Article

Quick Path Planning Based on Shortest Path Algorithm for Multi-UAV System in Windy Condition

Fuliang Zhou [1,2] and Hong Nie [1,2,*]

[1] State Key Laboratory of Mechanics and Control of Mechanical Structures, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China
[2] Key laboratory of Fundamental Science for National Defense-Advanced Design Technology of Flight Vehicle, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China
* Correspondence: hnie@nuaa.edu.cn;

Abstract: A graph based path planning method consisting of off-line part and on-line part for multi-UAV system in windy condition is proposed. In the off-line part, the task area is divided into grids. When two nodes is close enough, they are connected and weighted by the cost, obtained by solving an optimization problem based on difference method, between them. In order to ensure the accuracy of the difference method, the adjacent radius is set to be small enough. In order to ensure the generated paths close to analytical solution, the grid size is set to be much smaller than the adjacent radius. The shortest path algorithm is used to update the adjacent matrix and path matrix. In the on-line part, the target position assignment and optimal velocity can be obtained by accessing the adjacent matrix and path matrix. At the end of this paper, some numerical examples are taken to illustrate the validity of our method and the influence of the parameters.

Keywords: Path planning; Multi-UAV system; Shortest path algorithm

1. Introduction

In the past decades, cooperative coordination of multi-agent systems (MASs) has attracted adding attention from different domains [1]. Path planning of multi-unmanned aerial vehicle (multi-UAV) system is one of the key problems in MASs because of UAV's wide application ranges in searching, surveying, mapping, measurement and rescue missions [1] [2] . Path planning aims to find the path with the minimal cost between two points. Many path planning approaches have been proposed in literatures, which can be categorized into several types [3], such as graph-search methods, variational and nonlinear programming (NLP) methods, artificial potential field (APF) methods, and so on.

In graph based methods, the space is discretized as a graph, and the nodes represent a set of positions, the edges represent transitions between nodes. The optimal path is generated by making a search for a minimal cost path in such a graph. Then algorithms search the available nodes giving a solution if a path exist [4]. A* [5] [6] [7], D* [5] and rapidly random-exploring tree (RRT) [8] [9] [10] are the representatives of this class. In recent years, some improvements of the above-mentioned methods were published. Based on variable-step-length, paper [11] proposed an improvement of A* algorithm to overcome the drawbacks of the traditional A* algorithm, such as the large amount of steps and the non-optimal solution. A modifications in A* algorithm for reducing the processing time are proposed in [12]. For purpose of improving the real-time capability of UAV path planning, a method based on A* algorithm and virtual force is proposed in paper [13]. In paper [14], an algorithm named A*-RRT* is introduced. In A*-RRT* algorithm, an initial path is generated from the A* algorithm, and then used to guide the sampling process of the RRT* planner. Based on any-angle path biasing, the Thete*-RRT* algorithm, which can find a shorter path in shorter time than the RRT* and the A*-RRT* algorithms, is proposed in paper [15]. To overcome the limitaions of RRT, such as slow convergence and high memory consumption, combined with the convolutional neural network (CNN), paper [10] proposed the Neural RRT* algorithm.

A path planning problem can be converted to an optimization problem. Variational methods may be the most natural and direct for such problems. However, it is nearly impossible to solve complex problems using variational methods. Many papers solve such problems with NLP method, where the state and input variables sets are discretized into several intervals [16] [17] [18].

APF represents a class of effective motion planning methods, its algorithm structure is simple and suitable for navigation tasks with real-time requirements [19] [**?**]. Agents are simplified to particles moving along the current gradient direction of a potential created according to the target and the environment obstacles. Singularities and the convergence to local minimum are the most important problems still being studied [2].

In the past few years, the complexity of the path planning problem has increased. To deal with this complexity, researchers start to concentrate on nondeterministic algorithms [20]. In paper [21], a method based on vibrational genetic algorithm (GA) is proposed to improve the exploration and avoidance of local minima when searching for an optimal path. Combining with the concept of artificial immune system, the authors of [22] use a GA to maintain superior population diversity throughout the evolution process. In paper [23], the authors use a particle swarm optimization (PSO) to calculate the shortest path between reconnaissance UAV targets.

