Article
Collision-Free 4D Dynamic Path Planning for Multiple UAVs Based on Dynamic Priority RRT* and Artificial Potential Field

Yicong Guo 1, Xiaoxiong Liu 1,2,*, Wei Jiang 1 and Weiguo Zhang 1,2

1 School of Automation, Northwestern Polytechnical University, Xi’an 710129, China
2 Shaanxi Province Key Laboratory of Flight Control and Simulation Technology, Xi’an 710129, China
* Correspondence: liuxiaoxiong@nwpu.edu.cn

Abstract: In this paper, a four-dimensional (4D) dynamic cooperative path planning algorithm for multiple unmanned aerial vehicles (UAVs) is proposed, in which the cooperative time variables of UAVs, as well as conflict and threat avoidance, are considered. The algorithm proposed in this paper uses a hierarchical framework that is divided into a 4D cooperative planning layer and a local threat avoidance planning layer. In the cooperative planning layer, the proposed algorithm, named dynamic priority rapidly exploring random trees (DPRRT*), would be used for the 4D cooperative path planning of all UAVs involved in a given task. We first designed a heuristic prioritization strategy in the DPRRT* algorithm to rank all UAVs to improve the efficiency of cooperative planning. Then, the improved RRT* algorithm with the 4D coordination cost function was used to plan the 4D coordination path for each UAV. Whenever the environment changes dynamically (i.e., sudden static or moving threats), the proposed heuristic artificial potential field algorithm (HAPF) in the local threat avoidance planning layer is used to plan the local collision avoidance path. After completing local obstacle avoidance planning, the DPRRT* of the 4D cooperative planning layer is again called upon for path replanning to finally realize 4D cooperative path planning for all UAVs. The simulation and comparison experiments prove the feasibility, efficiency, and robustness of the proposed algorithm.

Keywords: multiple unmanned aerial vehicles (multi-UAVs); four-dimensional (4D) dynamic cooperative path planning; rapidly exploring random trees (RRT*); artificial potential field (APF)

1. Introduction
In recent years, UAVs have been widely used in both military and civilian fields, such as urban penetration, battlefield reconnaissance, aerial photography, disaster area search and rescue, etc. [1,2]. However, with the increasing diversification of missions and the complexity of flight environments, more and more users are reaching the limitations of single UAVs when performing missions [3]. Therefore, multiple UAV systems have gradually grown into a popular and challenging research topic [4]. As the most basic problem in the research of multi-UAV systems, the cooperative path planning technology of multiple UAVs directly affects the ability of UAVs to complete tasks smoothly, safely, and autonomously [5,6].

Multi-UAV systems are usually deployed for more complex tasks, such as coordinated multi-UAV attack and detection missions, which require all UAVs to have the same and shortest arrival time [7]. In addition, since there are usually a variety of threats in flight environments where multiple UAV systems perform tasks, the planning of a safe path that must simultaneously avoid environmental threats and collisions between individual UAVs is a difficult problem [8]. Furthermore, ground-based navigation systems are often unable to detect the precise location and configuration information of all threats in complex environments [9]. Therefore, it is particularly difficult to coordinate the paths of all UAVs involved and adjust their existing paths in a timely and efficient manner according to real-time environmental information [10], especially for some tasks that require accurate arrival time and dynamic conflict avoidance simultaneously.
The existing methods of solving the cooperative path planning problem of multiple UAVs include several collision avoidance methods and cooperative path generation methods [11]. Collision avoidance methods are often used to deal with local spatial conflicts. First, each UAV independently plans the shortest path while avoiding threats in the environment. During the path execution, each UAV regularly detects the position and speed of other UAVs around it. If there is a risk of collision, the UAVs adjust their paths in real time to avoid conflicts. Many local collision avoidance path planning algorithms based on collision avoidance methods have been proposed, such as the artificial potential field algorithm (APF) [12,13], the centralized control barrier function algorithm (CCBF) [14], the distributed control barrier function algorithm (DCBF) [15], and the optimal reciprocal collision avoidance algorithm (ORCA) [16]. Since collision avoidance is only performed in a local range, these algorithms usually have high computational efficiency and the ability to dynamically and rapidly replan paths. However, as a local collision avoidance method, with the increase in the number of UAVs and threats, these algorithms may fall into local minima and lead to the failure of cooperative path planning. Additionally, since such methods focus on space conflict avoidance, the consistency of flight time is not considered.

Cooperative path generation methods can be divided into two categories: coupling methods and decoupling methods [17]. The coupling method usually regards all UAVs as comprising an agent system and directly searches for the cooperative path of the agent system. Some popular coupling methods, such as increasing cost tree search (ICTS) [18] and conflict-based search (CBS) [19], have been extensively studied and many variant algorithms have been derived [20–23]. The coupling method directly deals with the path planning problem of the agent system and can usually obtain the optimal solution for the system. However, it is well-known that even a simple variant of the path planning problem for multiple UAVs, such as a planned cooperative path with the sudden appearance of a threat, interferes with the implementation of a cooperative path that encounters both conflict-free space and path execution time consistency. In addition, in computing, the coupling method is very difficult to solve [24].

Decoupling methods usually divide the problem of multi-UAV cooperative path planning into two steps, i.e., the coordination of the path planning sequence and the single UAV cooperative path planning [25]. As the most representative algorithm of the decoupling method, the prioritized planning algorithm (PP) [26] has become a very popular algorithm in cooperative path planning [27], along with other methods such as the revised prioritized planning algorithm (RPP) [28], the decentralized prioritized planning (DPP) [29], and the asynchronous decentralized prioritized planning algorithm (ADPP) [30]. In the prioritized planning algorithm, each UAV has its own priority and calculates its path according to the relative order of its priority. Therefore, the final path of each UAV is able to avoid static threats in the environment and UAVs with a higher priority than it. The common single UAV cooperative path planning algorithms in prioritized planning algorithms are the grid-based A* algorithm [31] and the sample-based RRT* algorithm [32,33]. The calculation time of the A* algorithm usually increases exponentially with the increase in spatial dimension and grid resolution, and it does not perform well in large planning spaces of UAVs [34]. The RRT* algorithm is able to reach the optimal solution with fewer samples than the fixed high-resolution grid method by sampling the extended tree structure. Therefore, RRT* is widely used in path planning in high-dimensional space [35]. Furthermore, if the priority can be assigned reasonably, the prioritized planning algorithm can usually plan the cooperative path for multiple UAVs more efficiently. The existing decoupling methods also focus on conflict avoidance and do not conduct in-depth research on the time consistency of all UAV planning paths. In addition, in complex environments with unknown threats, the ability of the existing decoupling methods to avoid dynamic sudden threats is unknown.

Therefore, combining the practical applications of the environment and the complex task categories of multiple UAV systems, as well as the possible improvement of existing
path planning algorithms for multiple UAVs, the main motivations of this paper are as follows.

1. The actual flight environment of UAVs is mostly wide open space. Due to the complexity of the actual flight environment, certain detected static threats can suddenly start to move and previously unknown static or moving threats can suddenly appear. Only relying on the known environment broadcast by the navigation system for cooperative path planning often carries the risk of unfeasible paths being followed.

2. The tasks performed by multi-UAV systems have gradually become more complex and diverse. Three-dimensional cooperative planning based on conflict-free space no longer meets the needs of most tasks.

3. Existing algorithms based on the local collision avoidance method, the coupling method, and the decoupling method are rarely studied in the field of 4D cooperative path planning [36,37]. In addition, the local collision avoidance method is able to adapt to the frequently changing dynamic environment but cannot ensure the success of global cooperative path planning. The decoupling method can quickly plan cooperative paths for multiple UAVs but it cannot guarantee the efficiency of cooperative replanning brought about by environmental changes.

Therefore, this paper focuses on the research of a 4D dynamic cooperative path planning algorithm for multiple UAVs. The algorithm needs to be consistent in terms of the conflict-free space and flight time of multiple UAVs simultaneously and must be able to adapt to frequent changes in the environment so that the algorithm can quickly avoid threats and suddenly encountered threats and ensure a lack of conflict between UAVs. More specifically, the algorithm proposed in this paper adopts a hierarchical strategy for the 4D cooperative path planning of multiple UAVs and the local cooperative collision avoidance planning for dynamic environments. The 4D cooperative planning algorithm based on DPRRT* consists of a heuristic dynamic prioritization strategy, priority planning order coordination, and an improved RRT* algorithm based on 4D cooperative constraints. The local threat avoidance planner based on HAPF effectively overcomes the problems of unreachable targets, the risks of falling into local minima, and the path oscillation which exists in traditional APF by improving potential field functions, heuristic optimal sub-target node selection strategies and memory resultant force strategies, respectively. In fact, the proposed algorithm based on a hierarchical strategy combines the advantages of the decoupling method and the collision avoidance method and overcomes their shortcomings. In addition, the algorithms based on the two methods are improved and upgraded to be used for more extensive practical applications in the future.

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

1. A 4D dynamic cooperative path planning algorithm based on a hierarchical strategy is proposed. The proposed upper layer based on DPRRT* is able to plan a 4D cooperative path for UAVs that simultaneously meets the requirements of conflict-free space and consistent flight time. The proposed lower layer based on HAPF is able to deal with online dynamic local collision avoidance planning in frequently changing environments.

2. In the proposed DPRRT* algorithm, a heuristic dynamic prioritization strategy was designed. Within a planned environment, the heuristic prioritization algorithm was used to reasonably and dynamically adjust the priorities of UAVs to efficiently and robustly plan their path generation sequences.

3. In the single-UAV planning algorithm of DPRRT*, a 4D cost function based on time cooperative constraints was proposed, which guides the expansion of new nodes and the optimization process of neighborhood nodes to quickly plan a path that satisfies the 4D cooperative constraints of UAVs.

4. In HAPF, a heuristic sub-target node selection strategy was designed to select the optimal sub-target node for the UAVs in local obstacle avoidance planning in order to get rid of the local minimum node. In addition, a memory resultant force strategy was
designed to improve path oscillation, so that the obstacle avoidance path is smoother and is able to meet the performance constraints of the UAVs.

This paper is organized as follows. Section 2 describes the 4D dynamic cooperative path planning problem. In Section 3, the 4D dynamic cooperative path planning algorithm proposed in this paper is explained in detail. Section 4 outlines the simulation and comparison experiments of the proposed algorithm and discusses the simulation results. Section 5 describes some of the conclusions reached, as well as future works.

### 2. Problem Formulation

The 4D multi-UAV dynamic cooperative path planning problem discussed in this paper refers to a problem in dynamic 4D space \( X(x, y, z, t) \subseteq \mathbb{R}^4 \) comprising sudden threats for each UAV in the group, \( \text{UAV}_N : (\text{UAV}_1, \ldots, \text{UAV}_n) \), composed of \( n \) UAVs, resulting in the planning of a path that meets 4D cooperative constraints and involves no collisions with threats in the dynamic environment.

Considering the diversity of tasks performed by UAV groups, the starting node or end node of multiple UAVs may overlap (such as in rendezvous tasks), so this paper assumes that if multiple UAVs have the same start node or goal, there will be no collision. In addition, each UAV has communication equipment to relay information. This paper assumes that communication is completely reliable and there are no delays.

