Received April 6, 2022, accepted April 23, 2022, date of publication April 26, 2022, date of current version May 4, 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3170583

Decentralized Multi-UAV Path Planning Based on Two-Layer Coordinative Framework for Formation Rendezvous

ZHEKUN CHENG[1], LIANGYU ZHAO[1], (Member, IEEE), AND ZHONGJIAO SHI[2]
[1]School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China
[2]School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China
Corresponding author: Zhongjiao Shi (shizhongjiao@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62103052, and in part by the National Natural Science Foundation of China under Grant 12072027.

ABSTRACT Unmanned aerial vehicle (UAV) formation rendezvous path planning problem is one of the important research topics in multiple UAV (multi-UAV) coordinated path planning. Aiming at solving low computational efficiency and poor scalability of the traditional multi-UAV path planning method, the decentralized multi-UAV path planning method suitable for obstacle environments is proposed. Firstly, the UAV rendezvous path planning problem with constraints such as the kinematics of UAVs and collision-free constraints is modeled as a non-convex optimal control problem. To minimize formation rendezvous time and energy consumption, a two-layer coordinative framework is developed to solve this problem. In the coordination layer, relying only on the information of neighboring UAVs, each UAV in the decentralized communication graph negotiates the desired flight time using a consensus protocol to achieve coordination among UAVs. In the planning layer, the initial non-convex formation rendezvous path planning problem is decoupled into several sub-problems, which can be solved in parallel by path planners distributed on each UAV using sequential convex programming. Finally, numerical simulations are carried out to verify the effectiveness and scalability of the proposed method. The results show that this decentralized multi-UAV path planning method can handle the minimum-time rendezvous path planning problem and optimize the energy consumption in flight, and the computing time does not increase significantly with the enlargement of the UAV swarm. This decentralized framework scales well with the number of UAVs and can be applied for future urban flight and supplies delivery tasks.

INDEX TERMS Unmanned aerial vehicle, path planning, decentralized coordination, sequential convex programming, collision-free.

I. INTRODUCTION

Unmanned aerial vehicle (UAV) swarms have attracted widespread attention, along with advances in processing, sensing, and communication technologies [1]. More flexible and robust than a single large UAV, UAV swarms are having a significant impact in many areas, including agriculture [2], powerline inspection [3], and Monitoring [4]. Multi-UAV coordinated path planning is one of the key technologies for UAV swarms to operate in complex environments such as urban canyons. Path planning methods need to generate a collision-free path from the starting position to the target position for each UAV in the swarm and minimize the total cost of the swarm while meeting the constraints such as kinematic and dynamic characteristics [5]. For UAVs taking off from different locations, to make UAVs reach a predetermined rendezvous state simultaneously, formation rendezvous path planning is one of the most critical technologies for UAV formation flight.

The difficulties of UAV formation rendezvous path planning are: 1) UAVs in the swarm are in the same working space and will affect each other; 2) due to the increase in the number of UAVs, the mathematical scale of the planning problem is usually larger, especially when specific optimality is required, the computational complexity of the problem will increase significantly; 3) it is necessary to consider the

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo.

coordination between different UAVs [6]. In recent years, researchers have carried out some explorations on UAV formation rendezvous path planning, many methods have been proposed to solve such problem, which can be roughly divided into four categories: 1) methods based on graph search, such as A* [7], [8], state lattices [9]; 2) methods based on sampling, such as Rapidly-exploring Random Trees (RRT) [10], Probabilistic Road Map (PRM) [11]; 3) methods based on mathematical models, such as nonlinear programming (NP) [12], mixed-integer linear programming (MILP) [13]–[15]; 4) methods based on artificial intelligence, such as genetic algorithm [16], ant colony optimization [17], and reinforcement learning [18].

Compared with other algorithms, the path obtained by the algorithm based on a mathematical model is generally more accurate, smoother, and more traceable due to the consideration of the UAV kinematics model and several other constraints, and optimization algorithms can be used to obtain optimal paths [19]. Although multi-UAV path planning is usually modeled as a non-convex optimization problem, sequential convex programming (SCP) can be used to approximate it as a series of convex optimization sub-problems to solve [20]. With the improvement of convex optimization methods and the development of computer technology, large-scale convex optimization problems have been able to obtain optimal solutions in a limited time [21].

In multi-UAV path planning, the constraints brought by multi-UAV coordination mainly include collision avoidance [22], [23], connectivity maintenance [24], synchronization requirements, heading coordination, and formation [25]. According to different communication topologies of UAV swarms, coordination among UAVs can be roughly divided into centralized and decentralized. Through coordination, the overall cost of the swarm is minimized, that is, the overall consumption of the swarm when completing tasks, which usually includes time and resource costs [26].

In this paper, we propose a decentralized multi-UAV path planning method based on the two-layer coordinative framework for the UAV formation rendezvous problem in a three-dimensional environment. The proposed method can plan coordinated collision-free paths for the UAV swarm in obstacle environments so that UAVs from different initial positions can arrive at the designated positions at the same time and form a predetermined formation. The main contributions of this paper are as follows:

1) Using consensus protocols and SCP, we proposed a decentralized multi-UAV path planning method based on the two-layer coordinative framework. The proposed method can handle the minimum time path planning problem and optimize the energy consumption in flight. In addition, since this method combines the consensus protocol and SCP, the UAVs in the swarm can negotiate the desired flight time autonomously, which avoids the unsolvable situation of the path planning problem and avoids the waste of flight time.
2) The decentralized path planning framework ensures that the increase in the number of UAVs does not lead to a significant boost in planning time, which enables the proposed method to scale well with the number of UAVs. Furthermore, in the obstacle environment, relying only on the information of neighboring UAVs, the proposed multi-UAV path planning method can realize the coordination of UAVs in a decentralized communication network.
3) The proposed method has a shorter computing time than the standard SCP, and this advantage becomes more pronounced as the number of UAVs increases. Meanwhile, paths planned by the proposed method have a shorter flight time than that planned by the decoupled SCP.

The remainder of this paper is organized as follows. Previous related research is presented in Section II. In Section III, some mathematical tools used in this paper are reviewed and the to-be-solved optimal control problem is established. Section IV introduces the main results of the paper. A decentralized multi-UAV path planning method based on the two-layer coordinative framework is proposed. The non-convex optimal control problem is transformed into a series of convex optimization sub-problems using the SCP. The numerical simulation results are presented in Section V. Finally, the conclusions are made in Section VI.

II. RELATED WORKS

To ensure that UAV swarms can successfully perform coordinative tasks, the UAV formation rendezvous path planning problem has been studied extensively, including many methods based on the Dijkstra algorithm, particle swarm optimization, reinforcement learning, and SCP.

Manathara and Ghose [27] applied velocity control and wandering maneuvers to obtain formation rendezvous paths for the UAV swarm based on the estimated time of arrival. An alternative method is to assume that all UAVs are flying at a constant and equal velocity, by generating paths of equal length to achieve simultaneous arrivals [28]. In addition, another approach is to plan the paths under the constraints of the rendezvous time. Ma *et al.* [29] used the improved Dijkstra algorithm to plan the rendezvous paths for UAVs in a two-dimensional environment. However, such methods are difficult to scale to high-dimensional systems. Swarm intelligent algorithms are suitable for handling such path planning problems when the problem is too complex or potentially uncertain [30]. Shao *et al.* [31] proposed a distributed cooperative particle swarm optimization algorithm with an elite keeping strategy, which was used for rendezvous path planning considering kinematic constraints. Reinforcement learning can enable UAVs to learn how to behave by interacting with the environment without plant and disturbances models. Using Q-Learning, Hung and Givigi [32] developed a path planning algorithm for the fixed-wing UAV swarm and tested the algorithm in a stochastic environment. These path planning methods have their advantages, but many of them can hardly obtain smooth, traceable, and precise optimal paths within a limited time.

Benefiting from the advantages of calculating efficiency, many researchers have applied convex optimization algorithms to the coordinative path planning of spacecraft and UAVs. Augugliaro *et al.* [33] applied SCP to obtain coordinated paths for UAV swarm under consideration of collision-free constraints. Lu and Liu [34] used a lossless relaxation technique for the original nonlinear rendezvous path planning problem and SCP to solve a series of subproblems to obtain paths. However, these methods are centralized and can not fully utilize the limited computing resources of each UAV in the swarm. Therefore, similar centralized UAV rendezvous path planning methods are not suitable for large-scale UAV swarms. To improve further the solving efficiency, Chen *et al.* [35] developed a decoupled incremental SCP to generate the optimal path for each UAV. However, in most of these SCP-based coordinative path planning research, a flight time will be pre-specified, which will increase the time for the UAV swarm to complete the task, and may even lead to unsolved path planning problems. Wang *et al.* [36] took the shortest flight time as the objective function and proposed a coupled SCP and a decoupled SCP for the UAV formation rendezvous path planning problem. The decoupled SCP will decompose the original problem into several decoupled subproblems, which can be solved in parallel, but this method requires the flight time information of all UAVs during time coordination, so it is not completely decentralized. Ramalho *et al.* [37] proposed an online decentralized rendezvous path planning method based on SCP, which uses only local information to plan two-dimensional collision-free paths for the multi-robot system. However, this method does not take into account the presence of obstacles in the environment.

III. PROBLEM STATEMENT

This section reviews some fundamental mathematical tools, including graph theory and notations used in this paper. In addition, constraints such as the kinematics model of UAV and collision-free constraints are analyzed, and the optimal control model is formulated. Assuming that obstacles exist in the environment, n homogeneous UAVs start from different initial positions and assemble in the predetermined airspace, as shown in Figure 1.

A. PRELIMINARIES

The rest of the paper uses the following notations. $\boldsymbol{I}_a$ is the a-dimensional identity matrix, $\mathbf{1}_{a\times b}$ is the a-by-b all-one matrix, and $\mathbf{0}_{a\times b}$ is the a-by-b all-zero matrix. We use bold font to denote matrices or vectors, $\|\cdot\|$ to denote the Euclidean norm of a vector, $(\cdot)^\top$ to denote the transpose of a vector or a matrix, $\max(\cdot)$ to denote the maximum element in the set, $\min(\cdot)$ to denote the minimum element in the set, $|\cdot|$ to denote the absolute value, g to denote the gravitational acceleration constant.

Ignoring the signal transmission delay, the UAV swarm communication network in this paper can be abstracted as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of nodes $\mathcal{V} = \{1, 2, \ldots, n\}$ and a group of edges $\mathcal{E}$. Nodes represent UAVs in the swarm and n is the number of UAVs in the swarm. Each edge $\{i, j\} \in \mathcal{E}$ is denoted by an unordered pair of distinct nodes. The edge set indicates whether an available communication connection exists between two UAVs:

$$\begin{cases} \{i,j\} \in \mathcal{E},\ 0 \le \|p_i - p_j\| \le r_{cmu} \\ \{i,j\} \notin \mathcal{E},\ otherwise \end{cases} \tag{1}$$

where i and j are the index of the UAV in the swarm, p_i, p_j are the position vectors of the ith UAV and jth UAV respectively, r_{cmu} is the maximum communication radius between two UAVs. The set of neighbors of node i is denoted by $\mathcal{N}_i = \{j | \{i,j\} \in \mathcal{E}\}$. The degree of node i is represented by $\deg(i)$, which is the number of neighbors of the node.

