J. Cent. South Univ. (2014) 21: 3079−3091
DOI: 10.1007/s11771-014-2279-7

Springer

Path planning for unmanned aerial vehicles in surveillance tasks under wind fields

ZHANG Xing(张兴)[1, 2], CHEN Jie(陈杰)[1, 2], XIN Bin(辛斌)[1, 2]

1. School of Automation, Beijing Institute of Technology, Beijing 100081, China;
2. State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China

Abstract: The optimal path planning for fixed-wing unmanned aerial vehicles (UAVs) in multi-target surveillance tasks (MTST) in the presence of wind is concerned. To take into account the minimal turning radius of UAVs, the Dubins model is used to approximate the dynamics of UAVs. Based on the assumption, the path planning problem of UAVs in MTST can be formulated as a Dubins traveling salesman problem (DTSP). By considering its prohibitively high computational cost, the Dubins paths under terminal heading relaxation are introduced, which leads to significant reduction of the optimization scale and difficulty of the whole problem. Meanwhile, in view of the impact of wind on UAVs' paths, the notion of virtual target is proposed. The application of the idea successfully converts the Dubins path planning problem from an initial configuration to a target in wind into a problem of finding the minimal root of a transcendental equation. Then, the Dubins tour is derived by using differential evolution (DE) algorithm which employs random-key encoding technique to optimize the visiting sequence of waypoints. Finally, the effectiveness and efficiency of the proposed algorithm are demonstrated through computational experiments. Numerical results exhibit that the proposed algorithm can produce high quality solutions to the problem.

Key words: unmanned aerial vehicle; path planning in wind field; Dubins traveling salesman problem; terminal heading relaxation; differential evolution

1 Introduction

The application of unmanned air vehicles (UAVs) in both civilian and military fields has experienced a remarkable rise of attention in recent years. Path planning for UAVs is a critical issue for their autonomous control which has been widely studied under specific backgrounds [1−3]. When UAVs perform MTST, collecting information from multi-target is a key problem, and the UAVs need to traverse through all the targets as quickly as possible. Different from the euclidean travelling salesman problem (ETSP), in order to make the planned path for UAVs flyable, kinematic constraints of UAVs should be considered in path planning. In this work, Dubins vehicle model is used to describe the aerodynamics of UAVs satisfying the following constraints [4].

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \dfrac{v}{r}u,\ u \in [-1,\ 1] \\ \dot{v} = 0 \end{cases} \tag{1}$$

where (x, y) and θ specify the planar position and the heading of the UAV, respectively, a state (x, y, θ) is termed as a configuration, v is the speed of the UAV, r is the minimal turning radius, and u is the control input.

It is known that there exist six possible shortest path patterns, called Dubins paths, between any two configurations: LSL, RSR, LSR, RSL, LRL and RLR, where L means turning left with the minimal turning radius (u=1), R means turning right with the minimal turning radius (u=−1), and S means flying along a straight line (u=0), respectively [4]. Figure 1 depicts the six possible shortest Dubins paths for a simple example. It is clear that the Dubins distance between two configurations depends not only on their relative position, but also on their headings.

It is known from above that, in order to get a Dubins tour, both the visiting sequence and headings of waypoints should be determined. Due to the strong coupling between the visiting sequence and headings of waypoints, some researchers employed decoupling methods to solve DTSP. TANG and ÖZGÜNER [5] proposed a heuristic method to determine the sequence

Foundation item: Project(61120106010) supported by the Projects of Major International (Regional) Joint Research Program Nature Science Foundation of China; Project (61304215, 61203078) supported by National Natural Science Foundation of China; Project(2013000704) supported by the Beijing Outstanding Ph.D. Program Mentor, China; Project(61321002) supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China
Received date: 2013−04−11; **Accepted date:** 2013−11−05
Corresponding author: XIN Bin, PhD; Tel: +86−010−68912463; E-mail: brucebin@bit.edu.cn.

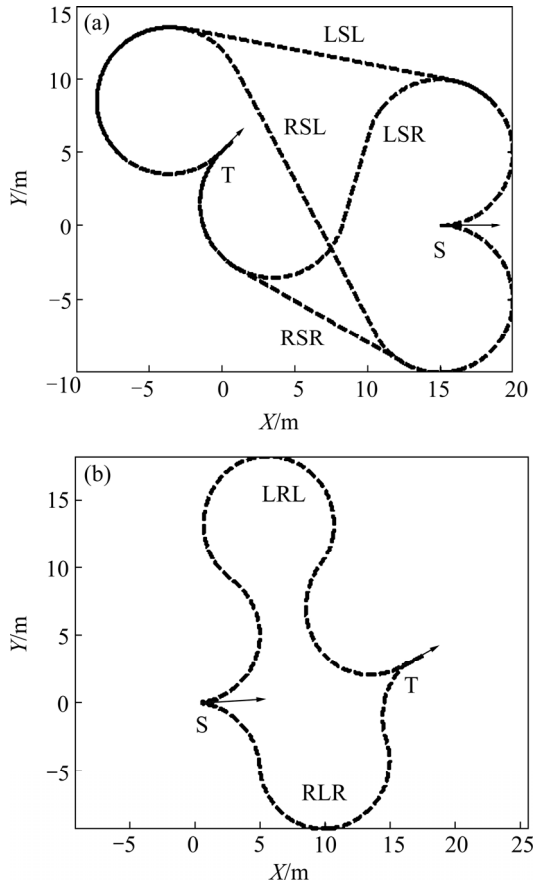

Fig. 1 Six possible shortest Dubins paths between two configurations

of waypoints, then, a gradient based approximation algorithm was applied to optimize the headings. MACHARET et al [6] used the order obtained by solving an angular-metric TSP as the sequence of DTSP, and an optimization method called C-GRASP was used to calculate the headings. SAVLA et al [7] identified the visiting sequence of waypoints by solving the corresponding ETSP, and a simple rule called alternating algorithm (AA) was used to obtain the tour. LENY et al [8] made an analysis and comparison of some recently proposed decoupling methods through corresponding hard instances. The performance of decoupling methods greatly depends on the similarity between Dubins distance and other approximate distance metrics. Some researchers transformed DTSP to TSP which can be easily solved by some prevailing TSP solvers such as LKH [9]. LENY [10] transformed the problem into a generalized TSP (GTSP) by sampling the headings of waypoints and then further converted it into an asymmetric travelling salesman problem (ATSP) based on noon-bean transformation. The precision of the solutions obtained by the transformation algorithm is closely related to the density of heading sampling. Some researchers solved the problem directly. YU and HUNG [11] adopted genetic algorithm to optimize the Dubins tours based on the integral solution encoding using both visiting sequence and headings of the waypoints. In addition, BOISSONNAT and BUI [12] conducted some researches on the optimal Dubins paths between two points without terminal heading constraint. RATHINAM et al [13] solved the DTSP based on the idea proposed in Ref. [12] where only the sequence of waypoints should be optimized, which can greatly reduce the optimization scale. Though it cannot guarantee the optimality of the solution, it makes every subpath of the tour optimal in the sense of terminal heading relaxation.

A lot of studies have been done on DTSP. On the contrary, there is very little research considering the impact of environmental factors (e.g., wind) on the paths especially for small UAVs. MCNEELY and IYER [14] presented conclusions on the existence and uniqueness of minimum-time solutions for a Dubins vehicle flying in wind field, and an iterative method used to calculate the solution was given. MCGEE et al [15] introduced the notion of virtual target to deal with the effect of wind on UAVs' paths, but it needs to calculate ten possible candidate paths for every pair of configurations. MCGEE and HEDRICK [16] assumed the visiting sequence was known, and the Dubins tour was obtained using the method described in Ref. [15], but knowing the visiting sequence is not always possible in practice. TECHY and WOOLSEY [17] studied the problem from a geometric point of view and gave the solution to the optimal Dubins path planning problem in wind.

