher information acquisition time for user 6 than the GAORT algorithm, this difference is very small. In summary, the BITPP algorithm can guarantee the higher priority users to obtain the information quickly while minimizing the time for the lower priority users to obtain the information.

5.2.3. Different Number of Users. Keeping UAV numbers and other parameters unchanged, we measure the variation in multiuser satisfaction when the user number increases. As seen in Figure 8, the multiuser satisfaction under each algorithm decreases as the number of users to be served increases. The fundamental reason is that UAVs need to reconnoiter more targets, and the total time that UAVs complete all user reconnaissance tasks is also increased. In addition, multiuser satisfaction showed different decreasing trends under different algorithms. Among the five algorithms, the decreasing trend of multiuser satisfaction under the BITPP algorithm was the slowest, while the satisfaction of the remaining four algorithms decreased faster. In GAORT and GRORT algorithms, UAVs need to reconnoiter all targets at one time before providing information for users. As the number of users increases and the targets for one-time reconnaissance increase, users’ time to obtain information increases together. In the GAMRT and GRMRT algorithms, UAVs prioritize reconnoitering targets of high priority users of interest. Thus, the information acquisition time for higher priority users is not affected as the number of users increases. However, the acquisition time for users with lower priority rankings is severely delayed. The BITPP algorithm ensures that the UAV priority reconnoiter the targets of interest to high priority users. It also dramatically reduces the impact on low priority users’ information acquisition time by avoiding the round trip time between the user area and the target area of the UAV. The above analysis shows that the BITPP algorithm can better cope with situations where the number of users is large or the number of UAVs is small.

5.2.4. Different Numbers of the Total Batch. Assume that each user needs 25 goals, and that user needs are not repeated. Setting the total number of batches from 1 to 6, Figures 9 and 10 are obtained by simulation. Figure 9 reflects the variation in multiuser satisfaction under different batch number settings in the BITPP algorithm, and Figure 10 reflects the change in each user’s information acquisition time. As shown in Figure 10, each user’s information acquisition time is close when the number of batches is 1. After setting the batch number to 2, UAVs serve users 1 and 2 in the first batch and 3-6 in the second batch. The information acquisition time of users 1 and 2 decreases substantially, while the remaining users’ information acquisition time increases slightly. At the time, multiuser satisfaction increases significantly in Figure 9 when the number of batches changes from 1 to 2. It shows that the multiuser satisfaction model is strongly influenced by the variation of the higher priority user’s information acquisition time.

When the number of batches is set to 3, UAVs serve only user 1 in batch 1, users 2 and 3 in batch 2, and users 4-6 in batch 3. Therefore, the information acquisition time of users 1 and 3 is further reduced. The information acquisition time of user 2 increases but is still smaller than the information acquisition time when the batch number is set to 1. In addition, the multiuser satisfaction still increases when setting the batch number from 2 to 3 in Figure 9, but the increase is smaller than the increase in the batch number from 1 to 2. These results show that the user satisfaction model takes the information acquisition time of high priority users as the main reference while taking into account the information acquisition time of lower priority users. As seen in Figure 9, the trend of multiuser satisfaction growth is faster and then slower as the number of batches increases. On the one hand, it reflects that when each user’s priority value is characterized by a uniform distribution, the multiuser satisfaction increases with the number of batch sets. On the other hand, the users whose information acquisition time is affected due to the increased number of batches are minority users, and most of them are lower priority users.
5.2.5. Different User Priority Distributions. The difference between the user priority values plays an important role in considering UAV planning; thus, we set four different priority value distribution types. As shown in Table 2, users’ priority values range from 10 to 100. In type 1, each user has the same priority value. The users’ priority values in type 2 are uniformly distributed, and the difference between adjacent users’ priority values is equal. In type 3, the difference between adjacent users’ priority values with the serial number from small to large is a descending arithmetic sequence with a tolerance of 6. In contrast to type 3, the difference between adjacent users’ priority values in type 4 from small to large is an increasing arithmetic sequence with a tolerance of 6. The first type represents the same importance of each user and the same urgency to obtain information, and UAVs do not need to pay more attention to any user than the others when providing service. The remaining three distribution types represent situations where the urgency of user needs varies. The second type represents a uniform distribution of urgency for each user to obtain information, the third type represents a scenario where a few users have high urgency to obtain information and most users have low urgency, and the fourth type represents a scenario where most users have high urgency to obtain information and a few users have low urgency to obtain information. Of these, the reconnaissance service environment for UAVs in type 3 is more relaxed because the urgency of the information needs of most users is concentrated toward the high value, and the reconnaissance service environment for UAVs in type 4 is more urgent because the urgency of most users is concentrated toward the high value.