For the communication graph of the UAV swarm involved in this paper, we make the following standard assumptions.

Assumption 1: Since UAVs in the swarm are often not far apart, and the communication distance of the communication equipment can be broad, it is assumed that the communication graph $\mathcal{G}$ is connected.

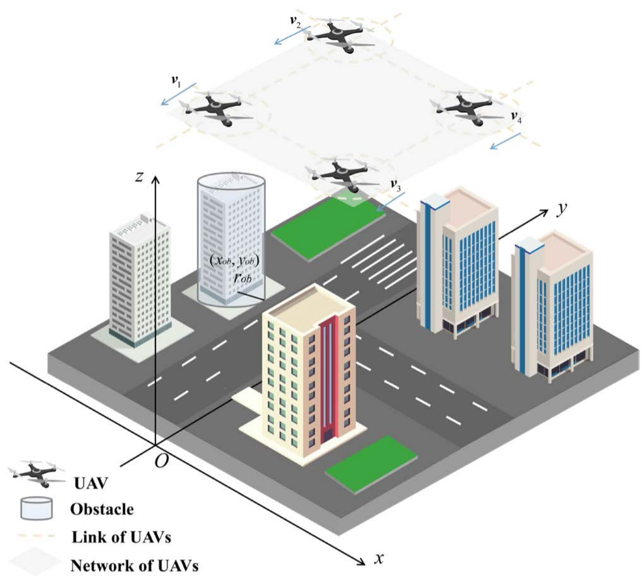

FIGURE 1. Schematic diagram of multi-UAV coordinative path planning in an obstacle environment.

B. DYNAMIC MODELING

The UAV considered in this paper is a type of multi-rotor UAV. Set $\boldsymbol{s} = \left[\boldsymbol{s}_1^\top, \boldsymbol{s}_2^\top, \ldots, \boldsymbol{s}_n^\top\right]^\top$ as the state matrix of the UAV swarm, and $\boldsymbol{u} = \left[\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top, \ldots, \boldsymbol{u}_n^\top\right]^\top$ as the control matrix of the UAV swarm, where $\boldsymbol{s}_i = \left[x_i, y_i, z_i, v_{ix}, v_{iy}, v_{iz}\right]^\top$ is the state vector of the ith UAV, $\boldsymbol{u}_i = \left[T_{ix}, T_{iy}, T_{iz}\right]^\top$ is the control vector of the ith UAV. To characterize the motion state of the UAV relative to the ground, the curvature of the earth is ignored and $e\,(Oxyz)$ is defined as the earth-fixed frame.

The origin O is located at a certain point in the departure area of the UAV swarm, Ox points east in the horizontal plane, Oz is perpendicular to the ground, and the right-hand rule defines Oy. In the earth-fixed frame $e\,(Oxyz)$, the simplified three-dimensional kinematics model of a single UAV can be expressed as follows:

$$\dot{\boldsymbol{s}}_i = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix} \boldsymbol{s}_i + \frac{1}{m_i} \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{u}_i + \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{g}, \quad (2)$$

where x_i, y_i, z_i are the position of the UAV in X, Y, and Z dimension in the Cartesian coordinates respectively, v_{ix}, v_{iy}, v_{iz} are the velocity of each dimension of the UAV, T_{ix}, T_{iy}, T_{iz} are the thrust of each dimension of the UAV, m_i is the mass of the UAV, $\boldsymbol{g} = [0, 0, -g]^\top$ is the gravitational acceleration vector.

C. CONSTRAINTS

1) PERFORMANCE CONSTRAINTS

To improve the feasibility of the planned path, the performance constraints for UAVs should be considered when planning, which gives

$$\begin{aligned} \|\boldsymbol{E}\boldsymbol{s}_i\| &\le v_{i,max}, \\ \|\boldsymbol{u}_i\| &\le T_{i,max}, \end{aligned} \quad (3)$$

where the role of $\boldsymbol{E} = [\boldsymbol{0}_{3\times3}, \boldsymbol{I}_3]$ is to extract the velocity information of the UAV, $v_{i\max}$ is the maximum flight speed of the ith UAV, and $T_{i\max}$ is the maximum thrust of the ith UAV. The above constraints are collectively referred to as performance constraints, which represent the maneuverability of UAVs and are used to avoid planning paths that the UAV cannot track. In addition, the constraints brought by the environment and tasks should also be considered when planning the paths.

2) OBSTACLE AVOIDANCE CONSTRAINTS

Many obstacles exist in complex environments such as cities or indoors. In practice, the shapes of obstacles are various, but they can be wrapped by a limited number of cylindrical obstacles. Although such an approximation would make the path planning problem more conservative, the obstacle is modeled as several cylindrical obstacle regions of infinite height for flight safety and to keep the planning algorithm simple [38], [39]. During the flight, all UAVs are restricted to stay outside the obstacle area all the time, which can be formulated as

$$\left\|\boldsymbol{F}\boldsymbol{s}_i - \boldsymbol{p}_{ob,m}\right\| \ge r_{ob,m} + r_{safe}, \quad (4)$$

where the role of $\boldsymbol{F} = [\boldsymbol{I}_2, \boldsymbol{0}_{2\times4}]$ is to extract the position in X and Y dimensions in the UAV state vector, $\boldsymbol{p}_{ob,m} = \left[x_{ob,m}, y_{ob,m}\right]^\top$ is the position of the center of the bottom surface of the cylindrical obstacle, $r_{ob,m}$ is the radius of the cylindrical bottom surface, r_{safe} is the safety radius of the UAV, $m \in \{1, 2, \ldots, M\}$ is the index of the obstacle area, and M is the number of obstacles in the environment.

3) FLIGHT ALTITUDE CONSTRAINTS

In addition to obstacles, the flight environment often brings flight altitude constraints for UAVs, such as the minimum flight altitude limit brought by the ground and the maximum flight altitude limit brought by the aviation control. Assuming that the minimum flight altitude is $h_{\min}$ and the maximum flight altitude is $h_{\max}$, the flight altitude constraints can be expressed as follows:

$$h_{\min} \le \boldsymbol{G}\boldsymbol{s}_i \le h_{\max}, \quad (5)$$

where the role of $\boldsymbol{G} = [0, 0, 1, 0, 0, 0]$ is to extract the position in the Z dimension in the UAV state vector.

4) COLLISION-FREE CONSTRAINTS

In multi-UAV path planning, the possible conflicts between UAVs should also be considered. If a collision exists between UAVs, it will not only affect the completion of the mission but also threaten the safety of other robots or humans in the environment. Therefore, it is necessary to introduce UAV collision-free constraints in the coordinative path planning, which gives

$$\left\|\boldsymbol{H}\boldsymbol{s}_i - \boldsymbol{H}\boldsymbol{s}_j\right\| \ge 2r_{safe}, \quad \forall j \in \mathcal{N}_i, \quad (6)$$

where $\boldsymbol{H} = [\boldsymbol{I}_3, \boldsymbol{0}_{3\times4}]$ is used to extract the position information of the UAV, and j is the index of another UAV in the swarm. Equation (6) shows that the ith UAV only avoids other UAVs within its communication range because UAVs outside the communication range are far away and will not pose a threat to the flight safety of the ith UAV.

5) INITIAL AND TERMINAL STATE CONSTRAINTS

In the rendezvous task, UAVs needs to arrive at the rendezvous area simultaneously from the different starting point and form in a predetermined formation. It can be assumed that the formation configuration to be formed is

$$\boldsymbol{P}_d = \left(\boldsymbol{p}_{1,d}, \boldsymbol{p}_{2,d}, \ldots, \boldsymbol{p}_{n,d}\right), \quad (7)$$

where $\boldsymbol{p}_{i,d} = \left(x_{i,d}, y_{i,d}, z_{i,d}\right)^\top$ is the desired position of the ith UAV in the predetermined formation. All UAVs need to reach their respective designated locations at the end of the designed paths. In addition, in order to ensure that the UAV swarm can fly in the predetermined formation after rendezvous, the velocity of UAVs at the terminal of the paths should also be the same. Here, assume that v_{fd} is the desired terminal velocity of the UAV swarm. To keep it concise, the constraints brought by the desired terminal position and the desired terminal velocity, together with the initial state constraints, can be expressed as the UAV initial and terminal state constraints. Assuming that UAVs depart at the same time, the constraint can be expressed as

$$\begin{aligned} \boldsymbol{H}\boldsymbol{s}_i(t_0) &= \boldsymbol{p}_{i,0}, \\ \boldsymbol{E}\boldsymbol{s}_i(t_0) &= \boldsymbol{v}_{i,0}, \\ \boldsymbol{H}\boldsymbol{s}_i\left(t_f\right) &= \boldsymbol{p}_{i,d}, \\ \boldsymbol{E}\boldsymbol{s}_i\left(t_f\right) &= \boldsymbol{v}_{fd}, \end{aligned} \quad (8)$$

where t_0 is the initial moment, t_f is the moment when the UAV reaches the desired position, $\boldsymbol{p}_{i,0}$ is the initial position of the ith UAV, and $\boldsymbol{v}_{i,0}$ is the initial velocity of the ith UAV.

D. PROBLEM FORMULATION

The coordinative path planning of UAVs can be formulated as an optimal control problem. The goal of this problem is to calculate the optimal control input so that UAVs in the swarm can arrive at the rendezvous area at the same time without any collisions. In this paper, to shorten the time of UAV rendezvous and save the energy of UAV, the weighted sum of the flight time and the square of the Euclidean norm of the control vector is selected as the objective function. Usually, choosing this objective function in a path planning problem can obtain a smoother path. To sum up, the multi-UAV coordinative path planning problem can be expressed as follows:

$$\begin{aligned} \text{P0}: \min\ J = nt_f + \sum_{i=1}^{n} a \int_{t_0}^{t_f} \|\boldsymbol{u}_i\|^2 \mathrm{d}t \\ s.t.\ (2)-(6), (8), \\ \forall i \in \mathcal{V}, \forall j \in \mathcal{N}_i, \quad \forall m \in \{1, 2, \ldots, M\}, \quad (9) \end{aligned}$$

where a is a constant coefficient.

IV. PATH PLANNING BASED ON THE TWO-LAYER COORDINATIVE FRAMEWORK

In this section, we propose a decentralized multi-UAV path planning method based on the two-layer coordinative framework. By consensus protocol and decomposition of constraints, P0 is decomposed into several sub-problems that can be solved in parallel by each UAV.