#### 2.1. Dynamic Cooperative Path Planning Constraints of UAVs

For the constraints of the general UAV’s 4D dynamic cooperative path planning, the UAV performance constraints, dynamic environment limitations, and 4D cooperative constraints should be considered. Each UAV \( i \) in \( \text{UAV}_N \) needs to take off from its own starting node at the same time. The planned path needs to avoid conflicts with the other UAVs \( \text{UAV}_N \setminus \text{UAV}_i \) while avoiding all threats (including radar, anti-aircraft artillery, anti-aircraft missiles, and terrain) and finally reaching the intended goal at the same time. The planned path needs to take into consideration the performance constraints of the UAV. In addition, sudden, undetected threats may appear during flight, i.e., the planning space is dynamic. Therefore, the UAVs affected by the change of environment need to carry out local path replanning to avoid these threats. Note that the local path replanning destroys the 4D cooperative constraints, making it also necessary to carry out cooperative replanning for each UAV \( N \) to allow the path to meet the cooperative path planning requirements of the UAV.

##### 2.1.1. UAV Performance Constraints

The cooperative path, \( \sigma_i \), of any UAV \( i \) in \( \text{UAV}_N \) is composed of a group of path nodes \( P_i : \{(p_1, \ldots, p_k)|p_1 = s_i, p_k = g_i, p_k = (x_k, y_k, z_k, t_k)\} \). \( \sigma_i \) needs to meet the performance constraints of the UAVs, including the maximum flight distance \( L_{\text{max}} \), the maximum steering angle \( \phi_{\text{max}} \), the maximum climbing/diving angle \( \gamma_{\text{max}} \), and the shortest path segment distance \( l_{\text{min}} \). UAV performance constraints are shown in Definition 1.

**Definition 1 (UAV performance constraints).** For any UAV \( i \) in-planning space \( X' \), the maximum flight distance constraint is:

\[
L_{\text{UAV}_i} = \sum_{i=0}^{k} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \leq L_{\text{max}}
\]

and the maximum steering angle constraint is:

\[
|\phi_{\text{UAV}_i}| = \frac{\lambda_i^T \lambda_{i+1}}{\|\lambda_i\|_{2} \|\lambda_{i+1}\|_{2}} \leq \phi_{\text{max}}
\]

where \( \lambda_i = [x_i - x_{i-1}, y_i - y_{i-1}]^T \).
The maximum climbing/diving angle constraint is:

\[ |\gamma_{UAV_i}| = \left| \arcsin \left( \frac{z_i - z_{i-1}}{\|\lambda_i\|_2} \right) \right| \leq \gamma_{\text{max}} \quad (3) \]

and the shortest path segment distance constraint is:

\[ l_{UAV_i} = \|p_{i+1} - p_i\|_2 \geq l_{\text{min}} \quad (4) \]

2.1.2. Environmental Limitations

In the dynamic planning environment, there are five different types of threats, i.e., terrain threats, radar, anti-aircraft artillery, missiles, and sudden threats. Of these, the sudden threats refer to threats that suddenly appear (i.e., radar, anti-aircraft artillery, and missiles). Sudden threats can be static or moving. The modeling process of the first four threats has been described in detail in our previous work [38]. This paper mainly describes sudden threats and environmental limitations.

Considering the time uncertainty and state uncertainty of sudden threats, we describe the sudden threats in Definition 2.

**Definition 2 (sudden threats).** For planning space \( X \), all UAVs in \( UAV_N \) fly according to the currently planned cooperative path, and the flight time interval is \( [t_s, t_g] \). At \( t \in [t_s, t_g] \), \( m \) new threats \( X_{\text{threat}}^{\text{new}} : \{X_{\text{threat}}^1, \ldots, X_{\text{threat}}^m\} \) are sudden threats. The position change rule \( X_{\text{threat}} \) is expressed as:

\[
\Delta X_{\text{threat}}^{\text{new}} = \bigcup_{i=1}^m \Delta X_{\text{threat}}^m, \text{ where } \Delta X_{\text{threat}}^m = \begin{cases} \emptyset & f(x, t_i) \\ \text{unpredictable} & \end{cases} \quad (5)
\]

where \( \Delta X_{\text{threat}}^m = \emptyset \) means that the sudden threat is static, \( \Delta X_{\text{threat}}^m = f(x, t_i) \) means that the dynamic sudden threat is moving, and \( \Delta X_{\text{threat}}^m = \text{unpredictable} \) means that the threat movement is unpredictable.

The definition of environmental limitations is given in Definition 3.

**Definition 3 (environmental limitations).** For any UAV \( i \) in-planning space \( X \), the limit of the planning environment is:

\[
\left\{ \begin{array}{l}
z_i - z((x_i, y_i)_{\text{DMG}}) \geq h_{\text{min}} \\
(p_i, p_{i+1}) \cap X_{\text{threat}} = \emptyset
\end{array} \right. \quad (6)
\]

where \( z((x_i, y_i)_{\text{DMG}}) \) represents the terrain threat altitude when the UAV is located at \( p_i \) and \( X_{\text{threat}} = \{X_{\text{known}}^{\text{threat}} \cup X_{\text{new}}^{\text{threat}}\} \) represents the known radar, anti-aircraft artillery, and missile threats, as well as sudden threats.

2.1.3. Four-Dimensional Coordination Constraints

The four-dimensional coordination constraints consist of spatial coordination constraints and time coordination constraints. In this paper, it is assumed that any UAV \( i \) in \( UAV_N \) flies at a uniform speed in any speed interval \([V_{\text{min}}, V_{\text{max}}]\); that is, the estimated time of arrival (ETA) interval of UAV \( i \) is \( \left[ t_{i, \text{min}}, t_{i, \text{max}} \right] = \left[ \frac{\sigma_i}{V_{\text{max}}}, \frac{\sigma_i}{V_{\text{min}}} \right] \).

**Definition 4 (four-dimensional coordination constraints).** The cooperative path of all UAVs in \( UAV_N \) must meet the following requirements:
1. At moment \( t \), the distance \( D_{ij}(t) \) between any two UAVs is:

\[
D_{ij}(t) = \| \sigma_i(t) - \sigma_j(t) \|_2 \geq D_{\text{min}} \quad i,j = 1,2,\ldots,k, i \neq j
\]  

where \( D_{\text{min}} \) is the minimum safe distance.

2. The set \( T \) of ETA of all UAVs is:

\[
T = \cap_{t=1}^n [t_{\text{min}}, t_{\text{max}}] \neq \emptyset
\]

2.2. Dynamic 4D Cooperative Path Planning Formulation

A set of 4D cooperative paths for UAV \( N \) can be obtained only when Definitions 1, 3, and 4 are met. Problem 1 describes the 4D dynamic cooperative path planning problem of the multiple UAVs studied in this paper.

Problem 1 (dynamic 4D cooperative path planning). In the current dynamic planning space \( \mathcal{X} \), the set starting nodes and goal nodes \( [s_1, g_1], \ldots, [s_n, g_n] \) of all UAVs in UAV \( N \) are given. A group of paths \( v_N = \{v_1, \ldots, v_n\} \) must be found so that all paths in \( v_N \) meet the necessary constraints of Definitions 1, 3, and 4.

3. Four-Dimensional Dynamic Cooperative Path Planning Algorithm

The overall frameworks of the 4D dynamic cooperative path planning algorithm, the DPRRT* algorithm, and the HAPF algorithm will be introduced below.

3.1. Overall Framework

Algorithm 1 shows the executive process of the 4D dynamic cooperative path planning algorithm for multiple UAVs.

In the pre-cooperative planning based on the initial environment (lines 02–07), the algorithm first initializes the search tree \( T_i \) and sets the start and goal \( [s_i, g_i] \) of any UAV \( i \), as well as the initial planning space \( \mathcal{X} \). Then, the classical RRT* shown in Algorithm 1 is used to plan the optimal path for all UAVs without considering the 4D cooperative constraints. After the UAV planning is completed, the shortest coordination time \( t_{\text{co}} \) of the UAVs is determined through the process ETA\_Len. Then, the DPRRT* algorithm is used to plan the paths of all UAVs that meet the 4D cooperative constraints in the initial planning space \( \mathcal{X} \). Finally, all UAVs execute their respective cooperative paths.

The purpose of the process ETA\_Len is to obtain the minimum coordination time \( t_{\text{co}} \) of all the UAVs by calculating the minimum flight time set for all UAVs and calculating the maximum value in the set. It should be noted that all UAVs can achieve the shortest path (Line 03) through classical RRT*. In other words, all UAVs are able to achieve the shortest flight time without considering 4D cooperative constraints. We assume that the minimum flight time UAV \( i \) is \( t_i \), the minimum flight time UAV \( j \) is \( t_j \), and that \( t_i > t_j \). Since \( t_i \) and \( t_j \) are already the shortest flight times, the only way to ensure that UAV \( i \) and UAV \( j \) satisfy the time coordination constraints and ensure the shortest coordination time is to increase \( t_i \) so that it is equal to \( t_j \).
Algorithm 1. Four dimensional Dynamic Cooperative Path Planning Algorithm for Multiple UAVs

01: MAIN ALGORITHM
02: initialization $X, T_i \leftarrow \emptyset, \sigma_i(t) \leftarrow \emptyset, [s_1, g_1], \ldots, [s_n, g_n], \sigma(t) \leftarrow [v_1(t), \ldots, v_n(t)],$
03: (Len_cost$[t], \sigma_i(t)) \leftarrow RRT*(X, T_i, s_i, g_i) / *Synchronous execution of all UAVs* /
04: wait for all UAVs to finish planning their own $\sigma_i$ and Len_cost$[t]$
05: $[t_{co}, l_{co}] \leftarrow ETA_{Len}(\sigma(t), Len_cost(t))$
06: $\sigma(t) \leftarrow DPRRT^*(X, t_{co}, l_{co}, [s_1, g_1], \ldots, [s_n, g_n], \sigma)$
07: all UAVs move along $\sigma(t)$
08: while goal not reach and $\Delta\chi_{\text{threat}}$ new detected
09: $X \leftarrow X \cup \Delta\chi_{\text{threat}}$ new
10: $(\sigma_{\text{conflict}, \text{UAV}_{\text{conflict}}}) \leftarrow \text{Conflicts}_\text{occurred}(\sigma, \Delta\chi_{\text{threat}})$
11: if $\sigma_{\text{conflict}} \neq \emptyset$
12: $t_0 \leftarrow \text{clock}()$
13: $HAPF(\text{UAV}_{\text{conflict}}, X \cup \sigma(t_0, t_{co}) \setminus \sigma_{\text{conflict}}(t_0, t_{co}))$
14: all UAV\textbackslash{\text{UAV}}_{\text{conflict}} and UAVs which finish HAPF move along their $\sigma_i$ \textbackslash{their\text{}}
15: else
16: all UAVs move along $\sigma(t_0, t_{co})$
17: end if
18: $t_0 \leftarrow \text{clock}()$
19: $[t_{co}, l_{co}] \leftarrow ETA_{Len}(\sigma(t_0, t_{co}), Len_cost(t_0, t_{co}))$
20: $\sigma(t_0, t_{co}) \leftarrow DPRRT^*(X, t_{co}, [v_1(t_0), g_1], \ldots, [v_n(t_0), g_n], \sigma)$
21: all UAVs move along $\sigma(t_0, t_{co})$
22: end while

If the UAV detects a sudden threat $\Delta\chi_{\text{threat}}$ new in planning space $X$ during flight, the cooperative path based on the previous planning space may be affected, resulting in the failure of the path to cooperate. Therefore, cooperative replanning needs to be executed in the new planning space containing $\Delta\chi_{\text{threat}}$ new. We use the Conflicts_occurred process to find the UAV set $\text{UAV}_{\text{conflict}}$ (and its conflict path $\sigma_{\text{conflict}}$) affected by $\Delta\chi_{\text{threat}}$ new from among all UAVs. Then, HAPF is applied to all UAVs in the set $\text{UAV}_{\text{conflict}}$ for the efficient online onboard replanning of paths. As HAPF is a local conflict avoidance algorithm, it can only ensure that the UAV meets the spatial coordination constraints in the 4D coordination constraints. Therefore, we choose the path node $\sigma_{\text{conflict}}$ which, as the generation node of the attractive potential field of $\text{UAV}_{\text{conflict}}$, is not reached and not affected by the threat, thus enabling $\text{UAV}_{\text{conflict}}$ to return to the previous 4D cooperative path after avoiding the sudden threat. At the same time, the UAVs not affected by $\Delta\chi_{\text{threat}}$ new continue to fly along the previously planned coordinated path. When the $\text{UAV}_{\text{conflict}}$ avoids sudden threats, the DPRRT* is used to replan the path from the current position of the UAVs to their destination, while meeting the 4D cooperative constraints for all UAVs. If the planning space is not changed, all UAVs fly along the current initial path until they reach the goal and the algorithm is terminated. Otherwise, when new sudden threats are detected, lines 08–22 of Algorithm 1 are executed until the goal is reached.

