Chinese Journal of Aeronautics, (2014),**27**(3): 622–633

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

A memetic algorithm for path planning of curvature-constrained UAVs performing surveillance of multiple ground targets

Zhang Xing, Chen Jie, Xin Bin*, Peng Zhihong

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

Received 6 May 2013; revised 1 July 2013; accepted 9 September 2013
Available online 2 May 2014

KEYWORDS

Approximate gradient;
Dubins traveling salesman problem with neighborhood;
Local search;
Memetic algorithm;
Unmanned aerial vehicles

Abstract The problem of generating optimal paths for curvature-constrained unmanned aerial vehicles (UAVs) performing surveillance of multiple ground targets is addressed in this paper. UAVs are modeled as Dubins vehicles so that the constraints of UAVs' minimal turning radius can be taken into account. In view of the effective surveillance range of the sensors equipped on UAVs, the problem is formulated as a Dubins traveling salesman problem with neighborhood (DTSPN). Considering its prohibitively high computational complexity, the Dubins paths in the sense of terminal heading relaxation are introduced to simplify the calculation of the Dubins distance, and a boundary-based encoding scheme is proposed to determine the visiting point of every target neighborhood. Then, an evolutionary algorithm is used to derive the optimal Dubins tour. To further enhance the quality of the solutions, a local search strategy based on approximate gradient is employed to improve the visiting points of target neighborhoods. Finally, by a minor modification to the individual encoding, the algorithm is easily extended to deal with other two more sophisticated DTSPN variants (multi-UAV scenario and multiple groups of targets scenario). The performance of the algorithm is demonstrated through comparative experiments with other two state-of-the-art DTSPN algorithms identified in literature. Numerical simulations exhibit that the algorithm proposed in this paper can find high-quality solutions to the DTSPN with lower computational cost and produce significantly improved performance over the other algorithms.

1. Introduction

Autonomous unmanned vehicles, e.g., unmanned aerial vehicles (UAVs), are increasingly being used in both civilian and military fields due to their low cost, high maneuverability, good survivability, and so on. Recently, there has been a growing interest in performing surveillance of multiple ground targets by UAVs. The primary goal of the path planning for UAVs is to fly

* Corresponding author. Tel.: +86 10 68912463.
E-mail addresses: zx218705@163.com (X. Zhang), chenjie@bit.edu.cn (J. Chen), brucebin@bit.edu.cn (B. Xin), peng@bit.edu.cn (Z. Peng).

Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER Production and hosting by Elsevier

1000-9361 © 2014 Production and hosting by Elsevier Ltd. on behalf of CSAA & BUAA.
http://dx.doi.org/10.1016/j.cja.2014.04.024

through all the targets in the shortest time. To make the planned paths for UAVs flyable, the Dubins vehicle model is used to approximate the dynamics of UAVs where the constraints of UAVs' minimal turning radius can be taken into account. Meanwhile, due to the effective observation scope of the sensors equipped on UAVs, they only need to pass through one point within a certain neighborhood of each target. In this paper, we focus on finding the minimum-length curvature-constrained closed path through a collection of regions in a 2D plane.

1.1. Related work

The path planning problem for UAVs has been studied for decades.[1–5] In this paper, we are concerned with the optimal path planning problem of UAVs monitoring multiple ground targets. The well-known traveling salesman problem (TSP)[6] is aimed at determining the shortest path for a salesman to visit a series of cities, which measures the distance between any two cities by the Euclidean metric. In contract, the traveling salesman problem with neighborhood (TSPN)[7] takes targets as regions instead of points. An interesting application of the TSPN is the data mule robots[8–11] which are widely used in wireless sensor networks (WSNs) to collect data from WSN nodes. Usually, the total download time and the transmission ranges of the sensor nodes are the main factors to consider in the data mule problem. In this paper, to make the planned paths for UAVs flyable, the Dubins model (Eq. (1)) is used to describe the dynamic of UAVs with the constraints of minimal turning radius.

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

where (x, y) and θ are the planar coordinates and the heading of the UAV, respectively, v is the speed of the UAV, r is the minimal turning radius, u is the control input, and a triplet (x, y, θ) is called as a configuration.

Dubins conducted some research on the model and drew conclusions on the shortest path between any two configurations. The shortest path from one configuration to another must be one of the six Dubins path patterns: RSL, LSR, RSR, LSL, RLR, and LRL,[12] where L means turning left with the minimal turning radius, R means turning right with the minimal turning radius, and S means moving along a straight line. From the conclusions, it can also be seen that the shortest Dubins path between two configurations relies on both their positions and their headings.

Many studies have been done on the Dubins traveling salesman problem (DTSP) which is another variant of the classic TSP for a Dubins vehicle. In contrast to the classic TSP, the Dubins distance between two points is used in the DTSP. It has been proved that the DTSP is NP-hard.[13] The existing methods to solve the DTSP can mainly be classified into two categories: decoupling methods and transformation methods. Decoupling methods determine visiting sequence and heading separately. Tang and Ozguner[14] determined the sequence by a heuristic method first, and then a gradient-based algorithm was used to optimize the heading of the UAV at the waypoints. Kenefic[15] determined the visiting sequence of waypoints by solving a corresponding Euclidean traveling salesman problem (ETSP), and then the particle swarm optimization (PSO) algorithm was used to optimize the heading of the UAV at these waypoints. Savla et al.[16] used the same method to determine the sequence and a method called alternating algorithm (AA) is used to construct the Dubins tour. The effectiveness of decoupling methods mainly relies on the similarity between the DTSP and its ETSP counterpart, which makes them unsuitable for the situations where the Euclidean distance between two points is not long enough as compared with the minimal turning radius of UAVs. Transformation methods[17–19] first sample the headings of waypoints, and then the DTSP is converted into an asymmetric traveling salesman problem (ATSP) by the noon-bean transformation which can be easily solved by some prevailing solvers, e.g., Lin-Kernighan heuristic LKH.[20] Transformation methods greatly depend on the sampling density of headings. To acquire a higher accuracy, they usually consume tremendous computational resources.

In real scenarios, considering the surveillance scope of the sensors equipped on a UAV, the UAV only needs to pass through a certain neighborhood of each target. The Dubins traveling salesman problem with neighborhood (DTSPN) is a more general variant of the DTSP which takes every target as a region. The literatures regarding the DTSPN are fairly limited. In contrast to the DTSP, the DTSPN is a more challenging problem. Obermeyer[21] used a genetic algorithm to solve the path planning problem for a UAV performing reconnaissance of static ground targets in terrain. Similarly, Guimaraes Macharet et al.[22] proposed a three-stage optimization process based on evolutionary algorithms to solve the DTSPN. Obermeyer et al.[23] proposed two sampling methods which extended the method used for the DTSP and transformed the problem into an ATSP. Isaacs et al.[24] developed a similar sampling method for scenarios with overlapped regions.

1.2. Main contributions

The main contributions of the paper are the following:

(1) Two approaches are proposed to predigest the solving difficulty of the DTSPN. Firstly, the Dubins paths in the sense of terminal heading relaxation are introduced, which makes the calculation of the Dubins distance easier. Secondly, based on the fact that a UAV must fly through the boundary of every target region, a boundary-based encoding scheme determining the visiting point of every target region is presented. These make the optimization scale of the DTSPN dropped from $4N$ to $2N$ (N is the number of targets), and the overall optimization difficulty is reduced dramatically.
(2) Based on the frame of memetic computing, a computationally efficient local search strategy based on approximate gradient with computational complexity $O(N)$ is developed. It can further improve the visiting point of each target region and enhance the quality of Dubins tours to a large extent. Meantime, a better tradeoff between exploration and exploitation in the solution space can be achieved by adjusting the executive strength of the local search strategy.
(3) The algorithm proposed for the DTSPN can be easily extended to deal with more complicated DTSPN variants with multiple UAVs and multiple groups of targets.