A. DECENTRALIZED COORDINATION AND DECOUPLING

The main difference between multi-UAV coordinated path planning and single UAV path planning is that UAVs in the swarm need to reach an agreement in a certain sense. In the considered UAV rendezvous task, v_{fd} and $\boldsymbol{P}_d$ can be pre-designated for the UAV swarm according to the mission requirements. However, the flight state of each UAV is unknown in advance, so it is difficult to pre-designate the desired flight time of the swarm. If the specified desired flight time is too large, energy may be wasted. If the desired flight time is too small, the problem may be unsolved. Therefore, a two-layer coordinative planning framework is adopted, which includes path planning at the bottom layer and coordinated planning at the top layer, as shown in Figure 2. The coordinated planning at the top layer is responsible for coordinating and managing the desired flight time of the UAV swarm, while the path planning at the bottom layer is completed by the local path planner scattered in each UAV.

In the coordination layer, information sharing is a necessary condition for multi-UAV coordination, and the minimum amount of information required to achieve coordination is called coordination variables [40], [41]. The concept of

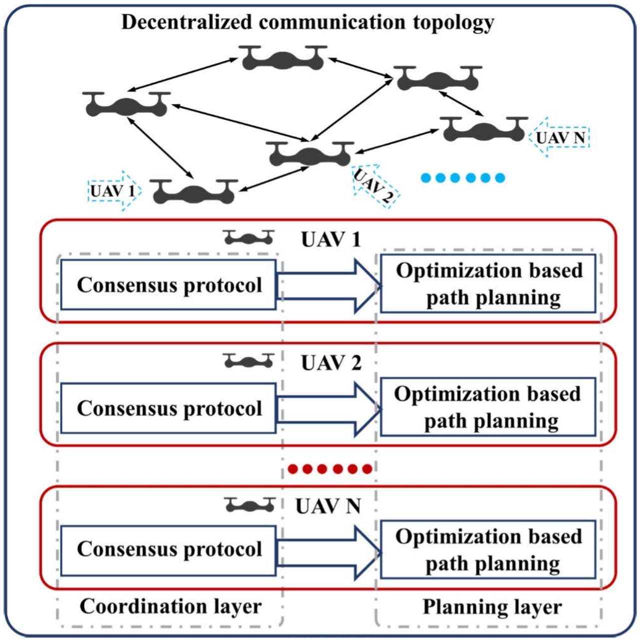

FIGURE 2. Schematic diagram of the two-layer coordinative path planning framework.

coordination variables can be found in many other works on coordination [42], [43]. For example, an "action reference" can be introduced, which would help maintain formation if every UAV in the swarm knew about it [44]. In the UAV rendezvous problem considered in this paper, the desired flight time t_{fd} is selected as the coordination variable, which defines mission-critical timing information. Set the average value $\left(1/n\right)\sum_{i=1}^{n}\hat{t}_{i,f}$ of the predicted flight time $\hat{t}_{i,f}$ of each UAV as the desired flight time t_{fd}, which is easy to achieve in a centralized communication network. However, in the decentralized communication network, the ith UAV can only communicate with UAVs around it, so it can not directly average all $\hat{t}_{i,f}$.

To reach a consensus on the desired flight time of each UAV, we employ a linear consensus protocol. Suppose the predicted flight time $\hat{t}_{i,f}$ is the initial value $t_{i,fd}(0)$ of the desired flight time of the ith UAV. At each step, the ith UAV exchanges the desired flight time with other UAVs around it. After several steps, each UAV gets all the initial values of all UAVs, so the average can be computed. The update formula for the desired flight time of the ith UAV is as follows:

$$t_{i,fd}(h+1) = W_{ii}t_{i,fd}(h) + \sum_{j\in\mathcal{N}_i} W_{ij}t_{j,fd}(h), \quad h = 0, 1, 2, \ldots \quad (10)$$

where h is the number of iterations, W_{ii}, W_{ij} is the weight on $t_{i,fd}$ and $t_{j,fd}$ at ith UAV, respectively. Set $W_{ij} = 0$ for $j \notin \mathcal{N}_i$, then all weights can form a weight matrix $\boldsymbol{W}$. For $t_{i,fd}(h)$ to quickly converge to the average value $\left(1/n\right)\sum_{i=1}^{n}\hat{t}_{i,f}$, a weight matrix with local-degree weights is chosen.

This method is to assign the weight according to the larger degree of two incident UAVs, as follows [45]:

$$W_{ij} = \frac{1}{\max\{\deg(i), \deg(j)\}}, \{i, j\} \in \mathcal{E}, \tag{11}$$

and then determine W_{ii} according to $\boldsymbol{W}\mathbf{1}_{n\times 1} = \mathbf{1}_{n\times 1}$. Since the weights at the ith UAV only depend on the degrees of the ith UAV and its neighbors, this method is very suitable for decentralized averaging. For the convenience of solving, a relatively simple consensus protocol is adopted here. To improve the performance of a certain aspect or complete a specific task, some more suitable consensus protocols can be designed [46], [47].

The planning layer consists of several local path planners, so the problem P0 needs to be decoupled and divided into several sub-problems that can be solved in parallel. It can be noticed that the objective function of problem P0 is additive. In fact, the team objective function is composed of the individual objective functions of each UAV in the swarm. The objective function and most of the constraints of P0 can be split directly, but the collision-free constraints involve not only the position information of the ith UAV itself but also the position of the jth UAV. During the path planning, the real position of the jth UAV is unknown because it is what will happen in the future. Therefore, the predicted state $\hat{\boldsymbol{s}}_j$ of the jth UAV is used instead of the real state $\boldsymbol{s}_j$, and the decoupled UAV collision-free constraint is expressed as follows:

$$\left\|\boldsymbol{H}\boldsymbol{s}_i - \boldsymbol{H}\hat{\boldsymbol{s}}_j\right\|_2 \geq 2r_{safe}, \quad \forall j \in \mathcal{N}_i. \tag{12}$$

After decoupling the problem P0, the path of each UAV is planned independently, so the flight time of different UAVs is likely to be different. For the swarm to achieve time coordination, the time coordination constraints should be considered when planning the path, as follows:

$$\left|t_{i,f} - t_{i,fd}\right| \leq \varepsilon_t, \tag{13}$$

where ε_t is a variable tolerance, which is used to enable the flight time to be progressively smaller in iterations. The specific value of ε_t will be described in detail later. Time coordination constraints, when satisfied, can be used to define the occurrence of coordination. In summary, the overall path planning problem is decomposed into several sub-problems $\mathrm{P0}_i$.

$$\begin{aligned}
\mathrm{P0}_i:\ \ & \min J_i = t_{i,f} + a\int_{t_0}^{t_{i,f}} \|\boldsymbol{u}_i\|^2 \mathrm{d}t \\
& s.t.\ (2)-(5), (12), (13), \\
& \boldsymbol{H}\boldsymbol{s}_i(t_0) = \boldsymbol{p}_{i,0}, \\
& \boldsymbol{E}\boldsymbol{s}_i(t_0) = \boldsymbol{v}_{i,0}, \\
& \boldsymbol{H}\boldsymbol{s}_i\left(t_{i,f}\right) = \boldsymbol{p}_{i,d}, \\
& \boldsymbol{E}\boldsymbol{s}_i\left(t_{i,f}\right) = \boldsymbol{v}_{fd}, \\
& \forall j \in \mathcal{N}_i, \quad \forall m \in \{1, 2, \ldots, M\},
\end{aligned} \tag{14}$$

where $t_{i,f}$ is the flight time of the ith UAV. In this paper, SCP is used to solve this set of sub-problems.

SCP is a local optimization method that can be used to solve non-convex optimization problems. SCP transforms the original non-convex optimization problem into a series of convex optimization sub-problems for an iterative solution, to obtain the approximate optimal solution of the original problem. Convex optimization theory has shown that as long as an optimization problem can be represented as a convex form, the global optimal solution of the problem can be obtained in polynomial time [48]. Problem $\mathrm{P0}_i$ is a typical non-convex optimization problem. If the SCP method is applied to solve it, the nonlinear and non-convex constraints need to be discretized and convex to establish the approximate convex problem of $\mathrm{P0}_i$.

B. CONVEXIFICATION

$\mathrm{P0}_i$ is a free-final-time optimal control problem. The flight time $t_{i,f}$ of the UAV is unknown, so the dynamic equation in $\mathrm{P0}_i$ cannot be discretized. Therefore, the flight time domain $\left[t_0, t_{i,f}\right]$ is mapped to the given interval [0, 1], as follows:

$$\tau_i = \frac{t - t_0}{t_{i,f} - t_0}, \quad t \in \left[t_0, t_{i,f}\right], \tag{15}$$

where τ_i is the new independent variable. After replacing the independent variables, (2) and (8) become

$$\begin{aligned}
\dot{\boldsymbol{s}}_i = t_{i,f}\begin{bmatrix} \mathbf{0}_{3\times 3} & \boldsymbol{I}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}\boldsymbol{s}_i + \frac{t_{i,f}}{m_i}\begin{bmatrix} \mathbf{0}_{3\times 3} \\ \boldsymbol{I}_{3\times 3} \end{bmatrix} \\
\times \boldsymbol{u}_i + t_{i,f}\begin{bmatrix} \mathbf{0}_{3\times 3} \\ \boldsymbol{I}_{3\times 3} \end{bmatrix}\boldsymbol{g},
\end{aligned} \tag{16}$$

$$\begin{aligned}
& \boldsymbol{H}\boldsymbol{s}_i(0) = \boldsymbol{p}_{i,0}, \\
& \boldsymbol{E}\boldsymbol{s}_i(0) = \boldsymbol{v}_{i,0}, \\
& \boldsymbol{H}\boldsymbol{s}_i(1) = \boldsymbol{p}_{i,d}, \\
& \boldsymbol{E}\boldsymbol{s}_i(1) = \boldsymbol{v}_{fd}.
\end{aligned} \tag{17}$$