3.2. Dynamic Priority RRT* Algorithm

In the cooperative path planning layer, we used the proposed DPRRT* to complete the 4D cooperative path planning of multiple UAVs based on the current initial planning space. The pseudo-code for DPRRT* is shown in Algorithm 2. The core idea of DPRRT* is to enable UAVs with lower priority to avoid conflict with the planned paths of the UAVs with higher priority when planning the cooperative paths in order to meet the requirements of 4D cooperative constraints.
Considering that the prioritization of UAVs usually has a great impact on the efficiency and robustness of cooperative path planning, we designed a dynamic heuristic prioritization strategy (i.e., the HeuDyn_Prioritization process) to assign the priorities of all UAVs, where UAV\(_i\) represents the priority of UAV\(_i\) as \(j\) and \(\sigma(t)\) represents the planned cooperative path of the UAVs based on the previously planned space. Thereafter, the cooperative paths of all UAVs are planned in order of priority from high to low. In addition, in order to simultaneously meet the time and space cooperative constraints, this paper improves the classical RRT* to an RRT* algorithm based on 4D cooperative constraints, named 4DRRT* (i.e., the 4DRRT* process), so that each UAV is able to quickly plan a path that meets the cooperative 4D constraints. Note that before planning the lower-priority UAV paths, the planned paths of the high-priority UAVs need to be added to the space-time occupied area \(H^t\) to ensure that the planned paths of the low-priority UAVs do not conflict with those of high-priority UAVs. The dynamic heuristic prioritization strategy and 4DRRT* algorithm will be introduced in detail below.

Algorithm 2. DPRRT* Algorithm

01: MAIN ALGORITHM
02: \(H^t \leftarrow \emptyset, \sigma^\ast \leftarrow \emptyset\)
03: \(j = 1\) to \(N\)
05: \(\sigma^t(\cdot) \leftarrow 4\text{DRRT}^*\left(X, t_{co}, H^t, s_t, g_t\right)\)
06: \(H^t \leftarrow \sigma^t(\cdot) \cup \sigma^t(t)\)
07: \(\sigma^\ast(\cdot) \leftarrow \sigma^t(\cdot) \cup \sigma^\ast\)
08: end for

01: Function HeuDyn_Prioritization
02: \(f^\text{HDP} \leftarrow \emptyset, k = 1\)
03: \(i = 1\) to \(N\)
04: \(f^\text{HDP} = \left\| t_{co} - \frac{\sigma(t)_{k|\text{co}}}{\|t_{co}\|} \right\|\)
05: \(f^\text{HDP} \leftarrow \arg\max \left\{ f^\text{HDP}, f^\text{AUV}_i \right\}\)
06: end for
07: while \(f^\text{HDP} \neq \emptyset\)
08: \(i = \arg\max f^\text{AUV}_i\)
09: \(f^\text{HDP} \leftarrow f^\text{AUV}_i \cup f^\text{HDP}\)
10: \(k = k + 1\)
11: end while

01: Function 4D_Extend
02: for all \(x_T \in T, T\) do
03: if \(\left\| x_T - x_{\text{rand}} \right\| \leq \delta\)
04: \(x_{\text{new}} \leftarrow x_{\text{rand}}\)
05: else \(\left\| x_T - x_{\text{rand}} \right\| > \delta\)
06: \(x_{\text{new}} \leftarrow x_T + \delta \frac{x_{\text{rand}} - x_T}{\left\| x_{\text{rand}} - x_T \right\|}\)
07: end if
08: if obstacle_free(x_T, x_{new}) and UAV_free(x_T, x_{new}) and Conflict_free([x_T, x_{new}], H_t)
09: \(f^\text{D}(x_{\text{new}}) \leftarrow \left| f_T(x_{\text{new}}) - l_{\text{co}} \right|\)
10: \(f^\text{D}_{\text{all}} \leftarrow f^\text{D}(x_{\text{new}})\)
11: end if
12: end for
13: \(x_{\text{new}} \leftarrow \arg\min f^\text{D}_{\text{all}}\)

01: Function 4DRT* (e.g., UAV\(_i\))
02: initialization \(T.V \leftarrow \{s_t\}; T.E \leftarrow \emptyset\)
04: for \(i = 1\) to \(n\)
05: if \(P : \text{random}([0, 1]) < 0.1\)
06: \(x_{\text{rand}} \leftarrow s_t\)
07: else
08: \(x_{\text{rand}} \leftarrow \text{Sample}(X)\)
09: end if
10: \(x_{\text{new}} \leftarrow 4\text{D_Extend}(x_{\text{nearest}})\)
11: \(T.V \leftarrow T.V \cup \{x_{\text{new}}\}\)
12: end if
13: \(x_{\text{nearest}} \leftarrow \text{Near}(x_{\text{new}}, T.V, \gamma)\)
14: for all \(x_{\text{new}} \in X_{\text{nearest}}\)
15: 4DRRT* \_\text{rewire}(x_{\text{nearest}}, x_{\text{new}})\)
16: end for
17: for all \(x_{\text{new}} \in X_{\text{nearest}}\)
18: 4DRT* \_\text{rewire}(x_{\text{new}}, x_{\text{nearest}})\)
19: end for
20: if \(x_{\text{new}} \in G_i\)
21: \(g_i \leftarrow x_{\text{new}}\)
22: end if
23: end for
24: while \(\left| \text{Cost}_T(s_{t|g_i}) - l_{\text{co}} \right| \geq \varepsilon\)
25: Update \(T.x_{\text{new}}\) with Line 23–39
26: end while
27: return \(\sigma_{i|g_i}\)

01: Function 4DRT* \_\text{rewire}
02: if obstacle_free(x_1, x_2)
03: \(\text{UAV}_\text{free}(x_1, x_2)\)
04: \(\text{Conflict}_\text{free}([x_1, x_2], H_t)\)
03: cost \(\leftarrow \left\| x_1 - x_2 \right\|_2\)
04: if \(\left| f_T(x_1) + \lambda_s h(x_2) - l_{\text{co}} \right| < f^\text{D}(x_2)\)
05: \(x_2_{\text{parent}} \leftarrow x_1\)
06: end if
07: end if
3.2.1. Dynamic Heuristic Prioritization Strategy

A key factor affecting the efficiency and robustness of DPRRT* is the prioritization strategy. If the prioritization is not carefully defined based on a specific problem scenario, it may lead to low-priority UAVs spending more computing time in planning to avoid the occupied area in space and time or even making the planning path infeasible. To solve this problem, a dynamic heuristic prioritization strategy is proposed. Whenever the planning space changes, this strategy can reasonably and dynamically adjust the priority of all UAVs so that the DPRRT* algorithm is able to find a 4D cooperative path planning solution with higher efficiency and robustness.

Note that the generation of cooperative paths needs to meet the 4D cooperative constraints based on time consistency and no conflicts of space. Since the HAPF described below can effectively avoid sudden threats in a local range, as well as avoid spatial conflicts between UAVs and allow the UAV to return to the cooperative path based on the previous spatial planning cycle, when DPRRT* is used to replan the cooperative path, not only does the original cooperative path pass through conflict-free space but the time coordination constraint is broken. If the flight time of the UAV, UAV\(_i\), along the original planned path is significantly different from the shortest cooperation time \(t_{co}\) based on the current planning space, the difference between the replanned cooperation path for UAV\(_i\) and the original path will be even greater. Furthermore, when the cooperative path is replanned for UAV\(_i\), the probability of conflict with other UAVs’ occupied space and time area is greater. Therefore, it is necessary to ensure that the priority of UAV\(_i\) is assigned to a higher level by planning its path first to improve planning efficiency. The prioritization evaluation function \(f_{HDP}^{UAV_i}\) of UAV\(_i\) is calculated as follows:

\[
f_{HDP}^{UAV_i} = \left| t_{co} - \frac{\sigma_i(t_0, t_{co})}{v_i} \right|
\]

where \(\sigma_i(t_0, t_{co})\) represents the original planned path of the UAV not flying, \(t_{co}\) represents the shortest cooperation time, and \(v_i\) represents the speed of UAV\(_i\). It is not difficult to see that the prioritization evaluation function \(f_{HDP}^{UAV_i}\) of UAV\(_i\) is positively related to the amount of time cooperation violations. Therefore, the priority ranking definition of UAVs based on prioritization evaluation functions can be given.

**Definition 5 (prioritization strategy based on prioritization evaluation function).** A group of UAVs \(UAV_N\) : \((UAV_1, \ldots, UAV_N)\) is given. The priority evaluation function set \(\bigcup_{i=1}^{N} f_{HDP}^{UAV_i}\) of UAV\(_N\) is mapped from large to small in order to generate the set \(\bigcup_{i=1}^{N} f_{HDP}^{UAV_i}\) → \(\max \rightarrow \min\), which in turn corresponds to the \(\{1, \ldots, N\}\) priority of UAV\(_N\).

Considering that there may be a situation where some UAVs are not affected by sudden threats, they will continue to fly along their originally planned path. The prioritization evaluation function of these UAVs may be the same. For such UAVs, their priority order remains unchanged. The steps of the dynamic heuristic prioritization strategy are shown as the HeuDyn_Prorationization function of Algorithm 2.

3.2.2. 4DRRT* Algorithm

As a decoupled cooperative path planning algorithm, the DPRRT* algorithm needs to carry out path planning for each UAV to meet the 4D cooperative constraints in order of priority after ranking the priorities of all UAVs in question. Because there is a space–time occupation area composed of high-priority UAV paths and the conflict-free space and time coordination constraints of paths need to be met simultaneously during planning, the classical RRT* cannot meet these planning requirements.

