A Memetic Algorithm for Curvature-Constrained Path Planning of Messenger UAV in Air-Ground Coordination

Yulong Ding, Bin Xin, *Member, IEEE*, Lihua Dou, *Member, IEEE*, Jie Chen, *Fellow, IEEE*, and Ben M. Chen, *Fellow, IEEE*

***Abstract*— This paper addresses a UAV path planning problem for a team of cooperating heterogeneous vehicles composed of one unmanned aerial vehicle (UAV) and multiple unmanned ground vehicles (UGVs). The UGVs are used as mobile actuators and scattered in a large area. To achieve multi-UGV communication and collaboration, the UAV, modeled as a Dubins vehicle, serves as a messenger to fly over the effective communication range of all UGVs to relay information. The curvature-constrained path planning of the messenger UAV is formulated as a Dubins Traveling Salesman Problem with Dynamic Neighborhood (DTSPDN) which is a complex optimization problem involving coupled variables and contains dynamic constraints. We design an effective memetic algorithm to find the shortest route that enables the messenger UAV to visit all moving UGVs. This algorithm combines the genetic algorithm procedure, two kinds of local search operators based on gradient search and uniform sampling respectively, and a gradient-based repair operator to repair the solutions violating dynamic constraints. During the evolutionary process, a special phenomenon may occur that changing some decision variables (i.e., visiting sequence and location) may not affect the evaluation function value, but may alter the feasible region of another decision variable (i.e., visiting time) due to the encounter constraint between the UAV and UGV. To track and utilize the change of the feasible region, a transformation procedure is proposed to change one solution to another with less visiting time by analyzing the encounter pattern between UAV and UGV. The computational results on random instances with different scales demonstrate that the proposed approach can effectively generate better curvature-constrained tours to encounter all moving UGVs when compared to other four competitive algorithms in the literature.**

***Note to Practitioners*—This paper studies an emerging path planning problem for a UAV which is used to provide communication service for multiple moving UGVs. These UGVs are required to execute tasks (e.g., firefighting, search and rescue) within a large area. Due to their limited communication capabilities, they may be unable to obtain necessary information from other UGVs. The UAV serves as a messenger to fly over the effective communication range of all moving UGVs to relay information. We propose a novel memetic algorithm to efficiently search for the shortest tour that enables the messenger UAV to visit all moving UGVs. The memetic algorithm combines the parallel global search virtue of genetic algorithm with efficient local search procedure to improve the generated tour. A gradient-based repair procedure is also employed to make sure that the planned tour can guide the UAV to sequentially encounter each moving UGV. Simulations exhibit that the proposed approach can effectively generate high-quality tours for messenger UAV to rapidly visit all UGVs, which assists UGVs to achieve collaboration in large area. In future work, the proposed memetic algorithm will be extended to plan tours for multiple messenger UAVs.**

***Index Terms*— Air-ground coordination, curvature-constrained path planning, memetic algorithm, Dubins traveling salesman problem.**

NOMENCLATURE

P^A	UAV's position (unit: m).
h^A	UAV heading angle (unit: rad).
P_0^A	UAV's initial position (unit: m).
h_0^A	UAV initial heading angle (unit: rad).
v^A	UAV flight speed (unit: m/s).
r_{min}	UAV's minimum turning radius (unit: m).
$D(\cdot)$	The length of Dubins path (unit: m).
$D_{min}(\cdot)$	The length of the shortest Dubins path (unit: m).
N_G	The number of UGVs.
I	Set of UGV IDs with $I = \{1, 2, \ldots, N_G\}$.
v_i^G	The speed of UGV i, $i \in I$ (unit: m/s).
v_{max}^G	The maximum speed of the UGV (unit: m/s).
$P_i^G(t)$	The location of UGV i, $i \in I$ at time t (unit: m).
$r_{i,com}^G$	The valid communication radius for UGV i, $i \in I$ (unit: m).

Manuscript received 13 November 2021; accepted 9 December 2021. Date of publication 22 December 2021; date of current version 13 October 2022. This article was recommended for publication by Associate Editor M. Franceschelli and Editor C. Seatzu upon evaluation of the reviewers' comments. This work was supported in part by the National Outstanding Youth Talents Support Program under Grant 61822304, in part by the Basic Science Center Programs of NSFC under Grant 62088101, in part by the National Postdoctoral Program for Innovative Talent under Grant BX20200175, in part by the China Postdoctoral Science Foundation under Grant 2020M682822, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Shanghai Municipal Commission of Science and Technology Project under Grant 19511132101, and in part by the National Natural Science Foundation of China under Grant 61873033. *(Corresponding author: Bin Xin.)*

Yulong Ding is with the Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen 518055, China.

Bin Xin and Lihua Dou are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, and also with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China (e-mail: brucebin@bit.edu.cn).

Jie Chen is with the Department of Control Science and Engineering, Tongji University, Shanghai 200092, China.

Ben M. Chen is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TASE.2021.3135044.

Digital Object Identifier 10.1109/TASE.2021.3135044

P_i The access location to visit the neighborhood of UGV i with $i \in I$ (unit: m).
h_i The UAV heading at P_i (unit: rad).
t_i^* The encounter time for UGV i, $i \in I$.
s_j The index of the j-th UGV.
S The visiting sequence about UGVs, $S = [s_1, s_2, \ldots, s_{N_G}]$.
Δt_{s_j} The time span for UAV to travel from UGV s_{j-1} to UGV s_j with $s_j \in I$ (unit: s).
$\underline{\Delta t_{s_j}}$ The lower bound of Δt_{s_j} (unit: s).

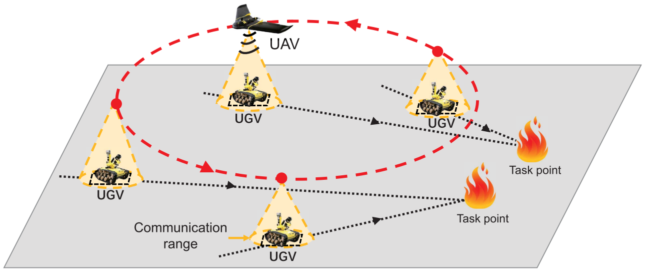

Fig. 1. A scenario for a UAGVS operating in a large area.

I. Introduction

COOPERATION between unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) provides a new breakthrough for the effective application of UAVs and UGVs. The strong complementarities between UAVs and UGVs in sensing, communication, and payload make a UAV-UGV coordination system (UAGVS) attractive [1]. UAGVS can be categorized into four classes according to the number of UGVs and UAVs in a UAGVS: one UAV with one UGV (see e.g., [2]–[4]), one UAV with many UGVs (see e.g., [5]), many UAVs with one UGV (see e.g., [6]), or many UAVs with many UGVs (see e.g., [7]). Chen *et al.* [1] reviewed a number of influential types of UAGVSs, and proposed a taxonomy for classifying existing UAGVSs. Our recent survey [8] systematically reviews advances in UAV-UGV coordination systems from 2015 to 2020, and offers a comprehensive investigation and analysis of recent research.

A typical UAGVS is shown in Fig. 1, where a UAV flies over multiple UGVs to provide communication services. The UGVs, as mobile actuators, are commonly deployed in application such as risk assessment [9], surveillance [10], and disaster relief [11]. When UGVs are required to execute tasks in a collective manner within a large area, they may be scattered and unable to obtain necessary information from other UGVs due to their limited communication capabilities. To address this issue, UAVs can be employed as a messenger to form a cooperative air-ground network [12]and to assist the UGVs in an environment where the communication infrastructure is not available and network connectivity is poor.

The speed and mobility of UAVs expands the effective communication range of the UGVs by collecting data and exchanging them with other UGVs or base stations beyond the transmitting ranges of the UGVs [13]. This messenger mechanism can broaden the UGV operating range effectively to enable multi-UGV collaboration in a large area.

In this paper, we investigate the path planning problem for the messenger UAV in the UAGVS mentioned above, which can be generally formulated as a *Dubins Traveling Salesman Problem with Dynamic Neighborhood (DTSPDN).*

A. Related Work

As mentioned above, the DTSPDN can be regarded as a generalized variant of the classic traveling salesman problem. If we fix each UGV to a certain local position, DTSPDN will be degenerated to the Dubins Traveling Salesman Problem with Neighborhood (DTSPN) described in [14], which is NP-hard [15].

The existing methods for solving DTSPN can be mainly grouped into four categories [16]: decoupling methods, transformation methods, unsupervised learning methods, and direct search methods. In the decoupling methods (see e.g., [17], [18]), the mixed variables, including the visiting sequence and the access location, are optimized separately. In the transformation methods (see e.g., [19], [20]), the UAV's positions and headings are uniformly sampled to construct a generalized traveling salesman problem (GTSP) firstly, and then the GTSP is effectively resolved by the existing techniques. In the unsupervised learning methods (see e.g., [21], [22]), the solution of the sequencing part of the problem is combined with the online sampling of the suitable positions and headings. In the direct search methods, the mixed variables are optimized simultaneously. Many optimization algorithms (see e.g., the double-loop hybrid algorithm [23], the variable neighborhood search [24], the memetic algorithm [25] and descent method [26]) can be used as search engine.

Similar to DTSPN, the DTSPDN is also an NP-hard problem with mixed variables. However, DTSPDN is more complex than DTSPN. DTSPDN exhibits strong coupling among the visiting sequence, the access locations and the encounter time for UGVs. The visiting sequence affects the encounter time for UAV to visit each UGV and further changes the access locations that vary with the UGVs' movements. On the other hand, DTSPDN contains dynamic constraints. Since the positions of the UGVs change with time, the UAV needs to predict the positions of UGVs and then find its appropriate access locations for all UGVs by solving a sequence of encounter problems. Therefore, the existing methods to resolve DTSPN are unsuitable for DTSPDN.

In our previous work [5], a heuristic algorithm is proposed by using a decoupling strategy to decompose the DTSPDN into two subproblems: determining the sequence of all UGVs for the UAV to visit by the receding horizon optimization, and optimizing the access locations in the neighborhood of each UGV by boundary sampling or center sampling. Since the visiting sequence and locations are interdependent, the heuristic algorithm in [5] does not perform well. Therefore, a more efficient approach is needed to address DTSPDN.

Note that the DTSPDN problem studied in this paper is different from the DDTSPN problem studied by Macharet *et al.* [27], [28]. The dynamic property in [27], [28] refers to dynamically arising target regions, instead of the neighbourhoods of dynamic UGVs with movement in DTSPDN.

B. Main Innovations and Contributions

Compared with existing efforts, the main innovations and contributions of the paper are as follows.

1) The path planning of the messenger UAV that flies over the communication neighborhood of all moving UGVs is modeled as the DTSPDN. By analyzing the characters of the DTSPDN, we design a novel encoding scheme and a fast decoding scheme. The encoding scheme integrating the visiting sequence and location information can effectively maintain population diversity. The decoding scheme utilizes the theoretical results of the expected path length of the Dubins vehicle and the path realization method to achieve rapid decoding.
2) A memetic algorithm tailored to the DTSPDN formulation is proposed, combining a genetic algorithm (GA) and an efficient local search strategy to achieve a better tradeoff between exploration and exploitation in the solution space. In addition, a gradient-based repair strategy is introduced to repair the individuals violating dynamic constraints by locating and tracking varying feasible regions over time.
3) During the evolutionary process, a special phenomenon may occur that when changing some decision variables (i.e. visiting sequence and location), evaluation function value cannot be changed but the feasible region of another decision variable (i.e., visiting time) may be altered due to the encounter constraint between UAV and UGV, which will affect the further optimization of the solution in local search procedure. To track and utilize the change of the feasible region when the decision variables change, a transformation procedure is proposed to improve one solution to a better one by analyzing the encounter pattern between UGV and UAV.
4) The proposed method is verified on random instances with different scales through extensive numerical simulations, which shows that the proposed approach can effectively generate better curvature-constrained tours to encounter all moving UGVs in various DTSPDN cases when compared with four other competitive algorithms in the literature.