2 Problem formulation

2.1 Optimization model of DTSP

To obtain a Dubins tour, both the visiting order and headings of waypoints should be determined. Assume that there are N points to be visited, which are denoted by P_1, P_2, $P_3\cdots$, and P_N, respectively. A general mathematical formulation of the DTSP is given as

$$\min\ f(s,\ h)=\sum_{i=1}^{N-1} d((p_{s_i},\ h_{s_i}),\ (p_{s_{i+1}},\ h_{s_{i+1}}))+d((p_{s_N},\ h_{s_N}),\ (p_{s_1},\ h_{s_1})) \tag{2}$$

where $s=[s_1,\ s_2,\ ...,\ s_N](s_i\in\{1,\ 2,\ \cdots,\ N\}$ and $s_i\neq s_j$ $(\forall i\neq j))$ and $h=[h_1,\ h_2,\ ...,\ h_N]$ $(h_i\in[0,\ 2\pi))$ are the permutation of waypoints and the headings of the UAV at these waypoints, respectively. $d((p_{s_i},\ h_{s_i}),\ (p_{s_{i+1}},\ h_{s_{i+1}}))$ is designated as the Dubins distance between the configurations $(p_{s_i},\ h_{s_i})$ and $(p_{s_{i+1}},\ h_{s_{i+1}})$. Obviously, s is a combinational variable while h is a continuous variable, which makes the DTSP a hybrid-variable optimization problem with $2N$ variables. The simultaneous optimization of two different types of variables brings great difficulty to the solving of the problem, and new algorithms have to be designed.

2.2 Dubins paths in sense of terminal heading relaxation

BOISSONNAT and BUI [12] studied the accessibility region for a Dubins car from a given configuration within limited time, where the optimal Dubins paths from an initial configuration to a target with free terminal heading were presented. As shown in Fig. 2, S is the initial position, and the initial heading is along Y-axis. Denote C_R and C_L as the right turning circle and the left turning circle with minimal turning radius at the initial configuration, and O_L and O_R as their corresponding centers, respectively. Due to the symmetry of Dubins paths about the Y-axis, we restrict our study to the points in the right-half plane in the rest of the work. The following conclusions can be obtained.

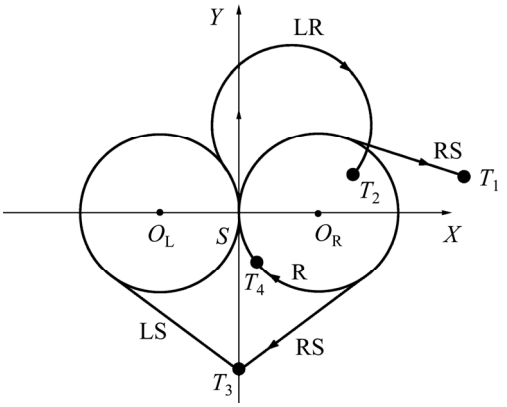

Fig. 2 Dubins paths under terminal heading relaxation

1) For the points outside C_R (e.g., the point T_1 in Fig. 2), the pattern of optimal path is RS.

2) For the points inside C_R (e.g., the point T_2 in Fig. 2), the pattern of optimal path is LR.

3) For the points positioned on the Y-axis (e.g., the point T_3 in Fig. 2), the pattern of optimal path is LS or RS, whose length of corresponding Dubins paths is the same.

4) For the points positioned on C_R (e.g., the point T_4 in Fig. 2), the pattern of optimal path is simplified to R.

2.3 Optimization model of DTSP based on terminal heading relaxation

Based on terminal heading relaxation, only the visiting sequence of waypoints should be determined, which can reduce the solution space by half and avoid optimizing two different types of variables simultaneously. By using the Dubins paths under terminal heading relaxation, the optimization model of DTSP can be described as

$$\min f(\boldsymbol{s}) = \sum_{i=1}^{N-1} \hat{d}\left((p_{s_i}, h_{s_i}), p_{s_{i+1}}\right) + \hat{d}\left((p_{s_N}, h_{s_N}), p_{s_1}\right) \tag{3}$$

where $\hat{d}((p_{s_i}, h_{s_i}), p_{s_{i+1}})$ specifies the Dubins distance from the configuration (p_{s_i}, h_{s_i}) to the target $p_{s_{i+1}}$ under terminal heading relaxation. In the sense of terminal heading relaxation, the heading at the target is determined once the initial configuration is given. Compared with the problem given in Section 2.1, the optimization scale of the whole problem decreases from $2N$ to N, which can effectively reduce the computational complexity.

3 Dubins path planning in wind under terminal heading relaxation

Environmental factors (e.g., wind) can make a great impact on UAVs' paths especially for small UAVs. The dynamics of UAVs in wind can be described approximately as [15]

$$\begin{cases} \dot{x} = v\cos(\theta) + v_{wx} \\ \dot{y} = v\sin(\theta) + v_{wy} \\ \dot{\theta} = \dfrac{v}{r}u,\ u \in [-1,\ 1] \\ \dot{v} = 0 \end{cases} \tag{4}$$

where v_w is the speed of wind, v_{wx} and v_{wy} are the speed components imposed on UAV by wind along X-and Y-axes, respectively. Besides, it is reasonable to assume that $v_w < v$.

3.1 Problem transformation with notion of virtual target

To plan the path from an initial configuration to a target in wind, a virtual target (VT) is introduced [15]. Its initial position is the same with the target and velocity is equal and opposite to that of the wind. Based on the idea, the problem is converted into an interception problem between the UAV and VT without wind.

As described in Ref. [15], the VT moves along the line in the opposite direction of wind. Any point $P(d)$ along the line can be expressed by the variable d which is the distance to the initial target position. Denote $T_1(d)$ as the time that the UAV needs to spend to arrive at the point $P(d)$, and by $T_2(d)$ as the time that the VT needs spend to arrive at the point $P(d)$. The solution d^* to the equation $T_1(d)=T_2(d)$ means that the UAV can intercept with the VT at the point $P(d^*)$. Assume that the coordinates of the target are (x_t, y_t). Then, we have

$$P_x(d) = x_t - d\frac{v_{wx}}{v_w},\ P_y(d) = y_t - d\frac{v_{wy}}{v_w}$$

where $(P_x(d), P_y(d))$ are the coordinates of the VT after it moves distance d.

$$T_1(d) = T_2(d) \rightarrow \frac{D(d)}{v} = \frac{d}{v_w} \tag{5}$$

where $D(d)$ is the Dubins distance from the initial

configuration to $P(d)$. So, the optimal path planning problem from an initial configuration to a target in wind can be solved by finding the minimal root of the equation. It is observed from Ref. [12] that $D(d)$ may be discontinuous, which makes Eq. (5) a discontinuous transcendental equation. In the following, the detailed method to find the minimal root of the equation is described.

3.2 Dubins distance from configuration to line

In the following, we present some basic conclusions which are essential for further analysis of Eq. (5).

Definition 1: The Dubins path from a configuration to a line (DPCL) refers to the Dubins path with shortest Dubins distance from the configuration to any point on the line in the sense of terminal heading relaxation. And the length of the DPCL is termed as Dubins distance from the configuration to the line (DDCL).

Lemma 1: If the line does not intersect with C_R or C_L corresponding to the configuration, the pattern of DPCL is RS, and the straight part of DPCL is perpendicular to the line.

As shown in Fig. 3(a), the DPCL in this condition is depicted by the red line.

Remark 1: If there exist two points on the line whose straight parts of corresponding Dubins paths from the configuration are both perpendicular to the line, the shorter one will be selected as the DPCL.

As shown in Fig. 3(b), the straight parts of optimal Dubins paths corresponding to points A_1 and A_2 are both perpendicular to the line L. Apparently, the Dubins distance from the configuration to A_2 is shorter than that to A_1, so the Dubins path from the configuration to A_2 is the DPCL.

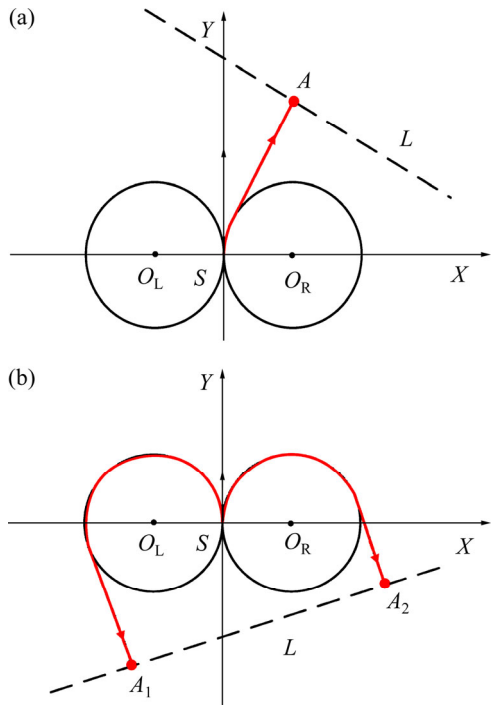

Fig. 3 Schematic of Dubins path from configuration to line in two different scenarios: (a) Scenario 1; (b) Scenario 2

Theorem 1: Given a line which is parallel to Y-axis but does not intersect with C_R or C_L, we denote P_0 as the point whose corresponding Dubins path from the configuration is the DPCL. It is also assumed that the VT is initially located at point P_0. When the VT moves along the line from P_0, the Dubins distance $D(d)$ from the configuration to the VT will increase. Besides, the derivative of $D(d)$ with respect to d will also increase, and $\lim_{d\to\infty} D'(d)=1$.