Although above-mentioned researches has gained notable achievement in many areas, there are still some deficiencies to be improved. The search strategy of some traditional graph based approaches can be summed up as "eight neighbourhood search", where the number of adjacent nodes of each node is limited 8, and the motion azimuth angle is limited as integral multiple of $\pi/4$. Under this limitation, the obtained path is probably not the optimal path. For some NLP methods and nondeterministic algorithms, it is hard to satisfy the requirement of the real-time capability because of their time consumption on repeated iteration or evolution. And some researches lack of the comparison between their results and the analytical solutions. In addition, the existing researches focus primarily on the path planning with obstacle avoidance, but the path planning of UAVs in windy conditions is rarely studied.

Fortunately, recently, a trajectory optimization algorithm that balances optimality and real-time performance has been proposed in literature [24]. Based on this algorithm, an improved graph based approach is proposed in this paper. To make the generated path converge to the optimal path, a concept named adjacent radius, a standard to judge whether two nodes is adjacent, is put forward. In order to balance the optimality and the real-time capability, our method consists of two parts: off-line part and on-line part. In the off-line part, the cost, such as time consumption and fuel consumption, between two adjacent nodes is calculated using difference method and stored in an adjacency matrix. After that, the shortest path algorithm, such as Floyd-Warshall algorithm, is taken to update the adjacency matrix. In the on-line part, the UAV is guided from the initial position to the target position according to the message stored in the adjacency matrix.

The rest of this paper is organized as follows. Section 2 presents some preliminaries and problem formulation. Section 3 presents the details of our method. Some numerical simulations are given in Section 4. The results are summarized in Section 5.

2. Preliminaries And Problem Formulation

2.1. Preliminaries

In graph theory, the shortest path problem is to find the path between two nodes in a graph such that the sum of the weights of its constituent edges is minimized. Two nodes are adjacent when they are both incident to a common edge. The adjacency relationship of a graph is always described by a adjacent matrix whose element in row i column j

presents the weight from the ith node to the jth node , and the weight is set to infinity if there is no edge start from vertex i to vertex j. For a undirected graph, the adjacent matrix is symmetric [25] [26].

Floyd-Warshall algorithm is a famous shortest path algorithm. It can find shortest paths in a weighted graph with no negative cycles, and it can find the summed weights of shortest paths between all pairs of nodes. With simple modifications, it is possible to reconstruct the paths. The Floyd-Warshall algorithm compares all possible paths through the graph between each pair of nodes, such that its time complexity and space complexity are $O(N^3)$ and $O(N^2)$ respectively [27]. In this paper, Floyd-Warshall algorithm is used to solve the shortest path problem.

2.2. Problem Formulation

We consider a planar system that consists of M UAVs and each UAV, $i \in \mathcal{V} = \{1, ..., M\}$, has the following dynamics in stagnant air:

$$\begin{cases} \dot{p}_i = u_i \\ p_i \in \Omega_p - \Omega_b \\ u_i \in \Omega_u \end{cases} \tag{1}$$

where p_i and u_i are the position and velocity of the ith UAV; Ω_u ,Ω_p are closed sets, and $\Omega_p = \left\{(x_1, x_2) | \delta_1^d \le x_1 \le \delta_1^u, \delta_2^d \le x_2 \le \delta_2^u \right\}$; Ω_b is a open set and represents the barrier. Ω_p is in a steady wind field. The wind speed at position x is denoted as

$$w = f(x) \tag{2}$$

Then the dynamics of the ith UAV in the wind field is

$$\dot{p}_i = f(p_i) + u_i \tag{3}$$

Let I_i denote the initial position of the ith UAV and $T = \{T_1, ..., T_M\}$ denote the set of target positions. The objective of this paper is to specify a target position, say T_{ki}, for the ith UAV, and design a control law $u_i = g(p_i)$ such that

$$J_{\text{sum}} = \begin{cases} \sum\limits_{i=1}^{M} \int_0^{t_{fi}} L(p_i, u_i) dt \\ s.t. \quad p_i(0) = I_i \\ \qquad p_i(t_{fi}) = T_{ki} \end{cases} \tag{4}$$

reach its minimum. The function $L(p_i, u_i)$ in Equation 4 is the cost function, and it is smooth about p_i and u_i.