2. Problem formulation

As illustrated above, the neighborhoods for the targets have to be determined beforehand. The shape of the neighborhoods depends on the specific task requirement and the environment. In a surveillance task, the neighborhoods of the targets should be defined to fit the footprint of the sensors equipped on UAVs. In a data collection task, the neighborhoods can be specified as the effective communication range of the targets. Considering scenarios with obstacles (e.g., high buildings and mountains), the neighborhoods may be irregular polygons. For the sake of clarity, we specify the neighborhood of each target as a disk centered at the target in this paper.

Let $\{D_1, D_2, \ldots, D_N\}$ be a set of N disk-shaped regions that a UAV should visit, where $D_i = \{Z, |Z - O_i| \leqslant R_i,\ R_i$ is the radius of the ith target region$\}$ is associated with the ith region centered at the target O_i. The optimization model of the DTSPN can be formulated as follows:

$$\min_{\boldsymbol{S},\boldsymbol{P},\boldsymbol{H}} J(\boldsymbol{S},\boldsymbol{P},\boldsymbol{H}) = d((p_0,h_0),(p_{s_1},h_{s_1})) + \sum_{i=1}^{N-1} d((p_{s_i},h_{s_i}),(p_{s_{i+1}},h_{s_{i+1}})) + d((p_{s_N},h_{s_N}),(p_0,h_0)) \tag{2}$$

where $\boldsymbol{S} = [s_1, s_2, \ldots, s_N]$ is the visiting sequence of the N target regions, $\boldsymbol{P} = [p_1, p_2, \ldots, p_N]$, $p_i = (x_i, y_i) \in D_i$, and $\boldsymbol{H} = [h_1, h_2, \ldots, h_N]$, $h_i \in [0, 2\pi)$. p_i is the visiting point corresponding to the ith target region, and h_i is the heading of the UAV at point p_i. p_0 and h_0 are the initial position and the initial heading of the UAV. $d((p_{s_i},h_{s_i}),(p_{s_{i+1}},h_{s_{i+1}}))$ denotes the Dubins distance between the configurations (p_{s_i},h_{s_i}) and $(p_{s_{i+1}},h_{s_{i+1}})$.

To get a solution to the DTSPN, the visiting sequence ($\boldsymbol{S}$), the visiting points ($\boldsymbol{P}$), and the headings ($\boldsymbol{H}$) of the UAV at these points should be determined. The optimization scale of the problem is $4N$. Besides, $\boldsymbol{S}$ is a combinatorial variable while $\boldsymbol{P}$ and $\boldsymbol{H}$ are continuous variables. The simultaneous optimization of hybrid variables is a great challenge in the DTPSN.

3. Memetic algorithm for DTSPN

Through the above-mentioned analysis, it is known that finding the optimal solution to the problem is computationally prohibitive. In this paper, Dubins paths in the sense of terminal heading relaxation are introduced to simplify the calculation of the Dubins distance, and a boundary-based encoding method is proposed to reduce the optimization scale of the problem. Then, a population-based evolutionary algorithm is presented. To achieve a desirable tradeoff between exploration and exploitation in the solution space,[25,26] a local search strategy based on approximate gradient is integrated into the evolutionary algorithm, constructing a so-called memetic algorithm.[27] The details are described as follows.

3.1. Dubins paths with terminal heading relaxation

Boissonnat and Bui[28] studied the accessible region for a Dubins vehicle in a given limited time and drew conclusions on the optimal Dubins paths with terminal heading relaxation. To be exact, the optimal Dubins paths with terminal heading relaxation can be taken as special cases of the Dubins paths with terminal heading constraint studied by Dubins.[12] As shown in Fig. 1, the initial position of the vehicle is located at the origin denoted by S, and the initial heading of the vehicle is along Y-axis. The target is denoted by T, and the centers of left and right turning circles corresponding to the initial configuration are denoted by O_L and O_R, respectively. Figs. 1(a)–(c) depict three typical Dubins paths corresponding to three points in the right-half plane. The paths shown in Figs. 1(a) and (b), path patterns of which are RS and LR, correspond to the two points inside and outside the right turning circle of the intial configuration, respectively. For the point on the right turning circle of the initial configuration shown in Fig. 1(c), the optimal Dubins path degrades to an arc. Specially, as for the point on the Y-axis shown in Fig. 1(d), the path lengths corresponding to the patterns RS and LS are the same. According to the symmetry of Dubins paths w.r.t. the Y-axis, similar conclusions hold for the points in the left-half plane.

It can be seen that, once the initial heading is fixed, the heading at the terminal point is determined by their relative position. In the sense of terminal heading relaxation, the calculation of Dubins paths can be remarkably simplified.

3.2. Encoding

Using the Dubins paths with terminal heading relaxation, only the visiting sequence and the visiting point of each target region should be determined. Based on the fact that the UAV must fly through the boundary of every target region, its boundary points are used to encode the visiting point. Any point on the boundary can be expressed by the polar angle w.r.t. its center. The schematic of the encoding scheme is described in Fig. 2.

Using the encoding scheme and the Dubins paths with terminal heading relaxation, a solution to the DTSPN can be described as $\begin{pmatrix} s_1 \\ \theta_{s_1} \end{pmatrix}$-$\begin{pmatrix} s_2 \\ \theta_{s_2} \end{pmatrix}$-$\cdots$-$\begin{pmatrix} s_N \\ \theta_{s_N} \end{pmatrix}$, where $\theta_{s_i} (i = 1, 2, \ldots, N)$ corresponds to the visiting point of the s_ith target region.

Remark 1. In real scenarios, surveillance or collecting information from a target cannot happen instantaneously. The target should lie within the effective range of the sensors equipped on the UAV for a period of time though it may be very short. In this case, an approximate method is used.

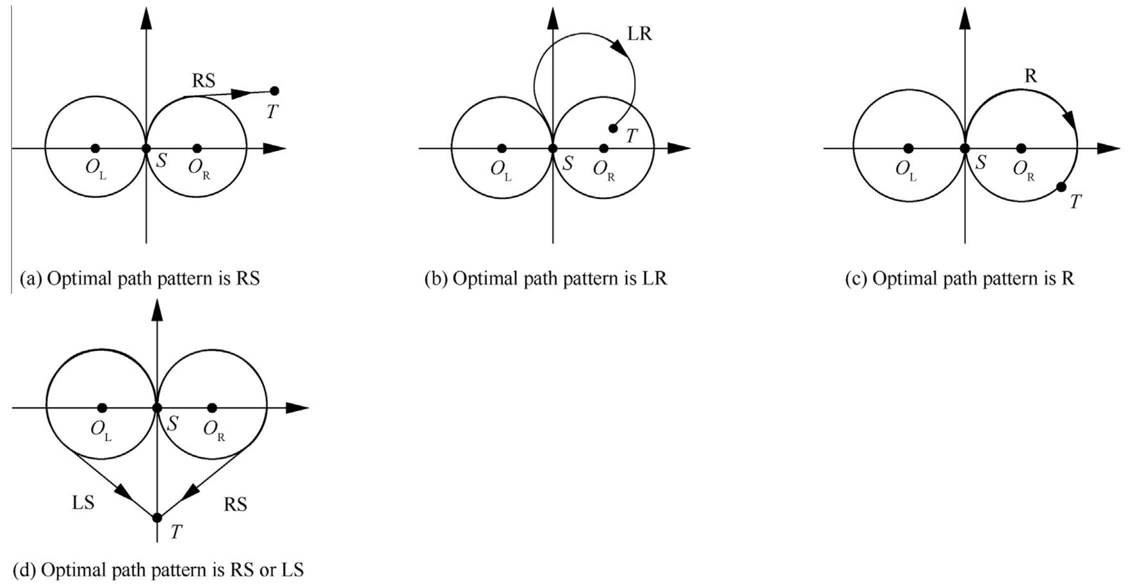
For convenience, it is assumed that the effective surveillance time should be longer than t_e. As shown in Fig. 3, the area within the dashed circle is the compressed neighborhood whose radius is r' while the area within the solid circle is the primitive neighborhood with radius R. The following relationship holds.

$$\frac{2(R - r')}{v} \geqslant t_\mathrm{e} \tag{3}$$

It can be noted that only if the UAV passes through one point within the compressed neighborhood, the effective surveillance time would be longer than t_e.

