

Article

HDP-TSRRT*: A Time–Space Cooperative Path Planning Algorithm for Multiple UAVs

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, China² Shaanxi Province Key Laboratory of Flight Control and Simulation Technology, Xi'an 710129, China

* Correspondence: liuxiaoxiong@nwpu.edu.cn

Abstract: This paper proposes a fast cooperative path planning algorithm for multiple UAVs that satisfies the time–space cooperative constraints, namely, the RRT* algorithm based on heuristic decentralized prioritized planning (HDP-TSRRT*), which takes into account the simultaneous arrival time variables of each UAV as well as the avoidance of conflicts and threats. HDP-TSRRT* is a hierarchical decoupling algorithm. First, all UAV pre-paths are planned simultaneously at the synchronous decentralized planning level. Second, at the coordination path level, the heuristic decentralized prioritized planning algorithm (HDP) is proposed to quickly complete the coordination process of the path planning sequence. This strategy assigns reasonable and robust priority to all UAVs based on the performance evaluation function composed of the number of potential collisions and the violation of collaboration time of the pre-planned path. Third, the time–space cooperative constraints-based RRT* algorithm (TSRRT*) is proposed at the single-machine cooperative path planning level. Based on this, the algorithm uses multiple sampling and cost evaluation strategies to guide the expansion of new nodes, and then optimizes neighborhood nodes based on the time coordination cost function so as to improve the efficiency of coordination path planning. Simulation and comparison show that HDP-TSRRT* has certain advantages in algorithm performance.

Keywords: multiple unmanned aerial vehicles (multi-UAVs); time–space cooperative path planning; decentralized prioritized planning; heuristic prioritization; rapidly exploring random trees (RRT*)



Citation: Guo, Y.; Liu, X.; Jiang, W.; Zhang, W. HDP-TSRRT*: A

Time–Space Cooperative Path Planning Algorithm for Multiple UAVs. *Drones* **2023**, *7*, 170. <https://doi.org/10.3390/drones7030170>

Academic Editors: Kaiyu Qin, Haitao Nie, Yikang Yang, Xue Li, Jinliang Shao, Lei Shi and Mengji Shi

Received: 3 February 2023

Revised: 25 February 2023

Accepted: 27 February 2023

Published: 28 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, with the complexity of unmanned aerial vehicle (UAV) flight environments and the diversification of mission load requirements, single UAVs with limitations of load and performance have been gradually replaced by multi-UAV cooperative systems with obvious advantages in scalability, applicability and robustness [1]. The application of multi-UAV cooperative systems in civil and military fields (e.g., geophysical survey, wide-area personnel search and rescue, high-risk target penetration and multi-target detection and strike) has received widespread attention. Multi-UAV cooperative path planning (MUCPP) is one of the most basic and important problems to realize the application of multi-UAV system [2].

This paper is devoted to the study of the MUCPP problem, which refers to planning collision-free paths from the start to the goal for multiple UAVs that satisfy the cooperative constraints [3]. As multi-UAV cooperative systems are increasingly used in more complex tasks, such as single- and multiple-target cooperative attacks or multi-source information acquisition at the same time, all UAVs need the same minimum estimated time of arrival (ETA) on the basis of meeting the requirements of path collision avoidance, so as to meet the mission requirements [4]. In other words, the path planning cooperative constraints that meet most of the mission requirements of UAVs need to be composed of both spatial and temporal constraints (i.e., time–space cooperative constraints), which makes it more difficult to deal with the MUCPP problem.

Since the configuration space is highly dimensional and the constraints are various (i.e., environmental, UAV performance and cooperative constraints), solving the MUCPP problem has become a challenging topic and has received widespread attention. In recent years, many different methods have been developed to solve this problem. These methods can be divided into two categories—reactive collision avoidance methods and active cooperative path generation methods. In reactive collision avoidance methods, each UAV only considers planning the optimal path that can avoid threats in the environment, without considering collision avoidance and time coordination with other UAVs. Then, when the risk of collision with other UAVs is detected, local re-planning is started to avoid collision [5]. Many algorithms based on reactive collision avoidance methods have been proposed, such as optimal reciprocal collision avoidance (ORCA) [6], distributed reactive collision avoidance (DRCA) [7] and some intelligent algorithms (e.g., PRIMAL₂ [8] and co-evolutionary genetic programming [9]). The reactive collision avoidance method has been widely used in practice because of its fast calculation speed [10]. However, it should be noted that such algorithms only plan the local collision avoidance path, so they may fall into the local optimal solution, meaning that the UAV cannot reach the goal. In addition, this local collision avoidance method struggles to meet the requirements of time–space constraints at the same time if the time coordination constraint is introduced.

Active cooperative path generation methods are divided into coupling methods and decoupling methods. Coupling methods usually design only one path planner for multiple UAVs. The planner includes the start and goal of all UAVs and can plan collision-free cooperative paths for all UAVs. Some coupling methods based on the reduction method obtain the cooperative path by simplifying the MUCPP problem to a problem that has been fully studied, such as linear programming (LP) [11], constraint satisfaction problem (CSP) [12], etc. In addition, some search-based coupling methods, such as the conflict-based search (CBS) [13] and incremental cost tree search (ICTS) [14], have also been proposed to transform the MUCPP problem into a global single-agent path search problem, and simultaneously generate the cooperative path of all UAVs. The coupling method can usually find the optimal path. However, with the increase in the number of collisions between UAVs and the increase in the dimensions and scope of the planning environment, the calculation time of the cooperative path obtained by the coupling method may increase exponentially. Furthermore, the coupling method has been proved to have a good effect when planning the conflict-free paths of multiple UAVs. However, if we want to deal with the MUCPP problem that meet the time–space cooperative constraints, the computational time of the algorithm will be further increased due to the introduction of time cooperative constraints. In addition, game theory-based optimal cooperative path planning has been deeply studied in recent years. The game-based particle swarm optimization (GPSO) algorithm [15] develops a hierarchical method based on the framework of game theory to extend the spherical vector-based particle swarm optimization (SPSO) algorithm [16]. Ref. [17] combines the Stag Hunt game approach with GSPO. These two algorithms can efficiently find the optimal cooperative path with the formation being maintained and have good performance. However, we are not sure whether these two algorithms can still perform well in a large number of UAV formations due to the formation constraints.

On the other hand, decoupling methods plan the path for each UAV and coordinate the planning order of all UAVs through path coordination methods to obtain the collision free path, such as hierarchical cooperative A* algorithm (HCA*) [18] and prioritized planning algorithm (PP) [19]. Among them, the PP algorithm has been widely studied recently because of its ability to transform the MUCPP problem into the single UAV path planning problem to quickly and efficiently obtain the cooperative path [20,21]. In the PP algorithm, the upper planning coordinator assigns different priorities to all UAVs and then uses the single UAV path planner to plan the path from high priority to low priority. Although the classical PP algorithm can obtain the cooperative path quickly, the centralized framework of PP can be further improved to raise the computational efficiency. The authors of [22,23] proposed an asynchronous decentralized prioritized planning algorithm (ADPPP), which

can make reasonable use of distributed computing resources to achieve faster algorithm convergence. Compared with the centralized prioritized planning algorithm (i.e., classical PP algorithm), the asynchronous decentralized prioritized planning algorithm (ADRPP) proposed in the literature [22,23] can usually complete the generation of cooperative paths with fewer cycles. In addition, compared with the synchronous decentralized prioritized planning algorithm (SDRPP) [24], ADRPP can start the next round of planning without waiting for all UAVs to complete the planning, so it has higher planning efficiency. However, ADRPP algorithm does not provide a prioritization approach that can improve efficiency, so there is some room for improvement. However, the ADRPP and SDRPP algorithm does not provide a prioritization approach that can improve efficiency, so there is some room for improvement. In addition, none of the decoupling methods have been used in in-depth research on the cooperative problem of multiple unmanned aerial vehicles simultaneously. However, after transforming the cooperative path planning problem of multiple UAVs into a single path planning problem, the realization of time–space cooperative constraints is relatively easy and has little impact on the computational efficiency. In addition, decoupling methods have not been used in in-depth research on the MUCPP problem that meets the time–space cooperative constraints. However, it should be noted that after transforming the MUCPP problem into a single-UAV path planning problem, the realization of time–space cooperative constraints is relatively easy, and the computational efficiency is relatively high.

1.1. Motivation

It can be seen from the analysis that the above algorithms still have some room and possibility for improvement:

1. In order to meet the needs of most cooperative tasks of multiple UAVs, it is necessary to introduce the time–space cooperative constraints to deal with the MUCPP. The above algorithms focus more on spatial conflict avoidance but do not consider spatial and temporal cooperative constraints at the same time.
2. The reactive collision avoidance methods mainly focus on conflict avoidance in local areas and struggle to achieve time coordination. The coupling method adds the time coordination constraint to the path planner of multiple UAVs, which will further increase the calculation time and reduce the calculation efficiency. The decoupling method can simplify the MUCPP problem to the single-UAV path planning problem. Compared with the coupling method, the decoupling method has the possibility of introducing time-space coordination constraints by adding less computation.
3. In the PP algorithm, if the random prioritization method is adopted, or only the collision number based on the space cooperative constraint is considered for prioritization, the calculation efficiency of the algorithm will be reduced, and the calculation time will be increased.
4. The existing single-UAV path planning algorithms usually take the shortest path as the optimization standard. Additionally, the RRT* algorithm has been proved to be able to quickly obtain the optimal solution in high-dimensional planning space [25]. However, the existing RRT* algorithm or its variant algorithms cannot meet the time–space coordination constraints of UAVs.

Therefore, in combination with the above motivations, this paper introduces the time–space cooperative constraints into the MUCPP problem so as to meet the needs of most multi-UAV systems to perform tasks. Meanwhile, the decentralized priority planning algorithm based on a decoupling method is further improved by combining the prioritization strategy suitable for time–space cooperative constraints. Furthermore, the RRT* algorithm is improved in the underlying single UAV planner to meet the time–space cooperative constraint, so that the path can be quickly planned for each UAV.

1.2. Contributions

In this paper, we propose the HDP-TSRRT* algorithm to solve the MUCPP problem based on time–space cooperative constraints. HDP-TSRRT* is a decoupled active coopera-

tive path generation algorithm and adopts a hierarchical architecture. First, HDP-TSRRT* uses a synchronous decentralized architecture based on the RRT* algorithm to quickly plan the optimal pre-planning path for each UAV without considering cooperative constraints, which is the foundation for cooperative planning. Second, in the path coordination level, the heuristic decentralized prioritized planning algorithm (HDP) is proposed to coordinate the path planning priorities of the multi-UAV. The algorithm ranks the priority of UAVs by combining the number of potential collisions and the violation of the cooperative time into a performance evaluation function, which makes the priority allocation more reasonable and can improve the efficiency of cooperative planning. The algorithm also uses the asynchronous decentralized prioritized planning framework to coordinate the path planning of multiple UAVs to further improve the computational efficiency. Third, the RRT* algorithm based on time-space cooperative constraints (TSRRT*) is proposed as the execution algorithm of the underlying single-UAV cooperative path planner. The TSRRT* algorithm designs a cost function based on time constraints to gradually optimize the path. In the TSRRT* algorithm, the strategy of multiple sampling and cost function evaluation of optimal sampling proposed in this paper is used to expand the new node, and the time-constrained cost function is used to update the neighborhood nodes to quickly plan the path that meets the time-space cooperative constraint for each UAV.

Compared with the existing cooperative path planning algorithm for multi-UAV, the HDP-TSRRT* proposed in this paper mainly includes the following contributions:

1. A decoupled UAV cooperative path planning hierarchical framework is designed, which is composed of a synchronous decentralized pre-planner, an asynchronous decentralized prioritized planning path coordinator based on heuristic prioritization, and a single-UAV cooperative path planner based on time-space coordination constraint RRT*, to quickly plan a UAV's time-space cooperative path.
2. In the path coordinator, a heuristic prioritization approach based on time-space coordination constraints is proposed to sort the UAVs, so as to ensure the rationality of the prioritization and reduce the calculation time of path coordination.
3. A cost function based on time coordination constraints is established to optimize the planned path of UAVs so as to ensure that the single-UAV cooperative path planner can find the path satisfying the time coordination constraint.
4. In the process of sampling and new node expansion of the single-UAV cooperative path planner, a multi-sampling and cost function evaluation strategy is proposed to expand the new node to improve the efficiency of cooperative path planning.
5. In the tree structure generation of the single-UAV cooperative path planner, a neighborhood node update method based on the time coordination cost function is proposed to guide the path generation meeting the time-space coordination constraints.

The rest of this paper is organized as follows: In Section 2, we introduce the related works; Section 3 describes the MUCPP problems based on the time-space coordination constraints; Section 4 introduces the HDP-TSRRT* algorithm in three aspects; Section 5 provides us with the simulation of the proposed algorithm and discusses the results; and Section 6 describes some conclusions and future work.