Therefore, this paper proposes that the 4DRRRT* in the single UAV cooperative path planning process of DPRRT* be used in order to ensure that all UAVs are able to plan paths
that meet the 4D cooperative constraints. Firstly, a cost function based on the 4D cooperation constraints is proposed to evaluate the expansion of the tree. In addition, we introduce the time variable into the RRT* algorithm in order to detect whether there is a spatial conflict between the growth tree and the space–time occupied area. On this basis, a tree node expansion method based on the 4D cost function is designed to guide tree growth. Similar to the tree node expansion method, we also used the 4D cost function to guide the updating and rewiring of nodes in the neighborhood. The 4DRRT* algorithm can ensure that the planned cooperative path meets the performance constraints, environmental constraints, and 4D cooperative constraints of UAVs at the same time to provide a 4D cooperative path based on the current planning space for multiple UAVs. Next, we introduce 4DRRT* in terms of two aspects: the 4D cost function and the algorithm process.

1. The 4D Cost Function Design Based on the 4D Cooperation Constraint

Through the process of ETA_Len, we can obtain the shortest cooperation time $t_{co}$ based on the current planning space and then calculate the 4D cooperation path length $l_{co}$ of any UAV $UAV_i$. If the path length planned by $UAV_i$ is essentially the same as that planned by $l_{co}$ and does not collide with threats and other UAVs, then it can be said that the planned path of $UAV_i$ meets the 4D cooperation constraints. Therefore, we designed a cost function that is able to meet the time cooperation constraints, namely the 4D cost function, to evaluate and optimize the tree structure constructed by the algorithm to ensure that the planning path can meet the 4D cooperation constraints.

Unlike the classical RRT* algorithm, the 4D cost function no longer aims to optimize the shortest path but to optimize the length of the cooperative path. The 4D cost function of any UAV is defined as follows:

$$f^{4D}_T(x) = \left| f_T(x) - l_{co} \right|$$

where $f_T(x) = g_T(x) + \lambda h(x)$. $g_T(x)$ represents the actual distance from the root node of tree $T_i$ to node $x$, $h(x)$ represents the estimated distance from node $x$ to the goal of the $UAV_i$, and $\lambda \in \mathbb{R}^+$ represents the coefficient.

2. The 4DRRT* Algorithm Process

Functions 4DRRT*, 4DRRT*_rewire, and 4D_Extandin Algorithm 2 show the overall flow of the 4DRRT* algorithm, taking $UAV_i$ as an example.

After obtaining the sampling nodes in the current planning space through to the end node-guided offset sampling strategy, the algorithm enters the process of tree node expansion based on the 4D cost function guidance (i.e., 4D_Extand). Similar to classical RRT*, 4DRRT* first calculates the expansion distance between the tree node $x_T$ and the sample node $x_{\text{rand}}$. It then uses the constraints defined in Definition 1 (corresponding to UAV_free), Definition 3 (corresponding to obstacle_free), and Definition 4 (corresponding to Conflict_free) to determine the tree node $x_T$. If the constraints are met, the 4D cost function is used to calculate the time cooperation cost. After all tree nodes are calculated, the tree node with the lowest cost for the expansion is selected.

After obtaining the extended new node $x_{\text{new}}$ guided by the 4D cost function, a neighborhood centered on $x_{\text{new}}$ is built and the node in the tree structure within its range is found in order to form the set $X_{\text{near}}$. Then, the optimization process of the neighborhood parent node is begun (i.e., 4DRRT*_rewire is called for the first time). First, the constraint judgment is performed on the edge $(x_{\text{near}}, x_{\text{new}})$ composed of the new node $x_{\text{new}}$ and the node $x_{\text{near}}$ in the neighborhood. If the UAV performance constraints, environmental constraints, and spatial cooperation constraints in the 4D cooperation constraints are met, the cost of taking $x_{\text{near}}$ as the potential parent node of $x_{\text{new}}$ is calculated. After all the neighborhood nodes are calculated, the region node $x_{\text{near}}$ with the lowest cost is selected as the optimal parent node of $x_{\text{new}}$. Similarly, the 4DRRT*_rewire function is called for the second time in order to complete the reconnection of the tree structure. The purpose of this
call is to optimize the parent node of \( x_{\text{near}} \) by taking \( x_{\text{new}} \) as the potential parent node of all neighborhood nodes \( x_{\text{near}} \) to reduce the cost.

When the new node \( x_{\text{new}} \) is extended to the terminal range of UAV\(_i\), a potential 4D cooperative path \( \sigma_i(s_i, g_i) \) is obtained through the backtracking tree structure. If the deviation between the path length \( \text{Cost}_T(s_i, g_i) \) of \( \sigma_i(s_i, g_i) \) and the cooperative path length is within the tolerance \( \epsilon \), the algorithm is terminated and outputs the 4D cooperative path with \( \sigma_i(s_i, g_i) \) as UAV\(_i\). Otherwise, the tree structure continues to expand in order to reduce the 4D cost until the tolerance between \( \sigma_i(s_i, g_i) \) and the cooperative path length \( l_{co} \) is less than \( \epsilon \).

3.3. Sudden Threat Avoidance Planning Based on the HAPF Algorithm

When sudden threats occur in the planning space and block the 4D cooperative path of the UAVs, local path adjustment is required for the affected UAVs to avoid these sudden threats and space conflicts with other UAVs. In addition, because the UAV is in flight, greater efficiency of local obstacle avoidance planning algorithms is usually required. As an online onboard local obstacle avoidance planning algorithm, the APF has the characteristics of simple calculation, high planning efficiency, and can deal with various state threats, which are able to meet the local cooperative obstacle avoidance path planning of multiple UAVs. However, the traditional APF still has some shortcomings, such as unreachable targets, the risk of falling into local minima, and path oscillation when approaching threats, which lead to the failure of local obstacle avoidance in cooperative path planning.

Therefore, in order to ensure the efficiency and robustness of local dynamic cooperative path planning, this paper proposes a heuristic artificial potential field algorithm (HAPF). The proposed algorithm can effectively solve the shortcomings of the traditional APF to ensure the success rate of local cooperative collision avoidance path planning. The HAPF algorithm will be introduced in detail regarding the following three aspects: the improved APF function, heuristic sub-target strategy, and memorial potential field force strategy.

3.3.1. Improved APF Function

It can be seen from the traditional APF method that the attraction force generated by the attraction potential field decreases with the continuous approach of UAVs, while the repulsion generated by the obstacle repulsion potential field increases when the UAV approaches an obstacle. If the threat appears near the end node of the local space, a situation will arise where the repulsion force is greater than the attraction force, resulting in the UAV not being able to continue to fly to the end node under the effect of the repulsion force. This situation is described as that of an unreachable target. In order to solve the problem, the potential field function needs to be improved to ensure that the attraction force can continue to pull the UAV to its goal even when the UAV is near the goal and threats are extant. The goal is guaranteed to be the global minimum node of the resultant potential field, thus enabling the UAV to stop stably when it reaches its goal.

Therefore, the design strategy of the repulsion potential field function in [39] is introduced in this paper, i.e., the distance between the end node and the UAV is added to the repulsion potential field function. The improved repulsion potential field function is as follows:

\[
U_{\text{rep}}(X) = \begin{cases} 
\frac{1}{2}k_{\text{rep}} \left( \frac{1}{\rho(X, X_o)} - \frac{1}{\rho_0} \right)^2 \rho^n(X, X_g) & \rho(X, X_o) \leq \rho_o \\
0 & \rho(X, X_o) > \rho_o 
\end{cases}
\]  

where \( n \in \mathbb{R}^+ \). Then the repulsion function is:

\[
F_{\text{rep}}(X) = -\nabla U_{\text{rep}}(X) = \begin{cases} 
F_{\text{rep}1} + F_{\text{rep}2} & \rho(X, X_o) \leq \rho_o \\
0 & \rho(X, X_o) > \rho_o 
\end{cases}
\]
where:
\[
\begin{align*}
F_{rep1}(X) &= k_{rep}\left(\frac{1}{\rho^n(X_{Xo})} - \frac{1}{\rho^n(X_{Xg})}\right) \rho^n(X_{Xo}) \frac{\partial \phi(X_{Xo})}{\partial X} \\
F_{rep2}(X) &= -\frac{k_{rep}}{2}\left(\frac{1}{\rho^n(X_{Xo})} - \frac{1}{\rho^n(X_{Xg})}\right) \rho^{n-1}(X, X_g) \frac{\partial \phi(X_{Xo})}{\partial X}
\end{align*}
\] (13)

In the above formula, it can be seen that \(F_{rep1}\) is the repulsive force of the obstacle against the UAV and \(F_{rep2}\) is the attractive force of the UAV to the goal. As the UAV approaches the goal, \(F_{rep1}\) gradually decreases to zero. Therefore, under the effect of the improved potential field resultant force, the UAV will still approach the end node even if there is a threat nearby and finally reach the goal.

Furthermore, considering the constraint problem of UAVs, it is necessary to add the constraints of the maximum steering angle \(\phi_{max}\) and the maximum climbing/diving angle \(\gamma_{max}\) of UAVs when calculating the resultant force direction. The steering angle constraint condition is \(-\phi_{max} \leq \Delta\phi \leq \phi_{max}\) and the constraint of climb/dive angle is \(-\gamma_{max} \leq \Delta\gamma \leq \gamma_{max}\). \(\Delta\phi\) and \(\Delta\mu\) are, respectively, the angular changes of the angle between the resultant forces of the current initial step and the next step in the horizontal plane and the vertical plane.

3.3.2. Heuristic Sub-Target Strategy

When using the APF for local conflict avoidance planning, the goal is designed as the global minimum of the potential field, so that the UAV moves in the descending direction of the potential field. However, in a complex environment involving other UAVs and various threats, there may be local potential field minima other than the goal which causes the UAVs to enter the local potential field minima by mistake, leading to the failure of path planning. Note that the reason for the local minimum is that the resultant force is zero. Therefore, the ideas to solve this problem are generally divided into three categories: optimization approaches based on control theory [40], changing repulsive potential field [41], and changing attractive potential field [42]. The optimization method based on control theory usually uses the optimal control method to get the UAV out of the trap when encountering the local minimum. However, this paper mainly studies the path planning algorithm of general cooperative UAVs and does not involve the description of some control variables. In addition, changing the repulsion potential field means placing a virtual obstacle between the UAV and the target point to make the UAV quickly disengage from the local minimum. This method has certain advantages in the path planning of a single UAV. However, for local collision avoidance planning of multiple UAVs, adding virtual obstacles may affect the obstacle avoidance planning of multiple UAVs. In other words, this method will force multiple UAVs to re-plan the path to increasing the calculation cost. Similarly, changing the attractive potential field refers to setting virtual sub-target points for the UAV trapped in the local minimum trap to quickly break away from the local minimum by changing the attractive potential field. After the UAV reaches the sub-target point, change the attractive potential field to the final target point to make the UAV continue to fly. This method has the same characteristics as adding virtual obstacles; that is, it is simple to calculate without additional calculation process. Therefore, real-time performance can be guaranteed. Meanwhile, the virtual sub-target point is only effective for the affected UAV, thus eliminating the disadvantage that the virtual obstacle affects other UAVs. However, random setting of sub-target points may lead to high cost of path length or large angle deflection of UAV when escaping from the local minimum. Therefore, in order to further improve the sub-target guidance strategy, this paper proposes a heuristic sub-target strategy to optimize the selection of sub-target points. This strategy designs a cost function that considers both path length and angle constraints, and further optimizes the path on the basis of ensuring that the UAV can escape the local minimum, so as to improve the path quality on the basis of ensuring real-time and reliability.