3. Method Details

3.1. Mesh the Task Area

The first step of our method is to mesh the task area Ω_p. The ith dimension of Ω_p is divided evnely by N_i nodes, then the mesh nodes set is

$$\begin{aligned} V = &\left\{ \delta_1^d + k \frac{\delta_1^u - \delta_1^d}{N_1 - 1} | k = 0, 1, ..., N_1 - 1 \right\} \\ &\times \left\{ \delta_2^d + k \frac{\delta_2^u - \delta_2^d}{N_2 - 1} | k = 0, 1, ..., N_2 - 1 \right\} \end{aligned} \tag{5}$$

where "$\times$" is Cartesian product. The mesh size can be defined as $s = \max\left(\frac{\delta_1^u - \delta_1^d}{N_1 - 1}, \frac{\delta_2^u - \delta_2^d}{N_2 - 1}\right)$. The coordinates of the ith node in V are:

$$v_i = \left(\delta_1^d + k_1^i \frac{\delta_1^u - \delta_1^d}{N_1 - 1}, k_2^i \frac{\delta_2^u - \delta_2^d}{N_2 - 1}\right) \tag{6}$$

where

$$\begin{aligned} k_1^i &= i//N_2 \\ k_2^i &= i\%N_2 \end{aligned} \tag{7}$$

"//" and "%" are exact division operator and modulo operator respectively. For any $x = (x_1, x_2) \in \Omega_p$, assume its nearest mesh point is v_i, then

$$i = \mathrm{Int}\left[\frac{x_1(N_1 - 1)}{\delta_1^u - \delta_1^d}\right] N_2 + \mathrm{Int}\left[\frac{x_2(N_2 - 1)}{\delta_2^u - \delta_2^d}\right] \tag{8}$$

Function $\mathrm{Int}(.)$ means the nearest integer of its inside, that is

$$\mathrm{Int}(x) = \arg\min_{i \in \mathbb{Z}} |x - i| \tag{9}$$

3.2. Calculate and Update the Adjacent Matrix

Define a graph over all nodes by connecting the ith node v_i and the jth node v_j if (as measured by Euclid norm $\|v_i - v_j\|_2$) they are closer than the adjacent radius r and there are no barriers on the line between v_i and v_j, then set the edge weights $c_{i,j}$ equal to the minimal cost from v_i to v_j, otherwise, $c_{i,j} = \infty$. If r is small enough, by using central difference method, $c_{i,j}$ can be obtained. See Equation (10).

$$c_{i,j} = \begin{cases} \left\{\begin{array}{l} \min\limits_u L\left(\frac{v_i + v_j}{2}, u\right)\Delta t \\ s.t.\ v_j = v_i + \left[f\left(\frac{v_i + v_j}{2}\right) + u\right]\Delta t \\ \qquad u \in \Omega_u \\ \qquad \Delta t \geq 0 \end{array}\right\}, & \begin{array}{l} \|v_i - v_j\|_2 \leq r \\ l_{v_i, v_j} \cap \Omega_b = \emptyset \end{array} \\ \infty, & otherwise \end{cases} \tag{10}$$

Where l_{v_i,v_j} is the line between v_i and v_j. Then, set the element at row i and column j of adjacent matrix D as $c_{i,j}$.

After that, use Floyd-Warshall algorithm to update the adjacent matrix D and generate path matrix P. Finally, $D_{i,j}$ is the cost of the shortest path in the graph start from v_i to v_j, and $P_{i,j}$ is the index of the first node passed by the shortest path from v_i to v_j.

The pseudocode of algorithm described in this subsection is shown in Algorithm 1. Literature [24] suggests that the shortest path from v_i to v_j converges to the optimal trajectory between v_i and v_j, if the adjacent radius r and the grid size s tend to infinitesimal and s is a higher infinitesimal of r.

3.3. On-line Control

For any positions $p_1, p_2 \in \Omega_p - \Omega_b$, the nearest nodes, say v_{i1} and v_{i2}, of them can be obtained by using Equation (8) and (9). Then $D_{i1,i2}$ is the approximate value of the minimal cost from p_1 to p_2. Therefore, the targets assignment problem can be solved by using Hungarian algorithm [28] or by traversing all permutations of set T. The outcome of this step depends on the accuracy of D. An inaccurate D always results in a bad assignment.