2. Related Works

This section provides a brief introduction to the classical PP algorithm and the classical RRT* algorithm.

2.1. Prioritized Planning Algorithm

PP is an algorithm that can quickly solve the cooperative path planning problems, which was first proposed in [26]. Algorithm 1 gives the framework of classical PP. The classical PP algorithm assigns different priorities to each robot. Then, the path planning of each robot is carried out from the highest priority 1 to the lowest priority n . During i th iteration, PP algorithm computes the path σ_i of the robot i that not only avoids the threats in the environment, but also the time-space occupied area set H_i occupied by robots

$1, \dots, i - 1$. The path of robot i is computed in function $\mathbf{A}^*(\mathcal{X}, H_i)$, which returns a path for robot i such that the robot avoids not only the threats in the environment, but also the paths that high-priority robots take. If the required path cannot be planned, the algorithm stops. Otherwise, the algorithm will continue to be executed until the robot with the lowest priority completes the planning.

Algorithm 1: Classical Prioritized Planning Algorithm

```

01 :  $\mathcal{X}$  : Environment;  $H_i \leftarrow \emptyset$ 
02 : for  $i = 1 \dots n$ 
03 :    $\sigma_i \leftarrow \mathbf{A}^*(\mathcal{X}, H_i)$ 
04 :   if  $\sigma_i = \emptyset$ 
05 :     report failure and terminate
06 :   end if
07 :    $H_i \leftarrow H_i \cup \sigma_i(t)$ 
08 : end for

```

In the classic PP, the A* algorithm [27] is used to complete the cooperative path planning of a single UAV. However, as a grid-based search algorithm, the planning time of A* usually increases exponentially with the increase in spatial dimensions and is rarely used in UAV path planning in large three-dimensional environments. Meanwhile, some local obstacle avoidance planning algorithms, such as artificial potential field (APF) [28] and interfered fluid dynamic system (IFDS [29]), may encounter the problem of local minima. Intelligent algorithms, such as genetic algorithm (GA) [30] and ant colony optimization (ACO) [31] algorithm, may only find sub-optimal solutions when there are complex or many constraints. In addition, the classic PP algorithm does not specify the approach of prioritization. However, the prioritization approach is one of the important factors affecting the quality and efficiency of the coordination path. Therefore, the design of prioritization is particularly important while improving the PP algorithm framework.

2.2. RRT* Algorithm

On the basis of the RRT algorithm [32], which can only guarantee the probability completeness, Karaman and Frazzoli proposed the RRT* algorithm [33], which introduced the neighborhood node selection mechanism, thus ensuring both the probability completeness and the asymptotic optimality. The framework of the goal-guided RRT* algorithm is shown in Algorithm 2. During i th iteration of tree growth, RRT* algorithm generates sampling node x_{rand} through the goal node bias strategy (i.e., the probability number P is randomly generated between 0 and 1. If $P > 0.1$, the x_{rand} is generated by random sampling; otherwise, the x_{rand} is set as the goal g_i). Then, the algorithm selects the node $x_{nearest}$ nearest to x_{rand} in the tree through function **Nearest**. After that, the new node x_{new} is obtained by extending a certain distance along the $x_{nearest} \rightarrow x_{rand}$ through the function **Steer**. If edge $(x_{new}, x_{nearest})$ meets the collision checking constraints (i.e., **obstacle_free**), the node x_{new} will be added to the tree $\mathcal{T}$. Then, the function **Near** x_{new} as the center and γ as the radius construct the neighborhood x_{new} , which selects the neighborhood node set $\mathbf{X}_{near}$. Then, the parent node $x_{new.parent}$ with the lowest cost of x_{new} is re-selected among all neighbor nodes x_{near} . Then, compare the cost between x_{new} as the potential parent node of all neighborhood nodes and the current parent node of neighborhood nodes. If the cost becomes low, the parent node of x_{near} is updated to x_{new} . The algorithm continues to iterate to expand the tree structure until a tree node falls into the goal region, and the path can be found retroactively.

Algorithm 2: RRT* Algorithm

```

01 : initialization Tree node :  $\mathcal{T}.V \leftarrow \{s_i\}$ ; Tree edge :  $\mathcal{T}.E \leftarrow \emptyset$ 
02 : for  $i = 1$  to  $n$ 
03 :   if  $P : \text{random}([0, 1]) < 0.1$ 
04 :      $x_{rand} \leftarrow g_i$ 
05 :   else
06 :      $x_{rand} \leftarrow \text{Sample}(\mathcal{X})$ 
07 :   end if
08 :    $x_{nearest} \leftarrow \text{Nearest}(x_{rand})$ 
09 :    $x_{new} \leftarrow \text{Steer}(x_{nearest})$ 
10 :   if  $\text{obstacle\_free}(x_{new}, x_{nearest})$ 
11 :      $\mathcal{T}.V \leftarrow \mathcal{T}.V \cup \{x_{new}\}$ 
12 :   end if
13 :    $X_{near} \leftarrow \text{Near}(x_{new}, \mathcal{T}.V, \gamma)$ 
14 :   for all  $x_{near} \in X_{near}$ 
15 :     if  $\text{obstacle\_free}(x_{near}, x_{new}) \wedge$ 
16 :        $\text{Cost}_{\mathcal{T}}(s_i, x_{near}) + \text{Cost}_{\mathcal{T}}(x_{near}, x_{new}) < \text{Cost}_{\mathcal{T}}(s_i, x_{new})$ 
17 :        $x_{new}.parent \leftarrow x_{near}$ 
18 :     end if
19 :   end for
20 :   for all  $x_{near} \in X_{near}$ 
21 :     if  $\text{obstacle\_free}(x_{new}, x_{near}) \wedge$ 
22 :        $\text{Cost}_{\mathcal{T}}(s_i, x_{new}) + \text{Cost}_{\mathcal{T}}(x_{new}, x_{near}) < \text{Cost}_{\mathcal{T}}(s_i, x_{near})$ 
23 :        $x_{near}.parent \leftarrow x_{new}$ 
24 :     end if
25 :   end for

```

The RRT* algorithm is widely used in the path planning of single UAVs due to its advantages of fast solutions in highly dimensional spaces. However, for the MUCPP problem, it is insufficient to rely only on RRT*, and the relevant research is relatively small. It is noted that the RRT* can introduce space cooperative constraints into the collision checking process, thus highly combining with the PP algorithm. Therefore, there are many possibilities and advantages to improve the RRT* to find the single-UAV cooperative path, which can meet the time–space cooperative path planning needs and effectively improve the efficiency of planning.

Therefore, this paper mainly studies the decoupling cooperative path planning approach based on PP and RRT*.

3. Problem Formulation

Consider the UAVs group $UAV_N : (UAV_1, \dots, UAV_n)$ composed of n UAVs flying in the planned space $\mathcal{X} \subseteq \mathbb{R}^3$ composed of a three-dimensional battlefield environment. In a cooperative flight, all UAVs in UAV_N take off at the same time and reach the goal at the same time. Any UAV_i needs to start flying from its start node s_i and avoid all threats, including radar, anti-aircraft gun, missile and terrain in flight (the model of the threats is described in our previous work [34]), as well as the remaining $UAV_{N/i} : UAV_N \setminus UAV_i$, and finally reach its goal node g_i .

Since the start node or goal node of multiple UAVs may overlap, we assume that when different UAVs are at the same start node or goal node, their positions do not coincide, and there will be no conflict. In addition, we assume that each UAV in the group has an independent path planner and wireless communication equipment. Meanwhile, we assume that the communication equipment is completely reliable and has no delay.

3.1. Cooperative Path Planning Constraints of UAVs

The related constraints of UAV cooperative path planning are mainly divided into three categories, namely UAV performance constraints, planning space constraints and multi-UAV cooperative constraints. Firstly, the performance constraints of UAV are defined as follows.

1. UAV performance constraints

Assume that the planned path σ_i of UAV_{*i*} consists of a set of path node $P : \{\langle p_1, \dots, p_k \rangle | p_k = (x_k, y_k, z_k)\}$. When planning the flight path of the UAV, it is necessary to take into account the maximum flight distance $L_{\max}$, the maximum steering angle $\phi_{\max}$ and the climbing/diving angle $\gamma_{\max}$, as well as the shortest flight distance before steering $l_{\min}$. Then, the performance constraints of UAV_{*i*} can be expressed as [35]:

$$\begin{cases} L_{UAV_i} = \sum_{i=0}^k \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \leq L_{\max} \\ |\phi_{UAV_i}| = \left| \frac{\omega_i^T \omega_{i+1}}{\|\omega_i\|_2 \|\omega_{i+1}\|_2} \right| \leq \phi_{\max} \\ |\gamma_{UAV_i}| = \left| \arcsin\left(\frac{z_i - z_{i-1}}{\|\omega_i\|_2}\right) \right| \leq \gamma_{\max} \\ l_{UAV_i} = \|p_{i+1} - p_i\|_2 \geq l_{\min} \end{cases} \quad (1)$$

where $\omega_i = [x_i - x_{i-1}, y_i - y_{i-1}]^T$.

2. Planning space constraints

The threat types set in the planning space include terrain threats, which is built by digital elevation model (DEM, i.e., the digital expression of terrain surface morphology), radar, anti-aircraft guns and missiles. Then, the constraints of the planning space on UAVs can be expressed as:

$$\begin{cases} z_i - z_{(x_i, y_i)_{DMG}} \geq h_{\min} \\ (p_i, p_{i+1}) \cap \mathcal{X}_{obs} = \emptyset \end{cases} \quad (2)$$

where $z_{(x_i, y_i)_{DMG}}$ represents the terrain threat level when the UAV is at p_i , and $\mathcal{X}_{obs}$ represents other threats.

3. Multi-UAV cooperative constraints

In this paper, we mainly study the MUCPP problem that simultaneously satisfies the time–space coordination constraints. We first describe the space coordination constraint.

If any two UAVs in UAV_{*N*} can maintain a certain safety distance $D_{\min}$ at any time in flight, then these two UAVs meet the space coordination constraint. Therefore, the space coordination constraint can be expressed as:

$$\|\sigma_i(t) - \sigma_j(t)\|_2 \geq D_{\min} \quad i, j = 1, 2, \dots, k, i \neq j \quad (3)$$

where $\sigma_i(t)$ and $\sigma_j(t)$, respectively, represent the positions of UAV_{*i*} and UAV_{*j*} at any time t . In this paper, we assume that all UAVs pass through the planning space at the same velocity. The path of UAV_{*i*} is discretized according to the time information to obtain a group of discrete path nodes. In order to determine the safe distance between UAVs more accurately, this paper sets the time interval between two adjacent discrete path nodes of UAV_{*i*} as 0.01s. The potential collision between UAV_{*i*} and UAV_{*j*} is detected by comparing the distance between two path nodes (i.e., $\|\sigma_i(t) - \sigma_j(t)\|_2$) and the minimum safety distance $D_{\min}$ at any same time t .

Meanwhile, in order to meet the mission requirements, UAVs need to take off at the same time and reach their respective goal at the same time to achieve time coordination. In this paper, we set all UAVs to fly at a constant speed and have the same speed range

$[\mathcal{V}_{\min}, \mathcal{V}_{\max}]$ —that is, the flight time of any UAV is $[t_{\min}^i = \sigma_i / \mathcal{V}_{\max}, t_{\max}^i = \sigma_i / \mathcal{V}_{\min}]$. The time coordination constraint that the UAV_N needs to meet is expressed as:

$$\bigcap_{i=1}^n [t_{\min}^i, t_{\max}^i] \neq \emptyset \quad (4)$$

3.2. Cooperative Path Planning Formulation

When dealing with the MUCPP problem, not only the performance constraints and planning space constraints of a single UAV should be met, but also the coordination constraints of all UAVs. Firstly, the description of satisfactory path problem of UAV_i is given.

Problem 1. (*Satisfying Path Planning of UAV_i*). In the planning space $\mathcal{X}$, for any UAV_i in the UAV_N, a satisfactory path σ_i is planned for the given start-goal set $[s_i, g_i]$ so that σ_i meets the constraint Equations (1) and (2).

If the satisfactory path σ_i and the satisfactory path σ_j of any UAV_j in UAV_{N/i} meet the space coordination constraint (i.e., Equation (3)), σ_i and σ_j are said to be *collision-free*. If σ_i and σ_j meet the time coordination constraint (i.e., Equation (4)), σ_i and σ_j are said to be *time-consistent*. Then, Problem 2 can be described as follows.

Problem 2. (*MUCPP Problem based on time–space coordination constraints*). Given the planning space $\mathcal{X}$, and the start-goal set $[s_1, g_1], \dots, [s_n, g_n]$ of UAV_N, find the satisfactory path $\sigma_1, \dots, \sigma_n$ of all UAVs, and the σ_i and σ_j is collision-free and time-consistent.