However, after changing the independent variables, the original convex objective function becomes non-convex, so it also needs to be convex. The convexification process and results of the objective function are given by

$$\begin{aligned}
J_i &= t_{i,f} + a\int_0^1 \|\boldsymbol{u}_i\|^2 t_{i,f}\mathrm{d}\tau, \\
&\Leftrightarrow \begin{cases} J_i = t_{i,f} + a\int_0^1 \alpha_{i,1}\mathrm{d}\tau, \\ \|\boldsymbol{u}_i\|^2 \leq \dfrac{\alpha_{i,1}}{t_{i,f}}, \end{cases} \\
&\Leftrightarrow \begin{cases} J_i = t_{i,f} + a\int_0^1 \alpha_{i,1}\mathrm{d}\tau, \\ \|\boldsymbol{u}_i\|^2 \leq 2\alpha_{i,1}\alpha_{i,2}, \alpha_{i,2} \geq 0, \\ 2\alpha_{i,2} - \dfrac{1}{t_{i,f}} \leq 0, \end{cases} \\
&\Leftrightarrow \begin{cases} J_i = t_{i,f} + a\int_0^1 \alpha_{i,1}\mathrm{d}\tau, \\ \|\boldsymbol{u}_i\|^2 \leq 2\alpha_{i,1}\alpha_{i,2}, \alpha_{i,2} \geq 0, \\ \left(2\alpha_{i,2}^* - \dfrac{1}{t_{i,f}^*}\right) + \begin{bmatrix} 2 & \dfrac{1}{t_{i,f}^{*2}} \end{bmatrix}\begin{bmatrix} \alpha_{i,2} - \alpha_{i,2}^* \\ t_{i,f} - t_{i,f}^* \end{bmatrix} \leq 0, \\ \left|\alpha_{i,2} - \alpha_{i,2}^*\right| \leq \rho_1, \\ \left|t_{i,f} - t_{i,f}^*\right| \leq \rho_2, \end{cases}
\end{aligned} \tag{18}$$

where ρ_1 and ρ_2 are trust regions. After changing the independent variable, the dynamic equation is still nonlinear, so the dynamic equation needs to be linearized at the nominal solution. In the SCP method, an initial guess is usually selected as the initial nominal solution, and the solution from the previous iteration is taken as the nominal solution for this iteration. Assuming that this iteration is the kth iteration, the state vector $\boldsymbol{s}_i^{(k-1)}$, control vector $\boldsymbol{u}_i^{(k-1)}$, and flight time $t_{i,f}^{(k-1)}$ obtained from the (k-1)th iteration will be used as the nominal solutions $\boldsymbol{s}_i^*$, $\boldsymbol{u}_i^*$, and $t_{i,f}^*$ for this iteration. According to the nominal state $\boldsymbol{s}_i^*$, the nominal control vector $\boldsymbol{u}_i^*$, and the nominal flight time $t_{i,f}^*$, the approximate dynamic equation after linearization is obtained as follows:

$$\dot{\boldsymbol{s}}_i = \boldsymbol{A}_i(t_{i,f}^*)_i\boldsymbol{s}_i + \boldsymbol{B}_i(t_{i,f}^*)\boldsymbol{u}_i + \boldsymbol{C}_i(\boldsymbol{s}_i^*, \boldsymbol{u}_i^*)t_{i,f} + \boldsymbol{D}_i(\boldsymbol{s}_i^*, \boldsymbol{u}_i^*, t_{i,f}^*), \quad (19)$$

where the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are as follows:

$$\begin{aligned} \boldsymbol{A}_i(t_{i,f}^*) &= t_{i,f}^* \begin{bmatrix} 0_{3\times3} & \boldsymbol{I}_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \\ \boldsymbol{B}_i(t_{i,f}^*) &= \frac{t_{i,f}^*}{m_i} \begin{bmatrix} 0_{3\times3} \\ \boldsymbol{I}_{3\times3} \end{bmatrix}, \\ \boldsymbol{C}_i(\boldsymbol{s}_i^*, \boldsymbol{u}_i^*) &= \begin{bmatrix} \boldsymbol{E}\boldsymbol{s}_i^* \\ \frac{1}{m_i}\boldsymbol{u}_i^* + \boldsymbol{g} \end{bmatrix}, \\ \boldsymbol{D}_i(\boldsymbol{s}_i^*, \boldsymbol{u}_i^*, t_{i,f}^*) &= -t_{i,f}^* \begin{bmatrix} \boldsymbol{E}\boldsymbol{s}_i^* \\ \frac{1}{m_i}\boldsymbol{u}_i^* \end{bmatrix}. \end{aligned} \quad (20)$$

The obstacle avoidance constraints in the problem are concave. Linearization is an effective convexity method for a class of non-convex optimization problems with concave constraints [49]. According to the nominal state $\boldsymbol{s}_i^*$, the obstacle avoidance constraint is linearized, and the approximate constraint is obtained as follows:

$$\left\|\boldsymbol{F}\boldsymbol{s}_i^* - \boldsymbol{p}_{ob,m}\right\| + \frac{\left(\boldsymbol{F}\boldsymbol{s}_i^* - \boldsymbol{p}_{ob,m}\right)^\top}{\left\|\boldsymbol{F}\boldsymbol{s}_i^* - \boldsymbol{p}_{ob,m}\right\|}\left(\boldsymbol{F}\boldsymbol{s}_i - \boldsymbol{F}\boldsymbol{s}_i^*\right) \geq r_{ob,m} + r_{safe}, \quad (21)$$

When dealing with collision-free constraints, the nominal state $\boldsymbol{s}_j^*$ of the jth UAV is chosen as its predicted state $\hat{\boldsymbol{s}}_j$, that is, only avoiding the paths of other UAVs obtained in the previous iteration. In the SCP, the result of the iterative calculation will gradually converge to a path, so when the algorithm converges, the processed collision-free constraints are equivalent to the initial constraints. At the nominal states $\boldsymbol{s}_i^*$ and $\boldsymbol{s}_j^*$, the collision-free constraints are linearized as follows:

$$\left\|\boldsymbol{H}\boldsymbol{s}_i^* - \boldsymbol{H}\boldsymbol{s}_j^*\right\| + \frac{\left(\boldsymbol{H}\boldsymbol{s}_i^* - \boldsymbol{H}\boldsymbol{s}_j^*\right)^\top}{\left\|\boldsymbol{H}\boldsymbol{s}_i^* - \boldsymbol{H}\boldsymbol{s}_j^*\right\|}\left(\boldsymbol{H}\boldsymbol{s}_i - \boldsymbol{H}\boldsymbol{s}_i^*\right) \geq 2r_{safe}, \quad (22)$$

where $j \in \mathcal{N}_i$. To ensure the effectiveness of linearization, trust-region constraints are introduced when solving approximate path planning problem.

$$\begin{aligned} \left|x_i - x_i^*\right| &\leq \rho_3, \\ \left|y_i - y_i^*\right| &\leq \rho_4, \\ \left|z_i - z_i^*\right| &\leq \rho_5, \\ \left|v_{ix} - v_{ix}^*\right| &\leq \rho_6, \\ \left|v_{iy} - v_{iy}^*\right| &\leq \rho_7, \\ \left|v_{iz} - v_{iz}^*\right| &\leq \rho_8, \end{aligned} \quad (23)$$

where $\rho_3 - \rho_8$ are trust regions. The above trust-region constraints are affine functions of the state variable, which do not affect the convexity of the problem.

C. DISCRETIZATION

Since computers cannot handle continuous-time problems, it is necessary to discretize the approximate path planning problem. Select the time step $\Delta\tau = 1/N$ to get $N+1$ discrete times $\tau_0, \tau_1, \ldots, \tau_N$, which satisfies $\tau_q = \tau_0 + q\Delta\tau, q \in \{0, 1, 2, \ldots, N\}$. Using the trapezoidal method to discretize the convex approximation problem P0$_i$, the continuous-time dynamic equation can be expressed as the following linear equation constraints on discrete-time state and control variables.

$$\begin{aligned} \boldsymbol{s}_{i,q} = \boldsymbol{s}_{i,q-1} + \frac{\Delta\tau}{2} &\times \big[\big(\boldsymbol{A}_{i,q-1}\boldsymbol{s}_{i,q-1} + \boldsymbol{B}_{i,q-1}\boldsymbol{u}_{i,q-1} + \boldsymbol{C}_{i,q-1}t_{i,f} + \boldsymbol{D}_{i,q-1}\big) \\ &+ \big(\boldsymbol{A}_{i,q}\boldsymbol{s}_{i,q} + \boldsymbol{B}_{i,q}\boldsymbol{u}_{i,q} + \boldsymbol{C}_{i,q}t_{i,f} + \boldsymbol{D}_{i,q}\big)\big], \end{aligned} \quad (24)$$

where $\boldsymbol{s}_{i,q} = \boldsymbol{s}_i(\tau_q)$ is the state vector of the UAV at time τ_q, $\boldsymbol{u}_{i,q} = \boldsymbol{u}_i(\tau_q)$ is the control vector of the UAV at time τ_q, $\boldsymbol{A}_{i,q} = \boldsymbol{A}\left(t_{i,f}^*\right)$, $\boldsymbol{B}_{i,q} = \boldsymbol{B}\left(t_{i,f}^*\right)$, $\boldsymbol{C}_{i,q} = \boldsymbol{C}\left(\boldsymbol{s}_i^*(\tau_q), \boldsymbol{u}_i^*(\tau_q)\right)$, $\boldsymbol{D}_{i,q} = \boldsymbol{D}\left(\boldsymbol{s}_i^*(\tau_q), \boldsymbol{u}_i^*(\tau_q), t_{i,f}^*\right)$. The discretized approximate obstacle avoidance constraint can be expressed as

$$\frac{\left(\boldsymbol{F}\boldsymbol{s}_{i,q}^* - \boldsymbol{p}_{ob,m}\right)^\top}{\left\|\boldsymbol{F}\boldsymbol{s}_{i,q}^* - \boldsymbol{p}_{ob,m}\right\|}\left(\boldsymbol{F}\boldsymbol{s}_{i,q} - \boldsymbol{F}\boldsymbol{s}_{i,q}^*\right) + \left\|\boldsymbol{F}\boldsymbol{s}_{i,q}^* - \boldsymbol{p}_{ob,m}\right\| \geq r_{ob,m} + r_{safe}, \quad (25)$$

where $\boldsymbol{s}_{i,q}^* = \boldsymbol{s}_i^*(\tau_q)$ is the nominal state at time τ_q. The discretized approximate collision avoidance constraint can be expressed as

$$\frac{\left(\boldsymbol{H}\boldsymbol{s}_{i,q}^* - \boldsymbol{H}\boldsymbol{s}_{j,q}^*\right)^\top}{\left\|\boldsymbol{H}\boldsymbol{s}_{i,q}^* - \boldsymbol{H}\boldsymbol{s}_{j,q}^*\right\|}\left(\boldsymbol{H}\boldsymbol{s}_{i,q} - \boldsymbol{H}\boldsymbol{s}_{i,q}^*\right) + \left\|\boldsymbol{H}\boldsymbol{s}_{i,q}^* - \boldsymbol{H}\boldsymbol{s}_{j,q}^*\right\| \geq 2r_{safe}, \quad (26)$$