Corollary 1: If a line intersects with C_R or C_L, the point on the line corresponding to DPCL does not fall within C_R or C_L.

The proofs of **Lemma 1**, **Theorem 1** and **Corollary 1** are presented in Appendix.

3.3 Analysis of Eq. (5)

3.3.1 Case 1: Virtual target path (VTP) does not intersect with C_R or C_L

Remark 2: Let $T(d)=T_1(d)-T_2(d)$. $T(d)$ is continuous when the VTP does not intersect with C_R or C_L, and

$$\begin{cases} T(0)=T_1(0)-T_2(0)=\dfrac{D(0)}{v}-\dfrac{0}{v_w}>0 \\ \lim\limits_{d\to\infty} T(d)<0 \end{cases} \tag{6}$$

Theorem 2: The equation $T(d)=0$ has one and only one solution d^* using optimal Dubins path, and

$$\frac{T_1(0)}{(\frac{1}{v_w}+\frac{1}{v})}<d^*<\frac{T_1(0)}{(\frac{1}{v_w}-\frac{1}{v})} \tag{7}$$

It means that the UAV can only intercept with the VT at a unique position using optimal Dubins path.

The proof for **Theorems 2** is presented in Appendix.

3.3.2 Case 2: Virtual target path (VTP) intersects with C_R or C_L

Theorem 3: Denote d_c as the Euclidean distance from the initial position to the target, and it is assumed that $d_c\geq 2r(1+\pi\dfrac{v_w}{v})$. Denote d' as the distance that the VT moves when it intersects with C_R or C_L for the first time, the minimal solution d^* to the equation $T(d)=0$ satisfies

$$\frac{T_1(0)}{(\frac{1}{v_w}+\frac{1}{v})}<d^*\leq\min(d',\ \frac{T_1(0)}{(\frac{1}{v_w}-\frac{1}{v})}) \tag{8}$$

Proof: It is obvious that $d'\geq d_c-2r$, so $T_2(d')=\dfrac{d'}{v_w}\geq\dfrac{d_c-2r}{v_w}\geq\dfrac{2\pi r}{v}$. Note that the point $p(d')$ is on the

circle which means $T_1(d') \leq \frac{2\pi r}{v}$. According to the conclusion of **Remark 2**, it can be known that the solution to the equation must lie within the interval $(0, d']$.

From the conclusion of **Theorem 2**, we have

$$\frac{T_1(0)}{(\frac{1}{v_w}+\frac{1}{v})} < d^* \leq \min(d', \frac{T_1(0)}{(\frac{1}{v_w}-\frac{1}{v})}) \tag{9}$$

Proof end.

Theorem 3 proves that if the Euclidean distance from the initial position to the target satisfies $d_c \geq 2r(1+\pi\frac{v_w}{v})$, then the UAV can intercept the VT before it intersects with C_R or C_L. Otherwise, the circumstance will be more complicated. On the basis of the conclusions of Ref. [12], it is clear that traversing through the boundaries of C_R or C_L above X-axis of the VT will make $D(d)$ discontinuous at the point on the circle, which leads to the discontinuity of $T(d)$ and may result in that there is no solution to Eq. (5) using optimal Dubins paths. When the VTP intersects with C_R or C_L, it can be divided into some segments by the circles. We separately calculate the solution on the segments along the VTP until one is attained. Two typical cases in which using optimal Dubins path cannot generate feasible solutions to Eq. (5) are illustrated as follows.

Case A: As shown in Fig. 4(a), the red line represents the VTP and it intersects with C_R. When the VT moves from the interior of C_R to the exterior through the point P, the Dubins distance $D(d)$ is shortened suddenly. As shown in Fig. 4(b), it is apparent that there are no intersection points between $T_1(d)$ and $T_2(d)$. In this case, the UAV should fly along a non-optimal Dubins path to meet the VT. Next, we focus on the path generation which can ensure the existence of the solution to Eq. (5).

For the point T shown in Fig. 5(a), the optimal Dubins path pattern is RS while there exist countless non-optimal paths. In this work, two non-optimal paths RL_{big} and RL_{small} depicted in Fig. 5(a) are selected as candidates.

Theorem 4: When the case shown in Fig. 4(a) occurs, one of the two non-optimal paths RL_{big} and RL_{small} can make the UAV intercept the VT.

Proof: As shown in Fig. 5(b), the target position is denoted by the red dot, and the points P_1 and P_2 on the VTP satisfy $|\overline{TP_1}| = d_1$ and $|\overline{TP_2}| = d_2$, respectively. According to the relationship shown in Fig. 4(b), it is clear that $T_1(d_1)_{RL_{big}} > T_2(d_1)$ and $T_1(d_1)_{RL_{small}} < T_2(d_1)$. However, it can be seen that the two non-optimal paths RL_{small} and RL_{big} become the same at point P_2 which means $T_1(d_2)_{RL_{big}} = T_1(d_2)_{RL_{small}}$. So, there exists a $d^* \in$

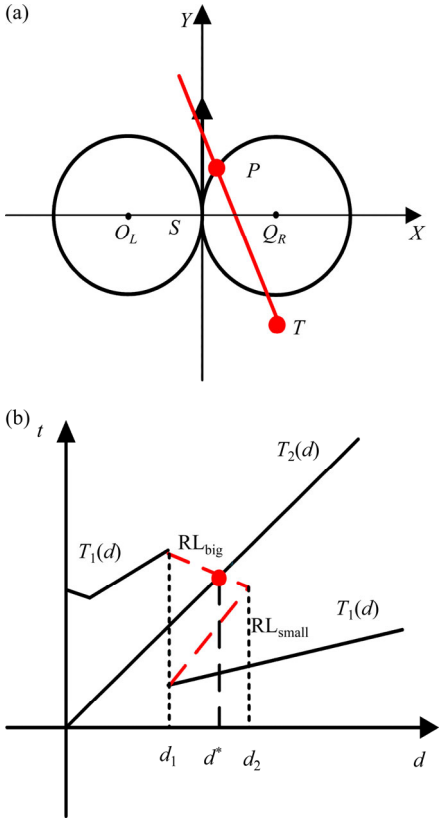

Fig. 4 Schematic diagram of Case A: (a) Scenario that VTP intersects with C_R (The red line represents VTP); (b) Relationship between $T_1(d)$ and $T_2(d)$ using optimal Dubins paths based on terminal heading relaxation in Case A

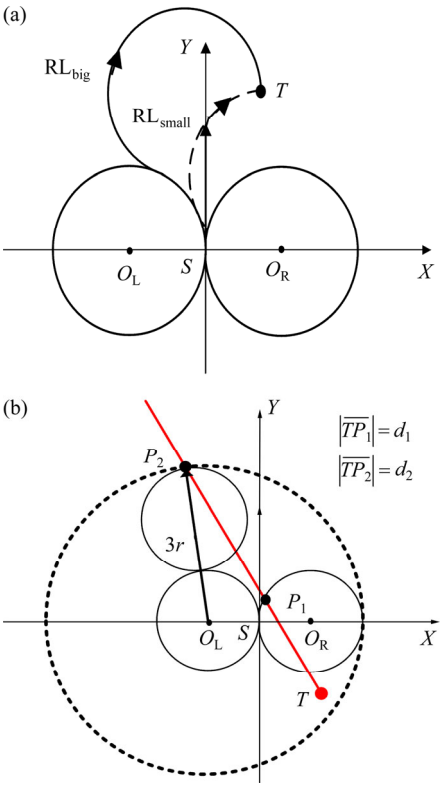

Fig. 5 Non-optimal Dubins paths in Case A: (a) Two non-optimal paths RL_{big} and RL_{small}; (b) Two points P_1 and P_2 satisfying $|\overline{TP_1}| = d_1$ and $|\overline{TP_2}| = d_2$