4. HDP-TSRRT* Algorithm

In this paper, the HDP-TSRRT* algorithm is proposed to quickly plan the path satisfying the time–space cooperative constraint of multi-UAV. As a decoupled path coordination algorithm, the algorithm realizes the path planning of a single UAV and the path coordination of multiple UAVs in a hierarchical manner. On the basis of finding the pre-planned path, the heuristic prioritization-based asynchronous decentralized prioritized planning algorithm (HDP) is used to allocate the cooperative path planning order of all UAVs. Then, the cooperative path planning of each UAV from high priority to low priority is completed by using the proposed TSRRT* algorithm so that the path can meet the time–space constraints at the same time. The HDP-TSRRT* algorithm will be introduced in detail through the overall framework of the HDP-TSRRT* algorithm, the HDP algorithm and the TSRRT* algorithm in the following subsections.

4.1. Overall Framework

The algorithm is mainly divided into three levels, i.e., the pre-planning level, the planning coordination level and the single-UAV cooperative path planning level. The pseudocode of HDP-TSRRT* is shown in Algorithm 3.

Algorithm 3: HDP-TSRRT* Algorithm

```

1 : initialization  $\mathcal{T}_i, \mathcal{T}^i \leftarrow \emptyset, \sigma_i, \sigma^i \leftarrow \emptyset, [s_1, g_1], \dots, [s_n, g_n], H_i \leftarrow \emptyset$ 
   / * pre-planning level * /
2 :  $\sigma_i \leftarrow \text{RRT} * (\mathcal{X}, \mathcal{T}_i, s_i, g_i)$  / * Synchronous execution of all UAVs * /
3 :  $\text{cost}_i \leftarrow \text{Length}(\sigma_i)$ 
4 : wait for all UAVs to finish planning their own  $\sigma_i$  and  $\text{cost}_i$ 
5 :  $\sigma \leftarrow (\sigma_1, \dots, \sigma_n)$ 
6 :  $\mathbf{t} \leftarrow (\text{cost}_1/v_1, \dots, \text{cost}_n/v_n)$ 
7 :  $t_{co} \leftarrow \max \mathbf{t}$ 
8 :  $\sigma^1 \leftarrow \arg \max(\text{Length}(\sigma)/v)$ 
   / * planning coordination level & single – UAV cooperative path planning level * /
9 :  $\langle \text{UAV}^1, \dots, \text{UAV}^N \rangle \leftarrow \text{Heu\_Prioritization}(\sigma)$ 
10 :  $H_i \leftarrow (H_i \cup (\text{UAV}^1, \sigma^1(t)))$ 
11 :  $\sigma^i \leftarrow \text{TSRRT} * (\mathcal{X} \setminus H_i, \mathcal{T}^i, s_i, g_i, t_{co})$  / * single – UAV cooperative
   / * path planning level * /
12 : communicate  $\text{Coor\_message}(\text{UAV}^i, \sigma^i(t))$ 
13 : wait for receiving  $\text{Coor\_message}$  of other UAVs
14 : while receiving  $\text{Coor\_message}(\text{UAV}^k, \sigma^k(t))$ 
15 :   if  $k < i$ 
16 :      $H_i \leftarrow (\{H_i \setminus (\text{UAV}^k, \sigma^k(t))\} \cup (\text{UAV}^k, \sigma^k(t)))$ 
17 :      $\sigma^{i*} \leftarrow \text{TSRRT} * (\mathcal{X} \setminus H_i, \mathcal{T}^{i*}, s_i, g_i, t_{co})$  / * single – UAV cooperative
   / * path planning level * /
18 :   if  $\sigma^{i*} \neq \sigma^i$ 
19 :      $\sigma^i \leftarrow \sigma^{i*}$ 
20 :   communicate  $\text{Coor\_message}(\text{UAV}^i, \sigma^i(t))$ 
21 :   else
22 :      $\sigma^i \leftarrow \sigma^i$ 
23 :     keep silent
24 :   end if
25 : end if
26 : if  $\sigma^1, \dots, \sigma^n$  are collision – free and time – consistent
27 :   terminate HDP-TSRRT*
28 : else
29 :   continue
30 : end if
31 : end while

```

The definitions of some symbols appearing in Algorithm 3 are as follows. $\mathcal{T}_i$ is the tree structure of any UAV_i . $\mathcal{T}^i$ is the tree structure of UAV^i with priority of i . σ_i is the current path of any UAV_i . σ^i is the current path of UAV^i with priority of i . H_i is the time–space occupied area set occupied by UAVs $1, \dots, i-1$. cost_i is the length of the path σ_i , which is calculated by function **Length**. σ and $\mathbf{t}$ are the path set and ETA set of UAVs, respectively. t_{co} the minimum ETA of the UAVs group. $\langle \text{UAV}^1, \dots, \text{UAV}^N \rangle$ represents the priority ranking set of all UAVs obtained through the function **Heu_Prioritization**(σ). The function **Coor_message** contains the path information of the UAV^k with priority k . *obstacle – free* and *time – consistent* are defined in Equations (3) and (4).

Figure 1 shows the overall framework of HDP-TSRRT * algorithm. As shown in Figure 1, the HDP-TSRRT * algorithm can be divided into three levels.

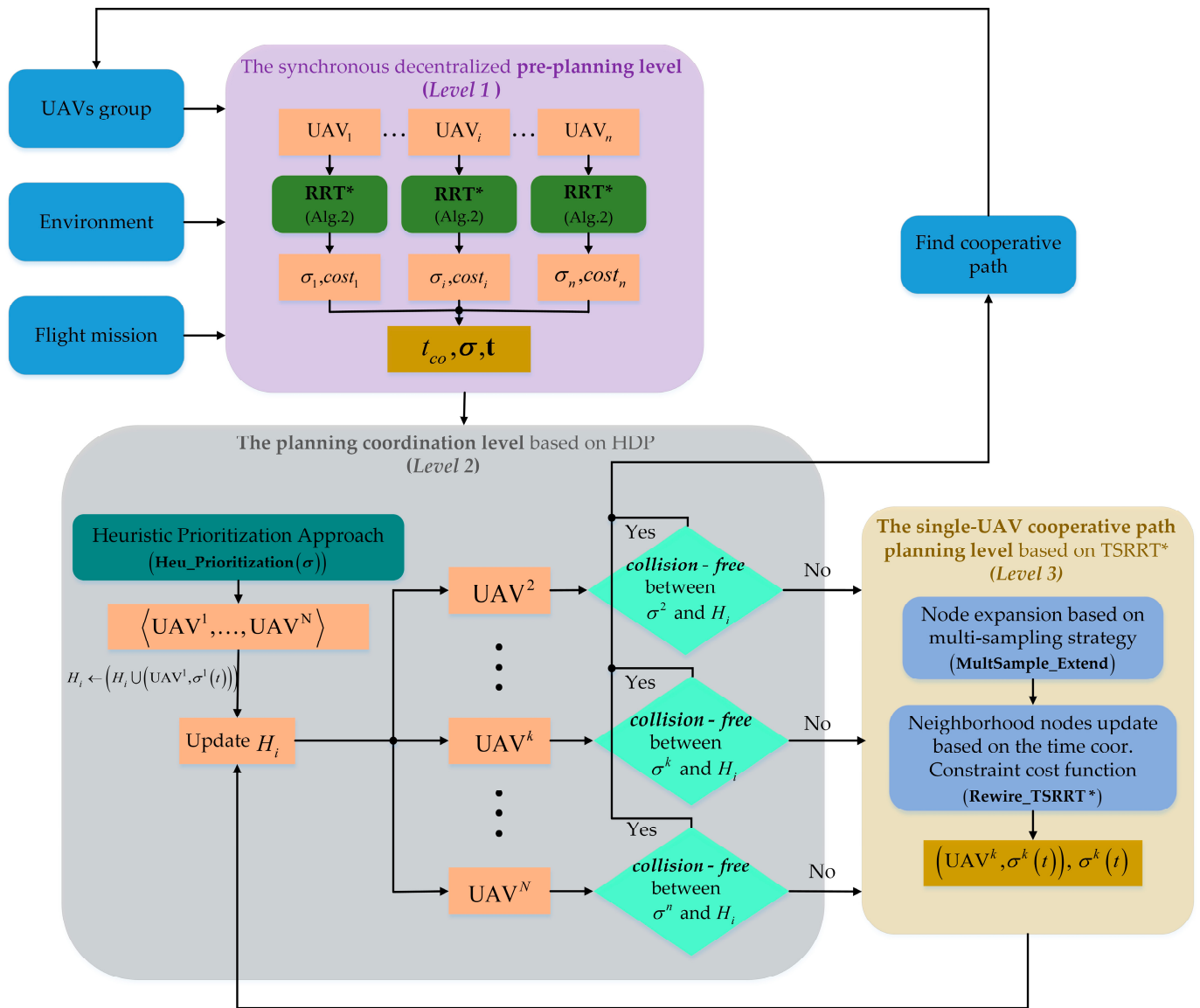


Figure 1. Overall framework of HDP-TSRRT* algorithm.

1. The pre-planning level

This level mainly includes the synchronous decentralized pre-planning process, which plans the initial optimal path for each UAV to obtain the shortest ETA and the initial parameters of the HDP algorithm. In this part, the UAVs do not need to consider the time–space coordination constraints when planning the path, and instead, only consider the performance constraints and planning space constraints of UAVs and use the RRT* algorithm (i.e., Algorithm 1) for optimal path planning. After each UAV obtains its optimal path, the path length is obtained through the $\text{Length}(\sigma_i)$ function, and then the flight time set $\mathbf{t} : (t_1, \dots, t_n)$ of all UAVs is obtained. It should be noted that all UAVs use the RRT* algorithm for optimal path planning, which means that all UAVs obtain the shortest flight time. However, for the MUCPP based on time–space coordination constraints, if the shortest flight time $t_{j,\min}$ of UAV_j is less than the shortest flight time $t_{i,\min}$ of the UAV_i , only by increasing the $t_{j,\min}$ to make it the same as the $t_{i,\min}$ can the two UAVs achieve time coordination. Therefore, it is necessary to take the maximum time in the flight time set $\mathbf{t}$ as the ETA, t_{co} , and set the UAV mapped by t_{co} as the UAV with the highest priority to avoid repeated planning of the same path.

2. The planning coordination level

This level mainly includes the coordination process of multi-UAV path planning priorities, which uses the proposed HDP algorithm to improve the efficiency of path coordination. First, the priority assignment process from 2 to n of all UAVs based on the heuristic prioritization approach is completed through the $\text{Heu_Prioritization}(\sigma)$ function. Since the cooperative constraints of UAVs are divided into two aspects, space and time, a priority performance evaluation function composed of potential collision times and cooperative time violations is designed in the algorithm. After $\text{Heu_Prioritization}(\sigma)$ outputs the priority order of all UAVs, the asynchronous decentralized prioritized planning is used to coordinate the generation of time-space cooperative paths. In the asynchronous decentralized prioritized planning, each low-priority UAV needs to avoid the time-space occupied area set $H_i = (\text{UAV}^i, \sigma^i(t))$ occupied by the high-priority UAV. In other words, the time-space area set of low-priority UAVs should contain high-priority paths to avoid collision with them.

3. The single-UAV cooperative path planning level

This level mainly includes the process of cooperative path planning for a single UAV, which is realized by the TSRRT* algorithm. Under the framework of the HDP algorithm, UAVs regard the time-space occupied area occupied by other UAVs as threats and use the TSRRT* to plan the path satisfying the time-space coordination constraint for each UAV. The TSRRT* algorithm evaluates the generated path through the cost function based on the time coordination constraint so that the planned path can meet the time coordination constraint on the basis of no collisions. In addition, we effectively reduce the computational loss through the proposed multiple sampling and cost function evaluation strategy.

4.2. HDP Algorithm

This section describes the HDP algorithm from two aspects: the heuristic prioritization approach and the framework of the HDP algorithm.

4.2.1. Heuristic Prioritization Approach

The priority level of a UAV affects the robustness and efficiency of cooperative path generation. When the prioritized planning algorithm is used for the path coordination of multiple UAVs, the inappropriate priority order will not only reduce the efficiency of cooperative path planning but also cause the failure of cooperative path planning. To solve this problem, the heuristic prioritization approach is proposed in this paper.