The objective function and its derived constraints can be discretized as follows:

$$J_i = t_{i,f} + a\Delta\tau \sum_{q=0}^{N} \alpha_{i,1,q}, \tag{27}$$

$$\begin{cases} \left\|\boldsymbol{u}_{i,q}\right\|^2 \le 2\alpha_{i,1,q}\alpha_{i,2,q},\ \alpha_{i,2,q} \ge 0, \\ \left(2\alpha^*_{i,2,q} - \dfrac{1}{t_{i,f}}^{*}\right) + \begin{bmatrix} 2 & \dfrac{1}{t^{*\,2}_{i,f}} \end{bmatrix} \begin{bmatrix} \alpha_{i,2,q} - \alpha^*_{i,2,q} \\ t_{i,f} - t^*_{i,f} \end{bmatrix} \le 0, \\ \left|\alpha_{i,2,q} - \alpha^*_{i,2,q}\right| \le \rho_{i,1}, \\ \left|t_{i,f} - t^*_{i,f}\right| \le \rho_{i,2}, \end{cases} \tag{28}$$

where $\alpha_{i,1,q}$ and $\alpha_{i,2,q}$ are auxiliary variables at time τ_q. The performance constraints, flight altitude constraints, trust-region constraints, and initial/terminal state constraints can be expressed as follows:

$$\left\|\boldsymbol{E}\boldsymbol{s}_{i,q}\right\| \le v_{i,max}, \tag{29}$$

$$\left\|\boldsymbol{u}_{i,q}\right\| \le T_{i,max},$$

$$h_{\min} \le \boldsymbol{G}\boldsymbol{s}_{i,q} \le h_{\max}, \tag{30}$$

$$\begin{aligned} \left|x_{i,q} - x^*_{i,q}\right| &\le \rho_{i,3}, \\ \left|y_{i,q} - y^*_{i,q}\right| &\le \rho_{i,4}, \\ \left|z_{i,q} - z^*_{i,q}\right| &\le \rho_{i,5}, \\ \left|v_{ix,q} - v^*_{ix,q}\right| &\le \rho_{i,6}, \\ \left|v_{iy,q} - v^*_{iy,q}\right| &\le \rho_{i,7}, \\ \left|v_{iz,q} - v^*_{iz,q}\right| &\le \rho_{i,8}. \end{aligned} \tag{31}$$

$$\begin{aligned} \boldsymbol{H}\boldsymbol{s}_{i,0} &= \boldsymbol{p}_{i,0}, \\ \boldsymbol{E}\boldsymbol{s}_{i,0} &= \boldsymbol{v}_{i,0}, \\ \boldsymbol{H}\boldsymbol{s}_{i,N} &= \boldsymbol{p}_{i,d}, \\ \boldsymbol{E}\boldsymbol{s}_{i,N} &= \boldsymbol{v}_{fd}, \end{aligned} \tag{32}$$

To sum up, in the kth iteration, the discretized convex approximation sub-problem $\mathrm{P1}_i^{(k)}$ can be expressed as

$$\begin{aligned} \mathrm{P1}_i^{(k)}:\ \min J_i^{(k)} &= t_{i,f} + a\Delta\tau \sum_{q=0}^{N} \alpha_{1,q} \\ s.t.\ &(13), (24)-(33), \\ &\forall j \in \mathcal{N}_i, \quad \forall m \in \{1, 2, \ldots, M\}, \\ &\forall q \in \{0, 1, 2, \ldots, N\}. \end{aligned} \tag{33}$$

Then the team objective function of the multi-UAV path planning problem in the kth iteration is $J^{(k)} = \sum_{i=1}^{n} J_i^{(k)}$.

D. SEQUENTIAL CONVEX PROGRAMMING

The SCP method converts the original non-convex optimization problem into a series of sub-problems and obtains the approximate optimal solution of the original problem by iteratively calculating these convex approximate sub-problems. During the iterations of SCP, the rendezvous paths are sequentially refined, and the additional conservatism brought by the convex approximation is gradually eliminated. Compared with the basic convex optimization method, the SCP method reduces the dependence on the initial guess. In practice, the SCP method can always obtain a feasible solution at a lower cost, which is suitable for the rapid planning of the UAV flight path. The proposed algorithm flow is shown in Figure 3.

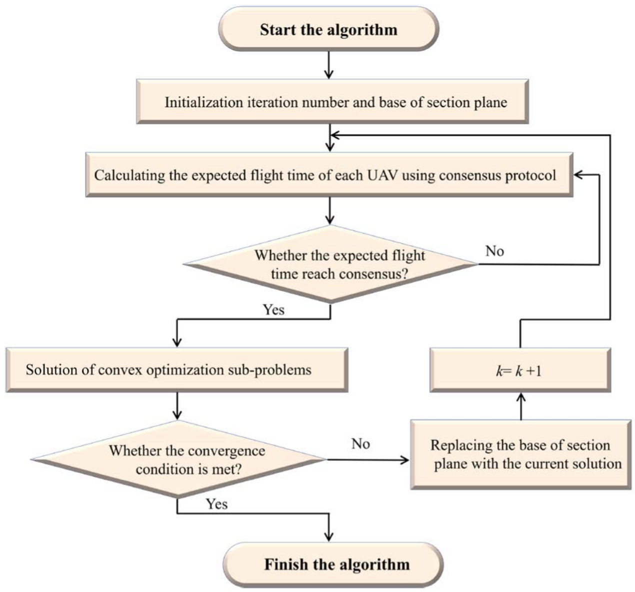

FIGURE 3. Flowchart of the proposed method.

In the SCP method, the initial guesses $\boldsymbol{s}^{(0)}$, $\boldsymbol{u}^{(0)}$, and $\boldsymbol{t}_f^{(0)} = [t_{1,f}^{(0)}, t_{2,f}^{(0)}, \ldots, t_{n,f}^{(0)}]^\top$ will be used as the nominal solutions for the first iteration, which will be used to convex the non-convex optimization problem, and the solutions $\boldsymbol{s}_i^{(k)}$, $\boldsymbol{u}_i^{(k)}$, and $t_{i,f}^{(k)}$ generated by each iteration will be used as the nominal solutions for the next iteration.

In each iteration, the coordination layer will calculate the coordination variables $t_{i,fd}$ of the swarm based on the nominal solutions $t^*_{i,f}$ and $t^*_{j,f}$, $\forall j \in \mathcal{N}_i$ The UAV swarm will independently negotiate the desired flight time, which not only improves the autonomy of planning but also reduces the possibility of no solution to the optimization problem. To ensure the convergence of the iteration while shortening $t_{i,f}$, ε_t in (13) should decrease as the iteration proceeds. Therefore, in the first iteration, give ε_t a value large enough to ensure that the first iteration is not unsolvable. Then, in the kth iteration, ε_t is assigned a value according to the $t_{i,f}^{(k-1)}$ and the $t_{i,f}^{(k-2)}$, as follows:

$$\varepsilon_t = \left|t_{i,f}^{(k-2)} - t_{i,f}^{(k-1)}\right|, \tag{34}$$

where $k \ge 2$. As the iteration proceeds, the solution of the convex approximation problem will gradually approach the optimal solution of the original problem. For the free-final-time path planning problem, the stopping condition of the iteration can be set as the maximum deviation between the paths obtained by the two iterations before and after, that is, when the path obtained in this iteration basically coincides with the path obtained in the previous iteration and the flight time of each UAV is basically the same, the algorithm can be considered to converge. In fact, to balance the optimality and rapidity of path planning, the convergence conditions of iteration can be appropriately relaxed, to plan a feasible sub-optimal path in a relatively short time. The pseudocode of the coordinative path planning algorithm is shown in Algorithm 1. Introduce the following sets:

$$\begin{aligned} \mathcal{X} &= \{\max(X_1), \max(X_2), \ldots, \max(X_n)\}, \\ \mathcal{Y} &= \{\max(Y_1), \max(Y_2), \ldots, \max(Y_n)\}, \\ \mathcal{Z} &= \{\max(Z_1), \max(Z_2), \ldots, \max(Z_n)\}, \\ \mathcal{T} &= \left\{\left|t_{1,f}^{(k)} - t_{1,f}^{(k-1)}\right|, \left|t_{2,f}^{(k)} - t_{2,f}^{(k-1)}\right|, \ldots, \left|t_{n,f}^{(k)} - t_{n,f}^{(k-1)}\right|\right\}, \end{aligned} \tag{35}$$

where

$$\begin{aligned} X_i &= \left\{\left|x_{i,0}^{(k)} - x_{i,0}^{(k-1)}\right|, \left|x_{i,1}^{(k)} - x_{i,1}^{(k-1)}\right|, \ldots, \left|x_{i,N}^{(k)} - x_{i,N}^{(k-1)}\right|\right\}, \\ Y_i &= \left\{\left|y_{i,0}^{(k)} - y_{i,0}^{(k-1)}\right|, \left|y_{i,1}^{(k)} - y_{i,1}^{(k-1)}\right|, \ldots, \left|y_{i,N}^{(k)} - y_{i,N}^{(k-1)}\right|\right\}, \\ Z_i &= \left\{\left|z_{i,0}^{(k)} - z_{i,0}^{(k-1)}\right|, \left|z_{i,1}^{(k)} - z_{i,1}^{(k-1)}\right|, \ldots, \left|z_{i,N}^{(k)} - z_{i,N}^{(k-1)}\right|\right\}. \end{aligned} \tag{36}$$

Algorithm 1 SCP Algorithm for Coordination of UAVs at kth Step

Initialization: Maximum number $k_{\max}$ of iterations for the SCP method, Maximum flight velocity $v_{i\max}$ and maximum thrust $T_{i\max}$ of all UAVs $i \in \mathcal{V}$, Minimum flight altitude $h_{\min}$ and maximum flight altitude $h_{\max}$, UAV safety radius r_{safe}, $\boldsymbol{p}_{ob,m}$ and $r_{ob,m}$ of all obstacles, predetermined formation $\boldsymbol{P}_d$, desired terminal velocity v_{fd}, accuracy $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)^\top$. Set the number of iterations $k = 1$. Design initial guesses $\boldsymbol{s}^{(0)}, \boldsymbol{u}^{(0)}$, and $\boldsymbol{t}_f^{(0)}$ as nominal solutions $\boldsymbol{s}^*, \boldsymbol{u}^*, \boldsymbol{t}_f^*$. The initially desired flight time $t_{fd}^{(0)}$ is determined according to the initial flight time guess value $\boldsymbol{t}_f^{(0)}$. Give ε_t a value large enough.

while $\max(\mathcal{X}) \le \varepsilon_1$ **and** $\max(\mathcal{Y}) \le \varepsilon_2$ **and** $\max(\mathcal{Z}) \le \varepsilon_3$ **and** $\max(\mathcal{T}) \le \varepsilon_4$ **do**

Calculate the coordination variables $t_{i,fd}$.
Solve each convex optimization sub-problem $\mathrm{P1}_i^k$ separately, and obtain the solutions $\boldsymbol{s}_i^{(k)}, \boldsymbol{u}_i^{(k)}$, and $t_{i,f}^{(k)}$ of each sub-problem, so as to indirectly obtain the solutions $\boldsymbol{s}^{(k)}, \boldsymbol{u}^{(k)}, \boldsymbol{t}_f^{(k)}$.
Taking $\boldsymbol{s}^{(k)}, \boldsymbol{u}^{(k)}$, and $\boldsymbol{t}_f^{(k)}$ as nominal solutions $\boldsymbol{s}^*, \boldsymbol{u}^*$, and $\boldsymbol{t}_f^*$ separately.
Calculate $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ and $\mathcal{T}$.
$k = k + 1$.
Calculate ε_t.

end while

V. NUMERICAL RESULTS

Numerical simulation is carried out for the multi-UAV coordinative path planning problem to evaluate the proposed coordinative path planning algorithm. The scenario used to test the effectiveness of the algorithm involves five UAVs. UAVs start with different initial conditions and are scheduled to form a diamond-shaped formation in the area near (60 m, 60 m, 60 m), thereby simulating the formation rendezvous task of the UAV swarm in an obstacle environment. The initial and final state of UAVs are shown in TABLE 1, and the desired flight velocity of the swarm is (2 m/s, 2 m/s, 0 m/s).