Algorithm 1 Calculate and update the adjacent matrix

Let D and P are matrices with shape $N_1N_2 \times N_1N_2$;
for $i \in \{0, 1, ..., N_1N_2 - 1\}$ **do**
    for $j \in \{0, 1, ..., N_1N_2 - 1\}$ **do**
        if $i = j$ **then**
            $D_{i,j} \leftarrow 0$;
            Continue;
        end if
        if $\|v_i - v_j\|_2 \le r$ and $l_{v_i,v_j} \cap \Omega_b = \emptyset$ **then**

$$c_{i,j} \leftarrow \begin{cases} \min\limits_u L\left(\frac{v_i+v_j}{2}, u\right)\Delta t \\ s.t. \quad v_j = v_i + \left[f\left(\frac{v_i+v_j}{2}\right) + u\right]\Delta t \\ \qquad u \in \Omega_u \\ \qquad \Delta t \ge 0 \end{cases}$$

            $D_{i,j} \leftarrow c_{i,j}$;
        else
            $D_{i,j} \leftarrow \infty$;
        end if
    end for
end for
Use Floyd-Warshall algorithm to update D and P;

Given the current position p_{cur} and the target position p_{tar} of a UAV, the nearest nodes, say v_a and v_b, of them can be obtained. Then the node numbered $P_{a,b}$ can be choosen as a temporary target, and the distance between temporary target and v_a is less than r. And the optimal velocity u^* of this UAV can be obtained through some simple calculations, see Equation (11).

$$u^* = \begin{cases} \left\{ \begin{array}{l} \arg\min\limits_u L\left(\frac{p_{cur}+v_{P_{a,b}}}{2}, u\right)\Delta t \\ s.t. \; v_{P_{a,b}} = p_{cur} + \left[f\left(\frac{p_{cur}+v_{P_{a,b}}}{2}\right) + u\right]\Delta t \\ \qquad u \in \Omega_u \\ \qquad \Delta t \ge 0 \end{array} \right\}, & \\ \qquad\qquad\qquad\qquad \|p_{cur} - p_{tar}\|_2 > s \\ \\ \left\{ \begin{array}{l} \arg\min\limits_u L\left(\frac{p_{cur}+p_{tar}}{2}, u\right)\Delta t \\ s.t. \; p_{tar} = p_{cur} + \left[f\left(\frac{p_{cur}+p_{tar}}{2}\right) + u\right]\Delta t \\ \qquad u \in \Omega_u \\ \qquad \Delta t \ge 0 \end{array} \right\}, & \\ \qquad\qquad\qquad\qquad \|p_{cur} - p_{tar}\|_2 \le s \end{cases} \tag{11}$$

See Fig. 1 for the intuitive interpretation.

4. Numerical Simulations

In this section, some simulations under different parameters are taken to analyze the influence of mesh size s and adjacent radius r on the solution, and to illustrate the validity of our method.

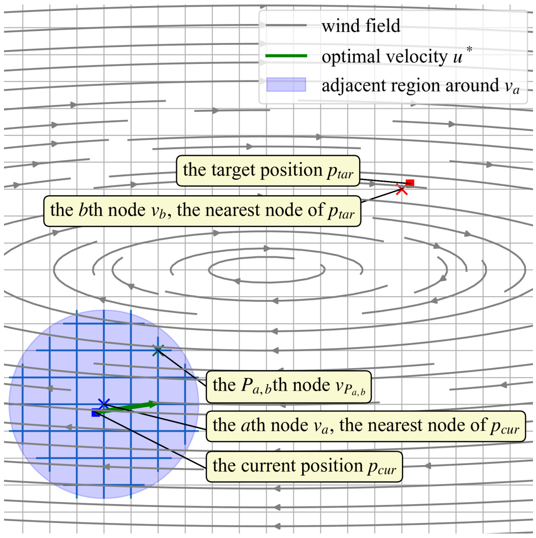


Figure 1. Intuitive interpretation of u^*

Consider a path planning problem for a multi-UAV system with 4 UAVs, the task area is $\Omega_p = \{(x_1, x_2)| -100 \le x_1, x_2 \le 100\}$. the limitation of UAV velocity is $\Omega_u = \{(u_1, u_2)| -10 \le u_1, u_2 \le 10\}$, and the wind field w is

$$\begin{cases} w_1 = 0.08x_2 \\ w_2 = -0.08x_1 \end{cases} \tag{12}$$

And the barriers set is $\Omega_b = \{(x_1, x_2)|x_2 \ge 8x_1 + 60,\ \ x_2 \ge -8x_1 + 60\}$. In the following examples, the nodes quantities in the two dimensions of task area are equal, that is $N_1 = N_2$.