Since the algorithm proposed in this paper is mainly used for path planning of multiple UAVs that satisfy the time-space coordination constraints, the design of heuristic prioritization approach should take into account the impact of both time and space coordination constraints. For the time coordination constraint, if the pre-planned path of UAV_i is significantly different from the minimum ETA time to be met, it is obvious that the cooperative re-planning path of the UAV_i will also change more. Furthermore, because the re-planning path of the UAV_i changes greatly, the change of its space and time occupied area set will be greater, and the impact on other UAVs will be greater. Therefore, we need to set the priority of the UAV_i higher, so that the UAV_i can carry out cooperative re-planning before other UAVs to reduce the impact on other UAVs. For the space coordination constraints, which are similar to the time coordination constraints, if the number of potential collisions between the pre-planned paths of UAV_i and other UAVs is more, the path of the UAV_i will change more during the collaborative re-planning, which means that more other UAVs will be affected. Therefore, if the priority of the UAV_i is assigned higher, the less other UAVs will be affected by its re-planning. It can be seen from the analysis that the heuristic prioritization approach based on time-space coordination constraints can improve the planning efficiency.

In the heuristic prioritization approach, we designed a performance evaluation function consisting of the number of potential collisions and the amount of cooperative time

violations of the pre-planned path to assign reasonable priority to all UAVs. The evaluation function is defined as follows:

$$\begin{aligned} f_{hp}^{UAV_i} &= \lambda_{col} f_{col}(UAV_i) + \lambda_{time} f_{time}(UAV_i) \\ s.t. \lambda_{col} + \lambda_{time} &= 1 \end{aligned} \quad (5)$$

where $f_{col}(UAV_i)$ is the space cost function, representing the normalized potential collision number cost between UAV_i and other UAVs. $f_{time}(UAV_i)$ is the time cost function, i.e., the normalized violation of UAV_i and cooperative ETA t_{co} . The final priority performance score of UAV_i is $f_{hp}^{UAV_i}$. λ_{col} and λ_{time} are weights.

The purpose of the heuristic prioritization approach is to allocate the priority for each UAV reasonably, and the performance evaluation function designed will be used to evaluate the priority level. In addition, it is not difficult to find that the $f_{hp}^{UAV_i}$ of any UAV_i is positively correlated with the number of collisions and the cost of cooperation time violation. Therefore, we give the prioritization definition of UAVs based on performance evaluation function as follows.

Definition 1. (Prioritization evaluation based on performance evaluation function) Given a group of $UAV_N : (UAV_1, \dots, UAV_n)$ and any group of weights $[\lambda_l, \lambda_t] : \lambda_l + \lambda_t = 1$, the performance evaluation function set $\left\{ \bigcup_{i=1}^{N-1} f_{hp}^{UAV_i} \right\}$ of UAV_N is ranked from largest to smallest to generate the set $\left\{ \bigcup_{i=1}^{N-1} f_{hp}^{UAV_i} \right\}_{\substack{\rightarrow \\ max \rightarrow min}}$, whose elements map the priority of $\{1, \dots, N\}$ of UAV_N in turn.

First, we analyze the calculation process of $f_{col}(UAV_i)$. Based on the space coordination constraint (Equation (3)), the potential collision number $Col_{i,j}$ between the UAV_i and any other UAV_j based on the pre-planned path can be calculated as follows:

$$\begin{cases} Col_{i,j} \parallel \sigma_i(t) - \sigma_j(t) \|_2 < D_{min} \\ 0 \parallel \sigma_i(t) - \sigma_j(t) \|_2 \geq D_{min} \end{cases} \quad (6)$$

$s.t. j = 1, \dots, N \text{ and } i \neq j$

Obviously, the number of collisions of the same UAV is zero. In addition, if the number of collisions between UAV_i and UAV_j is $Col_{i,j}$, the number of collisions between UAV_j and UAV_i is $Col_{j,i}$, where $Col_{i,j} = Col_{j,i}$. From this, the total number of collisions of UAV_N can be determined as:

$$f_{col}^{all} = \frac{\sum_{i=1, j=1}^N Col_{i,j}}{2} \quad (7)$$

Then, the $f_{col}(UAV_i)$ of UAV_i can be expressed as:

$$f_{col}(UAV_i) = \frac{\sum_{j=1}^{N-1} Col_{i,j}}{f_{col}^{all}} \quad (8)$$

For the calculation of cooperative time violation cost $f_{time}(UAV_i)$ based on the time coordination constraint of UAV_i , the path length L_i^0 needs to be obtained through the pre-planned path, so as to calculate the ETA time $t_i^0 = L_i^0 / v_i$. Then, the $f_{time}(UAV_i)$ is:

$$f_{time}(UAV_i) = \frac{|t_{co} - t_i^0|}{t_{co}} \quad (9)$$

It should be noted that there may be a specific situation, i.e., when the performance evaluation function of some UAVs may be equal, in which the priority of UAVs with equal f_{hp} can be randomly assigned.

Then, the heuristic prioritization approach framework is shown in Algorithm 4.

Algorithm 4: Heuristic prioritization approach (i.e., **Heu_Prioritization**(σ))

```

01 :  $UAV_{N-1} : UAV_N \setminus UAV_m$ 
02 :  $f_{hp}^{UAV_{N-1}} \leftarrow \emptyset$ 
03 :  $f_{col}^{all} \leftarrow \frac{\sum_{i=1, j=1}^N Col_{i,j}}{2}$ 
04 : for  $i = 1$  to  $N - 1$ 
05 :    $f_{hp}^{UAV_i} \leftarrow \lambda_{col} \frac{\sum_{j=1}^{N-1} Col_{i,j}}{f_{col}^{all}} + \lambda_{time} \frac{|t_{co} - t_i^0|}{t_{co}}$ 
06 :    $f_{hp}^{UAV_{N-1}} \leftarrow f_{hp}^{UAV_{N-1}} \cup \{f_{hp}^{UAV_i}\}$ 
07 : end for
08 : for  $k = 2$  to  $N$ 
09 :    $UAV^k \leftarrow \arg \max f_{hp}^{UAV_{N-1}}$ 
10 :    $f_{hp}^{UAV_{N-1}} \leftarrow f_{hp}^{UAV_{N-1}} \setminus \max f_{hp}^{UAV_{N-1}}$ 
11 : end for

```

4.2.2. The Framework of HDP Algorithm

This paper proposes the HDP algorithm, which improves the ADRPP algorithm. The HDP algorithm introduces a heuristic prioritization approach based on time–space coordination constraints, which can reasonably prioritize UAVs, thus improving the efficiency and robustness of path coordination. In addition, in order to deal with the MUCPP problem based on time–space cooperative constraints, we propose the TSRRT* algorithm in the single-UAV cooperative path planner of the HDP algorithm, which will be described in Section 4.3.

The framework of the HDP algorithm is shown in Algorithm 3. First, the priority of the UAV is ranked by the heuristic prioritization approach (i.e., function **Heu_Prioritization**(σ)). Then, any UAV_i^k with a priority of k only needs to receive and process the path change message of the $UAV_j^{m < k}$ with higher priority and response. If receiving the path change message from the $UAV_j^{m < k}$, UAV_i^k will update the information of the time–space occupied area H_i to the current path information of $UAV_j^{m < k}$. After that, UAV_i^k checks whether the planned path based on the current H_i can meet the space–time coordination constraints. If not, UAV_i^k uses the TSRRT* algorithm to re-plan the path that meets the time–space coordination constraints in combination with the current H_i . Otherwise, UAV_i^k does not need to re-plan the path and keeps the communication silent. The algorithm stops until all UAVs plan paths that meet the time–space cooperation constraints.

4.3. TSRRT* Algorithm

As a decoupled cooperative path planning algorithm, the HDP-TSRRT* algorithm needs to carry out path planning for each UAV that meets the time–space cooperative constraints after using the HDP algorithm to coordinate the planning of multi-UAV. In addition, it is noted that when the HDP algorithm is used for planning coordination, many rounds of path re-planning may be required for a single UAV to ensure that all UAVs can meet the coordination constraints, which is a great challenge to the efficiency of the path planning algorithm for a single UAV. If the efficiency of path planning is low, it will greatly increase the planning time of the cooperative path.

Therefore, this paper proposes the TSRRT* algorithm as the single-UAV cooperative path planning algorithm. The TSRRT* algorithm will be introduced in four parts, i.e., the

cost function based on time–space coordination constraints, the tree expansion based on multi-sampling strategy, the neighborhood node optimization approach and the framework of TSRRT*.

4.3.1. Evaluation of the Cost Function Based on Time-Coordination Constraint

After obtaining the shortest cooperative ETA (i.e., t_{co}), we can obtain the path length L_{co}^i of any UAV_{*i*} meeting the time cooperative constraint according to t_{co} , where $L_{co}^i = t_{co}v_i$. Therefore, in the TSRRT* algorithm, we introduce a new cost function that can constrain the length of the planned path, i.e., the cost function based on the time coordination constraint, so as to ensure that the planned path meets the time coordination constraint.

The cost function based on the time coordination constraint is no longer aimed at minimizing the path length but at minimizing the difference between the actual path length and the constrained path length. The cost function is defined as follows:

$$f_{\mathcal{T}}^*(x) = |f_{\mathcal{T}}(x) - L_{co}| \quad (10)$$

where $f_{\mathcal{T}}(x) = g_{\mathcal{T}}(x) + \lambda_{h_D} h_D(x)$. $g_{\mathcal{T}}(x)$ represents the actual distance from the start node s_i to the current tree node x . $h_D(x)$ represents the estimated distance from the x to the goal node g_i , and λ_{h_D} is the estimated distance coefficient. This paper takes the Euclidean distance between two nodes as the estimated distance $h_D(x)$, where $h_D(x) = \|x - g_i\|_2$.

4.3.2. Node Expansion Based on Multi-Sampling Strategy

In the tree structure of the classical RRT* algorithm, the expansion of a new node is only completed through one sampling. However, due to the large flight environment of UAVs, even though the generation of sampling points is guided by some offset sampling methods [36–38], there is still great randomness in the expansion, which affects the efficiency of path planning. For the decoupled cooperative path planning algorithm, the path planning efficiency of a single UAV is crucial.

Therefore, in order to ensure the efficiency of planning, this paper proposes a multi-sampling strategy, which directly selects the goal as the sampling point during sampling or selects a batch of sampling points for pre-expansion so as to reduce the probability of adding low-quality sampling nodes to the tree for expansions to improve efficiency. Algorithm 5 shows the process of the new node expansion approach based on a multiple sampling strategy.

When $P \geq 0.1$, the multi-sampling process is executed. First, we select a group of $K \in \mathbb{R}_+$ sampling nodes from the planning space, and any potential sampling point is x_{rand}^m , where $m \in \{1, 2, \dots, K\}$. Then, we calculate the corresponding potential expansion nodes $\{x_{temp}^1, \dots, x_{temp}^K\}$ for all potential sampling points, and the calculation formula for any potential expansion node x_{rand}^m is as follows:

$$\begin{aligned} x_{temp}^m &= x_{nearest}^m + l_{extend} \frac{(x_{rand}^m - x_{nearest}^m)}{\|x_{rand}^m - x_{nearest}^m\|_2} \\ \text{s.t. Eq.(1), (2), (3)} \end{aligned} \quad (11)$$

where x_{temp}^m , x_{rand}^m , $x_{nearest}^m$ are UAV position coordinates. For potential expansion nodes that meet UAV performance constraints, planning space constraints and space coordination constraints, the expansion length of nodes is determined by the expansion step l_{extend} . Then, we use the sampling node x_{rand}^m and the nearest node $x_{nearest}^m$ from x_{rand}^m in the tree to determine the potential expansion node x_{temp}^m . If x_{temp}^m does not meet the above constraints, it will be discarded.

