IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Data collection task planning of a fixed-wing unmanned aerial vehicle in forest fire monitoring

HAO ZHANG[1], LIHUA DOU[1,2], BIN XIN[1] (Member, IEEE), JIE CHEN[1,3] (Fellow, IEEE), MINGGANG GAN[1], and Yulong Ding[4]

[1]School of Automation, Beijing Institute of Technology, Beijing 100081, China
[2]Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing 100081, China
[3]State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China
[4]Peng Cheng Laboratory, Shenzhen 518052, China

Corresponding author: Bin Xin (e-mail: brucebin@bit.edu.cn), Minggang Gan (e-mail: agan@bit.edu.cn).

This work was co-supported by the National Outstanding Youth Talents Support Program (61822304), the Basic Science Center Programs of NSFC under Grant (62088101), the Peng Cheng Laboratory, the Consulting Research Project of the Chinese Academy of Engineering (2019-XZ-7), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant (61621063), the Projects of Major International (Regional) Joint Research Program of NSFC under Grant 61720106011.

ABSTRACT This article studies the data collection task planning for a fixed-wing unmanned aerial vehicle (UAV) in forest fire monitoring. Multiple wireless-based detection nodes (DNs) are distributed in high-risk areas of the forest to monitor the surrounding environment. The task of UAV is to circularly fly to them and collect the environmental data. Because of the kinematic constraints of UAV and the effective communication range between UAV and DN, this problem can be generally regarded as a Dubins traveling salesman problem with neighborhood (DTSPN). A bi-level hybridization-based metaheuristic algorithm (BLHMA) is proposed for solving this problem. At the first level, differential evolution (DE) optimizes the continuous-valued communication positions and UAV headings by the population-based search. For the asymmetric traveling salesman problem (ATSP) corresponding to the combination of the positions and headings generated by DE, a constructive heuristic based on self-organized multi-agent competition (SOMAC) is proposed to determine the discrete collection sequence. By competitive iterations in such a cooperative way in DE, a high-quality data collection tour can be generated. At the second level, a local search based on multistage approximate gradient is proposed to further refine the positions and headings, which accelerates the convergence of the BLHMA. Referring to a real-world scene of forest fire mornitoring, the simulation experiments are designed, and comparative results show that BLHMA can find significantly shorter data collection tours in most cases over three state-of-the-art algorithms. The proposed UAV data collection planning algorithm is conducive to the efficient execution of the forest fire monitoring data collection mission and the energy saving of UAV.

INDEX TERMS Unmanned aerial vehicle, Forest fire monitoring, Data collection task planning, Mixed-variable optimization, Differential evolution, Constructive heuristic

I. INTRODUCTION

FOREST fire is one of the most destructive disasters to the ecological environment, and the forest fire protection is receiving more and more attention from the governments of the world [1]. Thus, it is very meaningful to carry out prevention and control of the forest fire, and how to make use of high-tech means to monitor forest fire has become a hot topic [2], [3].

At present, there are mainly two kinds of high-tech means for forest fire monitoring. The first high-tech means are based on the unmanned aerial vehicle (UAV) surveillance. It is easier for a fixed-wing UAV to get a large range of ground information because of its top-view and high mobility [4], [5]. It has been increasingly applied in monitoring missions, which can be regarded as curvature-constrained path planning problems. In some research, the path planning of UAV is simplified as the traveling salesman problem (TSP) [6]–[10]. However, it is very important to consider the kinematic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