4.1. Time Optimal Path Planning

This subsection aims to specify a proper target position and plan path for each UAV to minimize the time consumption of the task. The initial positions set of UAVs is $I = \{(-100, -10), (-100, 10),\ \ (-80, -10), (-80, 10)\}$, and the target positions set is $T = \{(90, 50), (90, 25),\ \ (40, -70), (50, -80)\}$. In other words, mathematically, find $u_i(t)$ for the ith UAV, $i = 1, ..., 4$, to minimize the following expression:

$$J_{\text{sum}} = \begin{cases} \sum_{i=1}^{4} \int_0^{t_{fi}} 1dt \\ s.t.\ \ p_i(0) = I_i \\ \qquad p_i(t_{fi}) = T_{ki} \\ \qquad \dot{p} = w + u_i \\ \qquad p_i(t) \in \Omega_p - \Omega_b \\ \qquad u_i(t) \in \Omega_u \end{cases} \tag{13}$$

where T_{ki} is the specified position for the ith UAV. Four groups of parameters are tested, they are $s = 4\ r = 4\sqrt{2}$, $s = 2\ r = 2\sqrt{5}$, $s = 4/3\ r = 16/3$ and $s = 1,\ r = 5$. After the aversal of all the permutations of set T, see Algorithm 2, the correspondences between the UAVs and the target positions under the different parameters are shown in Table 1. For this problem, after identified the initial position and target postion for UAVs, the analytical solutions can be easily obtained. The simulation results are shown in Figure 2, the simulation time step is set as 0.001.

It can be seen from Table 1 and Figure 2 that as both the density of nodes and the quantity of adjacent points of each node increases, the planned paths generated by the proposed method approach to the analytical solutions, and the estimated time consump-

Table 1: Specified target position for each UAV under different parameters

Number of UAV	Initial Position	$s=4, r=4\sqrt{2}$		$s=2, r=2\sqrt{5}$		$s=4/3, r=16/3$		$s=1, r=5$	
		Target position	Estimated cost	Target position	Estimated cost	Target position	Estimated cost	Target position	Estimated cost
1	$(-100,-10)$	$(50,-80)$	17.71	$(90,50)$	13.58	$(90,50)$	13.37	$(90,50)$	13.27
2	$(-100,10)$	$(90,50)$	13.39	$(90,25)$	13.45	$(90,25)$	13.24	$(90,25)$	13.14
3	$(-80,-10)$	$(40,-70)$	14.94	$(40,-70)$	13.65	$(40,-70)$	13.23	$(40,-70)$	13
4	$(-80,10)$	$(90,25)$	12.39	$(50,-80)$	14.59	$(50,-80)$	14.04	$(50,-80)$	13.82

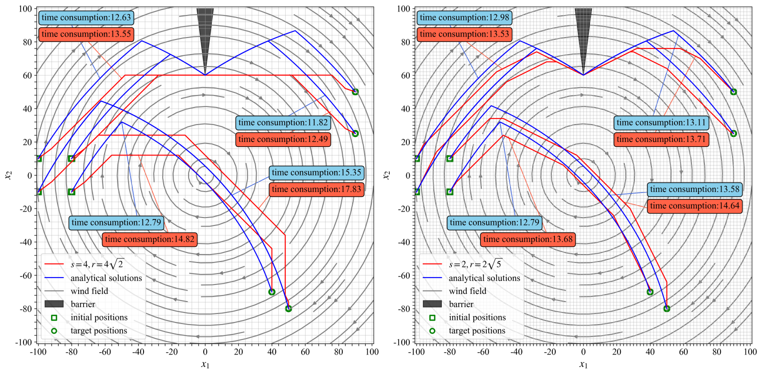
tion (the estimated cost in Table 1) and the real time consumption (the simulation time, shown in the red textboxes in Figure 2) of our method close to the time consumption of the analytical solutions (shown in the blue textboxes in Figure 2). Limited by the "eight neighbourhood search", the estimated costs in the case of $s=4, r=4\sqrt{2}s$ have relatively big error such that the method cannot specify the optimal target postion for every UAV, and the planned paths are far away from the analytical solutions.