Therefore, the disk with radius $r' = R - \dfrac{vt_\mathrm{e}}{2}$ can be taken as the new neighborhood for planning approximately.

Remark 2. For irregular neighborhoods, a similar encoding scheme can be utilized. Take the scenario with a polygonal neighborhood shown in Fig. 4 as an example, the point on the



(a) Optimal path pattern is RS

(b) Optimal path pattern is LR

(c) Optimal path pattern is R

(d) Optimal path pattern is RS or LS

Fig. 1 Four typical Dubins paths with terminal heading relaxation.

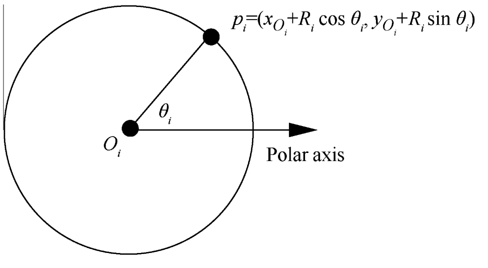

p_i=(x_{O_i}+$R_i$cos θ_i, y_{O_i}+$R_i$sin θ_i)
θ_i
O_i
Polar axis

Fig. 2 Schematic of the encoding scheme for the visiting point of a target region.

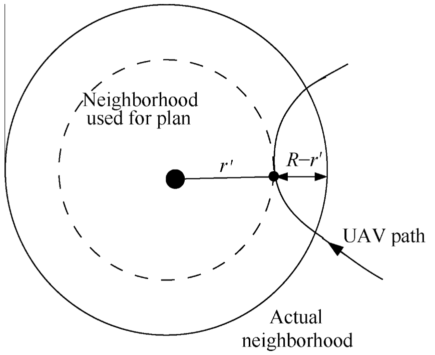


Fig. 3 Schematic of a compressed neighborhood.

boundary, $p(l)$, is encoded by its distance from $p(0)$ along the edges of the polygon.

Using the encoding scheme and the Dubins paths with terminal heading relaxation, the optimization model of the DTSPN can be simplified as:

$$\min_{\boldsymbol{S},\hat{\boldsymbol{P}}(\boldsymbol{\theta})} J(\boldsymbol{S},\hat{\boldsymbol{P}}) = \hat{d}((p_0, h_0), p_{s_1}) + \sum_{i=1}^{N-1} \hat{d}((\hat{p}_{s_i}, \hat{h}_{s_i}), \hat{p}_{s_{i+1}}) + \hat{d}((\hat{p}_{s_N}, \hat{h}_{s_N}), p_0) \tag{4}$$

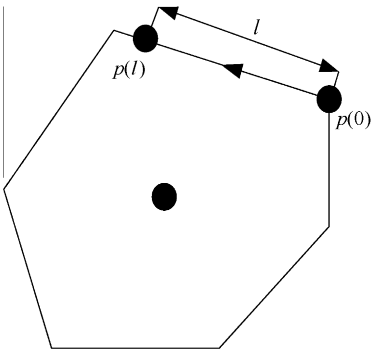


Fig. 4 Encoding for an irregular polygonal neighborhood.

where $\hat{d}((\hat{p}_{s_i}, \hat{h}_{s_i}), \hat{p}_{s_{i+1}})$ is the Dubins distance from the configuration $(\hat{p}_{s_i}, \hat{h}_{s_i})$ to the point $\hat{p}_{s_{i+1}}$ under terminal heading relaxation. $\hat{\boldsymbol{P}} = [\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_N]$, $\hat{p}_i = (R_i, \theta_i) \in B(D_i)$, and $B(D_i)$ denotes the boundary of the ith target region. Based on the Dubins paths with relaxed terminal heading and the encoding scheme, the optimization scale of the DTSPN is dropped from $4N$ to $2N$. What is noteworthy is that these approaches make the overall optimization difficulty of the DTSPN significantly reduced while the optimality of the generated solution is not lost obviously, which can be seen in Section 5.

3.3. Crossover operator

In this paper, an evolutionary algorithm is proposed to solve the DTSPN. Here, a crossover operator is developed to produce offspring. The basic process of the crossover operator is illustrated by an example with six targets as follows. Assume that the ith individual in the population is:

$$\begin{pmatrix} 2 \\ \theta_2^i \end{pmatrix} - \begin{pmatrix} 6 \\ \theta_6^i \end{pmatrix} - \begin{pmatrix} 1 \\ \theta_1^i \end{pmatrix} - \begin{pmatrix} 5 \\ \theta_5^i \end{pmatrix} - \begin{pmatrix} 3 \\ \theta_3^i \end{pmatrix} - \begin{pmatrix} 4 \\ \theta_4^i \end{pmatrix}$$

where the superscript i indicates the index of the individual in the population. The parent individuals are constituted by the *i*th individual and another randomly selected individual from the population.

$$\begin{pmatrix}4\\ \theta_4^j\end{pmatrix}\text{-}\begin{pmatrix}6\\ \theta_6^j\end{pmatrix}\text{-}\begin{pmatrix}2\\ \theta_2^j\end{pmatrix}\text{-}\begin{pmatrix}5\\ \theta_5^j\end{pmatrix}\text{-}\begin{pmatrix}3\\ \theta_3^j\end{pmatrix}\text{-}\begin{pmatrix}1\\ \theta_1^j\end{pmatrix}$$

Randomly generate an auxiliary vector having the same dimension as the parent individuals whose elements are 1 or 2. For example, given $\boldsymbol{v} = [1, 2, 2, 1, 1, 2]$, the value of the components in the auxiliary vector determines whose element in the two parent individuals will be selected as the element of the offspring. The first component in the auxiliary vector is 1, which means that the first gene in parent 1 will be selected to construct the offspring.

$$\text{Offspring}: \begin{pmatrix}2\\ \theta_2^i\end{pmatrix}\text{-}\cdots$$

Then, the selected gene will be deleted from both parent individuals. Removing the components corresponding to target numbered 2 from parents 1 and 2, then the parent individuals become

$$\text{Parent 1}: \begin{pmatrix}6\\ \theta_6^i\end{pmatrix}\text{-}\begin{pmatrix}1\\ \theta_1^i\end{pmatrix}\text{-}\begin{pmatrix}5\\ \theta_5^i\end{pmatrix}\text{-}\begin{pmatrix}3\\ \theta_3^i\end{pmatrix}\text{-}\begin{pmatrix}4\\ \theta_4^i\end{pmatrix}$$

$$\text{Parent 2}: \begin{pmatrix}4\\ \theta_4^j\end{pmatrix}\text{-}\begin{pmatrix}6\\ \theta_6^j\end{pmatrix}\text{-}\begin{pmatrix}5\\ \theta_5^j\end{pmatrix}\text{-}\begin{pmatrix}3\\ \theta_3^j\end{pmatrix}\text{-}\begin{pmatrix}1\\ \theta_1^j\end{pmatrix}$$

The second component in the auxiliary vector is 2, which means that the next gene in the offspring will be selected from parent 2.

$$\text{Offspring}: \begin{pmatrix}2\\ \theta_2^i\end{pmatrix}\text{-}\begin{pmatrix}4\\ \theta_4^j\end{pmatrix}\text{-}\cdots$$

Then, the target 4 is removed from both parent individuals, and the offspring individual will be constructed as the steps go on. Finally, the whole offspring individual is

$$\text{Offspring}: \begin{pmatrix}2\\ \theta_2^i\end{pmatrix}\text{-}\begin{pmatrix}4\\ \theta_4^j\end{pmatrix}\text{-}\begin{pmatrix}6\\ \theta_6^j\end{pmatrix}\text{-}\begin{pmatrix}1\\ \theta_1^i\end{pmatrix}\text{-}\begin{pmatrix}5\\ \theta_5^i\end{pmatrix}\text{-}\begin{pmatrix}3\\ \theta_3^j\end{pmatrix}$$

The mechanism of the crossover operator ensures that the generated offspring can inherit the genes of its parents.