Figure 11 shows the multiuser satisfaction results of different algorithms under the four types. The BITPP, GRMRT, and GAMRT algorithms all have the highest satisfaction level when dealing with type 3 and the lowest satisfaction level when dealing with type 2. The main reason is that these three algorithms can prioritize some users’ needs; so, they are more suitable for situations where there are differences in users’ priorities and situations where some users have very high priorities. The results show that the multiuser satisfaction under the GRORT and GAORT algorithms does not vary significantly with the change in priority distribution type. This is because the two algorithms require UAVs to reconnoiter targets needed by all users at one time. Thus, the change in user priority value has little effect on each user’s information acquisition time. The GRORT and GAORT algorithms are more suitable for situations where users have the same priority. BITPP combines the four algorithms’ advantages and can achieve higher multiuser satisfaction than the other four algorithms for different user priority distribution types. The algorithm works better in situations in which a few users have a very high priority value.

6. Conclusions

This paper studies a multi-UAV cooperative reconnaissance scenario where multiple users with different priorities need to be served. A multiuser satisfaction model based on users’ diverse priorities and a batchwise information backhaul strategy are proposed. This reconnaissance process is formulated as a cooperative path planning problem, and the path planning algorithm BITPP is proposed to maximize multiuser satisfaction. The simulation results show that the BITPP algorithm can make higher priority users obtain information faster than using one round trip reconnaissance strategy. In addition, compared to other multiple round trip strategies, the BITPP algorithm can make lower priority users obtain
information with less delay while ensuring higher priority users' faster information acquisition. In addition, the applicability of the BITPP algorithm is simulated and analyzed in several different scenarios. Simulation results show that it can still produce good results with increased users, and it is more effective in situations where a small number of users exists with very high priority values. In future studies, we will focus on cases where multiple UAVs are selected as ferries in each batch while exploring a more complex and effective UAV collaboration mechanism to cope with a dynamic environment and task flow.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

Zeyuan Liu and Cuntao Liu contributed equally to this work.

References

[1] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[2] D. Hong, S. Lee, Y. H. Cho, D. Baek, J. Kim, and N. Chang, “Least-energy path planning with building accurate power consumption model of rotary unmanned aerial vehicle,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14803–14817, 2020.

[3] H. Qin, Z. Meng, W. Meng et al., “Autonomous exploration and mapping System using heterogeneous UAVs and UGVs in GPS-denied environments,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1339–1350, 2019.

[4] O. M. Bushnaq, A. Celik, H. Elsawy, M.-S. Alouini, and T. Y. Al-Naffouri, “Aeronautical data aggregation and field estimation in IoT networks: hovering and traveling time dilemma of UAVs,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4620–4635, 2019.

[5] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, “Dynamic Spectrum interaction of UAV flight formation communication with priority: a deep reinforcement learning approach,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 892–903, 2020.

[6] C. Barroca, A. Grilo, and P. R. Pereira, “Improving Message Delivery in UAV-based Delay Tolerant Networks,” in 2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST), pp. 1–7, Lisboa, Portugal, 2018.