4.2. *Fuel Optimal Path Planning*

This subsection aims to specify a proper target position and plan path for each UAV to minimize the total fuel consumption of the task. The initial positions set of UAVs is $I=\{(100,0),(80,0),\ (60,0),(40,0)\}$, and the target positions set is $T=\{(-80,20),(-60,20),\ (70,20),(50,30)\}$. That is, find $u_i(t)$ for the ith UAV, $i=1,...,4$, to minimize the following expression:

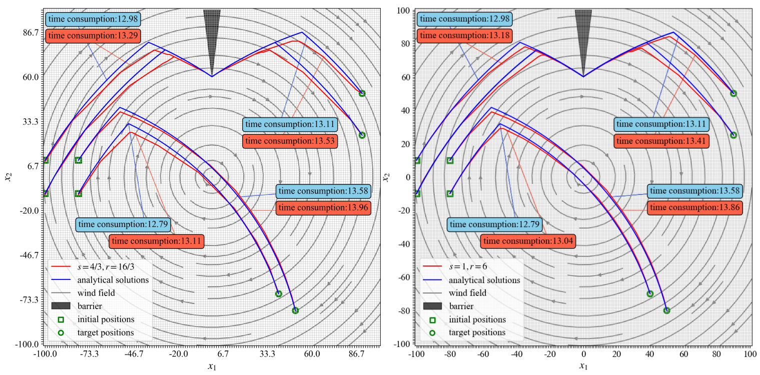
$$J_{\text{sum}} = \begin{cases} \sum_{i=1}^{4} \int_0^{t_{fi}} \|u_i\|_2 dt \\ s.t. \quad p_i(0) = I_i \\ \qquad p_i(t_{fi}) = T_{ki} \\ \qquad \dot{p} = w + u_i \\ \qquad p_i(t) \in \Omega_p - \Omega_b \\ \qquad u_i(t) \in \Omega_u \end{cases} \tag{14}$$

The parameter settings are same as the previous subsection. The results of target position assignment under different parameters are shown in Table 2. And the simulation results and the analytical solutions (not unique) are shown in Figure 3. It can be seen from Table 2 and Figure 3 that, similarly as the previous example, greater nodes density can significantly improve the performance of our method in terms of fuel consumption estimate accuracy and generated paths. In the cases of $s=4, r=4\sqrt{2}s$ and $s=2, r=2\sqrt{5}s$, the fuel consumption estimates have relatively big error, and the generated paths are very different from the optimal paths.



(a) Simulation result under $s = 4, r = 4\sqrt{2}$.

(b) Simulation result under $s = 2, r = 2\sqrt{5}$.



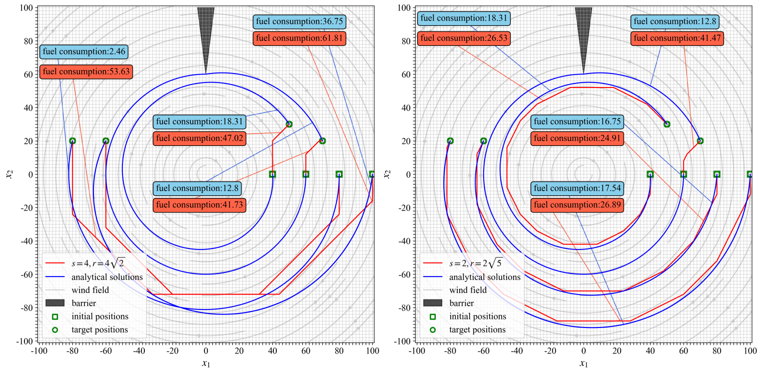
(c) Simulation result under $s = 4/3, r = 16/3$.

(d) Simulation result under $s = 1, r = 5$.

Figure 2. Simulation results for time optimal path planning under different parameters.