3.4. Mutation

In this paper, two mutation operators are provided. One is the swapping operator. Two indices $i, j \in \{1, 2, \ldots, N\}, i \neq j$ are chosen randomly, and then the corresponding genes $\begin{pmatrix}s_i\\ \theta_{s_i}\end{pmatrix}$ and $\begin{pmatrix}s_j\\ \theta_{s_j}\end{pmatrix}$ swap their positions in the chromosome. The other is the shift of visiting points. Randomly select a number $i \in \{1, 2, \ldots, N\}$, and then the polar angle corresponding to the visiting point of the *i*th target region is reset within the interval $(0, 2\pi]$.

It can be seen that there exists a coupling between the visiting sequence and the visiting points of all the target regions in the calculation of Dubins tours. To further enhance the quality of the solution obtained by the evolutionary algorithm and achieve a better tradeoff between exploration and exploitation in the solution space, an approximate gradient strategy is used to execute a local search to improve the visiting point of every target region.

3.5. An approximate gradient based local search to determine the visiting point of every target region

Given two solutions $\begin{pmatrix}\boldsymbol{S}\\ \boldsymbol{\theta}\end{pmatrix}: \begin{pmatrix}s_1\text{-}s_2\text{-}\cdots\text{-}s_i\text{-}\cdots\text{-}s_N\\ \hline \theta_{s_1}\text{-}\theta_{s_2}\text{-}\cdots\text{-}\theta_{s_i}\text{-}\cdots\text{-}\theta_{s_N}\end{pmatrix}$ and $\begin{pmatrix}\boldsymbol{S}\\ \boldsymbol{\theta}'\end{pmatrix}: \begin{pmatrix}s_1\text{-}s_2\text{-}\cdots\text{-}s_i\text{-}\cdots\text{-}s_N\\ \hline \theta_{s_1}\text{-}\theta_{s_2}\text{-}\cdots\text{-}(\theta_{s_i}+\Delta\theta)\text{-}\cdots\text{-}\theta_{s_N}\end{pmatrix}$ which have the same visiting sequence but different visiting points of the target region numbered s_i, denote by $J(\boldsymbol{S}, \boldsymbol{\theta})$ and $J(\boldsymbol{S}, \boldsymbol{\theta}')$ the corresponding lengths of their Dubins tours. The derivative of J w.r.t. θ_{s_i} can be approximated by forward difference.

$$\frac{\partial J}{\partial \theta_{s_i}} \approx \frac{J(\boldsymbol{S}, \boldsymbol{\theta}') - J(\boldsymbol{S}, \boldsymbol{\theta})}{\Delta\theta} \tag{5}$$

Strictly speaking, a change on the visiting point of a target region will affect all the subsequent Dubins paths in the tour. However, this effect weakens gradually as the distance between two targets in the visiting sequence increases. Therefore, to reduce the computational cost, only a portion of subsequent points in the sequence will be taken into account.

$$\frac{\partial J}{\partial \theta_{s_i}} \approx \frac{\hat{J}_{s_i,n}(\boldsymbol{S}, \boldsymbol{\theta}') - \hat{J}_{s_i,n}(\boldsymbol{S}, \boldsymbol{\theta})}{\Delta\theta} \tag{6}$$

where $\hat{J}_{s_i,n}(\boldsymbol{S}, \boldsymbol{\theta})$ is the total length of the Dubins paths of n subsequent points after the point p_{s_i} (w.r.t. θ_{s_i}) in the sequence $\boldsymbol{S}$. Using a bigger n means that more computation resources are needed, but higher accuracy can be achieved.

After deriving the approximate gradient at the point p_{s_i} (w.r.t. θ_{s_i}), the corresponding update equation for θ_{s_i} is

$$\theta_{s_i}(k+1) = \theta_{s_i}(k) - \rho\frac{\nabla \boldsymbol{J}(\theta_{s_i})}{\|\nabla \boldsymbol{J}(\theta_{s_i})\|} \tag{7}$$

where ρ is the step size and k the iteration number. The iteration goes on until the length of the Dubins paths corresponding to the solution no longer becomes shorter.

The pseudo-code of the memetic algorithm is presented in Algorithm 1.

Algorithm 1: The memetic algorithm for the DTPSN

```
initialize population
for individual $i$ = 1:NP
  compute the length of Dubins tour $J(X_i)$ ($J$ is the function to
  calculate the length of a Dubins tour)
end for
while the stopping criteria is not met do
  for generation $g$ = 1:$g_{\max}$
    for individual $i$ = 1:NP
      randomly choose an individual $\boldsymbol{Y}_i$ from the population
      randomly generate an auxiliary vector $\boldsymbol{v}_i$
      generate a new individual $\boldsymbol{C}_i$ by the crossover operator
      if $\delta < p_m$ ($p_m \in (0, 1)$ is the mutation probability and $\delta$ is a
      random number within (0, 1))
        perform the mutation operator on the individual $\boldsymbol{C}_i$ to get
        the offspring $\boldsymbol{C}_i'$
      end if
      if $J(\boldsymbol{C}_i') < J(\boldsymbol{X}_i)$
        $\boldsymbol{X}_i = \boldsymbol{C}_i'$
      end if
    end for
    if mod($g, k_l$) = 0 ($k_l$ is a constant)
      select $L_n$ best individuals and conduct local search on these
      selected individuals
    end if
  end for
end while
```

Remark 3. Regarding the local search, it is easy to see from Eq. (7) that the upper bound of the iteration number is $\left\lfloor \frac{2\pi}{\rho} \right\rfloor$, where $\lfloor \cdot \rfloor$ is the down-rounding function. The computational complexity of the local search strategy for an individual is $O\left(\left\lfloor \frac{2\pi}{\rho} \right\rfloor nN\right)$, or simply $O(N)$. In addition, the execution frequency of the local search strategy can be adjusted to achieve a better tradeoff between exploration and exploitation in the solution space.

4. Extensions

In the following, two extensions of the DTSPN are introduced. Modifications on the individual encoding scheme are made, and the algorithm proposed above is extended to deal with them.

4.1. Multi-UAV scenario

Sometimes, a single UAV is not competent for given tasks. For a surveillance task, a single UAV often consumes a large amount of time to finish information collection from all the targets especially for large-scale problems. Considering the requirement of real-time, the coordination of multiple UAVs to execute the task will be more efficient. Coordinating multiple UAVs to perform surveillance of multiple targets is essentially a combination of the problems of resource allocation and path planning.[29,30]

The path planning for a single UAV has been discussed above, so the key problem of the multi-UAV scenario is to assign targets to different UAVs effectively. Assume that there are $M(M < N)$ UAVs located at different positions that can be used, and then the total number of possibilities of assigning N targets to M UAVs is M^N. If any set of the targets assigned to a UAV cannot be empty, the total number of possibilities of assignments is[14]

$$S(N, M) = \sum_{i=0}^{M-1} (-1)^i C_M^i (M-i)^N$$

where $C_M^i = \frac{M!}{i!(M-i)!}$ is a binomial coefficient. It indicates that $S(N, M)$ increases exponentially as N and M increase. Denote by L_i $(i = 1, 2, \cdots, M)$ the Dubins tour length corresponding to the ith UAV. To pass through all the target regions as quickly as possible and minimize the total length of the Dubins tours, the optimization objective of the multi-UAV DTSPN can be defined as follows:

$$\min \quad J' = \alpha \max(L_i) + (1-\alpha)\sum_{i=1}^{M} L_i$$

where α is a weighting coefficient satisfying $0 \leqslant \alpha \leqslant 1$, max (L_i) is the maximal value of the tour lengths of all the UAVs, and $\sum_{i=1}^{M} L_i$ is the sum of the tour lengths of all the UAVs.