The paper is organized as follows: Section II provides the problem formulation and preliminaries. Section III presents the encoding and decoding scheme. In Section IV, we give a detailed description of the memetic algorithm. Section V evaluates the proposed algorithm through a series of computational experiments and analyses. Finally, Section VI concludes the paper.

II. Problem Statement and Preliminaries

A. Problem Statement

UGVs scattered in a large area[1] execute tasks in an obstacle-free work space. Each UGV arrives at task points along a straight-line path at different speeds, denoted by v_i^G with $i \in \{1, 2, \ldots, N_G\}$. Each UGV has a limited communication range. We specify the communication neighborhood of the UGV as a disk centered at the UGV. The valid communication radius for the UGV i with $i \in \{1, 2, \ldots, N_G\}$ is denoted by $r_{i,com}^G$.

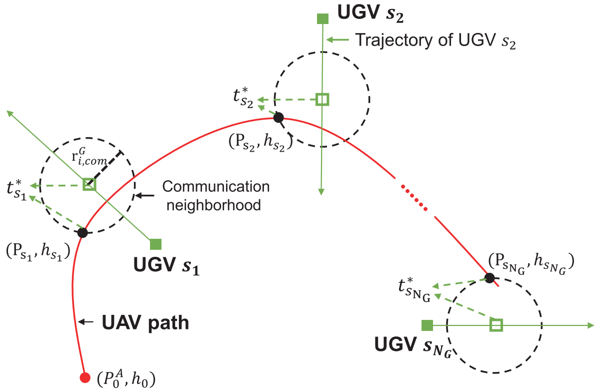

Fig. 2. The tour of a messenger UAV traversing the communication neighbourhoods of UGVs.

One small fixed-wing UAV flying in a 2D plane is chosen as the messenger UAV. The motion parameters of the UGVs (e.g., velocity and trajectory) are known by the UAV. Given the initial locations of the moving UGVs and their motion parameters, the messenger UAV is responsible for flying over each moving UGV to acquire this UGV information and transmit the information about the others. In practice, UGVs and UAVs generally have a limited communication range. A messenger UAV can transmit information to a UGV only if the UAV is located within the effective communication range (neighborhood) of the UGV, as shown in Fig. 2. It is assumed that information can be instantaneously transmitted from the UAV to each UGV since the amount of data to be transmitted including the location of UGVs is very small.

The UAV just needs to pass the neighborhoods of all UGVs, rather than flying over their precise locations. Hence, the UAV can plan a tour to traverse each UGV's neighborhood according to the planned sequence in the shortest time. The planned sequence of all UGVs for the UAV to visit (briefly called the visiting sequence of UGVs) is denoted by $S = [s_1, s_2, \ldots, s_{N_G}]$, where s_i $(i \in \{1, 2, \ldots, N_G\})$ is the index of the i-th UGV. For instance, $S = (3, 1, 2)$ means that the UAV needs to visit UGV3, UGV1, and UGV2 in sequence. The access location, the access heading, and the access time required to visit the neighborhood of each UGV i $(i \in \{1, 2, \ldots, N_G\})$ are denoted by P_i, h_i and Δt_i, respectively.

In order to acquire a smooth and feasible tour for the messenger UAV, the Dubins model [29] is used to describe the kinematic characteristics of the UAV in a two-dimensional plane:

$$\begin{cases} \dot{x}^A = v^A \cos(h^A) \\ \dot{y}^A = v^A \sin(h^A) \\ \dot{h}^A = \dfrac{v^A}{r_{min}} u^A, u^A \in [-1, 1] \end{cases} \tag{1}$$

[1] A large area means that the distance between UGVs is much larger than their effective communication range.

where u^A is the control input, and $\omega = (x^A, y^A, h^A)$ represents the status of the Dubins vehicle (UAV), also called Dubins configuration.

For the messenger UAV, the travel cost between every pair of UGVs can be measured by the length of the Dubins path between them. The objective of the problem is to determine a tour with the minimum cost for the messenger UAV in a 2D plane when the UAV traverses the valid neighborhood of each UGV. We model the messenger UAV's path planning as a DTSPDN in which each UGV's location changes dynamically. Therefore, the DTSPDN optimization model is represented as follows:

$$\arg\min_{[S,P,H]} J = D((P_0^A, h_0), (P_{s_1}, h_{s_1})) + \sum_{i=1}^{N_G} D((P_{s_i}, h_{s_i}), (P_{s_{i+1}}, h_{s_{i+1}})) \tag{2}$$

$$s.t. \begin{cases} \dot{x}^A = v^A \cos(h^A) \\ \dot{y}^A = v^A \sin(h^A) \\ \dot{h}^A = \frac{v^A}{r_{min}} u^A, u^A \in [-1, 1] \end{cases} \tag{3}$$

$$d(P_{s_i}, P_{s_i}^G(t)) < r_{i,com}^G, \forall i \in I \tag{4}$$

where $P = [P_{s_1}, P_{s_2}, \ldots, P_{s_{N_G}}]$ and $H = [h_{s_1}, h_{s_2}, \ldots, h_{s_{N_G}}]$, P_0^A and h_0 are the initial position and heading of UAV, respectively. $D((P_{s_i}, h_{s_i}), (P_{s_{i+1}}, h_{s_{i+1}}))$ denotes Dubins distance from configuration (P_{s_i}, h_{s_i}) to $(P_{s_{i+1}}, h_{s_{i+1}})$ and $P_{s_i}^G(t)$ denotes the location of UGV s_i at time t. Constraint (4) is a *dynamic constraint* that ensures the messenger UAV is located within the communication range of each moving UGV.

In this paper, Dubins paths with the terminal heading relaxation are introduced to simplify the calculation of the Dubins distance. The Dubins path with terminal heading relaxation, proposed by Bui and Boissonnat [30], refers to a Dubins path whose its initial configuration and its endpoint are fixed but its terminal heading is free. In the sense of the terminal heading relaxation, the calculation of the Dubins paths can be simplified [14].

B. The Shortest Path for UAV to Visit a Moving UGV

The problem of a messenger UAV to visit a moving UGV can be regarded as an encounter problem between the UAV and the UGV. The following Lemma provides a sufficient condition to achieve the encounter between the UAV and the UGV.

Lemma 1: (Theorem 1 in [5]) Assume that the UAV can obtain the UGV's trajectory $P^G(t) = (x^G(t), y^G(t))$, and the maximum speed of the UGV is less than the UAV speed ($v_{\max}^G < v^A$). There exists $t^* \in [t_{min}, t_{max}]$, such that

$$D((P_0^A, h_0), P^G(t^*)) = v^A t^*,$$

where

$$t_{min} = \frac{D_{min}((P_{s_{i-1}}, h_{s_{i-1}}), P^G(0))}{v^A + v_{max}^G}$$

$$t_{max} = \frac{D_{min}((P_{s_{i-1}}, h_{s_{i-1}}), P^G(0))}{v^A - v_{max}^G}.$$

Our previous study [5] provided a path planning method to visit a moving UGV, as shown in Algorithm 1. The existence interval $[t_{min}, t_{max}]$ of visiting time t^* is determined at first. A numerical approximation for $t^* \in [t_{min}, t_{max}]$ can then be obtained by a bisection approach (Lines 1-4 in Algorithm 1). This also obtains the UAV path to visit a moving UGV according to t^* and the rendezvous point $P^r(t^*)$.

Algorithm 1 The Bisection Algorithm for the Shortest Path of the Messenger UAV to Visit a Moving UGV

```
Require: Given the UGV's trajectory $P^G(t) = (x^G(t), y^G(t))$, the initial solution interval $[t_{min}, t_{max}]$, the threshold value $\epsilon_b$, and define $\delta(t) = D_{min}((P_0^A, h_0), P^G(t))/v^A - t$.
1: repeat
2:     Find the midpoint $t_c = (t_{min} + t_{max})/2$ and then calculate $\delta(t_{min})$, $\delta(t_c)$.
3:     If $\delta(t_{min}) \cdot \delta(t_c) > 0$ holds, then the new solution interval changes to $[t_c, t_{max}]$, otherwise take $[t_{min}, t_c]$ as a new interval.
4: until $|\delta(t)| < \epsilon_b$
5: $t^* = (t_{max} + t_{min})/2$
6: According to the numerical solution $t^*$, the path to visit a moving UGV is a Dubins path from $(P_0^A, h_0)$ to the rendezvous point $P^r(t^*)$.
```

C. UAV Path With Expected Length

The UAV needs to solve a sequence of encounter problems to visit all UGVs. When the UAV uses the shortest path from the current location to visit each UGV, the obtained tour of UAV may not be optimal. Therefore, it needs to control the visiting time for each UGV to obtain the optimal tour. Since the UAV's speed is constant, the messenger UAV needs to control its path length to access UGV at a given time.

Since the path length satisfies the monotonicity conditions (the monotonicity of the path length functions has been proved in [31]), the bisection algorithm can be used to solve the proper parameters for paths with the expected lengths, as presented in Algorithm 2. The inputs to the bisection algorithm are the destination points and the expected length; its outputs are the feasible path pattern and the adjustable parameters. The path patterns and the adjustable parameters to achieve the UAV path with the expected length can be found in our previous work [31].

III. Encoding and Decoding Scheme for DTSPDN

A. Encoding Scheme

Using the Dubins paths with terminal heading relaxation, only the visiting sequence and the visiting point of each UGV need to be determined. Assume that the visiting sequence is $S = [s_1, \ldots, s_{N_G}]$. The time interval to access from UGV s_{i-1} to UGV s_i with $i = \{2, \ldots N_G\}$ is denoted as Δt_{s_i},[2] as shown in Fig. 3. Based on the fact that the UAV should fly through the boundary of communication neighborhood of UGV, its boundary points are used to encode the visiting points

[2] Δt_{s_1} represents the time interval that accesses from ω_0 to UGV s_1.