TABLE 1. The initial and final state of the UAV swarm.

UAV identifier	Initial state (m, m, m, m/s, m/s, m/s)	Final state (m, m, m, m/s, m/s, m/s)
UAV_1	[0, 0, 0, 0, 0, 0]	[60, 60, 60, 2, 2, 0]
UAV_2	[-10, 10, 0, 0, 0, 0]	[57, 57, 60, 2, 2, 0]
UAV_3	[-20 ,20, 0, 0, 0, 0]	[57, 63, 60, 2, 2, 0]
UAV_4	[20, -20, 0, 0, 0, 0]	[63, 57, 60, 2, 2, 0]
UAV_5	[10, -10, 0, 0, 0, 0]	[63, 63, 60, 2, 2, 0]

Two infinitely high cylindrical obstacles exist in the simulated scenario, and the UAV can only fly around and cannot cross from above. The coordinates and radius of the center of the obstacle's bottom surface are shown in TABLE 2. The mass of the UAV $m_i = 1$ kg, i=1,2, . . . ,5, the safety radius is 0.5 m, the maximum flight velocity is 10 m/s, and the maximum thrust is 15 N. The line connecting the starting point and the ending point is used as the initial guess path, and it is assumed that the control inputs in both X and Y dimensions are 0 and the control inputs in the Z dimension are $m_i g$, that is, UAVs fly at a constant velocity along this path. The simulation uses a fixed number of discrete points, set predefined number of intervals $N = 50$, taking 51 points with equal distances on the path as the initial guess of the UAV position, taking the terminal flight velocity (2 m/s, 2 m/s, 0 m/s) as the initial guess of UAVs' flight velocity, and calculates the guess value of UAVs' initial flight time. The initial guesses of the auxiliary variables are set to $\alpha_{i,1,q}^{(0)} = t_{i,f}^{(0)} \left\|\boldsymbol{u}_{i,q}^{(0)}\right\|^2$ and $\alpha_{i,2,1}^{(q)} = 1\big/t_{i,f}^{(0)}$, and the weight coefficient $a = 0.1$ is set. To balance the solution efficiency while ensuring that the approximate problem has a solution, the trust regions for the kth iteration are shown in TABLE 3. The convergence condition is set to $\boldsymbol{\varepsilon} = [0.1\text{m}, 0.1\text{m}, 0.1\text{m}, 0.01\text{s}]^\top$.

TABLE 2. Information about obstacles.

Obstacle identifier	Bottom circle center coordinates (m, m)	Radius (m)
Obstacle 1	[50, 15]	12
Obstacle 2	[20, 40]	10

The simulations are implemented on a computer with an Intel Core i7-10750H Processor and 16GB of RAM.

TABLE 3. The trust regions for the kth iteration.

trust region	ρ_1 (1/s)	ρ_2 (s)	ρ_3 (m)	ρ_4 (m)	ρ_5 (m)	ρ_6 (m/s)	ρ_7 (m/s)	ρ_8 (m/s)
Value	$\frac{1}{2^{k-1}}$	$\frac{50}{2^{k-1}}$	$\frac{60}{2^{k-1}}$	$\frac{60}{2^{k-1}}$	$\frac{60}{2^{k-1}}$	$\frac{10}{2^{k-1}}$	$\frac{10}{2^{k-1}}$	$\frac{10}{2^{k-1}}$

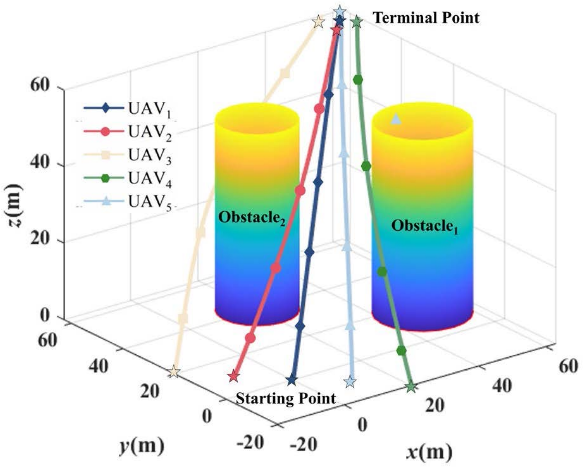

FIGURE 4. 3D view of the planned flight path.

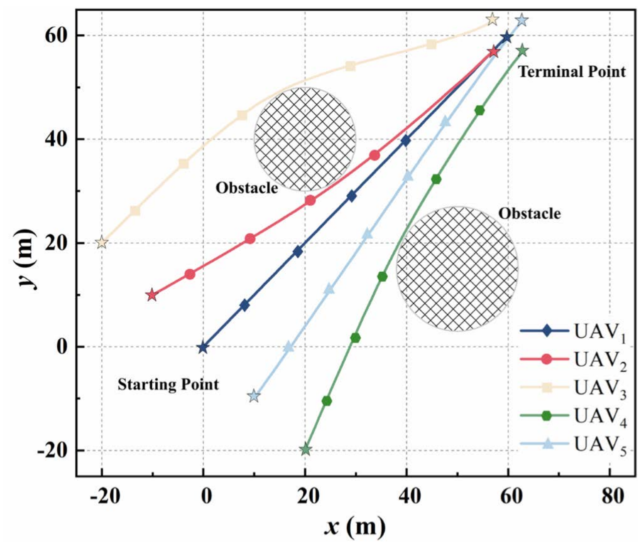

FIGURE 5. Top view of the planned flight path.

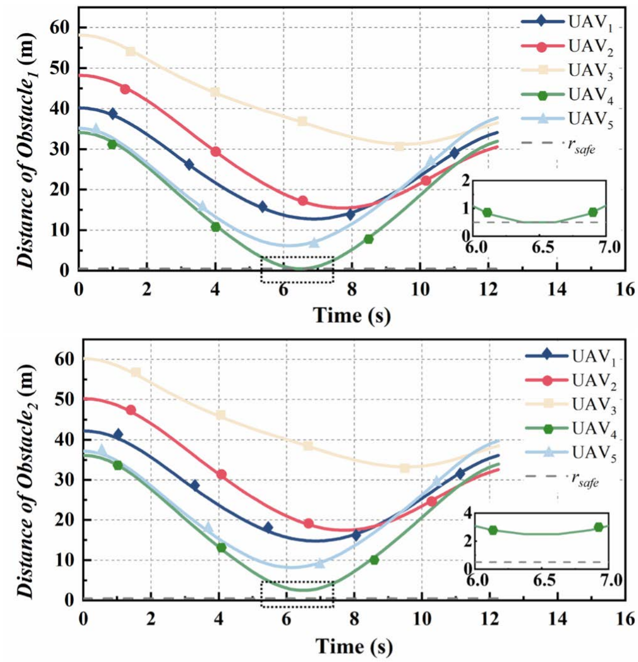

FIGURE 6. Distance of UAV from obstacle$_1$ and obstacle$_2$.

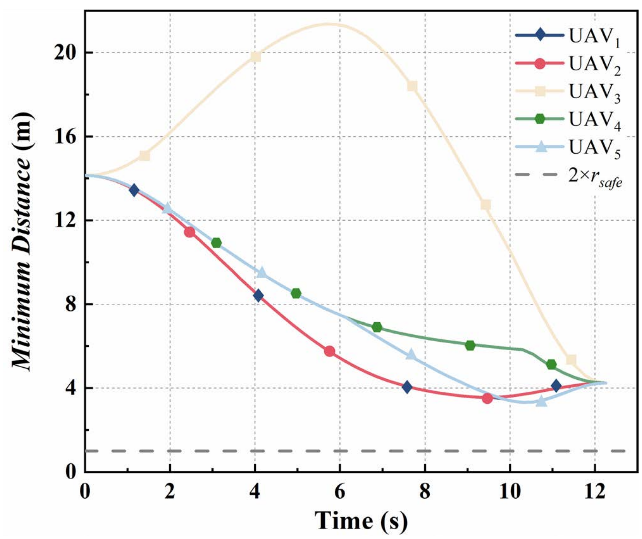

FIGURE 7. Minimum distance between the ith UAV and other UAVs.

The programming environment of numerical simulation is MATLAB 2020b, and the Parallel Computing Toolbox is used to solve each sub-problem $\mathrm{P1}_i^{(k)}$ in parallel. The YALMIP toolbox and MOSEK® solver are used for problem modeling and solving respectively. The flight paths obtained by the multi-UAV path planning algorithm based on the two-layer coordinative framework are shown in Figure 4 and Figure 5. The simulation results verified that the five UAVs started from different positions and finally formed the desired formation.

The distance between the UAV and the obstacle is shown in Figure 6, the minimum distance between the swarm and obstacle 1 and obstacle 2 is equal to or greater than the safety radius $r_{safe} = 0.5$ m, that is, all UAVs will not enter the obstacle area during flight, avoiding the collision between the swarm and the environment. Similarly, according to the simulation results shown in Figure 7, the minimum distance between the ith UAV and other UAVs in the swarm is greater than $2r_{safe}$, which meets the position coordination requirements for swarm flight.