The heuristic sub-target strategy adopts the idea of transforming multi-objective optimization problems into single-objective optimization problems to handle sub-target
point selection [43] and introduces it into the HAPF algorithm. The strategy includes four steps:

- Determining whether the UAV is trapped in a local minimum trap;
- If it is trapped, set the sub-target node selection range according to the UAV constraints;
- Set the heuristic function and selecting the sub-target node with the minimum cost as the generating node of the gravitational potential field;
- After reaching the sub-target node, change the target node to the final target node.

The main process and algorithm are as follows:

Before the UAV falls into the local minimum, the UAV always flies in the direction of low potential energy. If the UAV falls into the local minimum node, it will vibrate near this node; that is, the resultant energy will vibrate. Therefore, we can set the condition of falling into the local minimum as:

\[ U_{\text{all}}(t + \Delta t) - U_{\text{all}}(t) > \epsilon \]

where \( U_{\text{all}} \) is the combined potential energy at the current initial step and \( U_{\text{all}}(t + \Delta t) \) is the combined potential energy at the next step. When falling into the local minimum trap, the potential energy at the next time will be greater than that at the current initial time, so \( \epsilon \geq 0 \). If the decision is correct, escape processing will be carried out through the algorithm. Considering the flight constraints of UAVs, \( \epsilon \) is set to zero in this paper.

After detecting that the UAV is trapped in a local minimum, a partial virtual sphere \( R_t \) is established with the current path node \( X_t \) as the center and \( r_{\text{min}} \) as the radius, as shown in Figure 1, and the calculation of \( r_{\text{min}} \) is as follows:

\[ r_{\text{min}} = \min \rho(X_i, X_j) = 1, \ldots K \]  

(15)

![Figure 1. Heuristic sub-target strategy.](image)

The scope of the virtual sphere is determined by the constraints of the UAV and the scope of the sphere is determined by the maximum steering angle \( \phi_{\text{max}} \) and the climbing/diving angle \( \gamma_{\text{max}} \) of the UAV, respectively. In the horizontal plane, the range angle \( \phi_t \) is selected as \( \phi - \phi_{\text{max}} \leq \phi_t \leq \phi + \phi_{\text{max}} \). Then, in the vertical plane, the range angle \( \mu_t \) is selected as \( -\gamma_{\text{max}} \leq \gamma_t \leq \gamma_{\text{max}} \).

If the resultant force of UAV at path node \( X_j \) is \( F = (f_x, f_y, f_z) \), then:

\[ \phi = \arctan(f_y/f_x) \]

(16)

The partial sphere \( R_j \), which determines the range, is shown in Figure 1. \( R_t \) is placed around uniformly distributed \( n_0 \) nodes to construct a sub-target node set \( \{G_1, G_2, \ldots G_n\} \). \( \omega_1 \) is the angle between \( X_iG_i \) and \( X_iG_0 \), and \( \omega_2 \) is the angle between \( X_iG_i \) and \( G_iG_0 \). \( \Delta \omega = |\omega_2 - \omega_1| \) is set as the angle cost of selecting \( G_i \) as the sub-target node. The end node is \( G_0 \). Then, the heuristic function can be defined as:

\[ f = \hat{\xi}_1 D + \hat{\xi}_2 A \]

(17)

where \( \hat{\xi}_1 \) and \( \hat{\xi}_2 \) are the cost function weights. If \( \hat{\xi}_1 > \hat{\xi}_2 \), heuristic sub-target nodes with lower length costs can be obtained, but there may be significant angle changes. If \( \hat{\xi}_1 < \hat{\xi}_2 \), heuristic sub-target nodes with a lower cost of angle change can be obtained, but the
planned path length to the sub-target will increase. In this paper, we choose $\xi_1 = \xi_2 = 0.5$. $D$ is the cost function of the relative distance between the sub-target and the final target to reduce the path loss. $A$ is the cost function of angle conversion to smooth the path and avoid an overly large turning angle of the UAV. $G_i$ is any node in the node group. Of these parameters:

$$D = \frac{\rho(G_i, G_0) - \min \rho(G_i, G_0)}{\max \rho(G_i, G_0) - \min \rho(G_i, G_0)}$$

$$A = \frac{\Delta \omega - \min \Delta \omega}{\max \Delta \omega - \min \Delta \omega}$$

Then, the optimal sub-target node $G_i$ can be selected through the heuristic function:

$$G_i = \arg \min f$$

After the UAV reaches the sub-target, the attractive potential field is constructed at the goal, so that the UAV will fly toward the goal.

3.3.3. Memorial Potential Field Force Strategy

When using the potential field method to plan local collision avoidance near threats, UAVs will dominate the attraction of the current path node, and the planned path direction will tend toward the target node. However, at the next path node, due to the proximity to the threat, the repulsive force may dominate, and the planned path will tend to deviate from the direction of the target node. The cycle of the two situations may cause the planned path to oscillate.

UAV flight constraint optimization or some path smoothing methods are usually used to improve the local path oscillation problem existing in APF. Flight constraint optimization is to improve the oscillation problem by restricting the maximum turning angle of the UAV. However, when the vibration amplitude between adjacent nodes is large, the UAV will repeatedly turn back and forth at the maximum turning angle during flight and the vibration will still exist, which can also cause certain potential safety hazards. Therefore, the constraining of the flight angle is only applicable to some situations with small vibration amplitudes. In addition, the approach of trajectory optimization can be used to improve the vibration. Usually, the approach of trajectory optimization needs to obtain the vibration path and then perform a secondary calculation to smooth the trajectory, which does not meet the real-time requirements of online local obstacle avoidance planning in this paper. Furthermore, inspired by sliding mode control in the early work, we try to use the sigmoid function to smooth the oscillation path. This approach can obviously improve the vibration, but there is a certain lag. Therefore, we adopted the memory resultant strategy proposed in this paper to improve local oscillation. This approach has the following two advantages.

- Simple calculation, high efficiency, and good real-time.

The memory resultant force strategy only needs to record the resultant force of the previous step and fit it with the resultant force of the current step, without additional complex optimization or calculation processes, so it can ensure the real-time performance of the local obstacle avoidance planning level.

- Capable of adaptive adjustment.

The memory resultant force is calculated by the adjacent two-step resultant force. When the resultant force deviation of the two steps is large, i.e., when the potential planned path has a large oscillation, the direction of the memory resultant force can obtain a large deflection compared with the current step resultant force, so it can effectively improve the oscillation. When the resultant force deviation of the two steps is small; that is, when the potential planned path has a small oscillation, the resultant force deviation of the memory resultant force and the current step are also relatively small, so as to smooth the collision avoidance path.
Therefore, this paper proposes a memory resultant strategy to dynamically improve local path oscillations. This strategy can still suppress the vibration when the resultant direction of adjacent path nodes changes significantly. After UAV$_i$ calculates the resultant force direction to the next path node $X_{j+1}$, it still retains the resultant force direction to the path node $X_j$. The resultant force on the two path nodes is fused to reduce the variation of the resultant force angle and improve local vibration. The calculation method is as follows:

$$X_{j+1} = X_j + \delta t(\lambda_1 \frac{F_j}{||F_j||} + \lambda_2 \frac{F_{j+1}}{||F_{j+1}||})$$

(21)

where $F_j$ and $F_{j+1}$ are the resultant forces of the $j$ and $j+1$ path nodes, respectively. $\lambda_1$ and $\lambda_2$ are weights, and $\lambda_1 + \lambda_2 = 1$. The value of the weight depends on the change of the resultant force angle of path node $j$ and path node $j+1$, i.e., the amplitude of the oscillation. In the simulation, we found that if the resultant angle change $\Delta \alpha \leq \frac{\pi}{4}$ is small, then $\lambda_1 \leq \lambda_2$ should be adjusted. If the resultant angle change is $\frac{\pi}{4} < \Delta \alpha \leq \frac{\pi}{2}$, $\lambda_1 > \lambda_2$ should be adjusted.

3.4. Time Complexity Analysis

This section mainly analyzes the time complexity of the proposed algorithm (i.e., Algorithm 2). The proposed algorithm is mainly composed of three levels; namely, the pre-cooperative planning level, the cooperative path planning algorithm for a single UAV level based on DPRRT* (i.e., Algorithm 2), and the local dynamic threat avoidance planning level based on HAPF.

The purpose of the pre-cooperative planning level is to initialize the path of UAVs, i.e., to plan the optimal path for all UAVs without considering the 4D cooperative constraints. Therefore, in this level, we adopt the synchronous decentralized architecture and use RRT* algorithm to plan the optimal path for all UAVs. The pseudocode of the RRT* algorithm is shown in Algorithm A1 of “Appendix A”. As described in [44,45], the computational complexity of the RRT* algorithm is similar to that of the RRT algorithm and RRG algorithm, which run in time $O(n \log n)$. The other steps of the pre-planning level are only executed once, and the time complexity is very low. Therefore, it can be proved that the computational complexity of the pre-planning level of the proposed algorithm is $O(n \log n)$.

Then, we derive the time complexity of the DPRRT* algorithm (i.e., Algorithm 2). As can be seen from the main program (i.e., MAIN ALGORITHM) of Algorithm 2, DPRRT* is mainly composed of two parts; namely, the dynamic hydraulic prioritization strategy (FunctionHeuDyn_Prioritization) and the 4DRRT* algorithm (Function 4 DRRT*). The time complexity of the dynamic hydraulic prioritization strategy is $O(n)$. We derive the time complexity of 4DRRT* by comparing it with RRT* [44]. The main difference between the two algorithms is the cost function. The purpose of the cost function of RRT* is to calculate the shortest path, while the purpose of the cost function of 4DRRT* is to calculate the path with the shortest ETA error. The different cost functions lead to the difference between the expansion process of the two algorithms and the update process of the neighborhood optimal node. However, the highest order of the two cost functions is the same, so the time complexity of the above two processes is basically the same. In addition, the collision checking process (Conflict_free) and the sampling process (Algorithm 2 Line 5–9 and Algorithm A1 Line 3–7) of the two algorithms are the same, and they can be executed in a certain number of steps. Therefore, for the 4DRRT* and RRT*, since each iteration has only one sampling, they have the same number of nodes in the iteration $i$, i.e., $\gamma_i^{RRT*} = \gamma_i^{4DRRT*}$. Therefore, the two algorithms run in time $O(\log n)$ in iteration $i$. Therefore, when the number of iterations $i$ tends to infinity, similar to [40], we can prove that the time complexity of the two algorithms is basically the same, which runs in time $O(n \log n)$. Therefore, when the number of UAVs is $\delta$, the DPRRT* algorithm (Algorithm 2) runs in time $O(\delta n \log n)$.
The proposed HAPF algorithm is improved from the APF algorithm. Assume that the worst iteration number of HAPF is \( n \), where \( n \to \infty \). In other words, after \( n \) steps expansion, HAPF reaches the target point. Then, HAPF can run in time \( O(n) \).