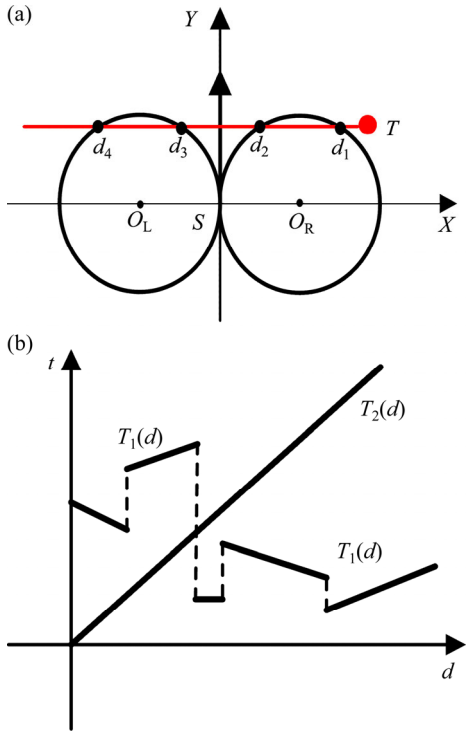

Fig. 6 Schematic diagram of Case B: (a) Scenario that VTP intersects with both C_R and C_L (The red line represents VTP); (b) Relationship between $T_1(d)$ and $T_2(d)$ using optimal Dubins paths based on terminal heading relaxation in Case B

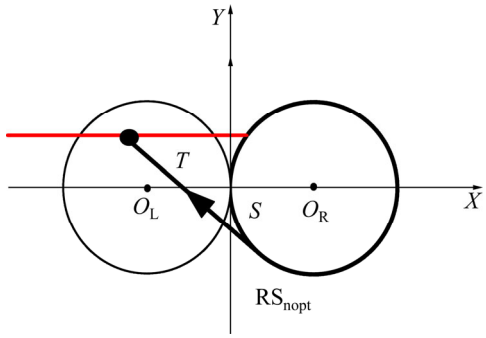

Fig. 7 Non-optimal Dubins paths in Case B

$(d_1, d_2]$ satisfying $T_1(d^*)_{RL_{big}\ or\ RL_{small}} = T_2(d^*)$. The process is presented by red dashed lines in Fig. 4(b).

Case B: If the VTP intersects with both C_R and C_L, then the VTP will be divided into five parts. As shown in Fig. 6(a), the intersection points between VTP and the turning circles are indicated by the distance that the VT moves from the initial target position. For the first three parts of the VTP, the method introduced above is adopted to calculate the solution. If there is no solution on the first three parts, we consider the fourth part. If $T_1(d) > T_2(d)$ holds for $d \in (d_3, d_4)$, the case is similar to case A. But if $T_1(d) < T_2(d)$ holds for $d \in (d_3, d_4)$ shown in Fig. 6(b), the UAV needs to fly along a longer path. For simplicity, we introduce another non-optimal Dubins path RS_{nopt} for the points satisfying $d>d_2$ which is shown in Fig. 7. According to the analysis in Section 3.3.1, it can be readily demonstrated that the UAV can intercept the VT by using this path.

3.4 Root-finding method

After determining the path pattern that the UAV intercepts the VT, a bisection algorithm is used to calculate the approximate numerical solution to the equation, as outlined in the following.

Given the initial solution interval $[a, b]$, the detailed steps of the bisection algorithm to calculate the solution are as follows:

Step 1: Find the midpoint $c=(a+b)/2$ and calculate the value of $T(c)$.

Step 2: If sign $(T(a)\cdot T(c))=1$ holds, the new solution interval changes to $[c, b]$, otherwise take $[a, c]$ as the new interval.

Step 3: Repeat steps 1 and 2 until the desired accuracy is derived.

Apparently, the accuracy of the solution depends on the number of iterations, which can reach $(b-a)/2^N$ after N iterations. According to the solution obtained by the bisection algorithm, the UAV can plan its path in wind which leads it to reach the target.

4 Optimization algorithm

4.1 Encoding

Random key is used to encode the visiting sequence of waypoints. A random key vector corresponds to a visiting order of waypoints [18]. The random key vector is sorted and the indexes of sorted elements in the original vector represent the visiting order in the tour. Now, a simple example is given to illustrate its principle. For a permutation problem with five points, a random key vector [0.5 0.2 0.4 0.7 0.6] is sorted in an ascending order as [0.2 0.4 0.5 0.6 0.7], the indexes of all elements in the original vector are 2, 3, 1, 5 and 4, respectively. So, the visiting order of the five points is 2-3-1-5-4.

4.2 Differential evolution algorithm

Differential evolution is a population based global optimization algorithm which shares the same framework with genetic algorithms [19]. Due to its high robustness and quick convergence speed, it has been widely used in scientific and engineering problems. Various mutation strategies such as DE/rand-to-best/1 and DE/best/2 have been developed to solve different problems. Now, the most successful DE variant, DE/rand/1, is described as follows.

It first randomly generates a population $\{\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D}) \mid i = 1, 2, \cdots, N_P\}$, where D is the dimension of the problem and N_P is the population size. The main operations include mutation, crossover and selection.

1) Mutation: For each target vector $\boldsymbol{x}_i^G$

($i = 1, 2, \cdots, N_P$, G is the current generation index), a mutant vector is generated as follows.

$$\boldsymbol{v}_i^G = \boldsymbol{x}_{r_1}^G + F \cdot (\boldsymbol{x}_{r_2}^G - \boldsymbol{x}_{r_3}^G) \tag{10}$$

where r_1, r_2 and r_3 are randomly chosen from the index set $\{1, 2, \cdots, N_P\}$ and different from i, and F is the scaling factor which is usually selected from the interval [0, 2].

2) Crossover: A crossover operation operates on the target vector and the mutant vector to form a trial vector $\boldsymbol{u}_i^G$.

$$\boldsymbol{u}_{i,j}^G = \begin{cases} \boldsymbol{v}_{i,j}^G, & \text{rand}_j(0, 1) \le C_r \ \text{ or } \ j = j_{\text{rand}} \\ \boldsymbol{x}_{i,j}^G, & \text{otherwise} \end{cases} \tag{11}$$

where C_r is the crossover probability within the interval (0, 1), rand_j is a random number generated within (0, 1), and j_{rand} is an integer randomly selected from [1, D].

3) Selection: The better individual between the target vector and the trial vector is selected to become a member of the next generation. For minimization problem, the selection operation is as follows.

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \boldsymbol{u}_i^G, & f(\boldsymbol{u}_i^G) \le f(\boldsymbol{x}_i^G) \\ \boldsymbol{x}_i^G, & \text{otherwise} \end{cases} \tag{12}$$

where $f(a)$ is the function to calculate the objective function value of individual a.

5 Computational experiments

The algorithms proposed were tested in Matlab environment on a PC with Intel(R) Core (TM) CPU i3-2120 3.3 GHz, and 2 GB RAM. Some typical experiments were conducted to verify the effectiveness and efficiency of the proposed path planning method. The parameter setting of the path planning problem and the DE algorithm used in the simulation are listed in Table 1.

Table 1 Parameter setting

Parameter	Value
Problem space/(m×m)	[0, 50]×[0, 50]
UAV speed/(m·s^{-1})	3
Minimal turning radius/m	5
Initial position	[0, 0]
Initial heading	$\pi/2$
N_P	$5D$
F	0.9
C_r	0.5

5.1 Dubins path between two points in wind

Four typical examples of the Dubins paths in wind are plotted in Fig. 8. The red lines indicate the paths of UAVs in moving air frame while the blue ones indicate the paths in the ground frame. As shown in Fig. 8(a), the UAV intercepts the VT by using optimal Dubins path labeled by the red line. Under the condition shown in Fig. 8(b), the VTP intersects with C_R, and there is no solution to Eq. (5) using optimal Dubins paths. As illustrated in Case A, the non-optimal path LR_{big} makes the UAV intercept the VT. Under the condition shown in Fig. 8(c), the VTP intersects with both two turning circles and the UAV flies along the optimal path RL to intercept the VT. Under the condition shown in Fig. 8 (d), the UAV intercepts the VT by using non-optimal path RS_{nopt}. It can be seen that the paths relative to air frame (air path) satisfy the minimum turning radius constraint while the paths relative to ground frame (ground path) do not.