Algorithm 2 The Bisection Algorithm for Calculating the Path Parameters With Expected Length

```
Require: The expected path length $d$, the location of destination point $P$, and the threshold value $\epsilon_e$.
Ensure: The parameter $x$ for expected path length.
Select one path pattern according to the classification about $P$ and $d$, and then determine the adjustable parameter $x$ and its given bound $[a, b]$, the monotonicity indicator $\alpha$ and the path length function $F(x)$.
$x_{min} = a$ and $x_{max} = b$;
$x_0 = (b + a)/2$;
while $|F(x_0) - d| > \epsilon_e$ do
    $K = F(x_0) - d$, $\eta = \frac{1-\alpha \mathrm{sgn}(K)}{2}$;
    $x_{min} \leftarrow \eta x_0 + (1 - \eta) x_{min}$;
    $x_{max} \leftarrow (1 - \eta) x_0 + \eta x_{max}$;
    $x_0 \leftarrow \frac{x_{max} + x_{min}}{2}$
end while
return $x_0$;
```

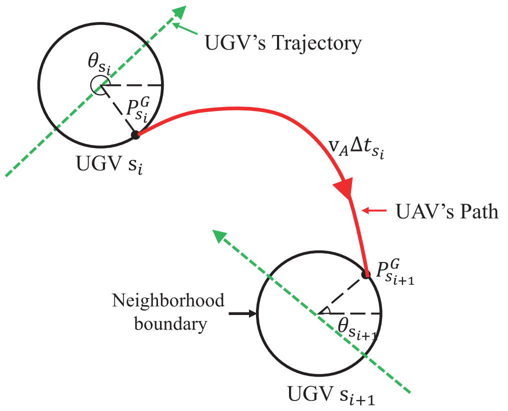

Fig. 3. Encoding strategy.

for the UGV. Any point on the boundary can be expressed by the polar angle $\theta_i \in [0, 2\pi]$ with respect to its center $P^G_i = [x^G_i, y^G_i]$. The visiting point P_{s_i} of UAV to visit UGV s_i can be calculated using Δt_{s_i} and θ_i as follows,

$$P_{s_i}(\theta_{s_i}, \Delta t_{s_i}) = [x^G_{s_i}(\sum_{k=1}^{i} \Delta t_{s_k}) + r^G_{i,com} \cos(\theta_{s_i}), \quad y^G_{s_i}(\sum_{k=1}^{i} \Delta t_{s_k}) + r^G_{i,com} \sin(\theta_{s_i})]. \tag{5}$$

Using boundary-based encoding, a solution of the DTSPDN can be described as

$$\begin{pmatrix} S \\ \Theta \\ \Delta T \end{pmatrix} = \begin{pmatrix} s_1 \\ \theta_{s_1} \\ \Delta t_{s_1} \end{pmatrix} - \begin{pmatrix} s_2 \\ \theta_{s_2} \\ \Delta t_{s_2} \end{pmatrix} \cdots \begin{pmatrix} s_{N_G} \\ \theta_{s_{N_G}} \\ \Delta t_{s_{N_G}} \end{pmatrix} \tag{6}$$

B. Decoding Scheme

Once an encoded string is generated with the encoding scheme, a corresponding decoding scheme is required to convert the encoded string into a feasible and appropriate planning solution. The encoded string only contains the information about the messenger UAV access to UGVs, and does not provide tour to visit all UGVs. In the study, a novel decoding scheme is proposed to generate a valid tour. To generate a UAV tour to visit all UGVs, it needs to determine the UAV tour to visit each UGV according to the visiting sequence. In this decoding scheme, the visiting point P_{s_i} for UGV s_i is firstly calculated on the basis of the vector $(s_i, \theta_{s_i}, \Delta t_{s_i})^T$ in the individual and eq. (5). The visiting path to UGV s_i can be then determined with the following two cases according to Δt_{s_i} and P_{s_i}:

1) If $D_{min}\left((P_{s_{i-1}}, h_{s_{i-1}}), P_{s_i}(\theta_{s_i}, \Delta t_{s_i})\right) < v_A \Delta t_{s_i} + \varepsilon$, then the UAV cannot visit $P_{s_i}(\theta_{s_i}, \Delta t_{s_i})$ even though using the shortest Dubins path. ε is a path margin to ensure the reliable communication between UAV and UGV. In this case, this solution is an infeasible solution.
2) If $D_{min}\left((P_{s_{i-1}}, h_{s_{i-1}}), P_{s_i}(\theta_{s_i}, \Delta t_{s_i})\right) \geq v_A \Delta t_{s_i} + \varepsilon$, then the UAV arrives at $P_{s_i}(\theta_{s_i}, \Delta t_{s_i})$ with the shortest Dubins path just right and in advance. In this case, the UAV needs to plan its path to ensure that the UAV can just arrive at P_{s_i} with path length $v_A \Delta t_{s_i} + \varepsilon$ according to Algorithm 2.

IV. Memetic Algorithm for DTSPDN

The proposed algorithms, named MA, is presented to solve DTSPDN problem in this section. Firstly, the evaluation function used in the MA is introduced in subsection IV-A. Then, the overall framework of MA is described in subsection IV-B. All major procedures of MA are presented in detail from subsection IV-C to IV-G. Finally, the computational complexity of MA is analyzed in subsection IV-H.

A. Evaluation Function

The evaluation function of the DTSPDN problem has two parts: The first part is the total flight tour of the UAV when visiting all UGVs, that is $\sum_{i=1}^{N_G} \Delta t_{s_i} v_A$. The second part reflects the degree of constraint violation.

Remark: Our previous work [31] discovered the realizability of Dubins path with an expected length in a two-dimensional plane. It is proved that, when the destination point is located in a special region, denoted by D_δ, there is a special length interval, denoted by (λ^-, β^-), for which no proper Dubins path exists.

An individual X violates constraints in the following cases:

Case 1. The UAV cannot meet the point P_{s_i} with the shortest Dubins path;

Case 2. P_{s_i} is located in the unrealizable area D_δ of the UAV, and $v_A \Delta t_{s_i} \in \left(\lambda^-, \beta^-\right)$.

The degree of constraint violation is denoted by c_{s_i} when the UAV needs to visit P_{s_i} with time t_{s_i}:

$$c_{s_i} = \begin{cases} D_{min}((P_{s_{i-1}}, h_{s_{i-1}}), P_{s_i}) - v_A \Delta t_{s_i}, & \text{Cases I and II} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Therefore, the evaluation function f of DTSPDN problem for X is

$$f(X) = v_A \sum_{i=1}^{N_G} \Delta t_{s_i} + \kappa \sum_{i=1}^{N_G} |c_{s_i}|, \tag{8}$$

where κ ($\kappa \gg \sum_{i=1}^{N_G} \Delta t_{s_i} v_A$) is a penalty parameter.

As can be seen from Eq. 8, there are three special properties of $f(X)$ as follows:

Property 1: When X violates constraints, $f(X)$ depends on s_i, θ_{s_i} and Δt_{s_i} with $s_i \in I$.

Property 2: When X does not violate constraints, $f(X)$ only depends on Δt_{s_i} with $s_i \in I$.

Property 3: When X does not violate constraints, changing θ_{s_i} and s_i cannot affect $f(X)$ but will change the location of P_{s_i}, and further affect $D_{min}\left(\left(P_{s_{i-1}}, h_{s_{i-1}}\right), P_{s_i}\right)/v_A$, that is the lower bound of Δt_{s_i}, denoted by $\underline{\Delta t_{s_i}}$.

B. Overall Framework

The memetic algorithm proposed by Moscato and Norman [32] is a powerful framework that combines the exploration power of a genetic algorithm (GA) and the exploitation strength of local search. This paper proposes a memetic algorithm based on GA procedure, repair procedure, transformation procedure and local search procedure to solve the DTSPDN problem. Algorithm 3 gives an overview of our memetic algorithm.

The GA refers to a class of adaptive search procedures based on the principles derived from natural evolution and genetics, see lines 4-12 in Algorithm 3. The initial population contains NP individuals that are constructed by randomly generated solutions. For each generation, each individual solution X_i is selected as parent p_A and another individual solution is randomly selected as parent p_B.

For each pair of parents, p_A and p_B, the crossover operator generates a new offspring solution X_{new}. Next, a mutation operation is performed on X_{new} with probability p_m. If X_{new} violates constraints, the repair procedure is invoked to try to repair X_{new}. During the evolutionary process, changing some decision variables (i.e., visiting sequence and location) may not affect the evaluation function value, but may alter the feasible region of another decision variable (i.e., visiting time) due to the encounter constraint between the UAV and the UGV. To track and utilize the change of the feasible region, a transformation procedure is proposed to improve one solution to another one with less visiting time. If X_{new} has a better objective function value than X_i, it will replace the individual X_i.

The local search procedure is applied to the elite individuals inherited from populations, see lines 14-19 in Algorithm 3. In order to reduce the computation time spent on local search, the local search procedure is performed every T_{lp} generation. The local search procedure selects N_{gen}/T_{lp} elite individuals randomly from the top 50% individual to implement local search.

C. Crossover Operation

The crossover operator plays a critical role in memetic search and defines the way to transmit the information from parents to offspring. A meaningful crossover operation should preserve good properties of parent individuals through its recombination process. In our case, an auxiliary-vector-based crossover operator is adopted to generate an offspring individual, which is in the same way as in [25]. Here is an example of four UGV cases to introduce the crossover operation. The parent individuals are constituted by the ith individual, denoted by p_A, and another selected from the population randomly, denoted by p_B:

Algorithm 3 The Memetic Approach for the DTSPDN

```
1: Initialization: Generate randomly $NP$ individual $X_i$ ($i = 1, 2, \ldots, NP$) and compute $f(X_i)$ for each $X_i$
2: $N_{gen} = 0.\%N_{gen}$ is evolutional generation
3: repeat
4:   for all $X_i$ do
5:     Select randomly another individual $X_i'$ and generate a new individual $X_{new}$ by the crossover operator
6:     Generate a random number $rand_i$ ($rand_i \in [0, 1]$), if $rand_i < p_m$, then perform the mutation operator on the individual $X_{new}$.
7:     Operate the repair procedure on $X_{new}$
8:     Operate the transformation procedure on $X_{new}$
9:     if $f(X_{new}) < f(X_i)$ then
10:      $X_i \leftarrow X_{new}$
11:    end if
12:  end for
13:  $N_{gen} \leftarrow N_{gen} + 1$
14:  if $mode(N_{gen}, T_{lp}) == 0$ then
15:    for $j = 1$ to $N_{gen}/T_{lp}$ do
16:      Select randomly an individual from top 50% individuals
17:      Operate the local search procedure on $X_j$
18:    end for
19:  end if
20: until The stopping criteria is met
21: return the best individual $X^*$ and $f(X^*)$
```

$$p_A = \begin{pmatrix} 1 \\ \theta_1 \\ \Delta t_1 \end{pmatrix} - \begin{pmatrix} 3 \\ \theta_3 \\ \Delta t_3 \end{pmatrix} - \begin{pmatrix} 2 \\ \theta_2 \\ \Delta t_2 \end{pmatrix} - \begin{pmatrix} 4 \\ \theta_4 \\ \Delta t_4 \end{pmatrix}$$

$$p_B = \begin{pmatrix} 4 \\ \theta_4' \\ \Delta t_4' \end{pmatrix} - \begin{pmatrix} 2 \\ \theta_2' \\ \Delta t_2' \end{pmatrix} - \begin{pmatrix} 3 \\ \theta_3' \\ \Delta t_3' \end{pmatrix} - \begin{pmatrix} 1 \\ \theta_1' \\ \Delta t_1' \end{pmatrix}$$

We randomly generate an auxiliary vector having the same dimensions as the parent individuals whose elements are 1 or 2. For example, given $v : 1-2-2-1$, the value of the components in the auxiliary vector determines which element from 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 p_A will be selected to construct the offspring X_{new}.

$$X_{new} = \begin{pmatrix} 1 \\ \theta_1 \\ \Delta t_1 \end{pmatrix}$$

Then, the selected gene will be deleted from both parent individuals. The components corresponding to UGV 1 from parents p_A and p_B are removed, which changes the parent individuals to

$$p_A = \begin{pmatrix} 3 \\ \theta_3 \\ \Delta t_3 \end{pmatrix} - \begin{pmatrix} 2 \\ \theta_2 \\ \Delta t_2 \end{pmatrix} - \begin{pmatrix} 4 \\ \theta_4 \\ \Delta t_4 \end{pmatrix}$$

$$p_B = \begin{pmatrix} 4 \\ \theta_4' \\ \Delta t_4' \end{pmatrix} - \begin{pmatrix} 2 \\ \theta_2' \\ \Delta t_2' \end{pmatrix} - \begin{pmatrix} 3 \\ \theta_3' \\ \Delta t_3' \end{pmatrix}$$

Repeating these steps constructs the offspring individual:

$$X_{new} = \begin{pmatrix} 1 \\ \theta_1 \\ \Delta t_1 \end{pmatrix} - \begin{pmatrix} 4 \\ \theta_4' \\ \Delta t_4' \end{pmatrix} - \begin{pmatrix} 2 \\ \theta_2' \\ \Delta t_2' \end{pmatrix} - \begin{pmatrix} 3 \\ \theta_3 \\ \Delta t_3 \end{pmatrix}$$

D. Mutation Operation

In order to increase the diversity in the population, the new individual X_{new} needs to be mutated according to a certain probability p_m. In this paper, the following three mutation operators are adopted with equal probability $(p_m/3)$:

Operator 1: Two indices $i, j \in \{1, 2, \ldots, N_G\}$ with $i \neq j$ are chosen randomly, and then the corresponding genes swap their positions in X_{new}.