**Corollary 1.** 4D Dynamic Cooperative path planning algorithm runs in time \( O(\delta n \log n) \), where \( n \) is the number of samples and \( \delta \) is the number of UAVs.

### 4. Simulation Results and Discussion

In this section, we carried out a simulation on the 4D dynamic cooperative path planning algorithm of UAVs and compared the proposed algorithm’s cooperative path planning layer and local anti-collision planning layer to relevant algorithms, respectively, thus proving the effectiveness, robustness, and high efficiency of the algorithm.

#### 4.1. Environments and Parameters

In the simulation of the proposed algorithm, we designed two types of simulated scenarios: the normal scenario (Scenario 1) and the complicated scenario (Scenario 2). For each UAV, we assume that they all have their own independent CPU computing units, which are able to achieve synchronous preplanning and local anti-collision planning. At the same time, each UAV has a completely reliable zero-delay communication unit, thus enabling the transmission and reception of path information. In addition, all UAVs in the two scenarios take off at the same time.

1. **Normal Scenario (Scenario 1).** In this scenario, we set up three UAVs with the same performance. All UAVs take off from different starting nodes at the same time and arrive at the same destination. Figure 2 shows the planning environment of the Normal Scenario, including radar, missiles, artilleries, no-fly towers, moving threats, and terrain threats based on elevation maps. Of these, moving threats are sudden unknown threats, and UAVs need to conduct dynamic cooperative path planning to avoid them during flight. See Table 1 for relevant parameters of general scenarios.

![Figure 2](image-url)

**Figure 2.** \( t = 0 \) s cooperative paths of UAVs, where (a) is the 3D view and (b) is the 2D view.
### Table 1. Normal Scenario.

<table>
<thead>
<tr>
<th>Environmental information</th>
<th>Center</th>
<th>Radius</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rader</strong></td>
<td>1:(100, 80, 0)</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>2:(100, 350, 5)</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>3:(170, 230, 20)</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>4:(280,200,20)</td>
<td>35</td>
</tr>
<tr>
<td><strong>Missile</strong></td>
<td>1:(70, 170, 0)</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>2:(170, 140, 0)</td>
<td>30</td>
</tr>
<tr>
<td><strong>Artillery</strong></td>
<td>1:(300, 100, 0)</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>2:(260, 180, 0)</td>
<td>25</td>
</tr>
<tr>
<td><strong>No-fly Tower</strong></td>
<td>1:(200, 290, 10)</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>2:(100, 275, 10)</td>
<td>20</td>
</tr>
<tr>
<td><strong>Sudden moving threat</strong></td>
<td>Center $(t = 0)$</td>
<td>Speed</td>
</tr>
<tr>
<td></td>
<td>(110.5, 300, 0)</td>
<td>3 m/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UAVs</th>
<th>Start position</th>
<th>Goal position</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV 1</td>
<td>(40, 40, 40)</td>
<td>(380, 200, 50)</td>
<td>19.91 m/s</td>
</tr>
<tr>
<td>UAV 2</td>
<td>(40, 200, 40)</td>
<td>(380, 200, 50)</td>
<td>20 m/s</td>
</tr>
<tr>
<td>UAV 3</td>
<td>(40, 360, 40)</td>
<td>(380, 200, 50)</td>
<td>18.53 m/s</td>
</tr>
</tbody>
</table>

2. Complicated Scenario (Scenario 2). In this scenario, we set up five UAVs with the same performance. All UAVs take off from different starting nodes at the same time and arrive at the same goal. Figure 8 shows the planning environment of the Complicated Scenario, including radar, missiles, artilleries, no-fly towers, moving threats, and terrain threats based on elevation maps. Of these, sudden threats appear many times at different times, and their status is static or moving. In addition, a sudden threat group (simulating the possible local minimum traps in the local collision avoidance planning) is set in Scenario 2. See Table 2 for the relevant parameters of the Complicated Scenario.

#### 4.2. Scenario 1: Normal Scenario

In Scenario 1, the relevant parameters of the HAPF are set as follows: the attractive potential field coefficient is $k_{att} = 8$ and the repulsive potential field coefficient is $k_{rep} = 30$. The influence distance of obstacle threat is $\rho_o = 20$ m, the unit time step is $t = 0.01$ s, and the adjustment factor is $n = 0.7$. The maximum steering angle is $\phi_{max} = 60^\circ$ and the maximum climbing/diving angle is $\gamma_{max} = 35^\circ$. The generating node of the attractive potential field is the path node $[157,296,50]$. Figure 2 shows the 4D cooperative path (i.e., the generation path of the initial cooperative planning layer) when the UAV does not detect the moving threat. Figures 3–6 show the collision avoidance path of the UAV at different times after detecting the moving threat (i.e., the generation path of local collision avoidance planning layer). The 4D dynamic cooperative replanning path of the UAV is shown in Figure 7 (i.e., the generation path of the cooperative planning layer). Table 3 shows the shortest cooperation time for 4D cooperative replanning. See Table 4 for the deviation ratio between the actual path length obtained by the 4D cooperative path algorithm and $l_{co}$. 
### Table 2. Complicated Scenario.

<table>
<thead>
<tr>
<th>Environmental information</th>
<th>Center</th>
<th>Radius</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rader</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:(200, 80, 0)</td>
<td></td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>2:(100, 350, 5)</td>
<td></td>
<td>35</td>
<td></td>
</tr>
<tr>
<td><strong>Missile</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>(70, 170, 0)</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td><strong>Artillery</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td></td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Radius</td>
<td></td>
<td>25</td>
<td>40</td>
</tr>
<tr>
<td>Height</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:(300, 100, 0)</td>
<td></td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>2:(260, 180, 0)</td>
<td></td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td><strong>No-fly Tower</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td></td>
<td>15</td>
<td>50</td>
</tr>
<tr>
<td>Radius</td>
<td></td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Height</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:(200, 290, 10)</td>
<td></td>
<td>15</td>
<td>50</td>
</tr>
<tr>
<td>2:(100, 275, 10)</td>
<td></td>
<td>20</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sudden moving threat</th>
<th>Center</th>
<th>Speed</th>
<th>Moving direction</th>
<th>Appearing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t = 4.7s)</td>
<td>(126, 125, 0)</td>
<td>3 m/s</td>
<td>Positive direction of Y-axis</td>
<td>t = 4.7 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sudden threats group</th>
<th>Center</th>
<th>Radius</th>
<th>Height</th>
<th>Appearing time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(t = 4.7s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:(260, 205, 0)</td>
<td></td>
<td>15</td>
<td>50</td>
<td>t = 16.5 s</td>
</tr>
<tr>
<td>2:(270, 215, 0)</td>
<td></td>
<td>15</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UAVs</th>
<th>Start position</th>
<th>Goal position</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV 1</td>
<td>(20, 100, 40)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UAV 2</td>
<td>(20, 150, 40)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UAV 3</td>
<td>(20, 200, 40)</td>
<td>(380, 200, 50)</td>
<td>15 m/s</td>
</tr>
<tr>
<td>UAV 4</td>
<td>(20, 250, 40)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UAV 5</td>
<td>(20, 300, 40)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It can be seen in Figures 3–6 that, under the guidance of the HPAF algorithm of the local anti-collision planning layer, UAV3 is able to evade sudden moving threats. It should be noted that the path of UAV2 in Figure 6 shows that the moving threat could involve a collision. However, HAPF is an online planning algorithm. Each time a collision avoidance path in a unit time step is planned, the UAV flies along the planned path. When UAV3 has flown for 6 s, it reaches the path node with coordinates [147.6170, 326.0181, 49.2858]. When \( t = 6 \) s, the path node avoids the moving threat, so there is no collision. Finally, after 7.59 s of flight, UAV3 reaches the sub-target node; that is, the preplanned path node [157, 296, 50]. When \( t = 7.59 \) s, UAV3 has completed the avoidance of the moving threats. While UAV 2 evades the sudden moving threat, other UAVs fly along the initial cooperative path.

At this point, the 4D cooperative planning layer comes into play. It can be seen in Table 3 that the shortest cooperation time is 12.22 s. Based on this, the \( l_{co} \) of each UAV can be obtained to ensure time coordination. It can be seen in Figure 7 that after HAPF is used to circumvent the sudden moving threat, the proposed algorithm uses the DPRRT* algorithm in the cooperative planning layer to replan the 4D cooperative path of all UAVs. Due to the constraints of spatial cooperation, the cooperative paths of the three UAVs do not collide, thus meeting the requirements of spatial cooperation. At the same time, combining the deviation between the expected cooperation path and the actual planned cooperation path in Table 4, it can be seen that the deviation percentage is within the allowable range, meeting the time cooperation requirements.
Due to the constraints of spatial cooperation, the cooperative paths of the three UAVs do not collide, thus meeting the requirements of spatial cooperation. At the same time, combining the deviation between the expected cooperation path and the actual planned cooperation path in Table 4, it can be seen that the deviation percentage is within the allowable range, meeting the time cooperation requirements.

Based on the above simulation results and analysis, the proposed algorithm is able to realize the planning of a 4D dynamic cooperative path of all UAVs in the Normal Scenario, thus proving the effectiveness of the proposed algorithm.

### Table 4.
The deviation ratio between the actual path length obtained by the 4D cooperative path algorithm and the expected path length.

<table>
<thead>
<tr>
<th>UAV</th>
<th>Expected 4D Path Length</th>
<th>Actual 4D Path Length</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV 1</td>
<td>243.36</td>
<td>244.85</td>
<td>0.61%</td>
</tr>
<tr>
<td>UAV 2</td>
<td>226.49</td>
<td>226.37</td>
<td>0.05%</td>
</tr>
<tr>
<td>UAV 3</td>
<td>244.42</td>
<td>244.42</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

**Figure 3.** \( t = 3 \) s cooperative paths of UAVs, where (a) is the 3D view and (b) is the 2D view.

**Figure 4.** \( t = 4.5 \) s cooperative paths of UAVs, where (a) is the 3D view and (b) is the 2D view.

**Figure 5.** \( t = 6 \) s cooperative paths of UAVs, where (a) is the 3D view and (b) is the 2D view.
Figure 6. $t = 7.59$ s cooperative paths of UAVs, where (a) is the 3D view and (b) is the 2D view.

Figure 7. 4D cooperative paths of UAVs, where (a) is the 3D view and (b) is the 2D view.

Table 3. Relevant parameters of 4D cooperative replanning after avoiding sudden threats.

<table>
<thead>
<tr>
<th>Current Position</th>
<th>Unexecuted Path</th>
<th>Speed</th>
<th>ETA</th>
<th>$l_{co}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV1</td>
<td>[177.88, 101.08, 38.88]</td>
<td>225.46</td>
<td>19.915</td>
<td>11.32</td>
</tr>
<tr>
<td>UAV2</td>
<td>[177.53, 184.60, 41.92]</td>
<td>209.83</td>
<td>18.534</td>
<td>11.32</td>
</tr>
<tr>
<td>UAV3</td>
<td>[157, 296, 50]</td>
<td>244.42</td>
<td>20</td>
<td><strong>12.22</strong></td>
</tr>
</tbody>
</table>

Table 4. The deviation ratio between the actual path length obtained by the 4D cooperative path algorithm and $l_{co}$.