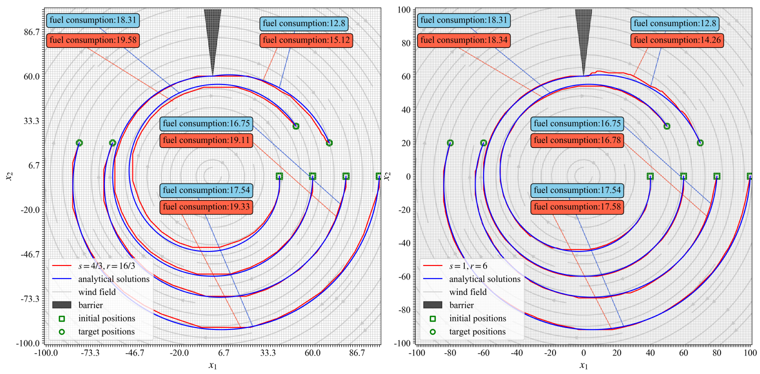
Table 2: Specified target position for each UAV under different parameters

Number of UAV	Initial Position	$s=4, r=4\sqrt{2}$		$s=2, r=2\sqrt{5}$		$s=4/3, r=16/3$		$s=1, r=5$	
		Target position	Estimated cost	Target position	Estimated cost	Target position	Estimated cost	Target position	Estimated cost
1	$(100,0)$	$(-60,20)$	60.75	$(-80,20)$	26.55	$(-80,20)$	19.24	$(-80,20)$	17.56
2	$(80,0)$	$(-80,20)$	53.52	$(-60,20)$	24.37	$(-60,20)$	18.67	$(-60,20)$	16.76
3	$(60,0)$	$(70,20)$	41.62	$(70,20)$	41.22	$(70,20)$	15.01	$(70,20)$	14.03
4	$(40,0)$	$(50,30)$	46.73	$(50,30)$	26.03	$(50,30)$	19.21	$(50,30)$	18.32



(a) Simulation result under $s = 4, r = 4\sqrt{2}$.

(b) Simulation result under $s = 2, r = 2\sqrt{5}$.



(c) Simulation result under $s = 4/3, r = 16/3$.

(d) Simulation result under $s = 1, r = 5$.

Figure 3. Simulation results for time optimal path planning under different parameters.

5. Conclusion

This paper proposed a graph based path planning method for multi-UAV system in steady windy condition. For both accuracy and real time capability, the method is divided into off-line and on-line parts. In the off-line part, the task area is divided into grids and a adjacent matrix and a path matrix are established. A pair of nodes are connected and weighted by the cost between them if they are closer than the adjacent radius. The cost is obtained by solving an optimization problem based central difference method and stored in the adjacent matrix. After that, Floyd-Warshall algorithm is used to update the adjacent matrix and path matrix. In the on-line part, the costs of different target position assignment schemes can be accurately estimated by access the adjacent matrix. And the optimal velocity of each UAV can be easily obtained by access the path matrix.

However, some shortcomings are found during the study. A large number of nodes always result in a lot of memory consumption. In addition, this method lacks a collision avoidance mechanism. These problems will be considered in our future work.