Operator 2: Randomly select a number $i \in \{1, 2, \ldots, N_G\}$, and then the polar angle θ_i in X_{new} is reset within the interval $(0, 2\pi]$.

Operator 3: Randomly select a number $i \in \{1, 2, \ldots, N_G\}$, and then the visiting time Δt_i in X_{new} is reset within the interval $[0, \overline{\Delta t}]$, where $\overline{\Delta t}$ is the upper bound of Δt_i.

E. Repair Operation

After the crossover and mutation operations, the generated individual X_{new} may violate the dynamic constraint in Eq. (4). The dynamics of constraints could cause changes in the shape, percentage, or structure of feasible/infeasible regions.

To accelerate the process of locating feasible regions, a gradient-based repair strategy is integrated into the algorithm. Since the messenger UAV can visit UGVs according to Lemma 1, changing ΔT in X_{new} can relocate the feasible region of X_{new}. The adopted repair method, utilizing the gradient information of constraints, can locate feasible regions rapidly.

It can be observed that the dynamic constraint only affects the current UGV to visit and the UGVs being visited without affecting the visited UGVs. Therefore, if the visiting time of one UGV (e.g., UGV s_j) is changed, Δt_{s_j} with $i \leq j \leq N_G$ may violate the dynamic constraint, but Δt_{s_j} with $1 \leq j < i$ can still satisfy it. Based on this character, the repair operation can use gradient information to locate the feasible regions of each UGV in visiting sequence $S = [s_1, \ldots, s_{N_G}]$.

The degree of constraint violation c_{s_i} with $i \in \{1, 2, \ldots, N_G\}$ can be obtained according to Eq. (7). Its gradient value ∇c_{s_i} is

$$\nabla c_{s_i} = \frac{1}{\eta}\left(c_{s_i}\left(\Delta t_{s_i} + \eta\right) - c_{s_i}\left(\Delta t_{s_i}\right)\right), \tag{9}$$

where η is a small positive value. Therefore, the visiting time Δt_{s_i} can be updated according to the following equation:

$$\Delta t_{s_i} := \Delta t_{s_i} - \eta_t c_{s_i} \nabla c_{s_i}^{-1}, \tag{10}$$

where η_t is the step size. The iteration goes on until c_{s_i} reaches desired accuracy ϵ_r.

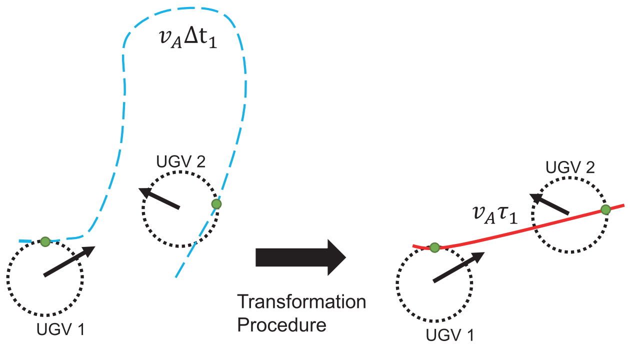

Fig. 4. Transformation procedure. The red solid curve represents the path that is treated by transformation procedure, and the blue dash curves represent the untreated path.

F. Transformation Procedure

If X_{new} does not violate constraints, $f(X_{new})$ only depends on Δt_{s_i} with $s_i \in I$ according to the property 1 of $f(X)$. In this case, only the changes of Δt_{s_i} to improve $f(X_{new})$ can be reserved during the evolutionary process but the changes of other components in the solution (i.e., S and Θ) by crossover and mutation operations will not affect $f(X_{new})$. However, Δt_{s_i} is restricted by its lower bound $\underline{\Delta t_{s_i}}$. According to the property 3 of $f(X)$, $\underline{\Delta t_{s_i}}$ is influenced by S and Θ. It means that if X_{new} does not violate constraints, changing S and Θ cannot alter $f(X_{new})$ but will change the feasible region of Δt_{s_i} and affect the further optimization of $f(X_{new})$ in local search procedure. The variants generated by changing the decision variables like S and Θ are called *silent variants* [33], in which the changes of some specific decision variables do not directly affect evaluation function value but will determine some properties of the solution. The evolutionary process containing silent variants can effectively maintain the population diversity [33] and is more likely to reach a global optimum without worrying about premature convergence [34].

To track the change of the feasible region with varying S and Θ and optimize the solution with silent variants, a transformation procedure is proposed to transform one solution to a better solution by adjusting the access time of UGVs, as shown in Fig. 4. Since there exists a positive correlation between the access time Δt_{s_i} $(s_i \in \{1, 2, \ldots, N_G\})$ and the value of the evaluation function, the proposed operation adjusts Δt_{s_i} in the new individual, affecting the evaluation function value of the new individual. In addition, since the repair operation is time-consuming, it needs to ensure that the transformation operation does not violate the dynamic constraint in Eq. (4).

According to Lemma 1, the UAV can always encounter the moving UGV. The encounter patterns fall into two categories according to the relative motion between the UAV and the UGV: the catch-up pattern and the meeting pattern.

Definition 1: (Catch-up pattern) If $D((P^A, h^A), P_G(t)) \leq D((P^A, h_A), P_G(t+\varepsilon_t))$ where ε_t is an arbitrary time interval, then the encounter patterns between UGV and UAV is defined as *catch-up*.

Definition 2: (Meeting pattern) If $D((P^A, h^A), P_G(t)) > D((P^A, h_A), P_G(t+\varepsilon_t))$, then the encounter patterns is defined as *meeting*.

The following proposition provides a sufficient condition to reduce the encounter time for the two UGVs to visit successively. The poof is postponed to Appendix A.

Proposition 1: Assume that a messenger UAV needs to visit UGV s_i and s_{i+1} successively if the encounter pattern between UAV and UGV s_{i+1} is catch-up, then reduce Δt_{s_i} such that UAV can still encounter UGV s_{i+1} with $\Delta t_{s_{i+1}}$. If the UAV visits UGV s_{i+1} with meeting pattern, then reducing Δt_{s_i} will lead to the UAV being unable to encounter s_{i+1} with $\Delta t_{s_{i+1}}$.

According to Proposition 1, if the encounter pattern between UAV and UGV s_{i+1} is catch-up, the access time of the UAV Δt_{s_i} for s_i can be reduced. Therefore, Δt_{s_i} for UGV s_i is reset to $\underline{\Delta t_{s_i}}$, that is the shortest visiting time. The details of the transformation procedure are shown in Algorithm 4.

Algorithm 4 Transformation Procedure

```
for all  Any adjacent UGVs $s_i$ and $s_{i+1}$ in the individual $X$ do
  if  UAV catches up with UGV $s_{i+1}$, then
    Calculate the shortest time to visit UGV $s_i$ according to Algorithm 1, denoted by $\tau_{s_i}$.
    $\Delta t_{s_i} = \tau_{s_i}$.
  end if
end for
```

G. Local Search Operation for Visiting Location

A gradient-based search (GS) operator and a sampling-based search (SS) operator are introduced to perform the local search procedure for visiting locations on the boundary of the UGV's neighborhood. The two local search operators have different search logic. The SS operator is a long-distance search algorithm while the GS operator is a short-distance search algorithm. The main reason for using two local search operators is to guide the elite individuals toward more promising areas in the solution space while avoiding being trapped in local optimum.

In accordance with the Ockham's Razor in Monte Carlo [35], the selected solution is processed either by the gradient-based or sampling-based search operator with an equal probability (0.5).

1) GS Operator: A component corresponding to a UGV is selected randomly in individual X, e.g., $g_{s_k} = (s_k, \theta_{s_k}, \Delta t_{s_k})^T$. The value of evaluation function of X with respect to θ_{s_k} is denoted by $f(X|\theta_{s_k})$. The approximate derivative of f depending on θ_{s_k} is

$$\nabla_{\theta_{s_k}} f(X) \approx [f(X|\theta_{s_k}) - f(X|(\theta_{s_k} + \Delta\theta))]/\Delta\theta, \qquad (11)$$

where $\Delta\theta$ is a small positive value. Note that the individual added by $\Delta\theta$ needs to perform the repair and transformation procedures.

TABLE I

Time Complexity Analysis of Each Procedure

	Procedure	Time Complexity
GA	Crossover	$\mathcal{O}(NP \cdot N_G)$
	Mutation	$\mathcal{O}(NP)$
	Repair	$\mathcal{O}(NP \cdot N_G \cdot N_G/\epsilon_r)$
	Transformation	$\mathcal{O}(NP \cdot N_G \cdot N_G)$
	Fitness	$\mathcal{O}(NP \cdot N_G)$
Local search	GS operator	$\mathcal{O}(N_{gen}/T_{lp} \cdot [2\pi/\rho_l] \cdot N_G)$
	SS operator	$\mathcal{O}(N_{gen}/T_{lp} \cdot N_s \cdot N_G)$

After deriving the approximate gradient θ_{s_k}, the corresponding update equation for θ_{s_k} is

$$\theta_{s_k} := \theta_{s_k} - \rho_l \nabla_{\theta_{s_k}} f(X) \qquad (12)$$

where ρ_l is step size. The iteration continues until the length of the Dubins paths corresponding to the solution does not improve. Then a new individual is generated by the above steps, denoted by X^g_{LS}. If $f(X^g_{LS}) < f(X)$, then X^g_{LS} will be retained.

2) SS Operator: For solution X, a UGV is selected randomly, and uniform sampling is then implemented on the boundary of the neighborhood of the UGV (the sampling number is N_s). For any sampling point m $(m = 1, 2, \ldots, N_s)$, the rendezvous point $P^r_m(\beta, t^*_m)$ and its corresponding rendezvous time t^*_m can be obtained by Algorithm 1. Furthermore, set $\Delta t_{s_k} = t^*_m$, perform the repair and transformation procedures on X, and generate one new individual, denoted by X_m.

For all sampling points, the best solution is reserved and denoted by X^s_{LS},

$$X^s_{LS} = \arg\min f(X_m). \qquad (13)$$

If $f(X^s_{LS}) < f(X)$, then X^s_{LS} will be retained.