Algorithm 5: Node expansion based on multi-sampling strategy (**MultSample_Extend**)

```

01 :  $P = \text{random}([0, 1])$ 
02 : if  $P < 0.1$ 
03 :    $x_{rand} \leftarrow g_i$ 
04 :    $x_{nearest} \leftarrow \text{Nearest}(x_{rand})$ 
05 :    $x_{new} \leftarrow \text{Steer}(x_{nearest})$ 
06 :   if  $\text{obstacle\_free}(x_{new}, x_{nearest}) \wedge \text{UAV\_free}(x_{new}, x_{nearest})$ 
07 :      $\mathcal{T}.V \leftarrow \mathcal{T}.V \cup \{x_{new}\}$ 
08 :   end if
09 : else
10 :    $\mathbf{X}_{rand}^K \leftarrow \bigcup_{m=1}^K \text{Sample}(\mathcal{X})$ 
11 :   for  $m = 1$  to  $K$ 
12 :      $x_{nearest}^m \leftarrow \text{Nearest}(x_{rand}^m)$ 
13 :      $x_{temp}^m \leftarrow \text{Steer}(x_{nearest}^m)$ 
14 :     if  $\text{obstacle\_free}(x_{temp}^m, x_{nearest}^m) \wedge \text{UAV\_free}(x_{temp}^m, x_{nearest}^m)$ 
15 :        $f_{\mathcal{T}}^*(x_{temp}^m) \leftarrow |f_{\mathcal{T}}(x_{temp}^m) - L_{co}|$ 
16 :        $f_{\mathcal{T},all}^* \leftarrow f_{\mathcal{T},all}^* \cup f_{\mathcal{T}}^*(x_{temp}^m)$ 
17 :     end if
18 :   end for
19 :    $x_{new} \leftarrow \text{argmin } f_{\mathcal{T},all}^*$ 
20 : end if

```

After obtaining the potential expansion nodes set $\mathbf{X}_{temp}$ that meet the constraints, the cost function based on time constraints is used to select the optimal potential expansion node as the new node x_{new} for expansion, that is,

$$x_{new} = \text{argmin}_{f_{\mathcal{T}}^*}(\mathbf{X}_{temp}) \quad (12)$$

It can be seen that the new node expansion method based on multiple sampling and cost function evaluation strategy is divided into three steps. First, we select the expansion method of the new node—that is, we select the target point as the sampling point or conduct multi-sampling. If multi-sampling is selected, a batch of nodes are randomly sampled through multi-sampling strategy for pre-expansion to ensure the randomness of tree structure expansion. Then, all potential expansion nodes that meet the constraints are evaluated using the cost function based on the time coordination constraint to obtain the optimal expansion node as a new node. This method can reduce the randomness of the expansion of new nodes to a certain extent and ensure that the expansion of the tree structure can meet the solution of the time coordination constraint so as to improve the planning efficiency and make the planned path close to the optimal path under the current cost function. In addition, the evaluation process directly discards some sampling points with poor performance so that these nodes will not be subject to subsequent neighborhood node updates and other operations with high computational complexity, which can also reduce the planning time.

4.3.3. Neighborhood Nodes Update Approach Based on the Cost Function of Time Coordination Constraint

It should be noted that the key to ensuring the progressive optimality of the traditional RRT* algorithm is the introduction of neighborhood node optimization. Therefore, in the TSRRT* algorithm, in order to plan a path that meets the time–space coordination constraints, on the basis of obtaining the x_{new} of the new node, the cost function based on the time coordination constraint should also be used in the neighborhood node update to guide the parent node's updating and rewiring of the new node to ensure that the planned path gradually meets the time coordination constraint.

Because the neighborhood node update includes the parent node update process and the rewiring of the new node x_{new} , the two processes are basically the same. Therefore, Algorithm 6 gives a general framework for neighborhood node optimization of the TSRRT* algorithm.

Algorithm 6: Neighborhood node update (i.e., **Rewire_TSRRT*** (x_1, x_2))

```

01 : if obstacle_free( $x_1, x_2$ )  $\wedge$  UAV_free( $x_1, x_2$ )
02 :   cost  $\leftarrow \|x_1 - x_2\|_2$ 
03 :   if  $|f_{\mathcal{T}}(x_1) + \text{cost} + h_D(x_2) - L_{co}| < f_{\mathcal{T}}^*(x_2)$ 
04 :      $x_2.parent \leftarrow x_1$ 
05 :   end if
06 : end if

```

First, we analyze the process of performing Rewire_TSRRT* for the first time—that is, the process of selecting the parent node with less cost based on the time cooperation constraint in the neighborhood for x_{new} . Firstly, it is determined whether the edge (x_{near}^k, x_{new}) composed of any node x_{near}^k in the neighborhood, and x_{new} meets the UAV performance constraints, planning space constraints and space coordination constraints. If the constraints are met, the distance between two nodes is calculated. Note that all nodes in the neighborhood have been added to the tree, so we can directly obtain the actual path length from the start node s_i to x_{near}^k in the tree. Then, the estimated distance $h_D(x_2)$ from x_{new} to the goal node g_i is calculated. Then, the cost $f_{\mathcal{T}}^*(x_{new})$ based on time coordination constraint from s_i through x_{near}^k to x_{new} is

$$f_{\mathcal{T}}^*(x_{new}) = |f_{\mathcal{T}}(x_1) + \text{cost} + h_D(x_2) - L_{co}| \quad (13)$$

After calculating the cost of all neighborhood nodes, the node with the lowest cost is selected as the parent node of x_{new} .

The process of the second Rewire_TSRRT* is basically similar to that of the first. The only difference is that this operation takes x_{new} as the potential parent node of all nodes in the neighborhood to optimize the neighborhood nodes.

4.3.4. The Framework of TSRRT* Algorithm

This section mainly analyzes the overall framework of the TSRRT* algorithm, which is shown in Algorithm 7. After designing the cost function based on the time coordination constraint as the evaluation standard in TSRRT*, the multiple sampling and cost function evaluation strategy is designed in the tree node expansion process to improve the planning efficiency. Then, the cost function based on time coordination constraint is used to guide the updating and rewiring of nodes in the neighborhood.

When the new node x_{new} extends into the goal region, a path $\sigma_i(s_i, g_i)$ can be found. If the deviation between the path length $\text{Cost}_{\mathcal{T}}(s_i, g_i)$ of $\sigma_i(s_i, g_i)$ and the L_{co} is greater than or equal to the given tolerance ε , the tree structure is updated using the Algorithm 7, Line 3–12 to optimize the $\sigma_i(s_i, g_i)$. When $\text{Cost}_{\mathcal{T}}(s_i, g_i) - L_{co} < \varepsilon$, the path satisfying the time–space coordination constraints is output.

Algorithm 7: TSRRT* algorithm

```

01 : initialization  $\mathcal{T}_i.V \leftarrow \{s_i\}; \mathcal{T}_i.E \leftarrow \emptyset$ 
02 :  $L_{co} \leftarrow t_{co}v_i$ 
03 : for  $i = 1$  to  $n$ 
04 :    $x_{new} \leftarrow \text{MultSample\_Extend}(\mathcal{T}_i)$ 
05 :    $\mathcal{T}_i.V \leftarrow \mathcal{T}_i.V \cup \{x_{new}\}$ 
06 :    $\mathbf{X}_{near} \leftarrow \text{Near}(x_{new}, \mathcal{T}_i.V, \gamma)$ 
07 :   for all  $x_{near} \in \mathbf{X}_{near}$ 
08 :     Rewire_TSRRT* $(x_{near}, x_{new})$ 
09 :   end for
10 :   for all  $x_{near} \in \mathbf{X}_{near}$ 
11 :     Rewire_TSRRT* $(x_{new}, x_{near})$ 
12 :   end for
13 :   if  $x_{new} \in \mathbf{G}_i$ 
14 :      $g_i \leftarrow x_{new}$ 
15 :   end if
16 : end for
17 : while  $|\text{Cost}_{\mathcal{T}}(s_i, g_i) - L_{co}| \geq \varepsilon$ 
18 :   Update  $\mathcal{T}_i.x_{new}$  with Line 03 – 12
19 : end while
20 : return  $\sigma_i(s_i, g_i)$ 

```

5. Simulation Results and Discussion

In this section, we show the simulation results of the HDP-TSRRT* algorithm and compare them with those of other algorithms to verify the feasibility, robustness and efficiency of the proposed algorithm in solving the MUCPP problem based on the time–space cooperative constraints.

5.1. Environments and Parameters

Due to the diversity of tasks performed by multi-UAV systems, the cooperative path planning problem of multiple UAVs has many forms. In order to comprehensively demonstrate the effectiveness and performance of the HDP-TSRRT* algorithm in dealing with MUCPP problems based on the time–space cooperative constraints, this paper designs a flight environment simulating real scenarios based on two different task categories to simulate and test the HDP-TSRRT* algorithm, i.e., rendezvous tasks [39] and allocation tasks [40].

For each planning example, we assume that the UAV can use its own independent CPU to calculate its path so as to achieve decentralized planning. In order to count and analyze the performance of HDP-TSRRT*, we use discrete events to simulate the parallel operation of the algorithm, including the simulation of cooperative message execution time and algorithm planning time between unmanned aerial vehicles to ensure that the process of executing the algorithm with the respective CPU of the UAV_N is consistent. Each UAV adopts a simulated idealized communication module, which can rely on zero-delay communication to achieve the sending and receiving of cooperative messages. The paths of synchronous decentralized pre-planning and the paths of single-UAV cooperative planning are the optimal paths. In addition, UAVs in all planning cases take off at the same time.

The following shows the simulation environment designed based on two types of tasks and the setting of some basic parameters.

1. Rendezvous tasks

The goal of MUCPP in this task is to find the path from different start nodes to the goal for each UAV in the planning space. Additionally, all UAVs must reach the same target position at the same time to perform specific tasks. Figure 2a shows the planning space based on rendezvous tasks, including the real terrain (DEM map), radar, missile,

anti-aircraft gun and conical no-flight tower. See Table 1 for the relevant parameters of threats.

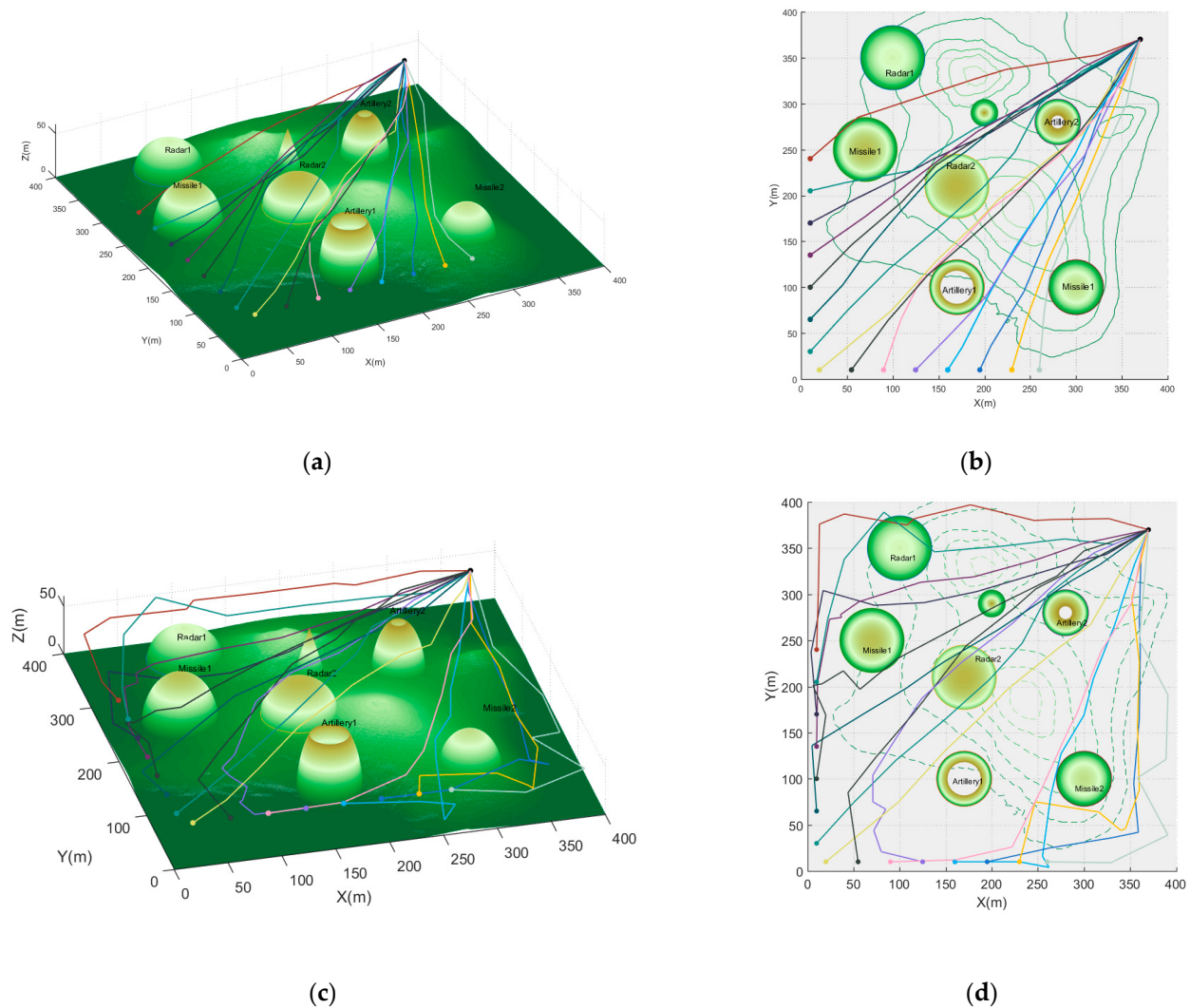


Figure 2. Rendezvous task scenario. The black dot indicates the target location of all UAVs. (a) The flight environment of rendezvous tasks, and the optimal path of 15 UAVs through RRT* synchronous decentralized pre-planning. (b) The top-down view of (a). (c) Fifteen UAVs use the HDP-TSRRT* algorithm to obtain the path that meets the time–space coordination constraints. (d) The top-down view of (c).