<table>
<thead>
<tr>
<th>Expected 4D Path Length</th>
<th>Actual 4D Path Length</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV 1</td>
<td>243.36</td>
<td>244.85</td>
</tr>
<tr>
<td>UAV 2</td>
<td>226.49</td>
<td>226.37</td>
</tr>
<tr>
<td>UAV 3</td>
<td>244.42</td>
<td>244.42</td>
</tr>
</tbody>
</table>
Based on the above simulation results and analysis, the proposed algorithm is able to realize the planning of a 4D dynamic cooperative path of all UAVs in the Normal Scenario, thus proving the effectiveness of the proposed algorithm.

4.3. Scenario 2: Complicated Scenario

In Scenario 2, the relevant parameters of the HAPF are set as follows: the attractive potential field coefficient is $k_{\text{att}} = 8$ and the repulsive potential field coefficient is $k_{\text{rep}} = 30$. The influence distance of obstacle threat is $\rho_o = 20$ m, the unit time step is $t = 0.01$ s, and the adjustment factor is $n = 0.7$. The maximum steering angle is $\phi_{\text{max}} = 60^\circ$ and the maximum climbing/diving angle is $\gamma_{\text{max}} = 35^\circ$. It can be seen in Table 2 that we have set up two kinds of sudden threats at different times. The first one is a mobile threat and the second one is a static threat group simulating a local minimum trap. Figures 8–10 show the pre-planned 4D coordination path and the satisfaction of 4D coordination constraints of the path when $t = 0$ s. Figure 11 shows the local collision avoidance planning path of some UAVs and the 4D cooperative replanning path of all UAVs after the first sudden moving threat. Figures 12 and 13 show the satisfaction of 4D cooperative constraints of the pre-planned path based on the current planning space. Figure 14 shows the dynamic planning path of the proposed algorithm when affected by the sudden threat group for the second time, including local collision avoidance planning and 4D cooperative path replanning. Figures 15 and 16 show the satisfaction of 4D cooperative constraints of the planned path.

![Figure 8](image1.png)

**Figure 8.** 4D cooperative pre-planning when $t = 0$ s in the complicated scenario. (a) is the 3D view and (b) is the 2D view.

![Figure 9](image2.png)

**Figure 9.** The shortest distance between all UAVs at any time.

![Figure 10](image3.png)
Figure 8. 4D cooperative pre-planning when t = 0 s in the complicated scenario. (a) is the 3D view and (b) is the 2D view.

Figure 9. The shortest distance between all UAVs at any time.

Figure 10. Time tolerance for all UAVs to reach the goal.

It can be seen in Figure 9 that the shortest distance between each pair of UAVs at the same time is larger than the minimum safety distance (2 m) we set, so it can be shown that the pre-planned path meets the space constraints in 4D cooperative constraints. At the same time, it can be seen in Figure 10 that the ETA tolerance of the five UAVs executing the pre-planned path is within 0.2 s, i.e., they meet the time cooperative constraints.

Figure 11. Local collision-checking planning and 4D coordinated path replanning when affected by the sudden moving threat for the first time. (a) is the 3D view and (b) is the 2D view.

Figure 12. (a) The shortest distance between UAV1 and UAV2 during local collision-checking planning. (b) The shortest distance between all UAVs in 4D coordinated path replanning.

When t = 4.7 s, a sudden moving threat appears in the planning space for the first time. Compared with Scenario 1, the sudden moving threat in Scenario 2 will affect two UAVs at the same time, which is more complex. The collision avoidance planning of two UAVs based on the HAPF algorithm will be executed online at the same time, and the planning needs to avoid sudden threats while preventing conflicts with other UAVs. Figure 11 shows the path of local collision-checking planning and 4D cooperative replanning.
Figure 11. Local collision-checking planning and 4D coordinated path replanning when affected by the sudden moving threat for the first time. (a) is the 3D view and (b) is the 2D view.

Figure 12. (a) The shortest distance between UAV1 and UAV2 during local collision-checking planning. (b) The shortest distance between all UAVs in 4D coordinated path replanning.

Figure 13. Time tolerance for all UAVs to reach the goal in the first replanning. It can be seen in Figures 11 and 12a that UAV1 and UAV2 can avoid threats in the environment during local collision-checking planning, and two UAVs can maintain a safe distance to avoid conflicts. After completing local collision-checking planning, all UAVs need to conduct cooperative path replanning to meet 4D coordination constraints. It can be seen in Figure 12b that the cooperative replanning path can meet the spatial constraints. In addition, Figure 13 proves that the collaborative replanning path can meet the time coordination constraints.

Figure 14. Local collision-checking planning and 4D coordinated path replanning when affected by sudden threats for the second time. (a) is the 3D view and (b) is the 2D view.

Figure 15. The shortest distance between all UAVs in 4D coordinated path replanning.
It can be seen in Figure 9 that the shortest distance between each pair of UAVs at the same time is larger than the minimum safety distance (2 m) we set, so it can be shown that the pre-planned path meets the space constraints in 4D cooperative constraints. At the same time, it can be seen in Figure 10 that the ETA tolerance of the five UAVs executing the pre-planned path is within 0.2 s, i.e., they meet the time cooperative constraints.

When $t = 4.7$ s, a sudden moving threat appears in the planning space for the first time. Compared with Scenario 1, the sudden moving threat in Scenario 2 will affect two UAVs at the same time, which is more complex. The collision avoidance planning of two UAVs based on the HAPF algorithm will be executed online at the same time, and the planning needs to avoid sudden threats while preventing conflicts with other UAVs. Figure 11 shows the path of local collision-checking planning and 4D cooperative replanning. It can be seen in Figures 11 and 12a that UAV1 and UAV2 can avoid threats in the environment during local collision-checking planning, and two UAVs can maintain a safe distance to avoid conflicts. After completing local collision-checking planning, all UAVs need to conduct cooperative path replanning to meet 4D coordination constraints. It can be seen in Figure 12b that the cooperative replanning path can meet the spatial constraints. In addition, Figure 13 proves that the collaborative replanning path can meet the time coordination constraints.

Similarly, when $t = 16.5$ s, the planning space changes for the second time. We set up a group of sudden threats to simulate the planning of escaping from the local minimum trap in local collision avoidance. It can be seen in Figure 14 that after being affected by threats, UAV3 can plan the path to escape from the local minimum. After that, all UAVs continue to replan the 4D cooperative path, so that all UAVs can reach the goal without conflict. It can be seen in Figures 15 and 16 that the second cooperative path replanning can plan the path satisfying 4D cooperative constraints for all UAVs.

Based on the above analysis, we can prove that the proposed algorithm can still complete the dynamic, 4D cooperative path planning in complex scenarios. Therefore, we prove the effectiveness of the proposed algorithm in different scenarios.

4.4. Comparison and Algorithm Performance Analysis

In this section, we compare the 4D cooperative path planning layer (DPRRT* algorithm) and local collision avoidance planning layer (HAPF algorithm) of the 4D dynamic cooperative path planning algorithm with some related algorithms. On the basis of comparison, the performance of the 4D dynamic cooperative path planning algorithm is analyzed.

4.4.1. Comparative Simulation and Performance Analysis of Cooperative Path Planning Layer

The DPRRT* algorithm is a decoupled cooperative planning algorithm. The generation of the cooperative path can be divided into two stages; namely, the planning sequence sequencing and the single machine cooperative path planning. At the same time, it should be noted that the existing priority planning algorithms and their derivative algorithms have not involved in-depth research on the 4D cooperative path planning problem. Therefore, in order to eliminate the impact of considering both time and space cooperation constraints
on the performance of the algorithm, we uniformly set the cooperation constraints that the path needs to meet in terms of 4D cooperation constraints in comparison with related algorithms to more accurately compare and analyze the performance of the DPRRT* algorithm. In the context of the scenario of cooperative path planning based on the decoupling method, we introduced the prioritized A* algorithm [31], the traditional priority planning algorithm (RPP) [26], and the coupling degree-based heuristic prioritized planning algorithm (CDH-PP) [46] for comparison.

This simulation compares the two types of mission scenarios mentioned in Sections 4.2 and 4.3. To fully analyze the performance of the DPRRT* algorithm in both the Normal Scenario and the Complicated Scenario, four algorithms are used for 4D cooperative path planning of three groups of UAVs with different numbers, where the number of UAVs is set as [2,4,6]. In the two scenarios, the starting and goal nodes of all UAVs are randomly generated in the planning space and 30 random problem situations are generated for each scenario.

The characteristics of the four comparisons are as follows:

1. The longest running time, the shortest running time, and the median running time of the algorithm (analyzing the computational efficiency and robustness of the algorithm).
2. Compared with the RPP algorithm (the single UAV planning layer uses the RRT* algorithm based on 4D constraints), the acceleration ratio is used to evaluate the impact of different heuristic priority ranking strategies or different underlying algorithms on the cooperative planning algorithm. The formula for calculating the acceleration ratio is \( \frac{t_i^{\text{RPP}}}{t_i^{\text{ALG}}} \). Where \( t_i^{\text{RPP}} \) represents the running time of the RPP algorithm and \( t_i^{\text{ALG}} \) represents the running time of the comparison algorithm.
3. The ETA time difference is used to evaluate the quality of the path planned by the algorithm. The calculation formula is \( t_{\text{ALG,MAX ETA}}^{\text{ALG}} - t_{\text{ALG,MIN ETA}}^{\text{ALG}} \) where \( t_{\text{ALG,MAX ETA}} \) represents the maximum flight time of the planned path of the four algorithms and \( t_{\text{ALG,MIN ETA}} \) represents the minimum flight time of the planned path of the four algorithms.
4. The success rate can be used to verify the effectiveness of the algorithm.

It can be seen from the comparison of the running time in Figure 17 that the median running time of the DPRRT* algorithm is the shortest and the difference between the maximum and minimum running times is relatively small. In addition, DPRRT* has the highest acceleration ratio. The comparison results of DPRRT* with RPP and CDH-PP prove that the heuristic prioritization strategy based on time cooperation constraints is more reasonable, efficient, and robust. In addition, the running time comparison between DPRRT* and Prioritized A* proves that the single machine cooperative path planning algorithm based on 4DRRT* in DPRRT* has higher efficiency and stability than the A* algorithm. Since the four algorithms have the ability to find the optimal solution, there is little difference in the ETA. In addition, the four algorithms are decoupled algorithms, so they all have a high success rate.

The comparison results of the four algorithms are shown in Figure 17.

4.4.2. Comparative Simulation and Performance Analysis of the Local Anti-Collision Planning Layer

We simulated the three improvements of the HAPF algorithm and compared them to the classic APF algorithm.

In the simulation, an improved repulsive force function was added to solve the problem of the unreachability of the target. The traditional artificial potential field was compared with HAPF in the path collision avoidance planning of obstacles near the target.

In order to verify the effectiveness of the algorithm and ensure the optimal performance of the traditional algorithm, the traditional algorithm was simulated using the best parameters. At the same time, in order to intuitively show the superiority of the improved potential field function, this simulation used the radar threat as the environment model for verification. The potential field parameters of the HAPF algorithm are \( k_{\text{att}} = 8 \), \( k_{\text{rep}} = 30 \), \( \rho_o = 2 \text{ km} \), \( l = 0.2 \text{ km} \), and \( n = 0.7 \). According to the simulation data, the
generation node of the repulsion potential field corresponding to the last path node is [7.1654, 9.9636, 1.9927] km. The relative distance between this node and the target node is 1.0541 km, which is less than the obstacle influence distance, so obstacle threat near the target node is extant.