The desired speed and desired thrust of the path are shown in Figure 8 and Figure 9 respectively. The algorithm

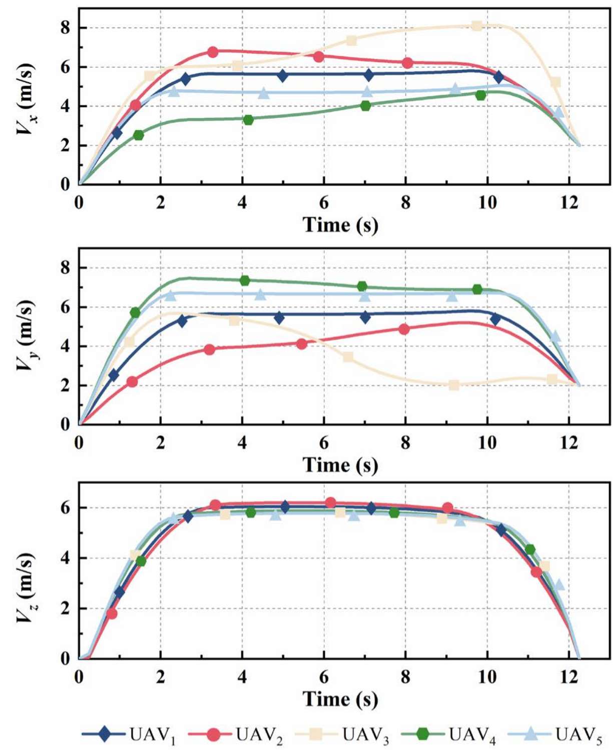

FIGURE 8. Desired flight velocity.

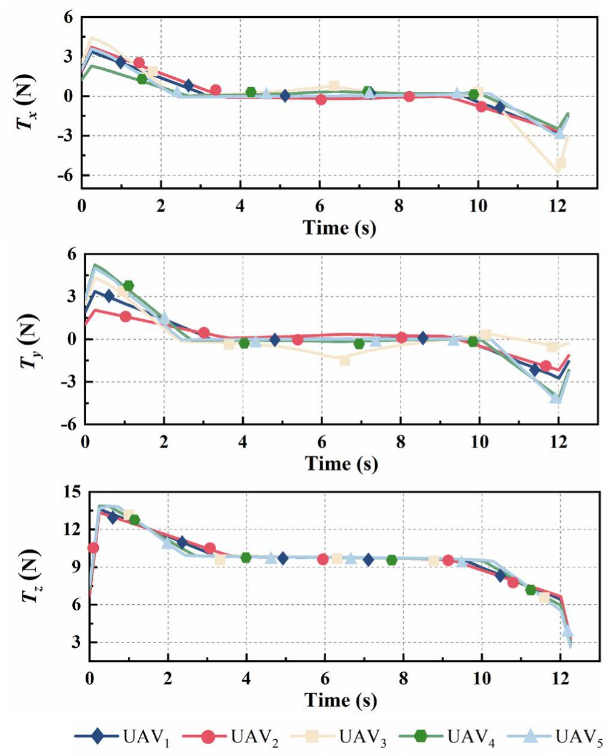

FIGURE 9. Desired thrust.

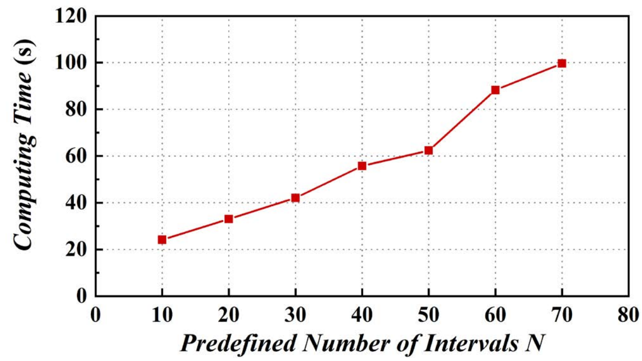

FIGURE 10. Relationship between computing time and predefined number of intervals N.

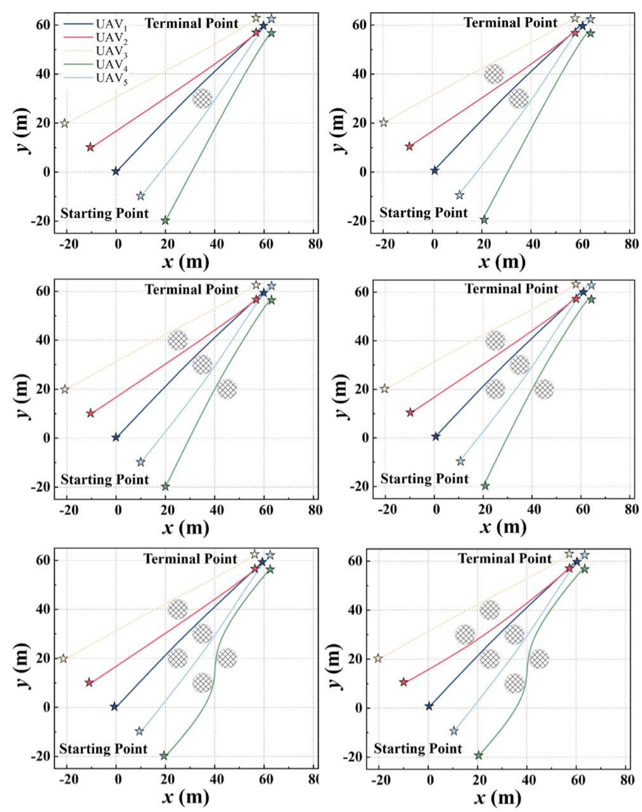

FIGURE 11. Simulation scenarios with different numbers of obstacles.

plans the desired control variables that satisfy the performance constraints for the five UAVs. The flight times of the five UAVs are 12.2558 s, 12.2559 s, 12.2594 s, 12.2594 s, and 12.2610 s respectively, and the maximum time difference is 0.0052 s. It can be considered that the five UAVs can reach the desired position at the same time, which meets the requirements of time coordination. Thus, the proposed method proves to be effective in solving the UAV formation rendezvous path planning problem.

A series of simulations are performed on the effect of the predefined number of intervals N and the number of obstacles on the proposed method. In the above simulation scenario, the effects of different N values on the computing time are shown in Figure 10. Here, the runtime of the simulation programs is used to represent the computing time. The simulation

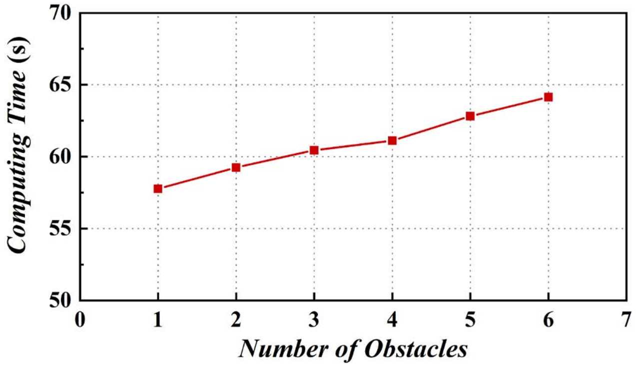

FIGURE 12. Relationship between computing time and the number of obstacles.

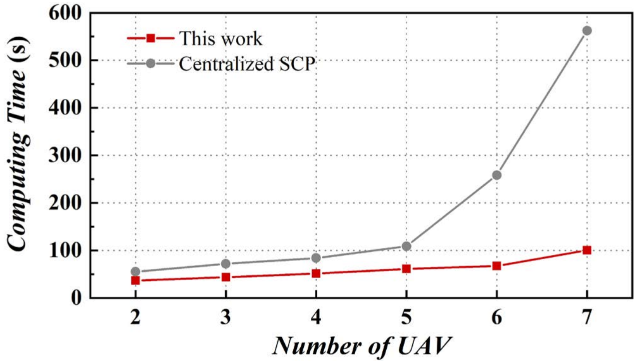

FIGURE 13. The comparison of computing time when using the proposed method and centralized SCP.

results show that with the increase of N, the computing time gradually becomes longer. However, this phenomenon is reasonable because the increase of N will lead to an increase in the scale of the sub-problem $\mathrm{P1}_i^{(k)}$, so it is necessary to select an appropriate number of intervals N in the application.

To compare the computing time of the proposed method under different numbers of obstacles, different numbers of cylindrical obstacles with a radius of 4 m are arranged in the above numerical simulation scenario, as shown in Figure 11. The relationship between computing time and the number of obstacles in Figure 12 shows that the computing time increases slightly with the number of obstacles. Furthermore, the computing time does not increase by more than 1.7 s for each additional obstacle. Therefore, to reduce the scale of the sub-problem, obstacles far from the UAV can be disregarded in path planning.

To demonstrate the scalability of the proposed method, we conduct comparative studies on the proposed decentralized path planning method and centralized SCP using the first simulation scenario on the same computer. Figure 13 displays the comparison of computing time when using the proposed method and centralized SCP, with different numbers of UAVs. The computing time of the centralized SCP is always longer than that of the proposed decentralized path planning method, and the computing time of the centralized SCP increases much faster with the increase of the number of UAVs.

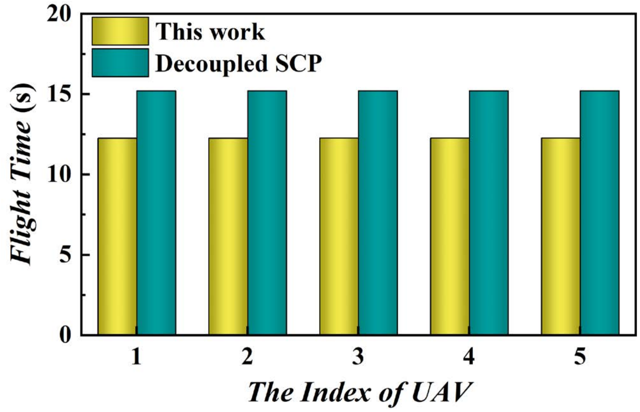

FIGURE 14. The comparison of flight time when using the proposed method and decoupled SCP.

Both the decoupled SCP [36] and the proposed method can solve the multi-UAV path planning sub-problems in parallel, and shortening the flight time is one of the optimization goals of both. In the initial simulation scenario of this chapter, the flight time of the paths planned by the two methods is compared. From Figure 14, paths planned by the proposed method have a shorter flight time than that planned by the decoupled SCP. Therefore, the proposed method has better optimality.

VI. CONCLUSION

This paper presents a decentralized multi-UAV path planning method with high computational efficiency and scalability. In the multi-UAV rendezvous task, all UAVs need to arrive at the designated position at the same time to form a formation and avoid collisions with obstacles or other UAVs in the swarm during flight. According to the mission requirements and physical constraints of formation rendezvous, a non-convex optimal control model is established for the coordinated path planning of multi-UAV. Then, a decentralized multi-UAV path planning method based on the two-layer coordinative framework is proposed, which decouples the original non-convex problem into several sub-problems that can be solved separately and uses the SCP method to solve them after linearization and discretization. In this method, the coordination layer uses the consensus protocol to calculate the coordination variable according to the nominal state of each UAV in the swarm. The planning layer is composed of path planners distributed in each UAV, which will plan a coordinated path according to the coordination variable. Finally, the effectiveness and scalability of the proposed method are verified by numerical simulation of a specific example. This method can realize swarm coordination and disperse the computational pressure on each UAV. Benefiting from the decentralized planning method, the increase in the number of UAVs will not significantly boost the solving time. Therefore, the method proposed in this paper is suitable for the cooperative path planning of large-scale UAV swarms.