[7] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. S. Shu, “Path planning for UAV-mounted Mobile edge computing with deep reinforcement learning,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5723–5728, 2020.

[8] M. Erdelji, “Wireless sensor networks and multi-UAV systems for natural disaster management,” *Computer Networks*, vol. 124, pp. 72–86, 2017.

[9] K. Anazawa, P. Li, T. Miyazaki, and S. Guo, “Trajectory and Data planning for mobile relay to enable efficient internet access after disasters,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, San Diego, CA, USA, 2015.

[10] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, “Age of information aware trajectory planning of UAVs in intelligent transportation systems: a deep learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12382–12395, 2020.

[11] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, “Stochastic optimization aided energy-efficient information collection in internet of underwater things networks,” *IEEE Internet of Things Journal*, p. 1, 2021.

[12] Y. Wang, Z. Hu, X. Wen, Z. Lu, and J. Miao, “Minimizing data collection time with collaborative UAVs in wireless sensor networks,” *IEEE Access.*, vol. 8, pp. 98659–98669, 2020.

[13] C. Zhan and Y. Zeng, “Completion time minimization for multi-UAV-enabled data collection,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4859–4872, 2019.

[14] H. Wang, J. Wang, G. Ding, J. Chen, F. Gao, and Z. Han, “Completion time minimization with path planning for fixed-wing UAV communications,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3485–3499, 2019.

[15] A. El Fallahi and I. Sefrioui, “A linear programming model and memetic algorithm for the emergency vehicle routing,” in *2019 4th world conference on complex systems (WCCS), Ouarzazate, Morocco*, 2019.

[16] K. Miyano, R. Shinkuma, N. B. Mandayam, T. Sato, and E. Oki, “Utility based scheduling for multi-UAV search systems in disaster-hit areas,” *IEEE Access.*, vol. 7, pp. 26810–26820, 2019.

[17] B. K. Mishra, T. N. Adhikari, K. Dahal, and Z. Pervez, “Priority-index based multi-priority relief logistics scheduling with greedy heuristic search,” in *in 2018 5th international conference on Information and communication Technologies for Disaster Management (ICT-DM)*, Sendai, Japan, 2018.

[18] J. Li, Y. Xiong, J. She, and M. Wu, “A path planning method for sweep coverage with multiple UAVs,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8967–8978, 2020.

[19] J. McMahon and E. Plaku, “Autonomous data collection with limited time for underwater vehicles,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 112–119, 2017.

[20] L. Sun, L. Wan, and X. Wang, “Learning-based resource allocation strategy for industrial IoT in UAV-enabled MEC systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5031–5040, 2021.

[21] H. Duan, J. Zhao, Y. Deng, Y. Shi, and X. Ding, “Dynamic discrete pigeon-inspired optimization for multi-UAV cooperative search-attack mission planning,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, pp. 706–720, 2021.

[22] A. T. Albu-Salih and S. A. H. Seno, “Energy-efficient data gathering framework-based clustering via multiple UAVs in deadline-based WSN applications,” *IEEE Access.*, vol. 6, pp. 72275–72286, 2018.

[23] M. Mashukar, Y. Nakamura, N. Suzuki, M. Inoguchi, and A. Morishima, “A route search system considering urgency and efficient coverage without complete information,” in *in 2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Kyoto, Japan, 2019.

[24] Z. Huang, C. Chen, and M. Pan, “Multiobjective UAV path planning for emergency information collection and
transmission,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6993–7009, 2020.

[25] L. Zhong, K. Garlichs, S. Yamada, K. Takano, and Y. Ji, “Mission planning for UAV-based opportunistic disaster recovery networks,” in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2018.

[26] J. Scherer and B. Rinner, “Multi-UAV surveillance with minimum information idleness and latency constraints,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4812–4819, 2020.

[27] J. Yu, “Intractability of optimal multirobot path planning on planar graphs,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 33–40, 2016.