Different from the single-UAV scenario, in the multi-UAV scenario, the policy of assigning targets to UAVs should also be determined. On the basis of the algorithm for a single UAV, a small modification is done to the individual encoding. The encoding scheme for the multi-UAV scenario is as follows:

$$\begin{pmatrix} U_{s_1} \\ s_1 \\ \theta_{s_1} \end{pmatrix} \text{-} \begin{pmatrix} U_{s_2} \\ s_2 \\ \theta_{s_2} \end{pmatrix} \text{-} \cdots \text{-} \begin{pmatrix} U_{s_N} \\ s_N \\ \theta_{s_N} \end{pmatrix}$$

where $U_{s_i} (i = 1, 2, \ldots, N)$ is the UAV to which the s_ith target region is assigned. Then, the same memetic algorithm can be applied.

4.2. Multiple groups of targets scenario

The above study assumes that the targets on the ground are isolated even though they are closely located. With the growing requirements of a task, a certain number of unmanned ground vehicles (UGVs) often form a group to carry out a task. In the group, a connected network is formed, and the environment and task information are shared. In this condition, a group can be taken as an ensemble, and its information can be transmitted to a UAV only if the UAV enters the communication range of any UGV in the group. A simple example is given in Fig. 5, where four UGVs are grouped to perform a task. The black dots denote the UGVs and the circles circumscribe their communication ranges. The dashed line describes the UAV flight path. It can be noted that a partial path of the UAV lies within the communication range of UGV_4, and the information of the group can be transmitted to the UAV by UGV_4.

In this case, a similar encoding scheme is adopted. The difference is that the range of the angle encoding the boundary point lies within $(0, 2k\pi]$ instead of $(0, 2\pi]$, where k is the number of UGVs in the group. The same evolutionary algorithm is used to solve this kind of problem.

5. Computational experiments

To verify the efficiency of the algorithm proposed in this paper, it was tested in MATLAB environment on a PC with Intel® Core™ CPU i3-2120 3.3 GHz and 2 GB RAM. In the following experiments, the mutation probability is set as $p_\mathrm{m} = 0.1$,[21] and the value of ρ is empirically set as $\rho = \pi/10$.

5.1. Typical DTSPN examples

The population size is set as NP = $20N$. The planned tours by the algorithm proposed in this paper in four typical environments are depicted in Fig. 6, where S denotes the initial position of the UAV. There are ten target regions in the first two environments, while 18 target regions in the last two environments. Figs. 6(a) and (c) are examples with uniformly distributed targets regions, and Figs. 6(b) and (d) are examples with overlapped target regions. It can be seen that the algorithm

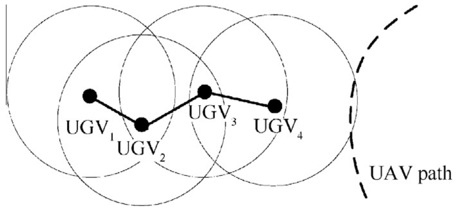
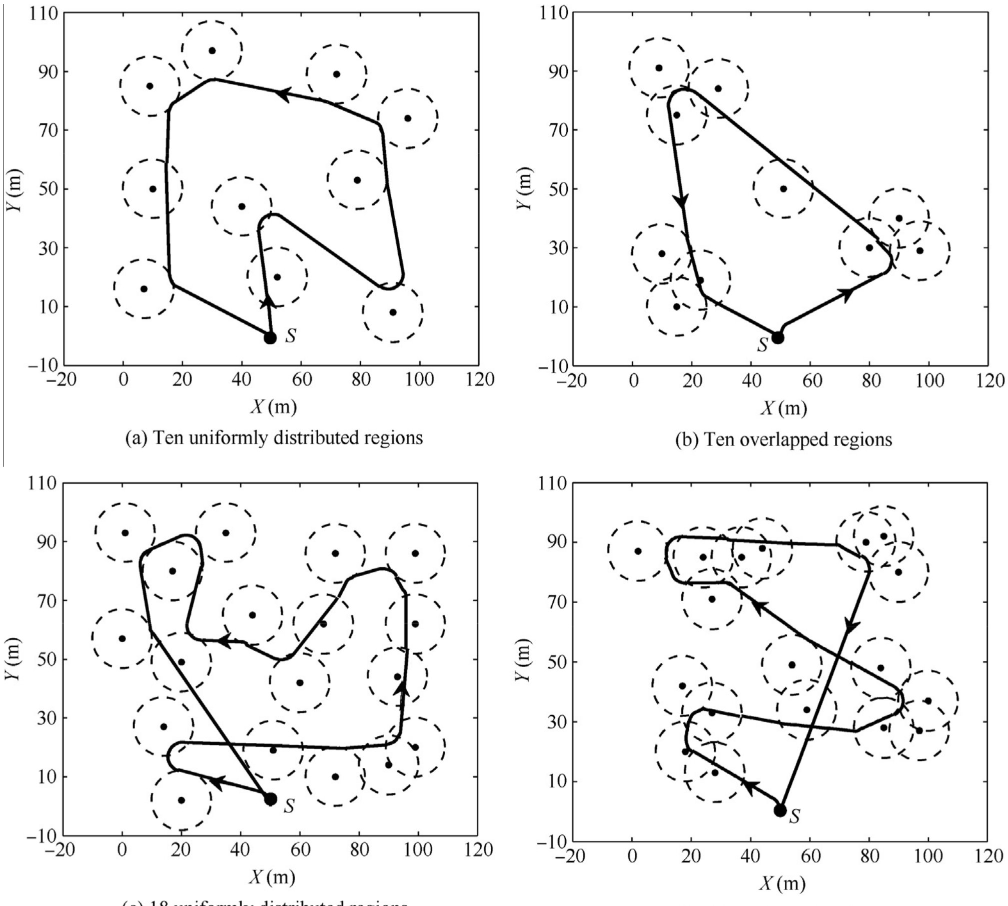


Fig. 5 Schematic of a UGV group.



(a) Ten uniformly distributed regions

(b) Ten overlapped regions

(c) 18 uniformly distributed regions

(d) 18 overlapped regions

Fig. 6 Planned tours in four typical environments with $r = 5$ m.

proposed in this paper generates feasible paths in all the four environments.

It is commonly known that the value of the minimal turning radius can make great impact on the Dubins paths. To show the effect of the minimal turning radius on the planned Dubins tours corresponding to the four environments, with different minimal turning radii, the algorithm was run 20 times and the average tour length was made a count. Fig. 7 shows the variation of the average length of Dubins tours w.r.t. the minimal turning radius in the four environments. It can be seen that, on the whole, the lengths of the planned Dubins tours grow as the minimal turning radius increases, except for the environment (a) when r changes from 6 m to 8 m.

5.2. Comparative experiments

To verify the efficiency of the algorithm proposed in this paper, other two state-of-the-art algorithms for the DTSPN (one was proposed by Obermeyer[21] (denoted by GA-com) and the other is a transformation method[23]) were applied to conduct the comparative experiments. For the transformation method, 30 entry points with random heading were sampled along the boundary of each target region. The comparative experiments were organized into two parts. The first part used 100 randomly generated instances with 10 target regions, and the second part tested the performance of the algorithm on different scale problems.

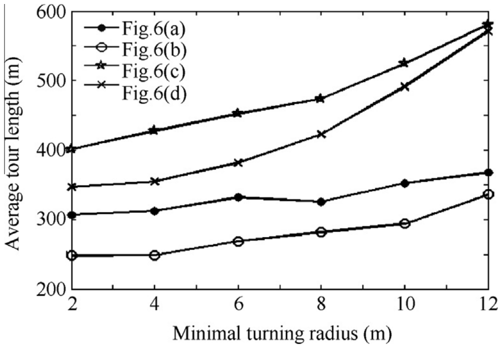


Fig. 7 Variation of the average tour length w.r.t. the minimal turning radius corresponding to the four environments shown in Fig. 6.

For the first part, for simplicity, it is assumed that the radii of all target regions are the same, and the minimal turning radius of the UAV is 3 m. 100 instances with 10 target regions were randomly generated within the square [0, 100] m × [0, 100] m. The three algorithms were applied to solve the 100 DTSPN instances, and the results were made a detailed comparison.