H. Computational Complexity Analysis

The time complexity analysis of the memetic algorithm proposed in this paper is presented in Table I.

For GA process, first it has a computational complexity of $\mathcal{O}(NP \cdot N_G)$ by using crossover and mutation method to generate new individuals; then, it has a computational complexity of $\mathcal{O}(NP \cdot N_G/\epsilon_r)$ to repair the new individuals since the iteration number of the gradient-based repair procedure for each UGV is $1/\epsilon_r$. In the transformation procedure, decoding operation is needed on each UGV and the computational complexity on each UGV is $\mathcal{O}(N_G)$. So the computational complexity of this procedure is $\mathcal{O}(NP \cdot N_G^2)$. Therefore, the computational complexity of GA process is

$$\mathcal{O}[NP \cdot (N_G + N_G/\epsilon_r + N_G^2 + N_G)] = \mathcal{O}(NP \cdot N_G^2). \qquad (14)$$

The local search process for visiting location employs two operators which consist in GS operator and SS operator. In GS operator, the upper bound of iteration number is $[2\pi/\rho_l]$ where $[\cdot]$ is down-rounding function according to [25]. So the computational complexity of GS operator is $\mathcal{O}(N_{gen}/T_{lp} \cdot [2\pi/\rho_l] \cdot N_G)$. In SS operator, the improved solution is generated via sampling N_s with a computational complexity of

$\mathcal{O}(N_{gen}/T_{lp} \cdot N_s \cdot N_G)$. The computational complexity of local search process is

$$\mathcal{O}(N_{gen}/T_{lp} \cdot \lceil 2\pi/\rho_l \rceil \cdot N_G + N_{gen}/T_{lp} \cdot N_s \cdot N_G) = \mathcal{O}(N_G). \tag{15}$$

Therefore, the total computational complexity for the proposed memetic algorithm is

$$\mathcal{O}(NP \cdot N_G^2 + N_G) = \mathcal{O}(NP \cdot N_G^2) \tag{16}$$

V. Computational Experiment

This section performs experiments to evaluate and test the proposed algorithm for the DTSPDN, named MA. This algorithm is implemented in MATLAB 2018b, and all experiments are executed on a PC with Intel Core(TM) i7-7700T 2.9 GHZ and 16 GB RAM.

To investigate the performance of the proposed algorithm, MA is compared with five algorithms as follows:

1) The heuristic algorithm with boundary sampling (HE-B) proposed in [5]. The visiting order of the UAV is determined by receding horizon optimization and the access locations are obtained by uniform sampling on the boundary of each UGV's communication neighborhood;
2) The heuristic algorithm with center sampling (HE-C) proposed in [5]. The approach to determine the visiting order of the UAV is the same with HE-B but the access location is the center point of each UGV's communication neighborhood;
3) A double sampling algorithm (DS). Stieber *et al.* [36] modelled a Euclidean version of DTSPDN, called the traveling salesman problem with moving targets (TSPMT), as a time-discrete model by discretizing the UGV trajectory and then used the branch-and-bound search to solve it. Similarly, for each possible visiting order, we uniformly discretize the trajectory and the neighborhood boundary of all UGVs, and find the shortest reachable tour by using a depth-first branch-and-bound search [37]. All possible visiting orders are exhausted to find the shortest reachable tour.
4) MAR is a memetic algorithm proposed by Zhang *et al.* [25] and is used for solving DTSPN in which the UGVs are static. The MAR applies an encoding scheme integrating the visiting sequence and location, and combines the GA and a local search strategy based on the approximate gradient. In this paper, we modified the algorithm in which the visiting paths for each UGV are replaced by the encounter paths with the shortest Dubins path.
5) Variable neighborhood search (VNS) was proposed by Mladenović and Hansen [38] in 1997. Pěnička *et al.* [25] proposed a new VNS for solving DTSPN. The VNS employs predefined neighborhood operators used for iterative improvement of the initial solution inside the shaking and local search procedures. In this paper, we modified the algorithm in which the initial solution is generated by HE-B and the visiting paths for each UGV are replaced by the encounter paths with the shortest Dubins path.

TABLE II
The Parameters of the Proposed Memetic Algorithm

Symbols	Description	Setting
The initial configuration of UAV	ω_0	$(0, 0, \pi/2)$
UAV flight speed	v_A	10
MA		
Mutation probability	p_m	0.1
Population size	NP	$10N_G$
Maximum number of generations	$maxGen$	20
The generation number to perform local search	T_{lp}	2
The step size in gradient-based local search operator	ρ_l	0.01
The sampling number in sampling-based search operator	N_s	20
The threshold value in Algorithm 2	ϵ	0.01
MAR		
Mutation probability	p_m^r	0.1
Population size	NP_r	$10N_G$
Maximum number of generations	$maxGen_r$	20
The generation number to perform local search	T_{lp}^r	2
The step size in gradient-based local search operator	ρ_l^r	0.01
HE-B		
The sampling number on boundary in HE-B	N_{HE-B}	100

The parameter settings of the proposed memetic algorithm are shown in Table II. The parameter settings of MA and MAR (e.g., mutation probability, population size, maximum number of generations) refer to those in [25]. For HE-B, the specific configurations are consistent with those in [5] and also shown in Table II.

A. Comparative Experiment on Random Instances With Different Scales

In this part, a series of random instances with different scales are used to analyze the performance of all tested algorithms. The test instances are labeled as B-N-i, where N is the number of UGVs and i is the instance identifier. For example, B-2-1 indicates the 1st instance with 2 UGVs. In this part, $r_{min} = 5$ and $r_{i,com}^G = 5$ with $i = 1, 2, \ldots, N_G$.

For small-scaled instances, the results calculated by the four algorithms over a set of the 20 benchmark instances are reported in Table III. MAR, VNS and MA are executed 25 times independently on each instance. The results are evaluated using three measures: the mean value (avg), the standard deviation (std) and the result of the Wilcoxon rank-sum test. The elapsed time for the computation of each algorithm recorded in Fig. 5 is the average time of running 25 times. The average distance among the neighborhood of UGVs, denoted by ρ, is described as the average distance among the initial positions of UGVs minus $2r_{i,com}^G$. The distribution sparsity of UGVs for UAV is described as the ratio of ρ to the UAV's minimum turning radius r_{min}, that is

$$\kappa_P = \frac{\rho}{r_{min}}.$$

TABLE III

COMPUTATIONAL RESULTS OF DS, HE-B, HE-C, MAR AND MA ON SMALL PROBLEM SCALES

Instances	κ_P	MA(a_1) Avg.±std.	DS	HE-B	HE-C	MAR(a_2) Avg.±std.	VNS(a_3) Avg.±std.	Test result h^+
B-2-01	-1.4	**12.80±2.30**	24	36.56	46.45	16.74±2.32	30.04±13.10	(1,1)
B-2-02	-0.74	**24.96±5.52**0	14	26.78	39.41	25.66±5.51	55.12±17.35	(1,1)
B-2-03	-0.04	16.08±6.51	34	33.23	44.42	**15.11±3.22**	37.92±6.04	(0,1)
B-2-04	0.21	**25.38±3.84**	56	47.96	63.79	29.06±7.26	49.49±4.76	(1,1)
B-2-05	0.89	**67.10±2.35**	56	77.66	74.85	67.41±3.10	67.14±0.05	(1,1)
B-2-06	2.91	**49.62±2.13**	42	99.28	119.08	50.10±4.79	91.62±0.27	(1,1)
B-2-07	3.02	35.90±6.26	**16**	43.07	57.27	35.92±5.81	54.86±0.15	(1,1)
B-2-08	4.25	**57.86±2.90**	48	62.69	78.96	58.94±4.34	56.35±0.92	(1,1)
B-2-09	10.03	114.83±3.29	130	133.69	149.81	**114.66±2.68**	127.97±0.08	(0,1)
B-2-10	17.15	197.63±1.94	200	219.92	219.92	197.52±2.10	**192.74±1.28**	(0,0)
B-3-01	-1.14	**33.98±10.08**	26	36.80	93.72	64.81±7.11	76.57±16.04	(1,1)
B-3-02	-1.01	48.43±9.34	**36**	69.85	95.06	50.63±1.53	79.81±6.67	(1,1)
B-3-03	0.62	**52.81±3.86**	52	48.79	82.70	116.37±13.78	58.19±10.23	(1,1)
B-3-04	1.87	**68.64±7.12**	56	61.79	112.01	87.49±0.96	62.86±0.97	(1,1)
B-3-05	1.29	52.17±6.09	**40**	51.04	101.29	76.44±6.98	48.06±0.32	(1,1)
B-3-06	2.04	51.44±4.66	**40**	45.71	71.84	68.80±3.23	61.29±9.72	(1,1)
B-3-07	2.87	**60.41±6.20**	54	100.11	134.60	82.84±7.91	70.37±2.49	(1,1)
B-3-08	2.84	**48.85±4.70**	42	49.33	58.13	58.21±6.12	54.68±4.24	(1,1)
B-3-09	5.29	**78.52±3.74**	76	81.17	96.16	107.94±11.01	80.71±0.00	(1,1)
B-3-10	13.19	**144.00±12.37**	138	198.20	214.13	144.75±12.42	159.66±18.96	(1,1)

Note that $h^+ = [h_{a_1:a_2}, h = [h_{a_1:a_3}]$ the result of Wilcoxon rank-sum test with a confidence level of 0.95.
$h_{a_i:a_j} = 1$ indicates that the algorithm a_i performs significantly better than the algorithm a_j.
$h_{a_i:a_j} = 0$ indicates that the algorithms a_i and a_j have no significant difference.
$h_{a_i:a_j} = -1$ indicates that the algorithm a_j performs significantly better than the algorithm a_i.

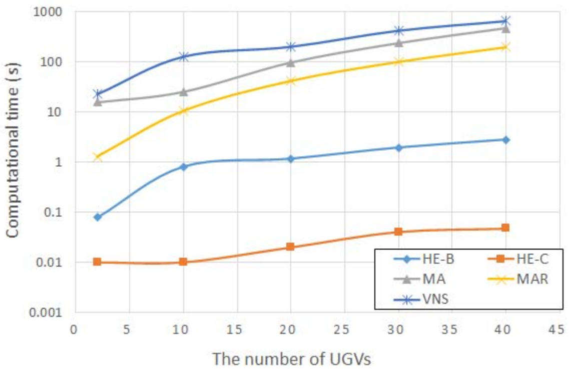

Fig. 5. Computational time of HE-C, HE-B, MAR, VNS and MA with the different number of UGVs.

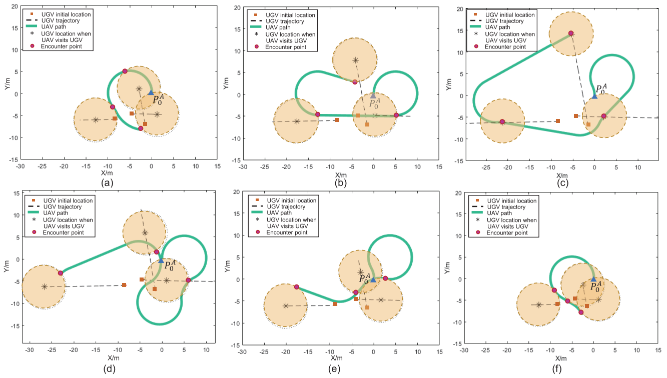

Fig. 6. The shortest UAV tours to visit three UGVs obtained by different methods for instance B-3-02. (a) DS, (b) HE-B, (c) HE-C, (d) VNS, (e) MAR, (f) MA.