From the analysis above, it is known that it is more complex when the VTP intersects with the turning circles. The path planning algorithm was executed for 10000 times with randomly selected targets in the problem space and wind fields satisfying v_w<3 m/s. The statistical results are shown in Table 2. It can be seen that the mean time consumption (MTC) of case 1 is minimal, and the time consumption of the method greatly depends on the allowed error of the bisection algorithm. The smaller the allowable error is, the more time is needed. Under all the conditions, the time consumed for the path planning of the UAV is very short, which makes the method very suitable for online path planning.

As shown in Fig. 9, five different paths under different wind fields for the same target were planned, and the time that the UAVs needed to reach the target was also calculated. It can be seen that the paths and time consumption under different wind fields are very different, which results in the difference in the planned tours for the same DTSP under different wind fields.

5.2 Dubins path for DTSP in wind

Based on the model that the UAV flies from an initial configuration to a target in wind, the DTSP considering wind field can be solved through optimizing the visiting sequence of waypoints by DE. Figure 10 shows the full air paths and ground paths for a simple MTST under two different wind fields. For both wind fields, the air paths shift and twist along the opposite direction of winds. The wind fields exhibit great impact on the UAVs' air paths.

Figure 11 exhibits the planned paths and their corresponding evolution plots in two more complex environments. There are 10 points to be visited in environment 1 and the DE converged by less than 100 generations while 15 points in environment 2 and the DE

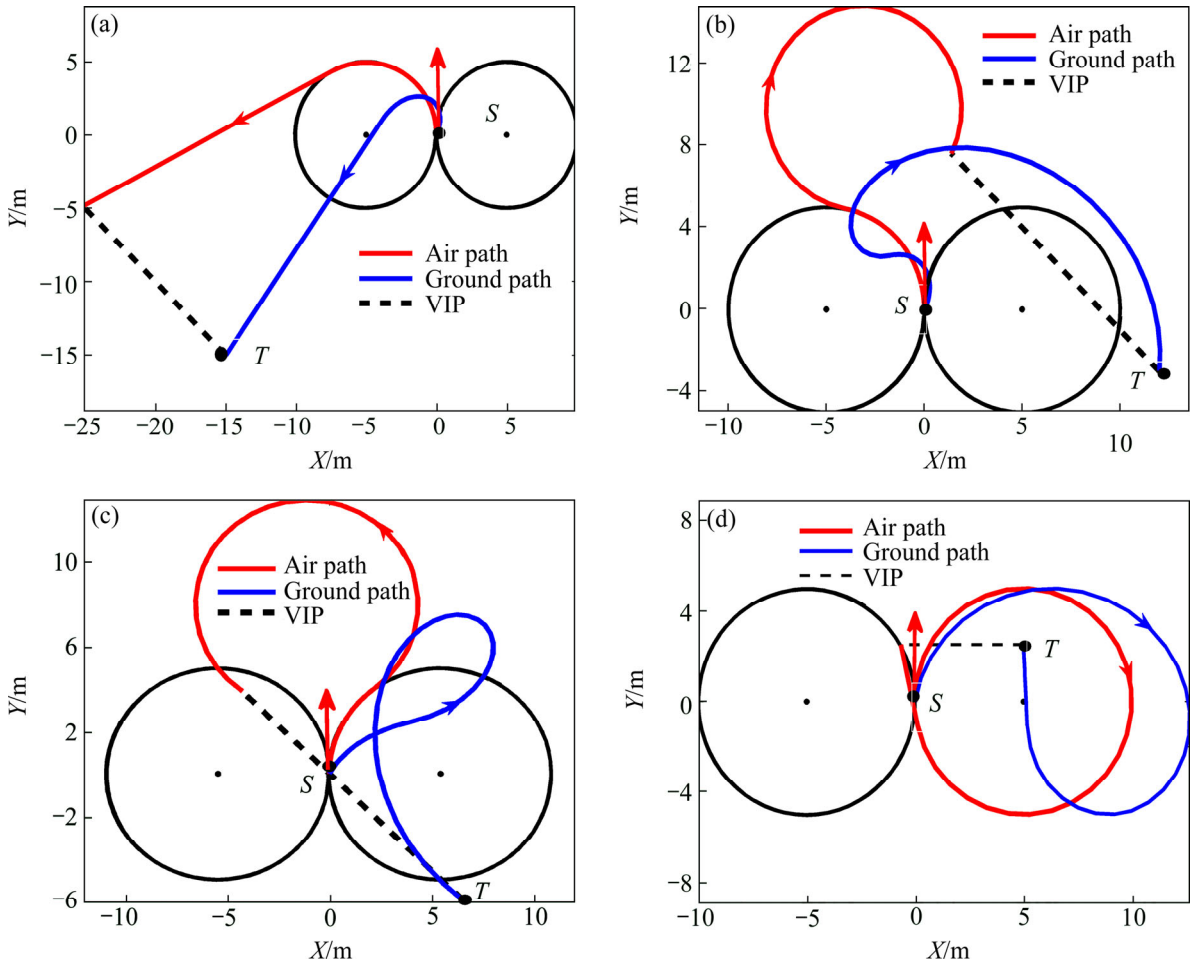

Fig. 8 Air paths and ground paths planned for four examples: (a) T=[−15 m,−15 m], v_w=[1 m/s, −1 m/s]; (b) T=[12 m, −3 m], v_w=[1 m/s, −1 m/s]; (c) T=[6 m, −6 m], v_w=[−1 m/s, 1 m/s]. (d) T=[5 m, 2.5 m], v_w= [1 m/s, 0 m/s]

Table 2 Statistical results based on Monto Carlo simulations

Case	Proportion/%	MTC (accuracy is 0.01)/ms	MTC (accuracy is 1)/ms
VTP does not intersect with turning circles	88.6	0.256	0.161
VTP intersects with one turning circle	8.8	0.417	0.315
VTP intersects with two turning circles	2.6	0.893	0.426

Note: Accuracy refers to accuracy of root obtained by bisection algorithm.

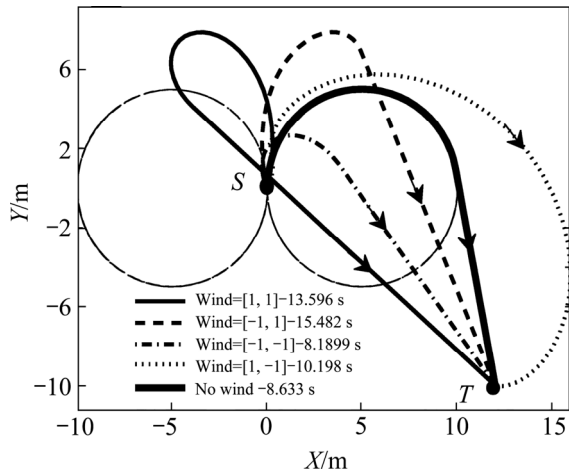

Fig. 9 Planned path under different wind fields for same target

converged by nearly 150 generations.

To verify the efficiency of the method above for DTSP, another two different algorithms for determining the visiting sequence of waypoints were also applied. One used the particle swarm optimization (PSO) to optimize the sequence and the other identified the sequence of waypoints by solving the corresponding ETSP. The comparative experiments were conducted on ten instances which are generated randomly in the problem space under different wind fields, and the three methods for determining the sequence of waypoints were executed 20 times with respect to each instance (the time that a UAV needs to finish the complete tour, called tour time, is recorded). For the instances with different scales,

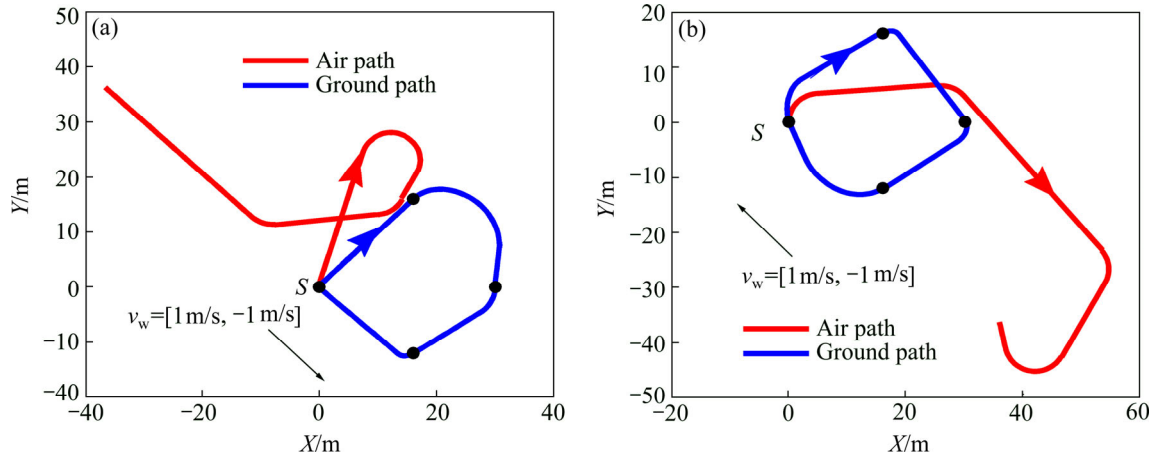

Fig. 10 Comparison of air paths and ground paths planned for MTST under different wind fields: (a) v_w=[1 −1]; (b) v_w=[−1 1]