Fig. 8 presents the histograms of the tour length reduction obtained by the method proposed in this paper compared with that by GA-com. For simplicity, the algorithm proposed in this paper is denoted by MA-relax. For MA-relax and GA-com, the algorithm terminated with the same number of function evaluation (NFE = 15000). The average simulation times taken by MA-relax and GA-com on these instances are 8.4 s and 23.6 s, respectively. From the results, it can be clearly seen that MA-relax generates better Dubins tours than GA-com in all instances, and the time consumed by MA-relax is less than half of that by GA-com with the same NFE.

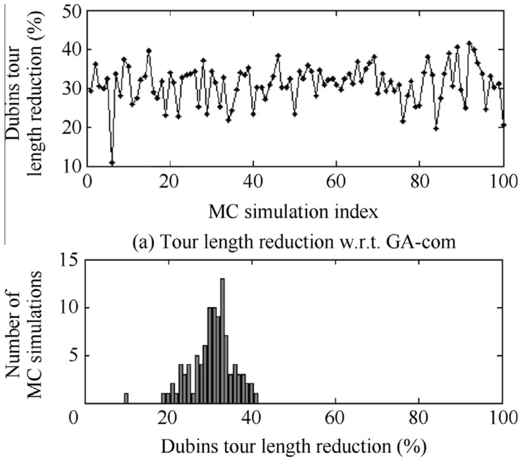
Fig. 9 presents the lengths of planned tours by both MA-relax and the transformation method for the 100 MC simulations. It can be seen that, in most cases, the lengths of their planned tours are very close. The average consumed time by the transformation method is 13.8 s.

It can be concluded that, in the simulation environments, MA-relax and the transformation method outperform GA-com, and the optimization results of the transformation method and MA-relax are very close while the time cost of MA-relax is 39% less than that of the transformation method on average.

In the second part, the environments with different numbers of targets were generated to analyze the performance of the algorithm on the problems with different scales. The vehicle's minimum turning radius and the radii of the target regions are randomly generated from given intervals: $r \in [1, 10]$ m and $R \in [5, 10]$ m. A total of eight instances were randomly generated to constitute a test suite. For every instance, all algorithms were executed 20 times, and the statistical results are presented in Table 1. In addition, to show the difference between the results planned by the three algorithms and the optimum values, the solutions found by the transformation method with adequate samplings along the boundary of all target regions are provided as the near-optimums for comparison.

From the statistical results, it can be seen that, on the whole, MA-relax is the best one as it can find the solutions closest to the near-optimums as compared with GA-com and the transformation method with lower computational costs in most cases. For all the instances, MA-relax outperforms GA-com in terms of both the length of the planned tours and the consumed time. For instances 1–5, 7, and 8, MA-relax outperforms the transformation method. For instances 6 and 7, MA-relax and the transformation method have similar time costs, and the transformation method outperforms MA-relax with minor advantages.

While summarizing the results of the experiments, it can be obviously observed that GA-com usually requires tremendous time to find a satisfactory solution due to its large amount of optimization variables. Therefore, with limited computation resources, it usually cannot derive satisfactory solutions. Supported by an efficient solving tool for the TSP, the sampling-based transformation method can quickly find the optimal solution in the sampling space. However, a large quantity of samplings usually consume a large amount of time, which is even unacceptable especially for real-time path planning. In contrast, though the method proposed in this paper loses optimality to some extent, it can greatly reduce the search space and improve the efficiency of solving the problem.



(a) Tour length reduction w.r.t. GA-com

(b) Tour length reduction histogram w.r.t. GA-com

Fig. 8 Dubins tour length reduction by MA-relax w.r.t. GA-com (100 Monte Carlo (MC) simulations).

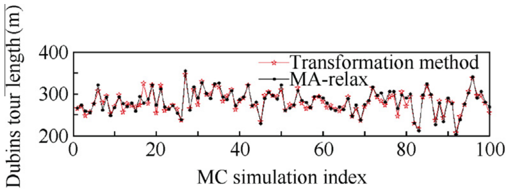


Fig. 9 Dubins tour lengths planned by MA-relax and the transformation method for the 100 MC simulations.

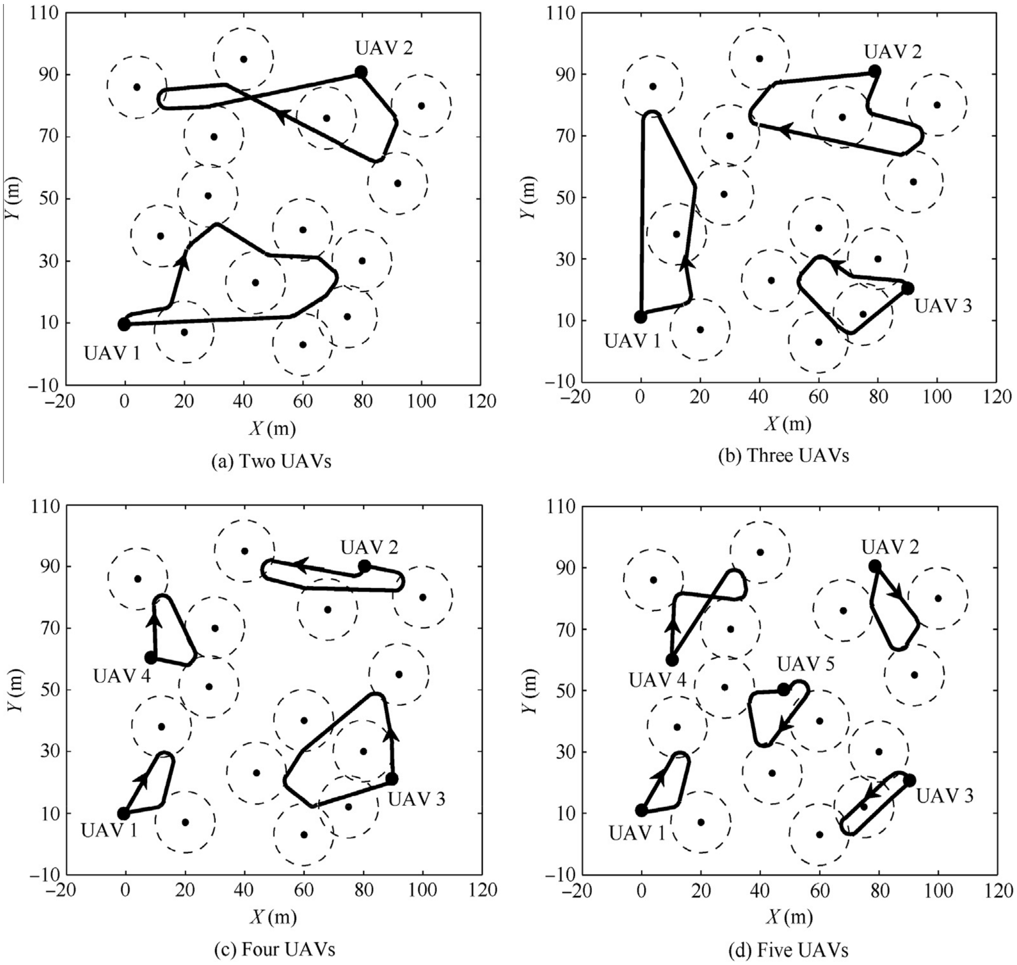
Table 1 Statistical results about the three algorithms in solving randomly generated instances.