Figure 17. Comparison results of the cooperative path planning layers under different scenarios.

Figure 18 shows the comparison between the two algorithms. It can be seen in Figure 18 that HAPF reaches the target node, while the traditional potential field method has the problem of target unreachability. The relative distances between different path nodes and target nodes of UAVs in the two algorithms are recorded separately. It can be seen that when there is an obstacle threat near the target, the traditional artificial potential field method oscillates near the target node and cannot reach the target node. The improved potential field method is able to reach the target node, which shows that the improved algorithm can solve the problem of the unreachable target.
Regarding the problem of the traditional artificial potential field method easily falling into the local minimum, this paper proposes a strategy to set heuristic target nodes in order to escape the local minimum. In the simulation, HAPF is compared with the traditional artificial potential field method. The potential field parameters are set as $k_{att} = 8$, $k_{rep} = 20$, $n = 0.7$, $\rho_o = 2$ km, and $l = 0.2$ km in the simulation. The starting node of the UAV is [0,0,0] km, and the goal is [22,22,5] km.

In order to fully verify the effectiveness of the algorithm, five combined obstacles with different obstacle threats are established in the simulation, and the five obstacles constitute the local minimum area. Figure 19a shows the traditional artificial potential field method and Figure 19b shows the HAPF algorithm. It can be seen in Figure 19a that when the UAV reaches the path node [13.9930, 14.5990, 4.7578] km, it falls into the local minimum value and oscillates back and forth near it, unable to escape it. In Figure 19b after the UAV has been trapped in the local minimum area, it successfully escapes and finally flies to the target node due to the guidance of heuristic sub-target nodes. The virtual spherical radius of the selected sub-target node is 3.1370 km, and the heuristic sub-target node coordinates are [13.1161, 17.5908, 4.9435] km. According to Figure 19 and the simulation data, after adding heuristic sub-target nodes, the UAV is able to escape the local minimum region, which solves the problem of the traditional potential field model easily falling into the local minimum.

In this paper, regarding the problem of the improved potential field easily generating local path vibrations when avoiding obstacles and planning trajectory collision avoidance, a memory resultant force strategy was proposed to improve this problem. In the simulation, by comparing the traditional artificial method with the potential field method, the effect of
resultant force memory on improving the local vibration is verified. In the simulation, the potential field simulation parameters are set as $k_{att} = 8, k_{rep} = 20, n = 0.7, \rho_0 = 2 \text{ km}$, and $l = 0.2 \text{ km}$. In the HAPF algorithm with a memory factor, the value of the memory factor is set to $\lambda_1 = 0.7$ and $\lambda_2 = 0.3$. Figure 20 simulates the flight path of the two algorithms.

![Figure 20](image_url)

**Figure 20.** A comparison of local shocks. (a) Is the complete view. (b) Is the enlarged view of the goal region.

It can be seen in Figure 20 that the UAV flight path was significantly improved after adding the memory resultant force without the local path oscillation. It can also be seen in Figure 20 that after adding the memory resultant force, the variations of resultant force angle between adjacent path nodes of the UAV become significantly smaller and there is no significant angle oscillation. This shows that the introduction of this strategy satisfies the flight constraints of UAVs, does not cause large angle changes between adjacent path nodes, makes the flight path smoother, and is suitable for the collision avoidance trajectory planning of UAVs.

5. Conclusions and Future Work

In this paper, a 4D dynamic cooperative path planning algorithm is proposed to solve the cooperative path planning problem under the 4D cooperative constraints of multiple UAVs. Each UAV shares the same arrival time. There is no conflict between them and they are able to successfully implement threat avoidance. The algorithm proposed in this paper adopts a hierarchical architecture; namely, a 4D cooperative path planning layer and a local anti-collision planning layer. First, the DPRRT* algorithm was used to complete the 4D cooperative preplanning based on the initially known environment before UAV flights. Later, if a UAV encounters sudden threats while in flight, the HAPF algorithm of the local anti-collision planning layer is used to avoid these sudden threats. Simultaneously, UAVs not affected by sudden threats continue to fly along their planned paths. It should be noted that the 4D cooperation constraint cannot be satisfied after the sudden threat avoidance. Therefore, after the sudden threat avoidance has been completed, the DPRRT* algorithm in the 4D cooperative path planning layer is used to replan the 4D cooperative path. Whenever a new threat appears, local obstacle avoidance planning and 4D coordinated path replanning of the threat are required until all UAVs reach the target at the same time. The simulation and comparison experiments show that the proposed algorithm is able to plan a cooperative path, thus satisfying the 4D cooperative constraints for UAVs, and the algorithm has high efficiency, stability, and robustness.

In future work, we plan to extend the proposed algorithm to other application levels; that is, we aim to use multiple UAVs for the actual experimental verification of the proposed algorithm. In addition, considering that the swarm intelligence cooperation composed of a large number of UAVs may be a hot topic in the near future, we plan to improve the
algorithm proposed in this paper and explore a 4D cooperative planning algorithm that can be synchronized, so that the algorithm is able to achieve efficient and robust cooperative path planning for 50 or even 100 UAVs.

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

**Funding:** This research work was funded by the National Natural Science Foundation of China, grant No. 62073266, and the Aeronautical Science Foundation of China, grant No. 201905053003.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** Gratitude is extended to the Shaanxi Province Key Laboratory of Flight Control and Simulation Technology.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Appendix A**

When dealing with the cooperative path planning problem of multiple UAVs in a dynamic threat environment, this paper adopts a hierarchical architecture path planning algorithm; that is, the proposed DPRRT* algorithm is used as the coordination of the cooperative path planning sequence, and the cooperative path planning itself and the improved HAPF is used as the local collision avoidance planner of the UAVs. Therefore, in this section, we introduce the classical RRT* algorithm and the classical APF algorithm.

1. **RRT* Algorithm**

   The sample-based RRT* algorithm, which is shown in Algorithm A1, was proposed by Karaman and Frazzoli [44]. The algorithm introduces the neighborhood node optimization mechanism, which enables the algorithm to find an asymptotic optimal solution on the basis of probability completeness.

   RRT* and its variant algorithms have been widely studied in the field of path planning for single UAVs. However, for the cooperative path planning of multiple UAVs, only relying on RRT* does not result in a good performance, and there is less related research on this topic. It should be noted that the multi-UAV cooperative path planning algorithm based on the decoupling method can be perfectly combined with RRT* to deal with the cooperative path planning problem. In addition, RRT*, based on the cost function guided tree node expansion and optimization, is able to meet specific planning requirements through improvement.

2. **Artificial Potential Field Algorithm**

   The basic idea of the APF is to establish attractive and repulsive potential fields for goals and obstacles, respectively. Then, a collision-free path is generated from the starting node to the goal by combining the effects of attraction and repulsion on the UAVs. The APF only considers the resultant field on the initial path node of the UAV and then calculates the next path node. Therefore, it requires high computing efficiency.
Algorithm A1. Classical RRT* Algorithm

\begin{algorithm}
\begin{algorithmic}[1]
\State initialization $\mathcal{T}, V \leftarrow \{s_i\}, \mathcal{T}, E \leftarrow \emptyset$
\For{$i = 1$ to $n$}
\If{$P : \text{random}([0,1]) < 0.1$}
    \State $x_{\text{rand}} \leftarrow s_i$
\Else
    \State $x_{\text{rand}} \leftarrow \text{Sample}(x_{\text{nearest}})$
\EndIf
\State $x_{\text{nearest}} \leftarrow \text{Nearest}(x_{\text{rand}})$
\State $x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}})$
\If{$\text{obstacle}\_\text{free}(x_{\text{new}}, x_{\text{nearest}})$}
    \State $\mathcal{T}, V \leftarrow \mathcal{T}, V \cup \{x_{\text{new}}\}$
\EndIf
\ForAll{$x_{\text{near}} \in X_{\text{near}}$}
\If{$\text{obstacle}\_\text{free}(x_{\text{near}}, x_{\text{new}}) \land \text{Cost}(s_i, x_{\text{near}}) + \text{Cost}(x_{\text{near}}, x_{\text{new}}) < \text{Cost}(s_i, x_{\text{new}})$}
    \State $x_{\text{near}}, \text{parent} \leftarrow x_{\text{new}}$
\EndIf
\EndFor
\EndFor
\end{algorithmic}
\end{algorithm}

The attractive potential field function $U_{\text{att}}$ and the repulsive potential field function $U_{\text{rep}}$ can be expressed as follows:

$$U_{\text{att}}(x) = \frac{1}{2}k_{\text{att}}\rho^2(x, g)$$  \hspace{1cm} (A1)

$$U_{\text{rep}}(x) = \left\{ \begin{array}{ll}
\frac{1}{2}k_{\text{rep}}(1 - \frac{1}{\rho(x, x_o)})^2 & \rho(x, x_o) \leq \rho_0 \\
0 & \rho(x, x_o) > \rho_0
\end{array} \right.$$  \hspace{1cm} (A2)

where $x = (x, y, z)$ is the coordinate vector of the current position of the UAV, $g = (x, y, z)$ is the coordinate vector of the goal, and $x_o = (x, y, z)$ is the obstacle coordinate vector. $k_{\text{att}}$ and $k_{\text{rep}}$ are the coefficients of the attractive potential field and the repulsive potential field, respectively, and $\rho_0$ is the maximum influence distance of threats, and $\rho(x, x_o) = ||x - x_o||_2$.

Attraction $F_{\text{att}}$ and repulsion $F_{\text{rep}}$ are the negative gradients of their corresponding potential fields:

$$F_{\text{att}}(X) = -\nabla(U_{\text{att}}) = -k_{\text{att}}\rho(x, g)\frac{\partial\rho(x, g)}{\partial x}$$  \hspace{1cm} (A3)

$$F_{\text{rep}}(x) = -\nabla(U_{\text{rep}}) = \left\{ \begin{array}{ll}
k_{\text{rep}}\left(1 - \frac{1}{\rho(x, x_o)}\right) & \rho(x, x_o) \leq \rho_0 \\
0 & \rho(x, x_o) > \rho_0
\end{array} \right.$$  \hspace{1cm} (A4)

The combined force $F_{\text{tot}}$ of the attraction of the UAV and the repulsive force generated by $n$ obstacles is:

$$F_{\text{tot}}(x) = F_{\text{att}}(x) + \sum_{i=1}^{n} F_{\text{rep}}(x)$$  \hspace{1cm} (A5)

Under the guidance of the combined force, the UAV is able to avoid threats and reach its goal. However, the classical APF also has some shortcomings, such as unreachable targets, the risk of falling into a local minimum, path oscillation, etc. These shortcomings may lead to the failure of the various UAVs; therefore, these shortcomings need to be improved to ensure the success of UAV local obstacle avoidance planning.
References

3. Elmeseiry, N.; Alshaer, N.; Ismail, T. A detailed surveyed and future directions of unmanned aerial vehicles (UAVs) with potential applications. Aerospace 2021, 8, 363. [CrossRef]

9. Li, S.; Zhang, R.; Ding, Y.; Qin, X.; Han, Y.; Zhang, H. Multi-UAV Path Planning Algorithm Based on BNN-HHO. Sensors 2022, 22, 9786. [CrossRef]


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