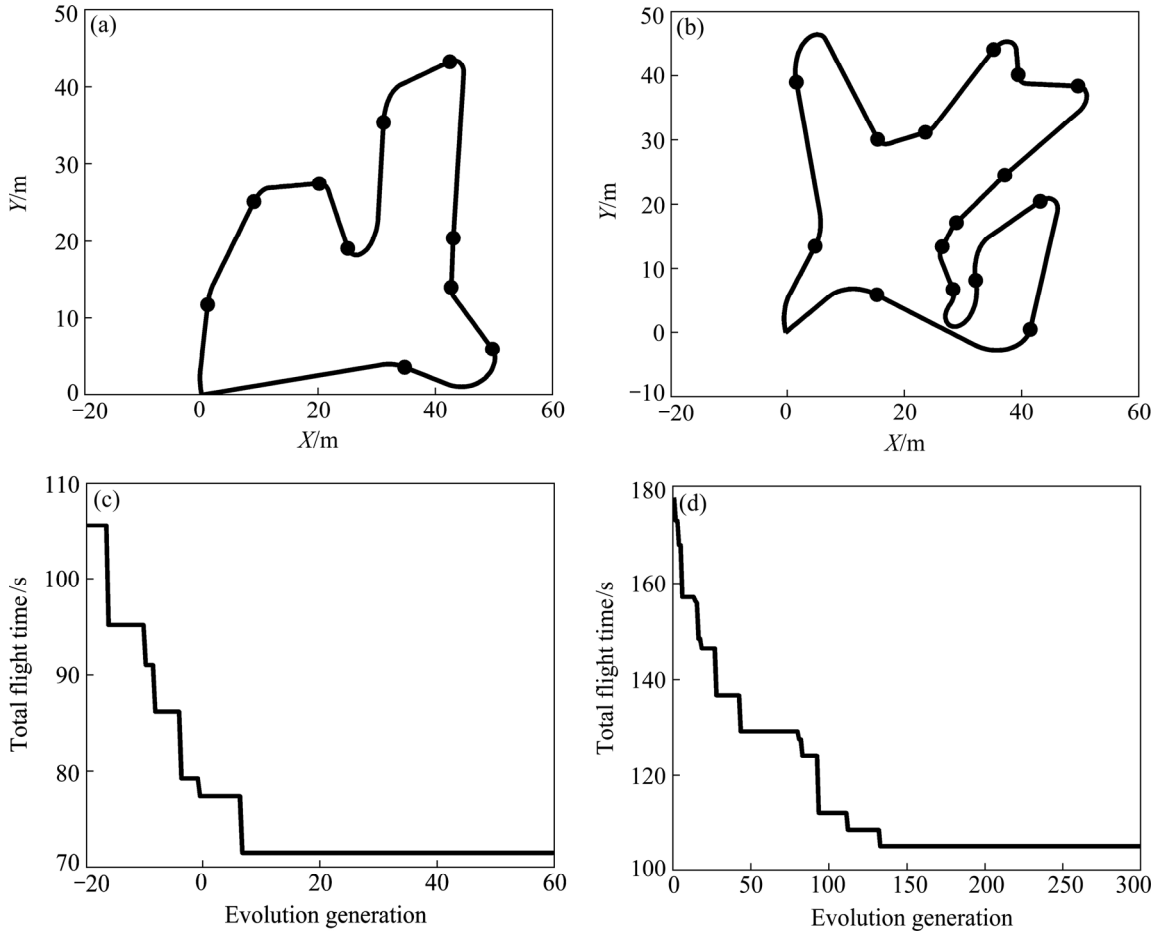

Fig. 11 Planned paths in two environments: (a) Planned paths in environment 1; (b) Planned paths in environment 2; (c) Convergence plots for environment 1; (d) Convergence plots for environment 2

different numbers of evolution generation (NEG) were used. The statistical results are presented in Table 3.

On the whole, the planned paths using the sequence generated by DE are better than that by PSO and by solving corresponding ETSP under the two wind fields. Though obtaining the sequence by solving ETSP consumes less time, its performance greatly relies on the difference between Dubins distance and Euclidean distance. When the distances between points are not big enough in comparison to the minimal turning radius of the UAVs, the generated tour whose visiting sequence is obtained by solving the corresponding ETSP may be bad. As listed in Table 3, the average tour time is used for comparison, and the average percentage of time reduction obtained by the DE based algorithm in comparison to the ETSP algorithm is approximately 20.5%.

With the same NEG, the average and standard

Table 3 Statistical results of 20 runs for ten randomly generated instances

Ins. No.	N	NEG	Wind	Algorithm			Algorithms comparison	
				ETSP/m	Random key-PSO (avg, std, max, min)/m	Random key-DE (avg, std, max, min)/m	Reduction -ETSP/%	Reduction -PSO/%
1	8	100	[1.2, −1.2]	93.3	(81.2, 8.4, 95.5, 72.5)	(73.0, 0.9, 74.7, 72.5)	21.8	10.1
			[0.2, −0.2]	77.4	(69.9, 7.3, 78.0, 59.4)	(60.1, 1.0, 61.4, 59.4)	22.4	14.0
2	9	100	[1.2, −1.2]	98.4	(96.9, 5.4, 107.5, 90.6)	(92. 4, 1.9, 96.5, 90.6)	6.1	4.6
			[0.2, −0.2]	77.5	(77.9, 4.5, 83.4, 69.4)	(74.9, 2.9, 79.9, 71.1)	3.4	3.9
3	12	200	[1.2, −1.2]	103.7	(111.8, 11.9, 130.3, 97.4)	(95.0, 1.3, 97.7, 94.0)	8.4	15.0
			[0.2, −0.2]	94.0	(94.5, 5.9, 107.5, 87.1)	(89.5, 1.9, 92.5, 86.9)	4.5	5.3
4	13	200	[1.2, −1.2]	118.8	(107.6, 7.0, 118.5, 92.9)	(89.1, 7.1, 100.1, 80.9)	25	17.2
			[0.2, −0.2]	104.0	(90.4, 8.1, 106.6, 81.9)	(86.4, 2.3, 89.8, 82.9)	16.7	4.4
5	13	200	[1.2, −1.2]	112.8	(94.6, 8.8, 108.0, 81.4)	(70.9, 3.4, 77.1, 68.7)	37.2	25.1
			[0.2, −0.2]	119.1	(85.9, 6.9 , 97.0, 73.4)	(72.8, 5.4, 84.5, 66. 9)	38.9	15.3
6	14	200	[1.2, −1.2]	120.4	(116.6, 8.3, 134.2, 107.3)	(103.2, 6.9, 114.2, 94.1)	14.3	11.5
			[0.2, −0.2]	120.4	(97.7, 10.8, 118.6, 82.8)	(83.8, 6.3, 94.8, 77.9)	30.4	14.2
7	15	300	[1.2, −1.2]	135.7	(110.7, 12.6, 127.1, 88.6)	(96.8, 6.1, 107.3, 91.5)	28.7	12.6
			[0.2, −0.2]	123.6	(96.6, 10.3, 116.1, 84.9)	(91.7, 5.6, 101.3, 84.5)	25.8	5.1
8	16	300	[1.2, −1.2]	114.3	(132.2, 8.5, 145.9, 114.4)	(98.6, 7.3, 108.8, 90.4)	13.7	25.4
			[0.2, −0.2]	120.3	(108.7, 6.6, 120.1, 100.6)	(100.5, 7.0, 114.1, 94.0)	16.5	7.5
9	19	300	[1.2, −1.2]	167.1	(150.3, 8.3, 162.9, 140.1)	(110.3, 9.4, 125.9, 99.7)	34.0	26.6
			[0.2, −0.2]	135.1	(124.9, 12.9, 144.4, 101.7)	(109.5, 9.4, 127.6, 98.5)	19.0	12.3
10	20	300	[1.2, −1.2]	161.0	(154.5, 15.0, 178.2, 131.2)	(127.0, 10.2, 141.1, 105.7)	21.1	17.8
			[0.2, −0.2]	159.3	(137.6, 8.9, 153.4, 122.9)	(124.1, 6.3, 132.9, 113.2)	22.1	9.8

Note: Avg, std, min, max: average, standard deviation, minimum, and maximum of the results in 20 runs; Reduction-ETSP, Reduction-PSO: the reduction of the average tour time obtained by DE based algorithm in comparison to the ETSP algorithm and the PSO based algorithm

deviations of the tour time optimized by DE are smaller than those optimized by PSO for most instances. DE obviously outperforms or at least performs comparatively against PSO in all instances except for instances 2, 4 and 7 in terms of the best result of 20 runs. Compared with PSO, DE shows higher robustness and better search ability for this kind of problems, and the average percentage of time reduction obtained by the DE based algorithm is approximately 12.9%.