References

1. Wu, Q.; Chen, Z.; Wang, L.; Lin, H.; Jiang, Z.; Li, S.; Chen, D. Real-Time Dynamic Path Planning of Mobile Robots: A Novel Hybrid Heuristic Optimization Algorithm. *Sensors* **2019**, *20*, 188. doi:10.3390/s20010188.
2. Chen, Y.; Luo, G.c.; Mei, Y.s.; Yu, J.q.; Su, X.l. UAV path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science* **2014**, *47*, 1–14. doi:10.1080/00207721.2014.929191.
3. D Amato, E.; Mattei, M.; Notaro, I. Bi-level Flight Path Planning of UAV Formations with Collision Avoidance. *Journal of Intelligent and Robotic Systems* **2018**, *93*, 193–211. doi:10.1007/s10846-018-0861-1.
4. Gonzalez Bautista, D.; Pérez, J.; Milanes, V.; Nashashibi, F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems* **2015**, pp. 1–11. doi:10.1109/TITS.2015.2498841.
5. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* **1968**, *4*, 100–107.
6. Sudhakara, P.; Ganapathy, V. Trajectory Planning of a Mobile Robot using Enhanced A-Star Algorithm. *Indian Journal of Science and Technology* **2016**, *9*. doi:10.17485/ijst/2016/v9i41/93816.
7. Meng, B.b.; Gao, X. UAV Path Planning Based on Bidirectional Sparse A* Search Algorithm. *Intelligent Computation Technology and Automation, International Conference on* **2010**, *3*, 1106–1109. doi:10.1109/ICICTA.2010.235.
8. LaValle, S.; Kuffner, J. Randomized Kinodynamic Planning. 1999, Vol. 20, pp. 473–479. doi:10.1109/ROBOT.1999.770022.
9. Lavalle, S. Rapidly-Exploring Random Trees: A New Tool for Path Planning **1999**.
10. Wang, J.; Chi, G.; Li, C.; Wang, C.; Meng, M. Neural RRT*: Learning-Based Optimal Path Planning. *IEEE Transactions on Automation Science and Engineering* **2020**, *PP*, 1–11. doi:10.1109/TASE.2020.2976560.
11. Da, K.; Xiaoyu, L.; Bi, Z. Variable-step-length A* algorithm for path planning of mobile robot. 2017, pp. 7129–7133. doi:10.1109/CCDC.2017.7978469.
12. Guruji, A.; Agarwal, H.; Parsediya, D. Time-efficient A* Algorithm for Robot Path Planning. *Procedia Technology* **2016**, *23*, 144–149. doi:10.1016/j.protcy.2016.03.010.
13. Dong, Z.; Chen, Z.; Zhou, R.; Zhang, R. A hybrid approach of virtual force and A* search algorithm for UAV path re-planning. *Proceedings of the 2011 6th IEEE Conference on Industrial Electronics and Applications, ICIEA 2011* **2011**, pp. 1140–1145. doi:10.1109/ICIEA.2011.5975758.
14. Brunner, M.; Brüggemann, B.; Schulz, D. Hierarchical Rough Terrain Motion Planning using an Optimal Sampling-Based Method. 2013. doi:10.1109/ICRA.2013.6631372.
15. Palmieri, L.; Koenig, S.; Arras, K. RRT-based nonholonomic motion planning using any-angle path biasing. 2016, pp. 2775–2781. doi:10.1109/ICRA.2016.7487439.
16. Qureshi, A.; Ayaz, Y. Potential Functions based Sampling Heuristic For Optimal Path Planning. *Autonomous Robots* **2015**, *40*, 1079–1093. doi:10.1007/s10514-015-9518-0.
17. Wang, J.; Chi, G.; Shao, M.; Meng, M. Finding a High-Quality Initial Solution for the RRTs Algorithms in 2D Environments. *Robotica* **2019**, *37*, 1–18. doi:10.1017/S0263574719000195.
18. Wang, J.; Li, X.; Meng, M. An improved RRT algorithm incorporating obstacle boundary information. 2016, pp. 625–630. doi:10.1109/ROBIO.2016.7866392.
19. Rickert.; Markus.; Brock.; Oliver.; Sieverling.; Arne. Balancing Exploration and Exploitation in Sampling-Based Motion Planning. *IEEE Transactions on Robotics A Publication of the IEEE Robotics And Automation Society* **2014**.
20. Roberge, V.; Tarbouchi, M.; Labonte, G. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *Industrial Informatics, IEEE Transactions on* **2013**, *9*, 132–141. doi:10.1109/TII.2012.2198665.
21. Pehlivanoglu, V. A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. *Aerospace Science and Technology - AEROSP SCI TECHNOL* **2011**, *16*, 47–55. doi:10.1016/j.ast.2011.02.006.
22. Cheng, Z.; Sun, Y.; Liu, Y. Path planning based on immune genetic algorithm for UAV **2011**. pp. 590–593. doi:10.1109/ICEICE.2011.5777407.
23. Bao, Y.; Fu, X.; Gao, X. Path planning for reconnaissance UAV based on Particle Swarm Optimization **2010**. pp. 28–32. doi:10.1109/CINC.2010.5643794.
24. Liao, W.; Wei, X.; Lai, J.; Sun, H. Numerical Method with High Real-time Property Based on Shortest Path Algorithm for Optimal Control. *International Journal of Control, Automation and Systems* **2021**, *19*. doi:10.1007/s12555-020-0196-0.
25. Ahuja, R.K.; Mehlhorn, K.; Orlin, J.B.; Tarjan, R.E. Faster Algorithms for the Shortest Path Problem. *Journal of the Acm* **1990**, *37*, 213–223. doi:10.1145/77600.77615.
26. Cherkassky, B.V.; Goldberg, A.V.; Radzik, T. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming* **1996**, *73*, 129–174. doi:10.1007/BF02592101.
27. W. Floyd, R. Algorithm 97: "Shortest Path. *Commun. ACM* **1962**, *5*, 345. doi:10.1145/367766.368168.
28. Kuhn, H.W. The Hungarian method for the assignment problem **2005**. *52*, 7–21.