Table 1. Rendezvous task scenario.

Rendezvous tasks Scenario	Radar	Center 1:(100,350,5) m 2:(170,210,20) m	radius 35 m 35 m	
	Missile	Center 1:(70,250,0) m 2:(300,100,20) m	Radius 30 m 30 m	Height 40 m 40 m
	Anti-aircraft gun	Center 1:(170,100,0) m 2:(280,280,20) m	Radius 30 m 25 m	Height 40 m 40 m
	No-fly Tower	Center 1:(170,100,0) m	Radius 15 m	Height 50 m

2. Allocation tasks

The goal of cooperative path planning of multiple UAVs in this mission is to plan the flight path from different starting positions to different target positions for each UAV in the flight environment. In addition, in order to meet the requirements of some tasks, such as a multi-target cooperative strike, all UAVs should be able to reach the target point at the same time. Figure 5a shows the planning space based on allocation tasks, including the real terrain (DEM map), radar, anti-aircraft missile, anti-aircraft gun and conical no-flight tower. See Table 2 for the relevant parameters of threats.

Table 2. Allocation task scenario.

Allocation tasks Scenario	Radar	Center 1:(360,230,5) m 2:(190,210,20) m	radius 35 m 35 m	
	Missile	Center 1:(90,290,0) m 2:(300,100,20) m	Radius 30 m 30 m	Height 40 m 40 m
	Anti-aircraft gun	Center 1:(90,110,0) m 2:(280,280,20) m	Radius 30 m 25 m	Height 40 m 40 m
	No-fly Tower	Center 1:(200,290,10) m	Radius 15 m	Height 50 m

Some common basic parameters of the HDP-TSRRT* algorithm in two kinds of task scenario and comparative simulation are shown in Table 3.

Table 3. Basic parameters of TSRRT*.

Parameter	Value
$[\lambda_{col}, \lambda_{time}]$	[0.4, 0.6]
D_{min}	2 m
v for all UAVs	8 m/s
λ_{hD}	1
K	16
ϵ	0.35 s
Node expansion length	30 m

5.2. Simulation of HDP-TSRRT* Algorithm in Different Categories of Task Scenarios

This section conducts benchmark simulation experiments on HDP-TSRRT* in a rendezvous task scenario and allocation task scenario, respectively, and shows the process of using HDP-TSRRT* to achieve time-space collaborative constraint paths in both scenarios. The simulation results show that HDP-TSRRT* can plan the cooperative path of multi-UAV satisfying the time-space cooperative constraints in two scenarios.

5.2.1. Rendezvous Task Scenario

In the rendezvous task scenario, we show the cooperative planning process of UAV_N, where $N = 15$. The pre-path obtained through the synchronous decentralized pre-planning process of the HDP-TSRRT* algorithm is shown in Figure 2a,b. Note that the purpose of pre-planning is to obtain the number of potential collisions and the shortest cooperative ETA so as to lay the foundation for the subsequent priority allocation and coordination path planning. The pre-planned path lengths of the 15 UAVs are 510.1156, 483.2191, 466.5199, 440.6697, 418.8264, 404.5157, 387.4666, 382.2667, 504.2721, 480.8086, 454.6412, 431.5821, 413.4772, 403.0816 and 391.2380 m. The number of potential collisions are 3, 2, 1, 2, 2, 0, 1, 1, 0, 2, 2, 2, 0, 1, 1 and 1. The shortest cooperative ETA in this instance is $t_{co} = 63.89$ s through the pre-planned path length set.

Figure 2c,d show the time–space cooperative path map finally obtained by UAVs using HDP-TSRRT* in this scenario. It is difficult to determine whether the path of each UAV meets the time–space coordination constraint only through Figure 2c,d. Therefore, Figure 3 shows the shortest paired distance $l_{co}^{\min}$ of the cooperative path of 15 UAVs at any time. The $l_{co}^{\min}$ of any t is:

$$l_{co}^{\min} = \min \left(\bigcup_{\substack{i,j=1 \\ i \neq j}}^N \|\sigma_i(t) - \sigma_j(t)\|_2 \right) \quad (14)$$

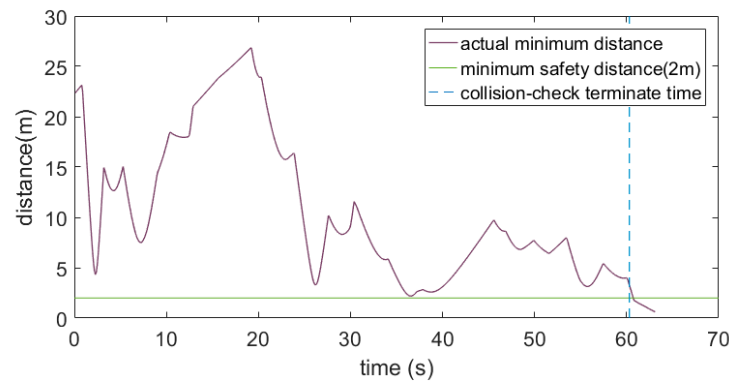


Figure 3. Minimum distance between any pair of UAVs. Note that the experimental scenario is a rendezvous task scenario, so collision checking ends when it reaches the path nodes before the target node (i.e., assuming there are no collisions in the target region).

Since the scenario of this planning example is a rendezvous task scenario, we assume that all UAVs will not collide at the target location. It can be seen in Figure 3 that at every moment, the actual shortest distance between any two UAVs is greater than the minimum safe distance $D_{\min}$, so it can be proved that all cooperative paths meet the spatial cooperative constraint. Figure 4 shows the collaboration time tolerance Δt_i of the coordination path. The coordination time tolerance of any UAV_{*i*} is,

$$\Delta t_i = \frac{L_i - L_{co}}{v_i} \quad (15)$$

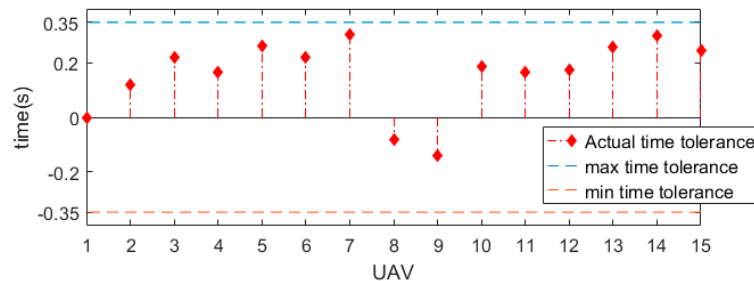


Figure 4. Time tolerance of 15 UAVs.

As can be seen from Figure 4, the time tolerance of all cooperative paths is $\Delta t_i < \varepsilon = 0.35$, which means that all paths meet the time cooperative constraint. Therefore, the simulation shows that HDP-TSRRT* can plan the cooperative path that meets the time–space cooperative constraints in the rendezvous tasks scenario.

5.2.2. Allocation Tasks Scenario

In the allocation task scenario, we show the cooperative planning process of UAV_N , where $N = 15$. The pre-planning path obtained through the synchronous decentralized planning layer of the HDP-TSRRT* algorithm is shown in Figure 5a,b. In order to better demonstrate the performance of the HDP-TSRRT* algorithm, in this planning example, we will divide 15 UAVs into 3 groups, with 5 in each group. The three groups of UAVs cross fly. As with the rendezvous task scenario, the pre-planned optimal path is obtained by the RRT* algorithm. The shortest cooperative ETA in this instance is $t_{co} = 56.21s$ through the pre-planned path length set. Figure 5c,d show the time-space cooperative path finally obtained UAV_N using HDP-TSRRT*. Figures 6 and 7, respectively, show the shortest paired distance of UAV_N at any time and the time tolerance in the allocation tasks scenario. It can be seen that the finally found path meets the time-space coordination constraints. Therefore, the simulation shows that HDP-TSRRT* can plan the cooperative path that meets the time-space coordination constraints in the allocation task scenario.

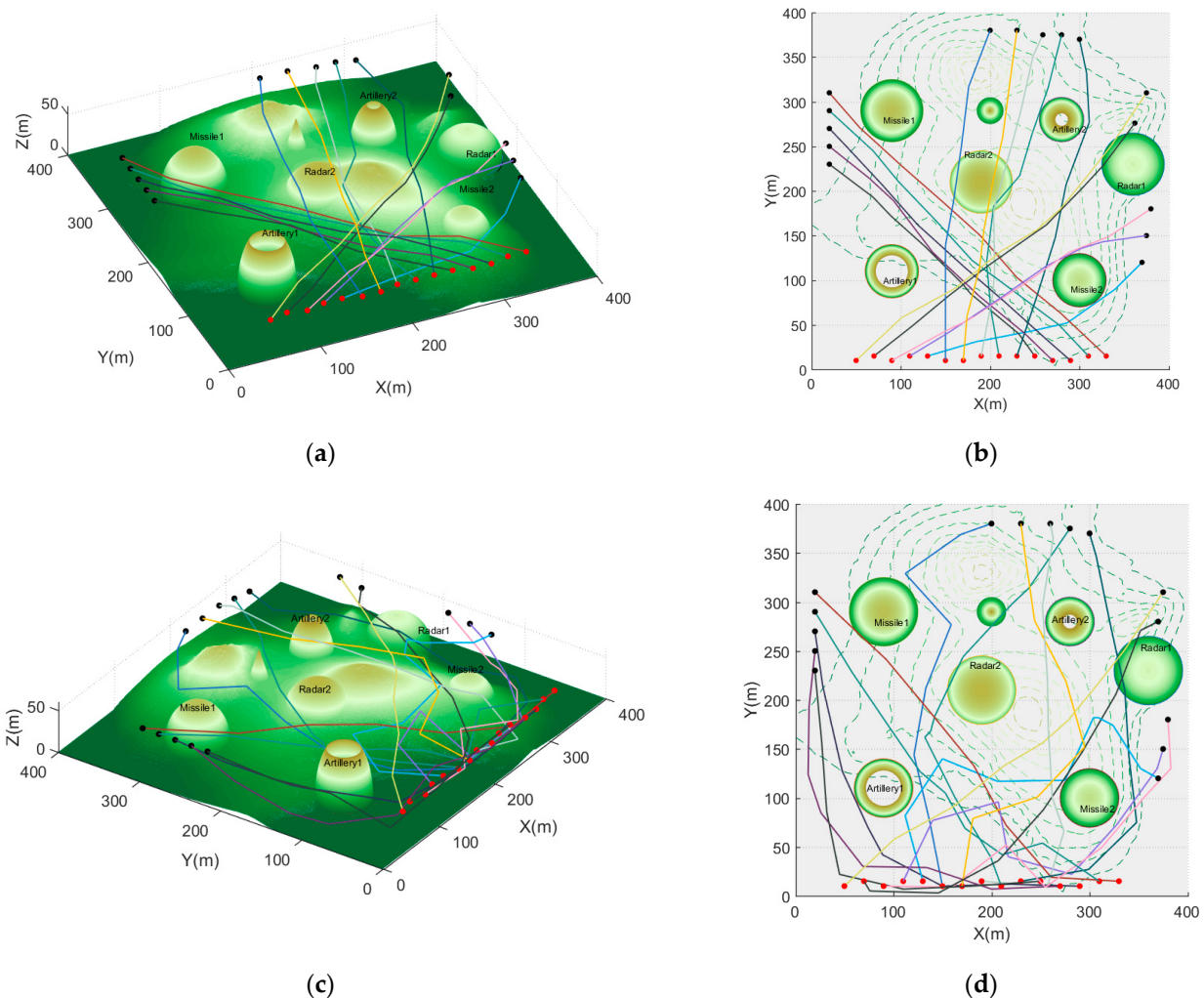


Figure 5. Allocation task scenario. The red dot represents the starting point of the UAV, and the black dot represents the ending point of the UAV. (a) The flight environment of allocation tasks, and the optimal path of 15 UAVs through RRT* synchronous decentralized pre-planning. (b) The top-down view of (a). (c) Fifteen UAVs use the HDP-TSRRT* algorithm to obtain the path that meets the time-space coordination constraints. (d) The top-down view of (c).

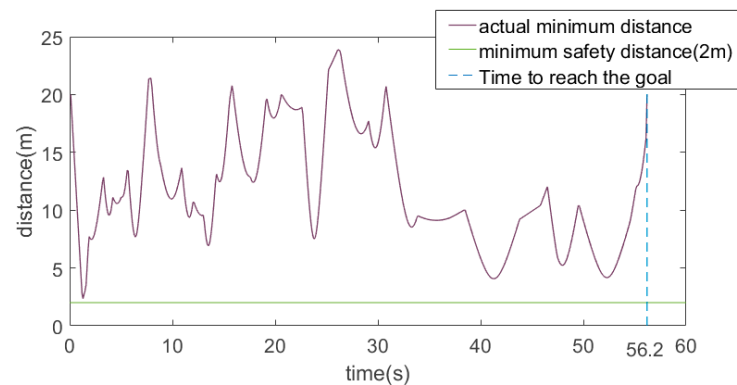


Figure 6. Minimum distance between any pair of UAVs in allocation tasks scenario.