The future work will focus on the problem of multi-UAV real-time path planning in a dynamic environment that includes other traffic participants. In this case, more consideration should be given to the signal transmission delay between UAVs and the interaction between the swarm and other traffic participants. Local path planning of UAV swarms in dynamic environments is also a technique that needs to be improved.

REFERENCES

[1] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 837–855, Aug. 2018.

[2] J. Kim, S. Kim, C. Ju, and H. I. Son, "Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications," *IEEE Access*, vol. 7, pp. 105100–105115, 2019.

[3] X. Shen, J. Sang, Y. Sun, and R. Liu, "Application of improved ant colony algorithm in distribution network patrol route planning," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Aug. 2016, pp. 560–563.

[4] S. K. K. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. G. Manyam, and D. Casbeer, "Optimal UAV route planning for persistent monitoring missions," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 550–566, Apr. 2021.

[5] M. D. Phung, C. H. Quach, Q. Ha, and T. H. Dinh, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Autom. Construct.*, vol. 81, pp. 25–33, Sep. 2017.

[6] H. Zhang, B. Xin, L. Dou, J. Chen, and K. Hirota, "A review of cooperative path planning of an unmanned aerial vehicle group," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 12, pp. 1671–1694, Dec. 2020.

[7] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[8] Z. Zhang, J. Wu, J. Dai, and C. He, "A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment," *IEEE Access*, vol. 8, pp. 122757–122771, 2020.

[9] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Robot.*, vol. 26, no. 3, pp. 308–333, Mar. 2009.

[10] X. Wang, X. Luo, B. Han, Y. Chen, G. Liang, and K. Zheng, "Collision-free path planning method for robots based on an improved rapidly-exploring random tree algorithm," *Appl. Sci.*, vol. 10, no. 4, p. 1381, Feb. 2020.

[11] H. Akbaripour and E. Masehian, "Semi-lazy probabilistic roadmap: A parameter-tuned, resilient and robust path planning method for manipulator robots," *Int. J. Adv. Manuf. Technol.*, vol. 89, pp. 1401–1430, Mar. 2017.

[12] A. Chamseddine, Y. Zhang, C. A. Rabbath, C. Join, and D. Theilliol, "Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 2832–2848, Oct. 2012.

[13] Y. Kuwata and J. P. How, "Cooperative distributed robust trajectory optimization using receding horizon MILP," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 2, pp. 423–431, Mar. 2011.

[14] I. K. Singgih, J. Lee, and B.-I. Kim, "Node and edge drone surveillance problem with consideration of required observation quality and battery replacement," *IEEE Access*, vol. 8, pp. 44125–44139, 2020.

[15] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot rendezvous planning for recharging in persistent tasks," *Trans. Robot.*, vol. 31, no. 1, pp. 128–142, 2015.

[16] Y. V. Pehlivanoglu and P. Pehlivanoglu, "An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems," *Appl. Soft Comput.*, vol. 112, Nov. 2021, Art. no. 107796.

[17] V. Sangeetha, R. Krishankumar, K. S. Ravichandran, and S. Kar, "Energy-efficient green ant colony optimization for path planning in dynamic 3D environments," *Soft Comput.*, vol. 25, no. 6, pp. 4749–4769, Jan. 2021.

[18] W. Zhao, Z. Fang, and Z. Yang, "Four-dimensional trajectory generation for UAVs based on multi-agent Q learning," *J. Navigat.*, vol. 73, no. 4, pp. 874–891, Feb. 2020.

[19] J. Xie, L. R. G. Carrillo, and L. Jin, "Path planning for UAV to cover multiple separated convex polygonal regions," *IEEE Access*, vol. 8, pp. 51770–51785, 2020.

[20] Z. Zhang, J. X. Li, and J. Wang, "Sequential convex programming for nonlinear optimal control problems in UAV path planning," *Aerosp. Sci. Technol.*, vol. 76, pp. 280–290, May 2018.

[21] Z. Wang and S. T. McDonald, "Convex relaxation for optimal rendezvous of unmanned aerial and ground vehicles," *Aerosp. Sci. Technol.*, vol. 99, Apr. 2020, Art. no. 105756.

[22] X. Sun, Y. Liu, W. Yao, and N. Qi, "Triple-stage path prediction algorithm for real-time mission planning of multi-UAV," *Electron. Lett.*, vol. 51, no. 19, pp. 1490–1492, Sep. 2015.

[23] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (UAVs): Collision avoidance systems and approaches," *IEEE Access*, vol. 8, pp. 105139–105155, 2020.

[24] D. Zhang and H. Duan, "Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning," *Neurocomputing*, vol. 313, pp. 229–246, Nov. 2018.

[25] Z. Wang, L. Liu, T. Long, and G. Xu, "Efficient unmanned aerial vehicle formation rendezvous trajectory planning using Dubins path and sequential convex programming," *Eng. Optim.*, vol. 51, no. 8, pp. 1412–1429, Aug. 2019.

[26] Y. Chen, J. Yu, Y. Mei, S. Zhang, X. Ai, and Z. Jia, "Trajectory optimization of multiple quad-rotor UAVs in collaborative assembling task," *Chin. J. Aeronaut.*, vol. 29, no. 1, pp. 184–201, Feb. 2016.

[27] J. G. Manathara and D. Ghose, "Rendezvous of multiple UAVs with collision avoidance using consensus," *J. Aerosp. Eng.*, vol. 25, no. 4, pp. 480–489, Oct. 2012.

[28] A. Ismail, B. Bagula, and E. Tuyishimire, "Internet-of-Things in motion: A UAV coalition model for remote sensing in smart cities," *Sensors*, vol. 18, no. 7, p. 2184, Jul. 2018.

[29] M. Pei-Bei, F. Zuo-E, and J. Jun, "Cooperative control of multi-UAV with time constraint in the threat environment," in *Proc. IEEE Chin. Guid., Navigat. Control Conf.*, Yantai, China, Aug. 2014, pp. 2424–2428.

[30] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[31] Z. Shao, F. Yan, Z. Zhou, and X. Zhu, "Path planning for multi-UAV formation rendezvous based on distributed cooperative particle swarm optimization," *Appl. Sci.*, vol. 9, no. 13, p. 2621, Jun. 2019.

[32] S.-M. Hung and S. N. Givigi, "A Q-learning approach to flocking with UAVs in a stochastic environment," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 186–197, Jan. 2017.

[33] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Algarve, Portugal, Oct. 2012, pp. 1917–1922.

[34] P. Lu and X. Liu, "Autonomous trajectory planning for rendezvous and proximity operations by conic optimization," *J. Guid., Control, Dyn.*, vol. 36, no. 2, pp. 375–389, Mar. 2013.

[35] Y. Chen, M. Cutler, and J. P. How, "Decoupled multiagent path planning via incremental sequential convex programming," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 5954–5961.

[36] Z. Wang, L. Liu, and T. Long, "Minimum-time trajectory planning for multi-unmanned-aerial-vehicle cooperation using sequential convex programming," *J. Guid., Control, Dyn.*, vol. 40, no. 11, pp. 2976–2982, Nov. 2017.

[37] G. M. Ramalho, S. R. Carvalho, E. C. Finardi, and U. F. Moreno, "Trajectory optimization using sequential convex programming with collision avoidance," *J. Control, Autom. Electr. Syst.*, vol. 29, no. 3, pp. 318–327, Mar. 2018.

[38] J. Zhang, J. Yan, P. Zhang, and X. Kong, "Collision avoidance in fixed-wing UAV formation flight based on a consensus control algorithm," *IEEE Access*, vol. 6, pp. 43672–43682, 2018.

[39] J. Zhang, J. Yan, and P. Zhang, "Fixed-wing UAV formation control design with collision avoidance based on an improved artificial potential field," *IEEE Access*, vol. 6, pp. 78342–78351, 2018.

[40] T. W. Mclain and R. W. Beard, "Coordination variables, coordination functions, and cooperative timing missions," *J. Guid. Control Dyn.*, vol. 28, no. 1, pp. 150–161, May 2005.

[41] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs," *Proc. IEEE*, vol. 94, no. 7, pp. 1306–1324, Jul. 2006.

[42] C. Zhang, J. Song, L. Huang, and G. Cai, "Cooperative guidance law considering the randomness of the unreliable communication network," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 233, no. 9, pp. 3313–3322, Jul. 2019.
[43] H. Mehrjerdi, J. Ghomman, and M. Saan, "Nonlinear coordination control for a group of mobile robots using a virtual structure," *Mechatronics*, vol. 21, no. 7, pp. 1147–1155, Oct. 2011.
[44] X. Wang, Y. Yu, and Z. Li, "Distributed sliding mode control for leader-follower formation flight of fixed-wing unmanned aerial vehicles subject to velocity constraints," *Int. J. Robust Nonlinear Control*, vol. 31, no. 6, pp. 2110–2125, Apr. 2021.
[45] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.
[46] M. Kishida, M. Ogura, Y. Yoshida, and T. Wadayama, "Deep learning-based average consensus," *IEEE Access*, vol. 8, pp. 142404–142412, 2020.
[47] Y. Wu, M. Xu, N. Zheng, and X. He, "Event-triggered resilient consensus for multi-agent networks under deception attacks," *IEEE Access*, vol. 8, pp. 78121–78129, 2020.
[48] S. Boyd, L. Vandenberghe, and L. Faybusovich, "Convex optimization," *IEEE Trans. Autom. Control*, vol. 51, no. 11, p. 1859, Nov. 2006.
[49] Y. Mao, M. Szmuk, and B. Acikmese, "Successive convexification of non-convex optimal control problems and its convergence properties," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Las Vegas, NV, USA, Dec. 2016, pp. 3636–3641.

ZHEKUN CHENG was born in Yancheng, Jiangsu, China, in 1996. He received the B.S. degree in exploration guidance and control from the School of Aerospace Engineering, Beijing Institute of Technology, Beijing, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include multi-agent path planning and distributed online convex optimization.

LIANGYU ZHAO (Member, IEEE) was born in Shangqiu, Henan, China, in 1981. He received the bachelor's and Ph.D. degrees from the Beijing Institute of Technology, Beijing, China, in 2003 and 2008, respectively. He is currently an Associate Professor with the School of Aerospace Engineering, Beijing Institute of Technology. He has published more than 30 papers in journals and conferences. His research interests include flight vehicle design, flight dynamics and control, and aerodynamics.

ZHONGJIAO SHI was born in Yantai, Shandong, China, in 1991. He received the B.Sc. and Ph.D. degrees in aerospace engineering from the Beijing Institute of Technology, Beijing, China, in 2014 and 2020, respectively. He is currently a Postdoctoral Research Associate with the School of Mechatronical Engineering, Beijing Institute of Technology. His current research interests include adaptive control, convex optimization, and logic-based path planning.