In contrast to the other two methods, the DE based algorithm exhibits a significant improvement in tour time, and it is very suitable to optimize the visiting sequence of waypoints.

6 Conclusions

1) Taking into account the prohibitively computational complexity of finding optimal solutions to DTSP, the Dubins paths in the sense of terminal heading relaxation were proposed. So, only the visiting sequence of waypoints should be determined, and the optimization scale of the whole problem reduced from $2N$ to N.

2) By introducing the notion of the virtual target, the optimal Dubins path planning problem in wind from an initial configuration to a target was converted into an interception problem which was solved by finding the minimal root of a transcendental equation. Then, the DTSP in wind can be solved using the aforementioned solving frame for classic DTSP.

3) The computational experiments verified the efficiency of the model and showed the great impact of wind on UAVs' paths. Meanwhile, it also exhibited that the DE-based algorithm can generate better visiting sequence of waypoints than the other two algorithms. The proposed algorithm has remarkable advantages over its competitors in both the quality of planned Dubins tours in wind and the computation cost.

Appendix

1) Proof for **Lemma 1**

Proof: Without loss of generality, the line L shown in Fig. 12 is taken as an example. Three points A, B and T lying on the line and their corresponding optimal Dubins paths from the initial configuration are depicted. In addition, the straight portion of the Dubins path with respect to T is perpendicular to L. The Dubins distances

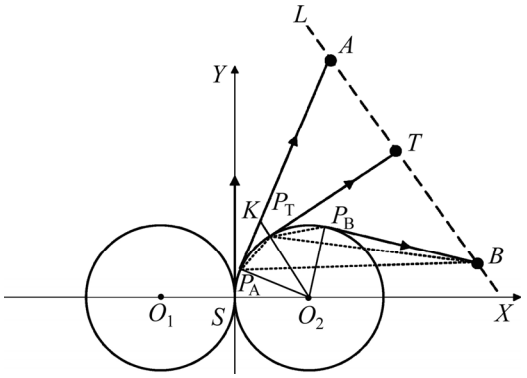

Fig. 12 Dubins path from a configuration to a line

from the initial configuration to A, B and T are $D(A)=\left|\overline{AP_A}\right|+\left|\widehat{SP_A}\right|$, $D(B)=\left|\overline{BP_B}\right|+\left|\widehat{SP_B}\right|$, $D(T)=\left|\overline{TP_T}\right|+\left|\widehat{SP_T}\right|$, respectively.

From Fig. 12, we have

$$D(A)-D(T)=\left|\overline{AP_A}\right|+\left|\widehat{SP_A}\right|-\left|\overline{TP_T}\right|-\left|\widehat{SP_T}\right|$$
$$=\left|\overline{AP_A}\right|-\left|\overline{TP_T}\right|-\left|\widehat{P_AP_T}\right|$$
$$=\left(\left|\overline{AK}\right|-\left|\overline{TP_T}\right|\right)+\left(\left|\overline{P_AK}\right|-\left|\widehat{P_AP_T}\right|\right)$$
$$=\left(\left|\overline{AK}\right|-\left|\overline{TP_T}\right|\right)+\left(R\cdot\tan(\angle P_AO_2K)-R\cdot\angle P_AO_2K\right)>0$$

Similarly, we also have

$$D(B)-D(T)=\left|\overline{BP_B}\right|+\left|\widehat{SP_B}\right|-\left|\overline{TP_T}\right|-\left|\widehat{SP_T}\right|$$
$$=\left|\overline{BP_B}\right|+\left|\widehat{P_TP_B}\right|-\left|\overline{P_TT}\right|>$$
$$\left|\overline{BP_B}\right|+\left|\overline{P_TP_B}\right|-\left|\overline{P_TT}\right|>\left|\overline{P_TB}\right|-\left|\overline{P_TT}\right|>0$$

So, T is the point corresponding to the shortest Dubins distance from the initial configuration to the line L.

Proof end.

2) Proof for **Theorem 1**

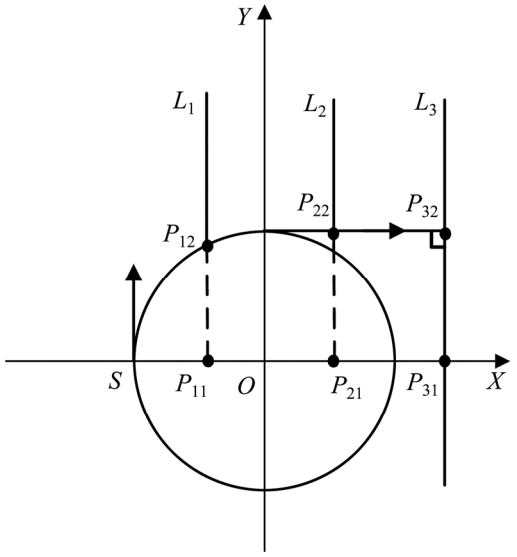

Fig. 13 Schematic of Dubins paths w.r.t. a point moving along a line

As shown in Fig. 13, the initial position is located at the point S, three lines L_1, L_2 and L_3 in parallel with the Y-axis are plotted which satisfy $0<\left|SP_{11}\right|\le r$, $r<\left|SP_{21}\right|\le 2r$ and $2r<\left|SP_{31}\right|$, respectively. We take the line L_3 as an example to prove the theorem. It can be seen from **Lemma 1** that the point corresponding to the DDCL for L_3 is P_{32}. The coordinates of a point on the line L_3 can be described as (x_0, y).

For the point on the line L_3 with $y>0$, the corresponding optimal Dubins distance can be calculated as follows.

$$l(y)=\sqrt{{x_0}^2+y^2-r^2}+r\left(\arccos\frac{-x_0}{\sqrt{{x_0}^2+y^2}}-\arccos\frac{r}{\sqrt{{x_0}^2+y^2}}\right)$$

Differentiating l with respect to y, we have

$$\frac{\mathrm{d}l}{\mathrm{d}y}=\frac{y\sqrt{{x_0}^2+y^2-r^2}-rx_0}{{x_0}^2+y^2}$$

The following conclusions can be drawn from the formula above.

$$\frac{\mathrm{d}l}{\mathrm{d}y}(y=r)=0,\ \lim_{y\to+\infty}\frac{\mathrm{d}l}{\mathrm{d}y}=1 \text{ and } \frac{\mathrm{d}^2l}{\mathrm{d}y^2}>0 \text{ for } y>0.$$

For the point on the line L_3 with $y\le 0$, the corresponding optimal Dubins distance is

$$l(y)=\sqrt{{x_0}^2+y^2-r^2}+r\left(2\pi-\arccos\frac{-x_0}{\sqrt{{x_0}^2+y^2}}-\arccos\frac{r}{\sqrt{{x_0}^2+y^2}}\right)$$

Differentiating l with respect to y, we have

$$\frac{\mathrm{d}l}{\mathrm{d}y}=\frac{y\sqrt{{x_0}^2+y^2-r^2}-rx_0}{{x_0}^2+y^2}$$

It can be seen that

$$\lim_{y\to-\infty}\frac{\mathrm{d}l}{\mathrm{d}y}=-1 \text{ and } \frac{\mathrm{d}^2l}{\mathrm{d}y^2}>0 \text{ for } y\le 0$$

From above, we can arrive at the conclusion that $\mathrm{d}l/\mathrm{d}y$ changes from −1 to 1 as y changes from $-\infty$ to $+\infty$, and $\mathrm{d}l/\mathrm{d}y=0$ holds when $y=r$. It means that when the VT moves along the line from P_{32} upwards or downwards, the Dubins distance $D(d)$ from the initial configuration to the VT will increase, $D'(d)$ will also increase and $\lim_{d\to\infty}D'(d)=1$.