When κ_P is large, the distance among UGVs relative to r_{min} is large and the distribution of UGVs is regarded as sparse.

As shown in Table III, the MA finds 14 best solutions out of 20 instances. In most cases, MA obtains the best Dubins path. The solutions obtained by DS are also very good. However, the computational complexity of DS is very high ($\mathcal{O}(N_G!(N_{NS})^{N_G}(N_{TS})^{N_G})$, where N_{NS} and N_{TS} are the sample numbers of the UGV's neighborhood and trajectory, respectively. When the number of UGVs is greater than 3, DS requires more than an hour to obtain a solution. Table III also shows that MA outperforms HE-C and HE-B for different instances in terms of the solution quality, no matter how sparse or dense the distribution of UGVs is. This is because MA takes advantage of the characteristics of the DTSPDN, combining individual learning and local search, which enables MA to better exploit and explore the search space. The optimal solutions obtained by MA are slightly inferior to those of MAR and VNS in some instances, such as with B-2-03, B-2-09 and B-2-10. For MAR and VNS, it uses the short encounter path between UGVs and contains local search to further optimize access locations. Thus, it can occasionally find better solutions. However, since MAR and VNS cannot adjust the visiting time for UGVs, they are prone to generate longer tour. For example, Fig. 6 shows the tours found by DS, HE-B, HE-C, MAR, VNS and MA on instance B-3-02.[3] In this case, for the tour generated by MAR and VNS, the UAV needs to turn a major arc along its minimal turning circle to access UGVs by applying the encounter paths with the shortest Dubins path. MA, however, can adjust the visiting time for UGVs to avoid generating major arcs.

[3] The video is available at https://youtu.be/rsNfggGD3zU

TABLE IV

COMPUTATIONAL RESULTS OF HE-B, HE-C, MAR AND MA ON MEDIUM AND LARGE PROBLEM SCALES

Instances	κ_P	MA (a_1) Avg.±std.	HE-B J^B	HE-C J^C	MAR (a_2) Avg.±std.	VNS (a_3) Avg. ±std.	*Gap* %	Test result h^+
B-10-01	7.90	**456.33**±**47.68**	1712.12	1997.87	961.7±184.2	575.79±73.55	73.3	(1,1)
B-10-02	6.38	**386.61**±**36.67**	1025.50	1221.73	878.9 ±148.0	597.99±105.26	62.3	(1,1)
B-10-03	7.06	**386.68**±**73.55**	1018.35	1278.77	896.1±270.8	541.41±135.97	62.0	(1,1)
B-10-04	6.54	**480.08**±**58.84**	982.18	1205.34	1084.3±291.2	592.52±146.08	51.1	(1,1)
B-10-05	5.12	**416.39**±**56.02**	1206.83	1739.14	987.9±236.3	526.12±59.17	65.5	(1,1)
B-10-06	3.93	**371.66**±**56.06**	821.17	893.92	780.8±205.5	459.68±81.22	54.7	(1,1)
B-10-07	2.35	**228.97**±**17.46**	347.86	477.60	462.8±92.6	354.891±39.63	34.1	(1,1)
B-10-08	2.42	**317.19**±**39.99**	318.49	516.46	598.0±109.9	418.321±60.63	22.9	(1,1)
B-10-09	1.02	**270.52**±**30.63**	671.63	1093.47	592.1±229.9	388.84±69.67	59.7	(1,1)
B-10-10	0.50	**205.36**±**24.85**	554.71	645.24	452.9±90.3	320.33±55.73	62.9	(1,1)
B-20-01	7.28	**1848.79**±**218.35**	6332.50	8393.63	2148.58±323.65	7726.22±2029.34	70.8	(1,1)
B-20-02	8.15	**2154.76**±**414.30**	9013.78	10554.59	2711.10±376.1	9496.55±2216.97	76.1	(1,1)
B-20-03	6.95	**2798.19**±**428.71**	9967.94	13889.17	3611.60±718.81	11045.50±2823.00	71.9	(1,1)
B-20-04	5.62	**2508.29**±**429.44**	9317.81	15303.03	3092.54±387.79	11064.93±5174.46	73.1	(1,1)
B-20-05	4.58	**2391.70**±**365.10**	5180.65	8437.20	2959.00±468.56	9740.99±3262.33	53.8	(1,1)
B-20-06	3.64	**3194.29**±**506.19**	20618.11	36623.40	3726.25±729.66	23148.22±7844.26	84.5	(1,1)
B-20-07	2.79	**2240.21**±**328.77**	5748.02	12695.01	2687.85±565.43	9379.01±4658.34	61.0	(1,1)
B-20-08	1.51	**2078.43** ±**310.19**	4201.01	12117.89	2544.44±402.12	10573.11±3772.07	50.5	(1,1)
B-20-09	1.01	**1769.54**±**335.08**	5654.23	11719.91	2543.27±355.41	11424.49±5721.41	68.7	(1,1)
B-20-10	0.35	**2459.97**±**511.73**	13843.43	21625.72	2685.72±494.50	15714.55±5993.11	82.2	(1,1)
		(Avg.±std.) × 10^4	$J^B \times 10^4$	$J^C \times 10^4$	(Avg.±std.) × 10^4	(Avg.±std.) × 10^4		
B-30-01	7.98	**2.72**±**0.65**	44.51	64.31	4.53 ±1.09	9.05±3.41	93.9	(1,1)
B-30-02	7.97	**2.96**±**0.45**	18.69	30.71	4.34 ±1.23	9.17±3.68	84.5	(1,1)
B-30-03	5.87	**2.58**±**0.60**	54.95	73.73	3.52±0.96	7.00±2.21	95.4	(1,1)
B-30-04	6.09	**2.34**±**0.34**	8.00	14.00	3.46±0.71	5.96±2.30	71.3	(1,1)
B-30-05	4.81	**1.96**±**0.45**	13.58	18.60	2.83±0.61	6.32±2.34	86.0	(1,1)
B-30-06	3.63	**1.97**±**0.41**	13.80	21.02	3.00±0.36	11.19±4.33	86.2	(1,1)
B-30-07	4.02	**2.52**±**0.28**	9.91	14.53	3.50±0.54	8.08±4.93	74.7	(1,1)
B-30-08	2.13	**1.99**±**0.42**	5.36	8.77	3.22±0.59	5.88±2.23	64.8	(1,1)
B-30-09	2.01	**0.96**±**0.16**	5.73	9.86	1.19±0.18	6.34±1.43	84.2	(1,1)
B-30-10	1.19	**1.42**±**0.25**	24.50	44.52	2.27±0.40	6.47±5.50	94.2	(1,1)
B-40-01	8.44	23.76±4.44	132.23	192.26	22.43±51.59	**13.48**±**4.29**	81.3	(0,0)
B-40-02	8.19	16.39±2.13	40.59	622.47	15.31±3.03	**13.13**±**6.15**	57.4	(0,0)
B-40-03	7.33	19.87±3.50	146.33	208.75	19.80±4.71	**13.58**±**5.17**	86.5	(0,0)
B-40-04	6.07	14.43±4.88	134.31	267.92	14.74±3.32	**12.37**±**6.29**	89.3	(1,0)
B-40-05	5.41	**10.13**±**1.67**	106.81	169.09	10.71±2.32	10.84±4.14	90.5	(1,1)
B-40-06	4.64	17.10±4.04	154.14	255.98	16.81±3.86	**12.88**±**3.77**	88.9	(0,0)
B-40-07	3.66	**12.97**±**1.90**	202.16	313.61	15.15±3.37	13.34±5.91	93.6	(1,1)
B-40-08	2.48	**13.06**±**2.87**	143.70	564.46	13.91±2.69	14.58±,8.28	90.5	(1,1)
B-40-09	1.77	**10.70**±**2.72**	66.37	117.36	13.37±3.35	14.73±4.09	83.6	(1,1)
B-40-10	1.41	**8.30**±**1.36**	84.56	186.02	9.56±1.93	12.36±4.83	90.2	(1,1)

Note that h^+ is consistent with that in Table III.

Table IV shows the performance of the four algorithms over 60 instances for medium-scale and large-scale instances. GAP (%) indicates the gap between the best solution T^* of the MA and the best solution out of HE-C and HE-B. It can be seen from the Table IV that MA outperforms its competitors in the medium-scaled and large-scaled instances. The GAP increases markedly with the increase in scale, ranging from 14.30% to 93.29%. Compared with the medium-scaled instances, the tours found by MA are much longer than those of HE-C and HE-B on the large instances. This is because the greedy nature of HE-B and HE-C makes it difficult to accommodate the complex spatial relationships among UGVs with the increasing problem scale. However, HE-C and HE-B have very short computation time since they use some heuristic rules to obtain solutions, as shown in Fig. 5.

As shown in Table IV, MA is superior to MAR and VNS on most medium-scaled and large-scaled instances. MAR and VNS lack effective adjustment for the visiting time, which results in solutions with lower quality. MA uses the transformation procedure to improve the visiting time of solutions by analyzing the encounter pattern between UAV and UGV. In addition, being equipped with local searches having different search logics, MA has a sufficient capability to intensify the search and explore the neighborhood of high-quality solutions more carefully, leading to higher quality solutions. MAR and VNS occasionally find better solutions when the distribution of UGVs is very sparse (B-40-01, B-40-02, and B-40-03 for MAR, and B-40-01, B-40-02, B-40-03 and B-40-06 for VNS). Since the difference between the Euclidean path and the Dubins path becomes indiscernible when the distribution of UGVs for UAV is very sparse (κ_P is large), the DTSPDN can be regarded as the Dynamic Euclidean Traveling Salesman Problem with Neighborhood (DETSPN). In this case, MAR and VNS directly use the shortest Dubins path, similar to the Euclidean path, to access all UGVs, reducing the search space and finding better solutions. However, MAR

TABLE V

STATISTICAL RESULTS OF FOUR ALGORITHMS FOR B-10-01 AND B-10-10 WITH VARIOUS r_{com}^{G} AND r_{min}

r_{min}	r_{com}^{G}	MA(a_1) Avg.±std.	HE-C	HE-B	MAR(a_2) Avg.±std.	VNS(a_3) Avg.±std.	Test result h^+
				B-10-01			
1	1	**367.45±19.46**	1560.24	1498.48	791.58±160.33	517.78±92.57	(1 1)
	5	**319.35±17.08**	1560.24	1263.79	822.72±166.87	473.83±95.52	(1 1)
	10	**291.84±29.61**	1560.24	1002.77	973.30±287.47	363.89±53.31	(1 1)
5	1	**422.26±35.79**	1997.64	1950.65	890.85±151.03	579.08±95.96	(1 1)
	5	**382.73±27.21**	1997.64	1712.12	981.15±246.33	575.79±73.55	(1 1)
	10	**369.99±50.09**	1997.64	1400.93	950.39±221.99	516.85±74.08	(1 1)
10	1	**501.47±26.95**	2598.54	2548.69	1100.74±226.41	766.16±88.03	(1 1)
	5	**464.03±41.49**	2598.54	2296.85	1085.11±164.85	851.11±105.59	(1 1)
	10	**430.55±56.55**	2598.54	1941.11	1100.60±204.94	740.16±100.67	(1 1)
				B-10-10			
1	1	**101.37±11.14**	407.54	225.92	247.26±49.79	140.99±7.76	(1 1)
	5	**84.68±14.58**	407.54	231.28	282.48±65.89	103.56±14.76	(1 1)
	10	**72.50±11.45**	407.54	231.74	325.51±101.39	102.98±14.99	(1 1)
5	1	**197.20±24.09**	644.86	533.75	430.10±77.84	383.46±58.45	(1 1)
	5	**223.49±20.84**	644.86	554.71	453.18±101.79	336.58±49.78	(1 1)
	10	**211.84±33.63**	644.86	427.23	490.63±122.30	303.43±30.61	(1 1)
10	1	**543.72±75.20**	1365.66	1116.97	780.21±112.20	866.41±235.66	(1 1)
	5	**498.72±40.36**	1365.66	893.48	750.71±110.40	758.87±118.28	(1 1)
	10	**464.64±62.89**	1365.66	851.43	873.96±277.93	714.15±107.29	(1 1)

Note that h^+ is consistent with that in Table III.

and VNS is only slightly superior to MA in the case of sparse distribution of UGVs. The video about the UAV's tour to visit ten UGVs obtained by different methods for instance B-10-02 is available at https://youtu.be/2UsZ10mxR0c.