Instance number	*N*	NFE	Near-optimum	Index	(avg, max, min, std)		
					MA-relax	GA-com	Transformation method
1	8	10000	276.0	*L* (m)	(**282.1**, **290.3**, **279.4**, 3.3)	(391.5, 408.4, 364.6, 13.8)	(286.8, 292.3, 283.4, **3.0**)
				t (s)	(**4.7**, **4.9**, **4.6**, **0.1**)	(11.4, 11.5, 11.2, 0.1)	(8.3, 8.4, 8.2, 0.1)
2	10	15000	298.7	*L* (m)	(**301.8**, **303.2**, **300.6**, **0.9**)	(444.6, 459.4, 432.7, 9.8)	(307.9, 314.2, 301.9, 3.7)
				t (s)	(**8.6**, **8.8**, **8.5**, **0.1**)	(21.5, 21.8, 21.3, 0.1)	(14.2, 14.5, 14.0, 0.2)
3	10	15000	252.2	*L* (m)	(**262.0**, 283.0, **255.3**, 8.5)	(462.3, 496.1, 428.9, 23.5)	(265.5, **270.2**, 262.6, **2.6**)
				t (s)	(**8.4**, **8.6**, **8.3**, **0.1**)	(21.1, 21.6, 20.8, 0.1)	(13.6, 13.9, 13.2, 0.2)
4	11	20000	291.0	*L* (m)	(**293.3**,**294.8**, **290.8**, **1.4**)	(425.3,449.4, 400.3, 13.7)	(301.3, 304.2, 299.5, 1.4)
				t (s)	(**13.0**, **14.9**, **12.3**, 0.8)	(32.3, 36.0, 30.6, 0.8)	(17.1, 17.2, 16.9, **0.1**)
5	12	25000	286.9	*L* (m)	(**293.7**, **299.4**, **289.1**, **3.5**)	(577.2, 614.0, 532.1, 28.5)	(304.3, 310.2, 294.1, 4.8)
				t (s)	(**16.9**, **18.5**, **16.3**, 0.7)	(42.5, 43.8, 41.6, 0.7)	(19.6, 19.8, 19.5, **0.1**)
6	14	35000	271.5	*L* (m)	(290.9, 300.7, **271.4**, 9.0)	(596.8, 619.4, 570.5, 16.9)	(**288.8**, **292.7**, 283.9, **2.7**)
				t (s)	(**26.5**, **26.9**, **26.1**, **0.2**)	(79.0, 79.2, 78.6, 0.2)	(28.9, 30.5, 28.1, 0.7)
7	15	40000	305.31	*L* (m)	(329.5, 372.0, **313.0**, 17.5)	(701.0, 722.3, 656.6, 18.3)	(**323.4**, **327.0**, 318.8, **3.2**)
				t (s)	(**32.2**, **33.8**, 31.5, **0.7**)	(81.5, 83.9, 80.7, 0.7)	(32.3, 34.4, **30.5**, 1.2)
8	17	50000	290.2	*L* (m)	(**292.7**, **302.5**, **288.5**, 4.4)	(754.8, 788.3, 699.4, 28.4)	(302.5, 304.1, 301.2, **1.0**)
				t (s)	(**44.9**, **45.4**, **44.4**, **0.3**)	(111.6, 111.9, 111.1, 0.3)	(46.2, 47.3, 44.6, 0.9)

Note: *L* is length of planned Dubins tour, and *t* is time cost; avg, std, min, max are average, standard deviation, minimum, and maximum of the results in 20 runs. In each case, the best result w.r.t. each index is highlighted in bold.

5.3. Multi-UAV scenario

As described in Section 4.1, the algorithm proposed for the single-UAV DTSPN can be extended to deal with the multi-UAV scenario. Fig. 10 shows the planned tours in an environment with different numbers of UAVs. The value of the weighting coefficient plays an important role in the generated solution. Here, the weighting coefficient is set as $\alpha = 0.7$. Meanwhile,



(a) Two UAVs

(b) Three UAVs

(c) Four UAVs

(d) Five UAVs

Fig. 10 Planned paths for scenarios with multiple UAVs in an environment ($r = 3$ m). For every scenario, the NFE is set to 50000.

considering the complexity of the multi-UAV scenario, the population size is set as NP = 30N.

From the results, it can be seen that the algorithm proposed in this paper can generate feasible paths for all the UAVs. The tour lengths of all the UAVs corresponding to the scenarios shown in Fig. 10 are recorded in Table 2, and the longest tour of all UAVs in every case is marked in bold. With the increase of the UAVs' number, the maximal tour length and the mean tour length of all UAVs become smaller. This is the advantage of the coordination by multiple UAVs.

5.4. Multiple groups of targets scenario

In this section, the condition described in Section 4.2 is tested by the proposed algorithm. The population size is set as NP = 30N. Fig. 11 plots two planned tours for two scenarios with multiple groups of targets. In every group, more than two targets are coordinated to perform a task, and the short solid lines denote the connectivity among them.

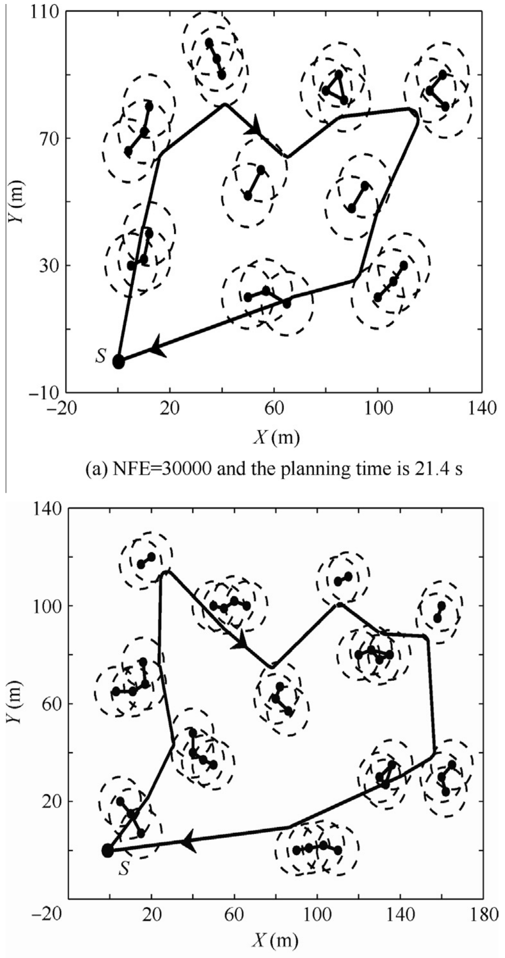
In every group, all the targets share the information. Different from the condition shown in Section 5.1, the UAV only needs to pass through one point within the neighborhood of any target in the group, and the information about the group can be transmitted to the UAV. It can be seen intuitively that using the algorithm proposed in this paper can generate feasible paths for the UAV in both cases.

From above, we can see that the algorithm proposed in this paper provides an effective and efficient tool to solve the DTSPN, and it can be easily extended to deal with more complicated DTSPN variants. Its efficiency mainly contains two

Table 2 Lengths of Dubins tours corresponding to the scenarios shown in Fig. 10.

Number of UAVs	Lengths of Dubins tours for all UAVs (m)	Mean length of all UAVs' tours (m)
2	(172.3, **186.9**)	179.6
3	(**154.7,** 144.8, 93.3)	130.9
4	(55.8, 104.2, **114.3,** 60.4)	83.7
5	(55.8, 64.1, 61.8, **91.0**, 64.8)	67.5

The longest tour of all UAVs in every case is marked in bold.



(a) NFE=30000 and the planning time is 21.4 s

(b) NFE=30000 and the planning time is 28.7 s

Fig. 11 Planned paths in two environments with multiple groups of targets (r = 3 m).

sides. Firstly, the employment of Dubins paths under terminal heading relaxation simplifies the calculation of the Dubins distance. It essentially compresses the search space of the original problem. Secondly, the boundary-based encoding scheme is also an approximation method which further compresses the search space of the visiting points of all target regions. In a word, we search for a suboptimal solution in a compressed space by the approximate approaches. From the simulations, it is clearly seen that, though these approaches may lose the optimality of the solution, they dramatically reduce the optimization scale of the whole problem and get feasible paths. For real applications, real-time performance is an important index especially in online planning. The developed algorithm outperforms GA-com and the sampling-based transformation method both in the planning time and the quality of the planned Dubins tours. At the same time, the method proposed in this paper can be easily extended to deal with scenarios with multiple UAVs and multiple groups of targets, and get satisfactory solutions.