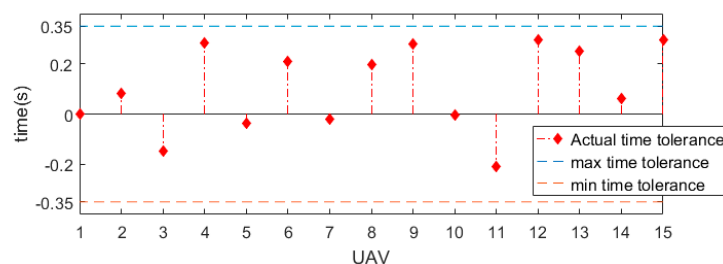


Figure 7. Time tolerance of 15 UAVs in the allocation task scenario.

5.3. Comparison and Algorithm Performance Analysis

In this section, we compare the planning coordination level and the single-UAV cooperative path planning level of HDP-TSRRT* with some existing algorithms and analyze the performance of HDP-TSRRT*.

5.3.1. Comparison and Performance Analysis of Planning Coordination Level of HDP-TSRRT*

Note that the existing asynchronous prioritized planning algorithms and their derivative algorithms do not consider both time and space coordination constraints. Therefore, in order to eliminate the impact of the single-UAV cooperative path planning algorithm and accurately verify the performance of the HDP-TSRRT* algorithm, we uniformly take TSRRT* as the single-UAV cooperative path planning algorithm and compare HDP-TSRRT* with the asynchronous decentralized prioritized planning algorithm based on three different prioritization approaches. These three algorithms are the asynchronous decentralized prioritized planning algorithm based on random prioritization (ADRPP) [23], asynchronous decentralized prioritized planning algorithm based on distance heuristic prioritization (DH-ADRPP) [26] and asynchronous decentralized priority planning algorithm based on collision heuristic prioritization (CH-ADRPP) [41].

This simulation compares the two types of task scenarios mentioned in Section 5.1 at the planning coordination level. In order to fully analyze the performance of the HDP-TSRRT* algorithm, in the rendezvous tasks scenario, the HDP-TSRRT* algorithm and the other three algorithms are, respectively, used to perform cooperative path planning in ten groups of different numbers of UAVs, including $N_{RT} = 3, 6, 9, 12, 15, 18, 21, 24, 27, 30$. In the allocation tasks scenario, ten groups of UAVs with a given number are also compared, including $N_{AT} = 4, 8, 12, 16, 20, 24, 28, 32, 36, 40$. Each group of UAVs with different numbers in the 2 kinds of mission scenarios are simulated 30 times.

We compared the following performance of HDP-TSRRT* with other three algorithms.

1. The average running time of the successful termination of the algorithm, which evaluates the efficiency and stability of the algorithm.

2. The average acceleration ratio of the four algorithms relative to the RPP algorithm, which evaluates the asynchronous execution effect of the algorithm using different prioritization approaches. The calculation formula of the acceleration ratio is $\frac{t_i^{RPP}}{t_i^{AD_ALG}}$, where t_i^{RPP} represents the running time of the RPP algorithm, and $t_i^{AD_ALG}$ represents the running time of the four comparison algorithms.
3. The average number of *Coor_message* sent by the UAV. Whenever the UAV running the algorithm re-plans the path, it will send new path information to other UAVs. Therefore, the number of *Coor_message* transmissions directly corresponds to the number of times the UAV is re-planned.
4. Success rate, which verifies the effectiveness of the algorithm.

We can see from Figure 8 that HDP-TSRRT* takes the shortest amount of time to plan the cooperative path in all different UAV groups in the two kinds of task scenarios. In addition, compared with the other three algorithms, HDP-TSRRT* has the highest acceleration ratio. This shows that the heuristic prioritization algorithm based on the time–space coordination constraint of HDP-TSRRT* is more reasonable, efficient and robust than the heuristic prioritization algorithm that only considers the length or the number of collisions and the random prioritization algorithm. Therefore, it can be proved that HDP-TSRRT* has the best planning efficiency and stability when dealing with the MUCPP problem based on the time–space coordination constraint.

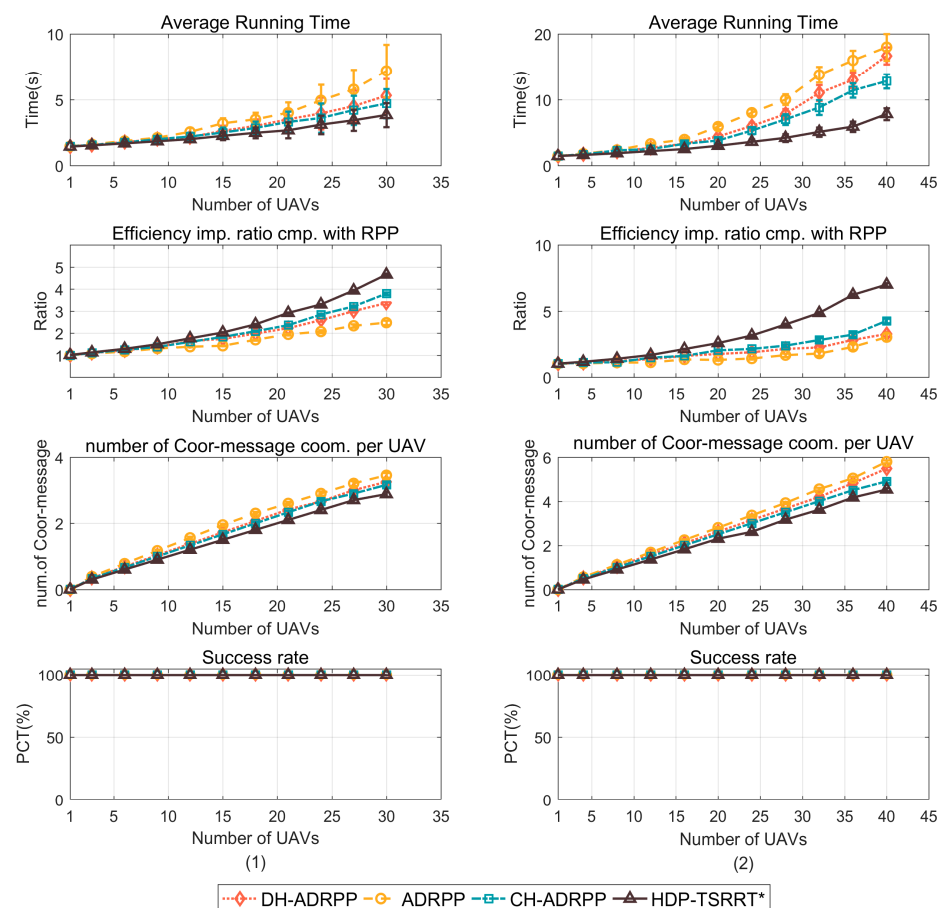


Figure 8. Comparison results of the planning coordination level. (1) Rendezvous tasks. (2) Allocation tasks.

Furthermore, it can be seen that the average number of *Coor_message* of each UAV in HDP-TSRRT* is also the lowest. That is to say, HDP-TSRRT* has the least path re-planning, so it can complete the cooperative path planning of all UAVs in the shortest time, and the planning efficiency is higher. Note that the four comparison algorithms are only different

in the prioritization algorithm, so it also proves that the prioritization approach of HDP-TSRRT* is more reasonable and efficient than other algorithms. As the four algorithms are asynchronous decentralized improved algorithms based on RPP algorithm, the success rate is basically the same.

5.3.2. Comparison and Performance Analysis of Single-UAV Cooperative Path Planning Level

The single-UAV cooperative path planning algorithm TSRRT* uses multiple sampling strategies to complete a batch of K pre-sampling nodes in sampling and new node expansion and optimizes the selection of sampling nodes in combination with the time-space coordination cost function. In this process, the selection of the pre-defined parameter K is crucial. If K is too small, the number of samples that can be evaluated by the time-space cost function is also very small, and there is a certain probability that the quality of this batch of samples is very low. On the other hand, if the K is too large, the cost of evaluating sampling nodes and the cost of the computing time of collision checking will also increase correspondingly. Therefore, whether the value of K is reasonable directly determines whether the TSRRT* algorithm can obtain better expansion nodes to complete the cooperative path planning faster, and then affects the efficiency of the HDP-TSRRT* algorithm to complete the cooperative path planning.

Therefore, we first simulated and evaluated the planning time of the TSRRT* algorithm with different values of K . In the experiment, 60 simulations were performed when $K = 1, 4, 8, 12, 16, 20, 24, 28$, including 30 runs for the rendezvous tasks scenario and 30 runs for the allocation tasks scenario. Figure 9 shows the relationship between the value of K and the running time. It can be seen that when K is between 16 and 20, the TSRRT* algorithm has the highest planning efficiency and the most stable robustness, and the planning time is basically the same. Therefore, we choose $K = 16$ to execute HDP-TSRRT*.

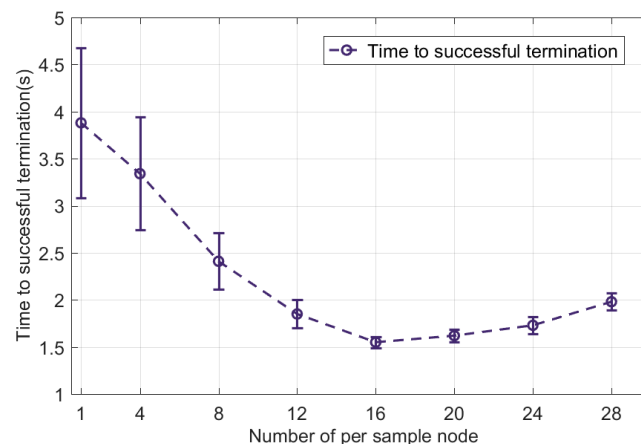


Figure 9. Simulation results under different numbers of per sample nodes.

When introducing the proposed algorithm, this paper theoretically demonstrates that the new node expansion strategy combined with the multi-sampling approach and time-space cost function optimization can improve the efficiency of cooperative path planning. In this section, we prove this view by comparing simulation results.

Because TSRRT* introduces a time-space-based cost function to evaluate the expansion of new nodes and the optimization of neighborhood nodes, in order to compare and discuss the performance of TSRRT* more accurately, when comparing TSRRT* with the classical RRT* algorithm (i.e., Algorithm 1) based on random sampling, this paper unifies the cost function of the two algorithms into the time-space-constrained cost function. The maximum planning time of the simulation is 10s, and the optimal path cost of TSRRT* is taken as the baseline.

Figure 10 shows the change in path cost based on multi-sampling and the time-space cost function of TSRRT* and the number of nodes in the tree structure, as well as the number of nodes in random sampling and the tree structure of the RRT* algorithm. As shown in Figure 10, using the multiple sampling and optimization strategy proposed in this paper, TSRRT* can find the first feasible path within a few milliseconds, and the cost will be reduced rapidly after that. In contrast, the RRT* algorithm takes nearly 1000 ms to find the first feasible path, and it takes longer to approach the optimal path. As we selected a reasonable number of multiple samples, the algorithm in this paper generates higher quality samples. Therefore, the algorithm in this paper is more likely to obtain the optimal solution with higher efficiency and faster convergence speed. In addition, from the distribution of the number of nodes in the tree structure, it can be seen that the quality of the tree structure generated by the algorithm in this paper is higher than that of RRT*, and the optimal path can be obtained faster in the generation of fewer tree nodes. This also proves that TSRRT* has advantages in planning efficiency.

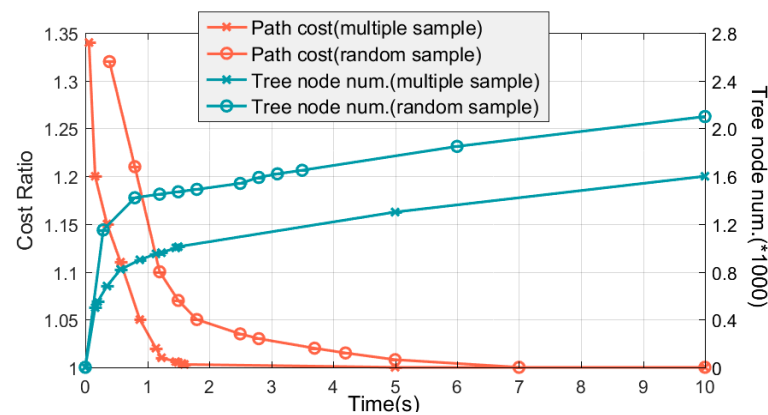


Figure 10. Comparison of multiple sampling strategy and random sampling method.