Computation time of MA is higher than that of MAR. For small instance-scale, the difference in computation time for MA and MAR is not very significant. When the number of UGVs is less than 10, the computation time difference is less than 15s. However, the time difference tends to widen as the problem size increases, as shown in Fig. 5. Computation time for VNS is much higher than that for MA and MAR since the evaluations of candidate solutions are time-consuming in VNS.

B. Comparative Experiment on Random Instances With Different Neighborhood and Turning Radii

This subsection investigates the performance of all algorithms in DTSPDN instances with various $r_{com}^{G} \in \{1, 5, 10\}$ and $r_{min} \in \{1, 5, 10\}$. When r_{com}^{G} and r_{min} take a small value, DTSPDN is approximate to TSPMT. The longer r_{com}^{G} and r_{min} are, they exert greater influence on the tour length. Increasing the turning radius can enlarge the tour length while larger neighborhood radius can reduce tour length due to the distance savings. Both effects can be observed in Tables V by comparing $r_{min} = 1$ and $r_{min} = 10$ for the turning radius, and $r_{com}^{G} = 1$ and $r_{com}^{G} = 10$ for the neighborhood radius.

As shown in Table V, MA performs better than its competitors. For HE-B, when the turning radius (r_{min}) of the UAV is very small with respect to the distance between any two UGVs, the instance implies a weak coupling effect between the visiting sequence of the UAV and its visiting locations. The decoupling strategy used in HE-B fits the instances with the weak coupling effect. The weak coupling effect will disappear when r_{min} increases to a large value. Thus, the performance gap between MA and HE-B becomes larger with the increase of r_{min}. In addition, the method of determining the visiting sequence in HE-B is based on the Euclidean distance. As r_{min} increases, the difference between the Euclidean path and the Dubins path is significant. As a result, the performance of HE-B becomes worse with the increase of r_{min}. The solution quality of MA is also higher than that of MAR and VNS, with its advantages mainly reflected in the mean value and the standard deviation, especially when r_{min} is large ($r_{min} = 10$). In this case, DTSPDN exhibits strong coupling between visiting sequence, access locations and encounter time. MAR and VNS relax the coupling relationship between the encounter time and other decision variables, which results in longer tours.

C. Analysis About the Transformation Procedure

To test the effect of the transformation procedure, more experiments are carried out. The performance of MA without transformation procedure (MA-T) and MA is compared with the same initialization method and constraint handling strategy. MA outperforms MA-T as shown in Table VI. Thus, MA with the transformation procedure can improve the quality of solutions and leads to significant improvement in performance.

D. Parametric Sensitivity Analysis

The generation number to perform local search T_{lp} determines how frequently the local search procedure is executed. In the preceding experiments, T_{lp} is set to 2. In order to verify the effect of different T_{lp} on the algorithm performance,

TABLE VI
RESULTS OF THE COMPARISON BETWEEN MA AND MA-T

Instance	MA (Avg.,min,max,std.)	MA-T (Avg.,min,max,std.)
B-2-01	**(23.2, 7.6, 39.8, 7.5)**	(30.5, 23.7, 41.5, 5.5)
B-2-10	**(198.8, 195.7, 207.5, 2.9)**	(202.0, 195.7, 216.2, 5.4)
B-3-01	**(33.9, 10.0, 50.1, 10.0)**	(38.5, 28.4,55.0,6.4)
B-3-10	**(149.0, 136.4, 174.6, 13.5)**	(152.9, 136.6, 183.9, 2.4)
B-10-01	**(456.3, 388.5, 593.9, 47.6)**	(598.2, 419.9, 812.7, 9.1)
B-10-10	**(205.3, 161.7, 247.3, 24.8)**	(272.7, 196.7, 376.3, 5.4)
B-20-01	**(1.8, 1.3, 2.1, 0.2)**$\times 10^3$	(3.4, 2.4, 4.3, 0.5)$\times 10^3$
B-20-10	**(2.4, 1.7, 3.3, 0.5)**$\times 10^3$	(4.5, 2.3, 7.5, 1.3)$\times 10^3$
B-30-01	**(2.7, 2.0, 4.2, 0.6)**$\times 10^4$	(4.9, 2.6, 6.1, 0.8)$\times 10^4$
B-30-10	**(1.4, 0.8, 1.8, 0.2)**$\times 10^4$	(2.2, 1.3, 3.1, 0.5)$\times 10^4$
B-40-01	**(24.8, 16.9, 30.9, 4.3)**$\times 10^4$	(42.4, 28.3, 57.4, 6.9)$\times 10^4$
B-40-10	**(8.3, 5.9, 9.9, 1.3)**$\times 10^4$	(12.5, 8.3, 16.6, 2.1)$\times 10^4$

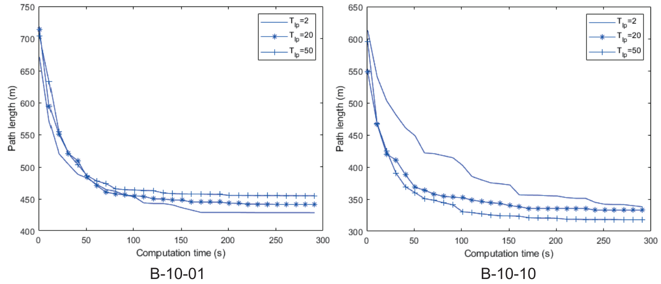

Fig. 7. Comparison of the tour length for $T_{lp} \in \{2, 10, 20\}$ over the computational time for instances B-10-01 and B-10-10.

set $T_{lp} = \{2, 10, 20\}$ to verify the effect on the algorithm performance.

The variants of instances B-10-01 and B-10-10 with different $T_{lp} = \{2, 10, 20\}$ are selected, and each algorithm runs 300s for 20 executions independently. Within each run, we store the objective value of the best individual thus far every 10s. The experiment is repeated under the same parameter setting with subsection V-A for instances B-10-1 and B-10-10.

Fig. 7 shows how the different T_{lp} values for instances B-10-01 and B-10-10 affect the algorithm coverage. The convergence curves indicate that the values of T_{lp} do not significantly affect the performance of MA. This means that MA is not very sensitive to the local search frequency.

VI. CONCLUSION

In this paper, a path planning problem, modeled as DTSPDN, that arises in dynamic coordination between UGVs and a messenger UAV is studied. By analyzing the characters of the DTSPDN, we introduce a novel encoding scheme and a corresponding effective decoding scheme. We then propose a novel memetic algorithm to effectively solve DTSPDN. In the memetic algorithm, a gradient-based repair strategy is used to repair individuals that violate dynamic constraints while a transformation procedure is also employed to find a better solution by analyzing the changes of feasible region caused by silent variants. On this basis, the memetic algorithm framework is used to achieve the exploration-exploitation trade-off by combining GA and two local search operators with different search logics. The computational results on random instances with different scales demonstrate that the proposed approach offers a high-quality solution in a reasonable time compared to four other competitive algorithms in the literature. In future work, we can adapt the memetic algorithm to plan tours for multiple messenger UAVs, allowing the messenger UAVs to provide relay services for each UGV with higher frequency of information update.

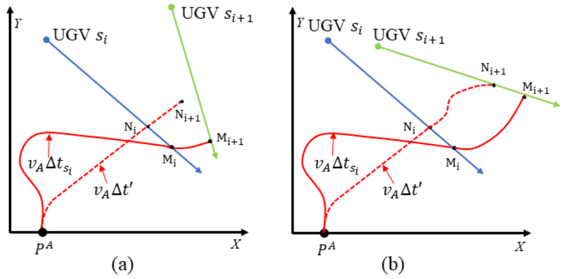

Fig. 8. Meeting pattern between UAV and UGV s_i. (a) Meeting pattern between UAV and UGV s_{i+1}; (b) Catch-up pattern between UAV and UGV s_{i+1}.

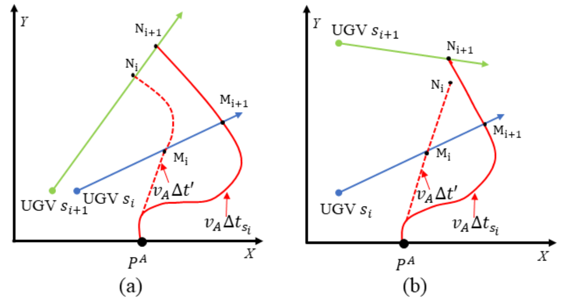

Fig. 9. Catch-up pattern between UAV and UGV s_i. (a) Catch-up pattern between UAV and UGV s_{i+1}; (b) Meeting pattern between UAV and UGV s_{i+1}.

APPENDIX A
PROOF OF PROPOSITION 1

Assume that the encounter pattern between UAV and UGV s_i is meeting, and the UAV encounters UGV s_i at the point M_i (see the curve $P^A M_i$ in Fig. 8) and encounters UGV s_{i+1} at the point M_{i+1} (see the curve $M_i M_{i+1}$ in Fig. 8). According to whether the encounter pattern between UAV and UGV s_{i+1} is meeting pattern or catch-up pattern, two cases are discussed as follows:

Case 1 (Meeting Pattern Between UAV and UGV s_{i+1}): If Δt_{s_i} is reduced to $\Delta t'$ with $0 < \Delta t' \leq \Delta t_{s_i}$, UAV will encounter UGV s_i at ahead of time. In this case, the length of the encounter path between UAV and UGV s_{i+1} will be larger than the length $|M_i M_{i+1}| = v_A \Delta t_{s_{i+1}}$ according to the definition of the meeting pattern in Definition 2, as shown in Fig. 8 (a). Therefore, the UAV cannot visit UGV s_{i+1} with $\Delta t_{s_{i+1}}$.

Case 2 (Catch-Up Pattern Between UAV and UGV s_{i+1}): If Δt_{s_i} is reduced to $\Delta t'$, the length of the encounter path between UAV and UGV s_{i+1} will be less than $|M_i M_{i+1}| = v_A \Delta t_{s_{i+1}}$ according to the definition of catch-up pattern in Definition 1. Therefore, UAV can still encounter UGV s_{i+1} with $\Delta t_{s_{i+1}}$, as shown in Fig. 8 (b).