6. Conclusions

In this paper, a memetic algorithm for the DTSPN is presented. By using the Dubins paths with terminal heading relaxation and the boundary-based encoding scheme, the optimization scale of the DTSPN decreases by half. To enhance the quality of planned Dubins tours, a local search strategy based on approximate gradient is used to improve the visiting point of every target region. Finally, the algorithm is easily extended to scenarios with multiple UAVs and multiple groups of targets. Numerical results demonstrate that the method proposed in this paper, compared with other two state-of-the-art algorithms, can generate better Dubins tours within shorter time, and it is also suitable to address the other two extended DTSPN scenarios.

Acknowledgements

The authors are grateful to the anonymous reviewers for their critical and constructive review of the manuscript. This study was co-supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61321002), the Projects of Major International (Regional) Joint Research Program NSFC (No. 61120106010), Beijing Education Committee Cooperation Building Foundation Project, the National Natural Science Foundation of China (No. 61304215), and Beijing Outstanding Ph.D. Program Mentor (No. 20131000704).

References

1. Dong ZN, Zhang RL, Chen ZJ, Zhou R. Study on UAV path planning approach based on fuzzy virtual force. *Chin J Aeronaut* 2010;**23**(3):341–50.
2. Wang YY, Wei TT, Qu XJ. Study of multi-objective fuzzy optimization for path planning. *Chin J Aeronaut* 2012;**25**(1):51–6.
3. Zhang X, Chen J, Xin B, Fang H. Online path planning for UAV using an improved differential evolution algorithm. In: *Proceedings of the 18th IFAC world congress 2011 Aug 28–September 2*. Milano, Italy. Australia: IFAC Secretariat; 2011. p. 6349–54.
4. Zhang X, Bai YQ, Xin B, Chen J. Differential evolution based receding horizon control for UAV motion planning in dynamic environments. *Robot* 2013;**35**(1):107–14.
5. Yu HL, Beard RW, Argyle M, Chamberlain C. Probabilistic path planning for cooperative target tracking using aerial and ground vehicles. In: *Proceedings of the American Control Conference 2011 Jun 29–July 1*. San Francisco, USA. Piscataway: IEEE; 2011. p. 4673–8.
6. Applegate DL, Bixby RE, Chvatal V, Cook WJ. *The traveling salesman problem: a computational study*. Princeton: Princeton University Press; 2011, p. 16–24.
7. Dumitrescu A, Mitchell JSB. Approximation algorithms for tsp with neighborhoods in the plane. *J Algorithms* 2003;**48**(1):135–59.
8. Tekdas O, Isler V, Lim JH, Terzis A. Using mobile robots to harvest data from sensor fields. *IEEE Wireless Commun* 2009;**16**(1):22–8.
9. Bhadauria D, Tekdas O, Isler V. Robotic data mules for collecting data over sparse sensor fields. *J Field Robot* 2011;**28**(3):388–404.
10. Isaacs JT, Venkateswaran S, Hespanha J, Madhow U, Burman J, Pham T. Multiple event localization in a sparse acoustic sensor network using UAVs as data mules. In: *Proceedings of the IEEE Globecom Workshops 2012 December 3–7; Anaheim, USA*. Piscataway: IEEE; 2012. p. 1562–7.
11. Lai YL, Jiang JR. A genetic algorithm for data mule path planning in wireless sensor networks. *Appl Math Inf Sci* 2013;**7**(1):413–9.
12. Dubins LE. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Am J Math* 1957;**79**(3):497–516.
13. Le Ny J, Feron E, Frazzoli E. On the Dubins traveling salesman problem. *IEEE Trans Autom Control* 2012;**57**(1):265–70.
14. Tang ZJ, Ozguner U. Motion planning for multitarget surveillance with mobile sensor agents. *IEEE Trans Robot* 2005;**21**(5):898–908.
15. Kenefic RJ. Finding good Dubins tours for UAVs using particle swarm optimization. *J Aeros Comput Inf Commun* 2008;**5**(2):47–56.
16. Savla K, Frazzoli E, Bullo F. On the point-to-point and traveling salesperson problems for Dubins' vehicle. In: *Proceedings of the 2005 American Control Conference; 2005 Jun 8–10; Portland, USA*. Piscataway: IEEE; 2005. p. 786–91.
17. Le Ny J, Feron E. An approximation algorithm for the curvature-constrained traveling salesman problem. In: *Proceedings of the 43rd Annual Allerton Conference on Communications, Control and Computing*; 2005. p. 620–9.
18. Le Ny J. *Performance optimization for unmanned vehicle systems dissertation*. Cambridge (MA): Massachusetts Institute of Technology; 2008.
19. Le Ny J, Frazzoli E, Feron E. The curvature-constrained traveling salesman problem for high point densities. In: *Proceedings of 46th IEEE Conference on Decision and Control; 2007 December 12–14; New Orleans, USA*. Piscataway: IEEE; 2007. p. 5985–90.
20. Helsgaun K. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *Eur J Operat Res* 2000;**126**(1):106–30.
21. Obermeyer KJ. Path planning for a UAV performing reconnaissance of static ground targets in terrain. 2009. Report No.: AIAA-2009-5888.
22. Guimaraes Macharet D, Alves Neto A, Fiuza da Camara Neto V, Montenegro Campos M. An evolutionary approach for the Dubins' traveling salesman problem with neighborhoods. In: *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference 2012 July 7–11; Philadelphia, USA*; 2012. p. 377–84.
23. Obermeyer KJ, Oberlin P, Darbha S. Sampling-based roadmap methods for a visual reconnaissance UAV. 2010. Report No.: AIAA-2010-7568.
24. Isaacs JT, Klein DJ, Hespanha JP. Algorithms for the traveling salesman problem with neighborhoods involving a Dubins vehicle. In: *Proceedings of American Control Conference; 2011 June 29–July 1; San Francisco, USA*. Piscataway: IEEE; 2011. p. 1704–9.

25. Chen J, Xin B, Peng ZH, Dou LH, Zhang J. Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization. *IEEE Trans Syst Man and Cybernet A Syst Humans* 2009;**39**(3):680–91.
26. Xin B, Chen J, Zhang J, Fang H, Peng ZH. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. *IEEE Trans Syst Man Cybernet C Appl Rev* 2012;**42**(5):744–67.
27. Chen J, Xin B, Peng ZH, Dou LH, Zhang J. Evolutionary decision-makings for the dynamic weapon-target assignment problem. *Sci China Ser F Inf Sci* 2009;**52**(11):2006–18.
28. Boissonnat JD, Bui XN. Accessibility region for a car that only moves forward along optimal paths. Research Report INRIA 2181; 1994.
29. Rathinam S, Sengupta R, Darbha S. A resource allocation algorithm for multivehicle systems with nonholonomic constraints. *IEEE Trans Automat Sci Eng* 2007;**4**(1):98–104.
30. Edison E, Shima T. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Comput Operat Res* 2011;**38**(1):340–56.

Zhang Xing received his Bachelor's degree in 2009 from University of Science and Technology Beijing. He is currently working towards his Ph.D. degree in the School of Automation, Beijing Institute of Technology, Beijing, China. His research interests include intelligent optimization, path planning, and decentralized coordination.

Chen Jie received his B.S., M.S., and Ph.D. degrees in control theory and control engineering in 1986, 1993, and 2000, respectively, from Beijing Institute of Technology, Beijing, China. He is currently a professor of control science and engineering at Beijing Institute of Technology. His main research interests include complex system multi-objective optimization and decision, intelligent control, constrained nonlinear control, and optimization methods.

Xin Bin received his B.S. degree in information engineering and the Ph.D. degree in control science and engineering, both from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively. He is currently a lecturer of control science and engineering at Beijing Institute of Technology. He has been an academic visitor in the Decision Sciences Research Center of Manchester Business School since February 2011. His research interests include search and optimization, intelligent decision-making methods, evolutionary computation, operations research, and combinatorial optimization.

Peng Zhihong received her Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2000. She is currently a professor in the Key Laboratory of Complex System Intelligent Control and Decision, School of Automation, Beijing Institute of Technology, Beijing, China. Her research interests include genetic algorithms, intelligent information processing, bioinformatics, and intelligent control.