Furthermore, we conducted 60 cooperative path planning simulations involving two types of task scenarios (every scenario runs 30 trials) using TSRRT*, RRT* (in the cooperative path planning) and PRRT* [42] (in the cooperative path planning) algorithms based on the time-space coordination. The simulation results are shown in Figure 11. It can be seen that TSRRT* has the shortest and most stable planning time to achieve the optimal time-space path.

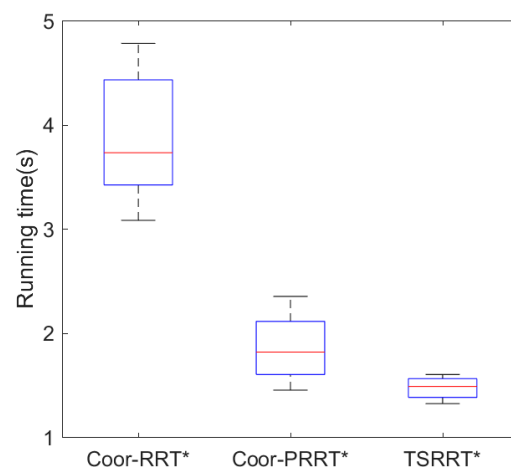


Figure 11. Comparative simulation of running time.

Therefore, through comparative simulations, we have proved that TSRRT* can plan the cooperation path that meets the time–space cooperation constraints, the planning efficiency is high, and the stability is good.

6. Conclusions

In this paper, a decoupling cooperative path planning algorithm named HDP-TSRRT* is proposed to solve the MUCPP based on the time–space cooperative constraints. Each UAV shares the same arrival time, and there is no conflict or threat avoidance among them. The proposed algorithm adopts a hierarchical architecture. The first is the pre-planning level, which aims to obtain the relevant parameter information and lay the foundation of cooperative path planning. Then, at the coordination planning level, an HDP algorithm is proposed, which introduces the heuristic prioritization approach based on time–space coordination constraints so as to assign reasonable priority to each UAV to improve the planning efficiency and robustness. Lastly, at the single-UAV cooperative path planning level, we propose the TSRRT* algorithm, which can quickly and stably plan a path that meets the time–space cooperative constraints for a single UAV.

In future work, we plan to expand the HDP-TSRRT* algorithm to the multi-UAV system with local communication to complete the real outdoor experiment. In real experiments, the proposed algorithm requires each UAV to have a separate communication unit to achieve real-time information reception and broadcast. However, ensuring the reliability and real-time of communication between UAVs is also a difficult problem to be faced in the implementation of this algorithm. In addition, we plan to use the decoupling prioritized planning algorithm or its derivative algorithm to deal with the MUCPP problem in a dynamic environment.

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 number No. 62073266, the Aeronautical Science Foundation of China, grant number 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.

Nomenclature

The nomenclature of all acronyms in the paper is shown in the following:

Acronyms	Definition
UAVs	Unmanned aerial vehicles
Multi-UAVs	Multiple unmanned aerial vehicles
MUCPP	Multi-UAV cooperative path planning
RRT*	Rapidly exploring random trees
HDP	Heuristic decentralized prioritized planning algorithm
TSRRT*	the time–space cooperative constraints-based RRT*
ETA	Estimated time of arrival
DRCA	Distributed reactive collision avoidance
ORCA	Optimal reciprocal collision avoidance
CSP	Constraint satisfaction problem

LP	Linear programming
ICTS	Incremental cost tree search
CBS	Conflict-based search
HCA*	Hierarchical cooperative A* algorithm
PP	Prioritized planning algorithm
ADRPP	Asynchronous decentralized prioritized planning
DEM	Digital elevation model map
DH-ADRP	Asynchronous decentralized prioritized planning algorithm based on distance heuristic prioritization
CH-ADRPP	Asynchronous decentralized priority planning algorithm based on collision heuristic prioritization
PRRT*	Potential functions based RRT*

References

- Madridano, Á.; Al-Kaff, A.; Martín, D.; de la Escalera, A. Trajectory planning for multi-robot systems: Methods and applications. *Expert Syst. Appl.* **2021**, *173*, 114660. [\[CrossRef\]](#)
- Shang, Z.; Bradley, J.; Shen, Z. A co-optimal coverage path planning method for aerial scanning of complex structures. *Expert Syst. Appl.* **2020**, *158*, 113535. [\[CrossRef\]](#)
- Wang, H.; Chen, W. Multi-Robot Path Planning with Due Times. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4829–4836. [\[CrossRef\]](#)
- Wang, Z.; Liu, L.; Long, T.; Wen, Y. Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding. *Chin. J. Aeronaut.* **2018**, *31*, 339–350. [\[CrossRef\]](#)
- Best, A.; Narang, S.; Manocha, D. Real-time reciprocal collision avoidance with elliptical agents. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016.
- Van Den Berg, J.; Guy, S.J.; Lin, M.; Manocha, D. Reciprocal n-body collision avoidance. In *Robotics Research*; Springer Tracts in Advanced Robotics; Springer: Cham, Switzerland, 2011.
- D’Amato, E.; Mattei, M.; Notaro, I. Distributed Reactive Model Predictive Control for Collision Avoidance of Unmanned Aerial Vehicles in Civil Airspace. *J. Intell. Robot. Syst. Theory Appl.* **2020**, *97*, 185–203. [\[CrossRef\]](#)
- Damani, M.; Luo, Z.; Wenzel, E.; Sartoretti, G. PRIMAL2: Pathfinding Via Reinforcement and Imitation Multi-Agent Learning-Lifelong. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2666–2673. [\[CrossRef\]](#)
- Kala, R. Multi-robot path planning using co-evolutionary genetic programming. *Expert Syst. Appl.* **2012**, *39*, 3817–3831. [\[CrossRef\]](#)
- Zu, C.; Yang, C.; Wang, J.; Gao, W.; Cao, D.; Wang, F.Y. Simulation and field testing of multiple vehicles collision avoidance algorithms. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 1045–1063. [\[CrossRef\]](#)
- Liu, Z.; Wei, H.; Wang, H.; Li, H.; Wang, H. Integrated Task Allocation and Path Coordination for Large-Scale Robot Networks With Uncertainties. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 2750–2761. [\[CrossRef\]](#)
- Panescu, D.; Pascal, C. A constraint satisfaction approach for planning of multi-robot systems. In Proceedings of the 2014 18th International Conference on System Theory, Control and Computing, ICSTCC 2014, Sinaia, Romania, 17–19 October 2014.
- Lee, H.; Motes, J.; Morales, M.; Amato, N. Parallel Hierarchical Composition Conflict-Based Search for Optimal Multi-Agent Pathfinding. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7001–7008. [\[CrossRef\]](#)
- Sharon, G.; Stern, R.; Goldenberg, M.; Felner, A. The increasing cost tree search for optimal multi-agent pathfinding. *Artif. Intell.* **2013**, *195*, 470–495. [\[CrossRef\]](#)
- Van Nguyen, L.; Phung, M.D.; Ha, Q.P. Game Theory-Based Optimal Cooperative Path Planning for Multiple UAVs. *IEEE Access* **2022**, *10*, 108034–108045. [\[CrossRef\]](#)
- Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [\[CrossRef\]](#)
- Nguyen, L.V.; Herrera, I.T.; Le, T.H.; Phung, D.M.; Aguilera, R.P.; Ha, Q.P. Stag hunt game-based approach for cooperative UAVs. In Proceedings of the International Symposium on Automation and Robotics in Construction, Bogotá, Colombia, 13–15 July 2022; pp. 367–374.
- Silver, D. Cooperative pathfinding. In Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference, AIIDE 2005, Marina Del Rey, CA, USA, 1–2 June 2005.
- Li, N.; Fan, A.Z.; Fischer, R.; Kontar, W.; Ran, B. A Prioritized Trajectory Planning Algorithm for Connected and Automated Vehicle Mandatory Lane Changes. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, Indianapolis, IN, USA, 19–22 September 2021.
- Ma, X.; Jiao, Z.; Wang, Z.; Panagou, D. 3-D Decentralized Prioritized Motion Planning and Coordination for High-Density Operations of Micro Aerial Vehicles. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 939–953. [\[CrossRef\]](#)
- Tazaki, Y.; Suzuki, T. Constraint-based prioritized trajectory planning for multibody systems. *IEEE Trans. Robot.* **2014**, *30*, 1227–1234. [\[CrossRef\]](#)
- Cap, M.; Novak, P.; Selecky, M.; Faigl, J.; Vokffnek, J. Asynchronous decentralized prioritized planning for coordination in multi-robot system. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013.

23. Cap, M.; Novak, P.; Kleiner, A.; Selecky, M. Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 835–849. [\[CrossRef\]](#)
24. Velagapudi, P.; Sycara, K.; Scerri, P. Decentralized prioritized planning in large multirobot teams. In Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010—Conference Proceedings, Taipei, Taiwan, 18–22 October 2010.
25. Liao, B.; Wan, F.; Hua, Y.; Ma, R.; Zhu, S.; Qing, X. F-RRT*: An improved path planning algorithm with improved initial solution and convergence rate. *Expert Syst. Appl.* **2021**, *184*, 115457. [\[CrossRef\]](#)
26. Van Den Berg, J.P.; Overmars, M.H. Prioritized motion planning for multiple robots. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Edmonton, AB, Canada, 2–6 August 2005.
27. Deng, X.; Li, R.; Zhao, L.; Wang, K.; Gui, X. Multi-obstacle path planning and optimization for mobile robot. *Expert Syst. Appl.* **2021**, *183*, 115445. [\[CrossRef\]](#)
28. Rasekhipour, Y.; Khajepour, A.; Chen, S.K.; Litkouhi, B. A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1255–1267. [\[CrossRef\]](#)
29. Yao, P.; Wang, H.; Su, Z. Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment. *Aerosp. Sci. Technol.* **2015**, *47*, 269–279. [\[CrossRef\]](#)
30. Wu, Y.; Wu, S.; Hu, X. Cooperative Path Planning of UAVs UGVs for a Persistent Surveillance Task in Urban Environments. *IEEE Internet Things J.* **2021**, *8*, 4906–4919. [\[CrossRef\]](#)
31. Yang, H.; Qi, J.; Miao, Y.; Sun, H.; Li, J. A New Robot Navigation Algorithm Based on a Double-Layer Ant Algorithm and Trajectory Optimization. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8557–8566. [\[CrossRef\]](#)
32. An, S.; Park, M.; Oh, H. Receding-horizon RRT-Infotaxis for autonomous source search in urban environments. *Aerosp. Sci. Technol.* **2022**, *120*, 107276. [\[CrossRef\]](#)
33. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
34. Jiang, W.; Lyu, Y.; Li, Y.; Guo, Y.; Zhang, W. UAV path planning and collision avoidance in 3D environments based on POMPD and improved grey wolf optimizer. *Aerosp. Sci. Technol.* **2022**, *121*, 107314. [\[CrossRef\]](#)
35. Guo, Y.; Liu, X.; Liu, X.; Yang, Y.; Zhang, W. FC-RRT*: An Improved Path Planning Algorithm for UAV in 3D Complex Environment. *ISPRS Int. J. Geo-Inform.* **2022**, *11*, 112. [\[CrossRef\]](#)
36. Gammell, J.D.; Barfoot, T.D.; Srinivasa, S.S. Informed Sampling for Asymptotically Optimal Path Planning. *IEEE Trans. Robot.* **2018**, *34*, 966–984. [\[CrossRef\]](#)
37. Akgun, B.; Stilman, M. Sampling heuristics for optimal motion planning in high dimensions. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 2640–2645.
38. Li, Y.; Wei, W.; Gao, Y.; Wang, D.; Fan, Z. PQ-RRT*: An improved path planning algorithm for mobile robots. *Expert Syst. Appl.* **2020**, *152*, 113425. [\[CrossRef\]](#)
39. Liu, Y.; Zhang, X.; Zhang, Y.; Guan, X. Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach. *Chin. J. Aeronaut.* **2019**, *32*, 1504–1519. [\[CrossRef\]](#)
40. Ming, Z.; Lingling, Z.; Xiaohong, S.; Peijun, M.; Yanhang, Z. Improved discrete mapping differential evolution for multi-unmanned aerial vehicles cooperative multi-targets assignment under unified model. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 765–780. [\[CrossRef\]](#)
41. Li, H.; Long, T.; Xu, G.; Wang, Y. Coupling-Degree-Based Heuristic Prioritized Planning Method for UAV Swarm Path Generation. In Proceedings of the 2019 Chinese Automation Congress, CAC 2019, Hangzhou, China, 22–24 November 2019.
42. Brunner, M.; Bruggemann, B.; Schulz, D. Hierarchical rough terrain motion planning using an optimal sampling-based method. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 5539–5544.

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.