In the same way, it can be easily verified that if the UAV encounters UGV s_i and UGV s_{i+1} with catch-up pattern and Δt_{s_i} is reduced to $\Delta t'$, then UAV can still encounter UGV s_{i+1} with $\Delta t_{s_{i+1}}$, as shown in Fig. 9(a). If the UAV encounters UGV s_i with catch-up pattern and UGV s_{i+1} with meeting pattern, then reducing Δt_{s_i} will lead to the fact that the UAV cannot encounter UGV s_{i+1} with $\Delta t_{s_{i+1}}$, as shown in Fig. 9(b).

Acknowledgment

The authors thank Dr. Robert Pěnička for his assistance in reproducing his VNS algorithm.

References

[1] J. Chen, X. Zhang, B. Xin, and H. Fang, "Coordination between unmanned aerial and ground vehicles: A taxonomy and optimization perspective," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 959–972, Apr. 2016.

[2] J. Li *et al.*, "A memetic path planning algorithm for unmanned air/ground vehicle cooperative detection systems," *IEEE Trans. Autom. Sci. Eng.*, early access, Mar. 10, 2021, doi: 10.1109/TASE.2021.3061870.

[3] Z.-H. Wang, K.-Y. Qin, T. Zhang, and B. Zhu, "An intelligent ground-air cooperative navigation framework based on visual-aided method in indoor environments," *Unmanned Syst.*, vol. 9, no. 3, pp. 237–246, Jul. 2021.

[4] T. Baca *et al.*, "Autonomous landing on a moving vehicle with an unmanned aerial vehicle," *J. Field Robot.*, vol. 36, no. 5, pp. 874–891, 2019.

[5] D. Yulong *et al.*, "Path planning of messenger UAV in air-ground coordination," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8045–8051, Jul. 2017.

[6] K. Peng *et al.*, "A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery," *IEEE Access*, vol. 7, pp. 49191–49200, 2019.

[7] H. Qin *et al.*, "Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1339–1350, Feb. 2019.

[8] Y. Ding, B. Xin, and J. Chen, "A review of recent advances in coordination between unmanned aerial and ground vehicles," *Unmanned Syst.*, vol. 9, no. 2, pp. 97–117, Apr. 2021.

[9] H. Huang *et al.*, "A probabilistic risk assessment framework considering lane-changing behavior interaction," *Sci. China Inf. Sci.*, vol. 63, no. 9, pp. 1–15, Sep. 2020.

[10] T. Wang, P. Huang, and G. Dong, "Modeling and path planning for persistent surveillance by unmanned ground vehicle," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1615–1625, Oct. 2021.

[11] W. Gao, J. Luo, W. Zhang, W. Yuan, and Z. Liao, "Commanding cooperative UGV-UAV with nested vehicle routing for emergency resource delivery," *IEEE Access*, vol. 8, pp. 215691–215704, 2020.

[12] W. Lu *et al.*, "Resource and trajectory optimization in UAV-powered wireless communication system," *Sci. China Inf. Sci.*, vol. 64, no. 4, pp. 1–14, Apr. 2021.

[13] Q. Song, Y. Zeng, J. Xu, and S. Jin, "A survey of prototype and experiment for UAV communications," *Sci. China Inf. Sci.*, vol. 64, no. 4, pp. 1–21, Apr. 2021.

[14] B. Xin, J. Chen, D.-L. Xu, and Y.-W. Chen, "Hybrid encoding based differential evolution algorithms for Dubins traveling salesman problem with neighborhood," *Control Theory Appl.*, vol. 31, no. 7, pp. 941–954, Jul. 2014.

[15] J. T. Isaacs, D. J. Klein, and J. P. Hespanha, "Algorithms for the traveling salesman problem with neighborhoods involving a Dubins vehicle," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 1704–1709.

[16] H. Zhang, L. Dou, B. Xin, J. Chen, M. Gan, and Y. Ding, "Data collection task planning of a fixed-wing unmanned aerial vehicle in forest fire monitoring," *IEEE Access*, vol. 9, pp. 109847–109864, 2021.

[17] S. G. Manyam, S. Rathinam, D. Casbeer, and E. Garcia, "Tightly bounding the shortest Dubins paths through a sequence of points," *J. Intell. Robotic Syst.*, vol. 88, nos. 2–4, pp. 495–511, Dec. 2017.

[18] J. Faigl, P. Vana, and J. Deckerova, "Fast heuristics for the 3-D multi-goal path planning based on the generalized traveling salesman problem with neighborhoods," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2439–2446, Jul. 2019.

[19] K. J. Obermeyer, P. Oberlin, and S. Darbha, "Sampling-based path planning for a visual reconnaissance unmanned air vehicle," *J. Guid., Control, Dyn.*, vol. 35, no. 2, pp. 619–631, Mar. 2012.

[20] J. Le Ny and E. Feron, "An approximation algorithm for the curvature-constrained traveling salesman problem," in *Proc. 43rd Annu. Allerton Conf. Commun., Control Comput.*, 2005, pp. 620–629.

[21] J. Faigl, P. Váňa, R. Pěnička, and M. Saska, "Unsupervised learning-based flexible framework for surveillance planning with aerial vehicles," *J. Field Robot.*, vol. 36, no. 1, pp. 270–301, Jan. 2019.

[22] J. Faigl and P. Vana, "Unsupervised learning for surveillance planning with team of aerial vehicles," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 4340–4347.

[23] Z. Yang *et al.*, "A double-loop hybrid algorithm for the traveling salesman problem with arbitrary neighbourhoods," *Eur. J. Oper. Res.*, vol. 265, no. 1, pp. 65–80, Feb. 2018.

[24] R. Pěnička, J. Faigl, M. Saska, and P. Váňa, "Data collection planning with non-zero sensing distance for a budget and curvature constrained unmanned aerial vehicle," *Auto. Robots*, vol. 43, no. 8, pp. 1937–1956, Dec. 2019.

[25] X. Zhang, J. Chen, B. Xin, and Z. Peng, "A memetic algorithm for path planning of curvature-constrained UAVs performing surveillance of multiple ground targets," *Chin. J. Aeronaut.*, vol. 27, no. 3, pp. 622–633, May 2014.

[26] Z. Chen, C.-H. Sun, X.-M. Shao, and W.-J. Zhao, "A descent method for the Dubins traveling salesman problem with neighborhoods," *Frontiers Inf. Technol. Electron. Eng.*, vol. 22, no. 5, pp. 732–740, May 2021.

[27] D. G. Macharet, A. A. Neto, V. F. da Camara Neto, and M. F. M. Campos, "Dynamic region visit routing problem for vehicles with minimum turning radius," *J. Heuristics*, vol. 24, no. 1, pp. 83–109, Feb. 2018.

[28] D. G. Macharet, A. A. Neto, V. F. da Camara Neto, and M. F. M. Campos, "Efficient target visiting path planning for multiple vehicles with bounded curvature," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3830–3836.

[29] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957.

[30] X.-N. Bui and J.-D. Boissonnat, "Accessibility region for a car that only moves forwards along optimal paths," Res. Rep. INRIA, Rocquencourt, France, Tech. Rep. 2181, 1994.

[31] Y. Ding, B. Xin, and J. Chen, "Curvature-constrained path elongation with expected length for Dubins vehicle," *Automatica*, vol. 108, Oct. 2019, Art. no. 108495.

[32] P. Moscato and M. G. Norman, "A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems," *Parallel Comput. Transputer Appl.*, vol. 1, pp. 177–186, Sep. 1992.

[33] M. R. Karim and C. Ryan, "Degeneracy reduction or duplicate elimination: An analysis on the performance of attributed grammatical evolution with lookahead to solve the multiple knapsack problem," in *Nature Inspired Cooperative Strategies for Optimization (NICSO)*. Berlin, Germany: Springer, 2011, pp. 247–266.

[34] I. Harvey and A. Thompson, "Through the labyrinth evolution finds a way: A silicon ridge," in *Proc. Int. Conf. Evolvable Syst.* Berlin, Germany: Springer, 1996, pp. 406–422.

[35] F. Caraffini, F. Neri, G. Iacca, and A. Mol, "Parallel memetic structures," *Inf. Sci.*, vol. 227, no. 4, pp. 60–82, 2013.

[36] A. Stieber, A. Fügenschuh, M. Epp, M. Knapp, and H. Rothe, "The multiple traveling salesmen problem with moving targets," *Optim. Lett.*, vol. 9, no. 8, pp. 1569–1583, Dec. 2015.

[37] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[38] N. Mladenović and P. Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997.

Yulong Ding received the bachelor's degree from the Qilu University of Technology, Jinan, China, in 2012, the master's degree from Xiamen University, Xiamen, China, in 2015, and the Ph.D. degree from the Beijing Institute of Technology, China, in 2020. He is currently working with the Peng Cheng Laboratory as a Post-Doctoral Researcher. His current research interests include UAV-UGV collaborative systems, heterogeneous multi-agent systems, and task planning of multirobot systems.

Bin Xin (Member, IEEE) received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively. He was an Academic Visitor with the Decision and Cognitive Sciences Research Centre, The University of Manchester, from 2011 to 2012. He is currently a Professor with the School of Automation, Beijing Institute of Technology. His current research interests include search and optimization, evolutionary computation, combinatorial optimization, and multi-agent systems. He is also an Associate Editor of the journal *Unmanned Systems*, the *Journal of Advanced Computational Intelligence and Intelligent Informatics*, and the *Advanced Control for Applications: Engineering and Industrial Systems* (Wiley).

Lihua Dou (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from the Beijing Institute of Technology, Beijing, China, in 1979, 1987, and 2001, respectively. She is currently a Professor of Control Science and Engineering with the Beijing Institute of Technology. Her current research interests include command and control, pattern recognition, and *ad hoc* networks.

Jie Chen (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in control theory and control engineering from the Beijing Institute of Technology in 1986, 1996, and 2001, respectively. From 1989 to 1990, he was a Visiting Scholar with California State University, USA. From 1996 to 1997, he was a Research Fellow with the School of E&E, University of Birmingham, U.K. He is currently a Professor of Control Science and Engineering with Tongji University, China. He is also an Academician with the Chinese Academy of Engineering. He has coauthored four books and more than 200 research articles. His main research interests include intelligent control and decision in complex systems, multi-agent systems, and optimization methods. He serves as the Managing Editor for the journal *Unmanned Systems* and the *Journal of Systems Science and Complexity* from 2014 to 2019, and an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS from 2016 to 2021 and many other international journals.

Ben M. Chen (Fellow, IEEE) is currently a Professor of Mechanical and Automation Engineering with The Chinese University of Hong Kong (CUHK) and a Professor of Electrical and Computer Engineering (ECE) with the National University of Singapore (NUS). He was a Provost's Chair Professor with the ECE Department, NUS, where he also served as the Director of Control, Intelligent Systems and Robotics Area, and the Head of the Control Science Group, NUS Temasek Laboratories. He was an Assistant Professor with the State University of New York at Stony Brook from 1992 to 1993. He has authored/coauthored more than 450 journal and conference papers, and 12 research monographs in control theory and applications, unmanned systems, and financial market modeling. His current research interests are in unmanned systems, robust control, and control applications.

Dr. Chen has received a number of research awards. His research team has actively participated in international UAV competitions and won many championships in the contests. He had served on the Editorial Board of 12 international journals, including IEEE TRANSACTIONS ON AUTOMATIC CONTROL and *Automatica*. He currently serves as the Editor-in-Chief for *Unmanned Systems*.