For the lines L_1 and L_2 shown in Fig. 13, we study the parts that are outside C_R and the similar conclusions can be derived.

Proof end.

3) Proof for **Corollary 1**

Take the lines in the right half plane which are intersected with C_R as examples.

① Case a: There exists at least one intersection point above the X-axis.

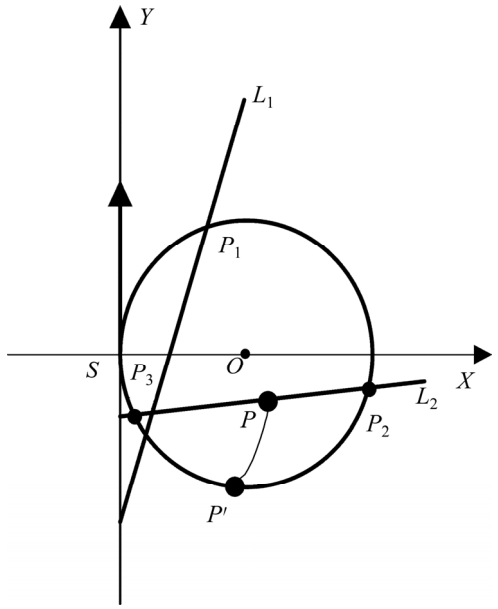

Fig. 14 Schematic provided for proof of Corollary 1

As shown in Fig. 14, the line L_1 intersects with C_R where one intersection point is above the X-axis and the other is not. It is evident that the Dubins distances from the initial configuration to the points inside C_R are bigger than πr, and the Dubins distance to the intersection point P_1 above the X-axis is smaller than πr. So the point corresponding to DDCL must not be inside the circle.

② Case b: Both the intersection points are below the X-axis.

As shown in Fig. 14, the line L_2 intersects with C_R and both intersection points are below the X-axis. From Ref. [12], it is known that the points inside C_R which have the same Dubins distances form a series of arcs of cardioids. So for any point P on the line which lies inside the circle，its corresponding Dubins distance must be the same with that of a point P' on the circle from P_2 to P_3 below the X-axis. So the Dubins distance with respect to P is longer than that of P_2.

In light of the conclusions of **Lemma 1**, it can be concluded that the point on a line corresponding to DDCL must be outside the turning circles or on the turning circles.

Proof end.

4) Proof for **Theorem 2**

It is known from above that

$$T(d)=T_1(d)-T_2(d)=\frac{D(d)}{v}-\frac{d}{v_w}$$

Differentiating T with respect to d, we have

$$T'(d)=\frac{1}{v}D'(d)-\frac{1}{v_w}$$

According to **Theorem 1**, it is known that $D'(d)<1$, so $T'(d)=\frac{1}{v}D'(d)-\frac{1}{v_w}<\frac{1}{v}-\frac{1}{v_w}<0$.

According to the conclusion of **Remark 2**, it can be concluded that the equation $T(d)=0$ has one and only one solution. Figure 15 plots two lines L_1 and L_2, they both go through the point $(0, D(0)/v)$ and their corresponding slopes are $1/v$ and $-1/v$, respectively. From **Theorem 1**, we know that $\frac{-d}{v}\le T_1(d)\le\frac{d}{v}$ which means that $T_1(d)$ must lie within the shaded region shown in Fig. 15. From Fig. 15, it can be readily calculated that $d_{\min}=T_1(0)/(\frac{1}{v_w}+\frac{1}{v})$ and $d_{\max}=T_1(0)/(\frac{1}{v_w}-\frac{1}{v})$.

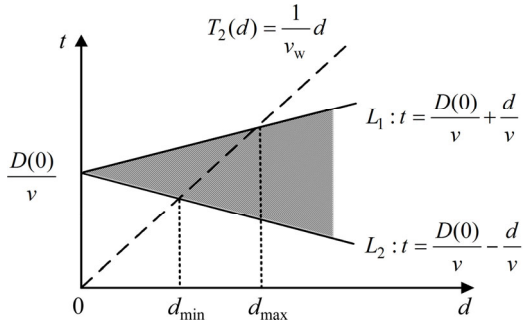

Fig. 15 Solution to equation under condition with VTP not intersect with C_R or C_L

Proof end.

References

[1] PENG Zhi-hong, WU Jin-ping, CHEN Jie. Three-dimensional multi-constraint route planning of unmanned aerial vehicle low-altitude penetration based on coevolutionary multi-agent genetic algorithm [J]. Journal of Central South University of Technology, 2011, 18(5): 1502−1508.

[2] ZHANG Xing, BAI Yong-qiang, CHEN Jie, XIN Bin. Differential evolution based receding horizon control for UAV motion planning in dynamic environments [J]. Robot, 2013, 35(1): 107−114.

[3] YU H L, Beard R W, ARGYLE M, CHAMBERLAIN C. Probabilistic path planning for cooperative target tracking using aerial and ground vehicles [C]// Proceedings of the American Control Conference. Piscataway, NJ, USA: IEEE, 2011: 4673−4678.

[4] DUBINS L E. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents [J]. American Journal of Mathematics, 1957, 79(3): 497−516.

[5] TANG Zhi-jun, ÖZGÜNER Ü. Motion planning for multiagent surveillance with mobile sensor agents [J]. IEEE Transactions on Robotics, 2005, 21(5): 898−908.

[6] MACHARET D G, ALVES NETO A, DA CAMARA NETO V F, CAMPOS M F M. Nonholonomic path planning optimization for Dubins' vehicle [C]// Proceedings of the IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2011: 4208–4213.

[7] SAVLA K, FRAZZOLI E, BULLO F. On the point-to-point and traveling salesperson problems for Dubins' vehicle [C]// Proceedings of the American Control Conference. Piscataway, NJ, USA: IEEE, 2005: 786−791.

[8] LE NY J, FERON E, FRAZZOLI E. On the Dubins travelling salesman problem [J]. IEEE Transactions on Automatic Control,

2012, 57(1): 265−270.

[9] HELSGAUN K. An effective implementation of the Lin-Kernighan traveling salesman heuristic [J]. European Journal of Operational Research, 2000, 126(1): 106–130.

[10] LE NY J. Performance optimization for unmanned vehicle systems [D]. Cambridge: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2008.

[11] YU X, HUNG J Y. A genetic algorithm for the dubins traveling salesman problem [C]// Proceedings of the IEEE International Symposium on Industrial Electronics. IEEE, 2012: 1256−1261.

[12] BOISSONNAT J D, BUI X N. Accessibility region for a car that only moves forward along optimal paths [R]. France: INRIA, 1994.

[13] RATHINAM S, SENGUPTA R, DARBHA S. A resource allocation algorithm for multivehicle systems with nonholonomic constraints [J]. IEEE Transactions on Automation Science and Engineering, 2007, 4(1): 98−104.

[14] MCNEELY R L, IYER R V. Tour Planning for an unmanned air vehicle under wind conditions [J]. Journal of Guidance, Navigation and Control, 2007, 30(5): 1299−1306.

[15] MCGEE T G, SPRY S, HEDRICK J K. Optimal path planning in a constant wind with a bounded turning rate [C]// Proceedings of AIAA Guidance, Navigation, and Control Conference. San Francisco: AIAA, 2005: 3456−3466.

[16] MCGEE T G, HEDRICK J K. Path planning and control for multiple point surveillance by an unmanned aircraft in wind [C]// Proceedings of the American Control Conference. Piscataway, NJ, USA: IEEE, 2006: 4261−4266.

[17] TECHY L, WOOLSEY C A. Minimum-time path planning for unmanned aerial vehicles in steady uniform winds [J]. Journal of Guidance, Control and Dynamics, 2009, 32(6): 1736−1746.

[18] MENDES J J M, GONÇALVES J F, RESENDE M G C. A random key based genetic algorithm for the resource constrained project scheduling problem [J]. Computers & Operations Research, 2009, 36(1): 92−109.

[19] DAS S, SUGANTHAN P N. Differential evolution: A survey of the state-of-the-art [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4−31.

(Edited by DENG Lü-xiang)