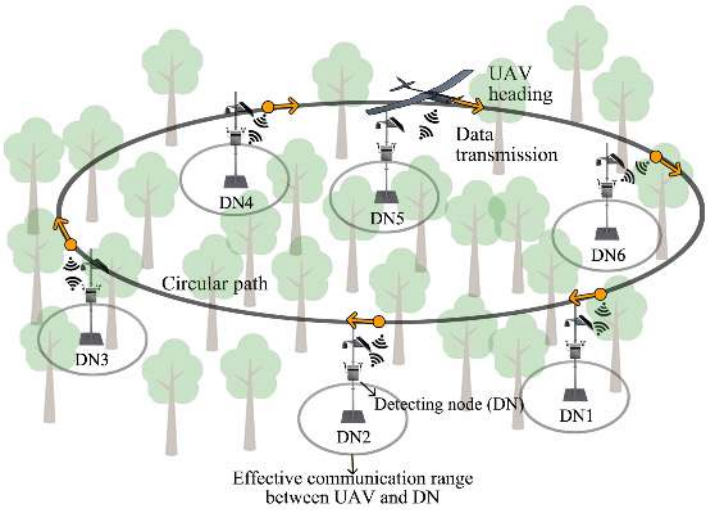


FIGURE 1. UAV data collection task in forest fire monitoring

constraints from the practical aspects, and the Dubins vehicle model is usually used to approximate the dynamic model of UAV [11]–[14]. In the path planning, a UAV circularly collects images above the forest and detects the fire source through image analysis [15]–[18]. The second high-tech means are based on wireless sensor network (WSN). Multiple wireless sensors are distributed in the forest and form a monitoring network [19]. These sensors continuously monitor the surrounding environment and collect environmental data to achieve forest fire monitoring.

Both means have advantages and disadvantages. For the UAV-based means, a UAV can quickly obtain a large range of ground information, but its monitoring effectiveness is greatly affected by the image quality, which is easily affected by weather factors, such as dust, fog, etc. In addition, in the early stage of the forest fire, the fire is so small that there is no obvious burning phenomenon, so it is difficult to monitor the fire at a high altitude. For the WSN-based means, WSN can quickly detect the possible fire area in the early stage of the fire by monitoring the temperature, humidity, and harmful gases. But in the deep forest of the mountainous area, the transmission signal may be very poor, and a large number of sensors need to be arranged to cover the forest. This will bring great economic costs and maintenance difficulties.

This paper combines the advantages of the two means. Multiple wireless-based, low-power, and self-sustaining detection nodes (DNs) are distributed in high-risk areas of the forest to perceive the surrounding environment and record environmental data. Because of their poor signal transmission condition in the mountain forest, they have limited communication capability and can only perform short-range communication. Thus, a small, low-altitude, and low-cost fixed-wing UAV which is equipped with a transceiver device is adopted to circularly collect the latest data of DNs, as shown in Fig. 1.

At the beginning of the forest fire monitoring, the command center will plan a data collection tour for the UAV, and command the UAV to circularly collect the latest data from all DNs to obtain the information of monitoring areas. When the UAV arrives at the planned collection position of each DN, it sends an application to the DN to receive the latest data. After receiving the application, the DN sends the data to the UAV. Then the UAV transmits the data back to the command center, and flies to the next planned collection position.

In this sense, UAV needs to enter the communication range of each DN with an appropriate sequence to collect data, and its flight path is constrained by curvature, so the data collection task planning for a UAV in forest fire monitoring can be generally formulated as a Dubins traveling salesman problem with neighborhood (DTSPN) [20], [21].

A. RELATED WORK

As mentioned above, the UAV data collection task planning in forest fire monitoring is formulated as a DTSPN which can be regarded as a generalized variant of the classic TSP. In addition, there are two other generalized variants of TSP. On one hand, if the traveling salesman in TSP becomes a mobile agent which can be treated as a Dubins vehicle, the resulting TSP variant is called Dubins TSP (DTSP) [22], [23]. In the DTSP, the distance between two neighboring targets to be visited is not Euclidean distance. It should be measured by the Dubins paths which rely not only on the visiting positions but also on the headings of the vehicle at the positions [12]. On the other hand, if the targets to be visited are extended from points to regions, TSP will be transformed into its well-known variant, named the traveling salesman problem with neighborhood (TSPN) [24]–[26]. DTSPN shares the characteristics of DTSP and TSPN, and also inherits their difficulties.

Similar to DTSP and TSPN, the DTSPN is also an NP-hard optimization problem with mixed variables [27], [28].

Solving the DTSPN involves optimizing the discrete collection sequence, the continuous communication positions, and the UAV headings simultaneously, which makes it more challenging.

The existing methods for solving this class of problems can be mainly grouped into four categories: decoupling methods, transformation methods, unsupervised learning methods, and direct search methods. In the decoupling methods, mixed variables are optimized separately [11], [22], [28]–[31]. To some extent, the decoupling method reduces the difficulty of solving the problem. However, the effectiveness of decoupling methods mainly relies on the similarity between problems, which makes them unsuitable for the situations where the Euclidean distance between two points is obviously shorter than the minimal turning radius of UAV when solving DTSP and DTSPN [32]. In the transformation methods, positions and headings are uniformly sampled to construct the generalized traveling salesman problem (GTSP) firstly, and then the GTSP is converted into an asymmetric traveling salesman problem (ATSP) by the Noon-Bean transformation [33]–[37]. The transformation method can obtain a global optimum when taking a large number of samples, but it greatly increases the amount of calculations. Decoupling methods and transformation methods share the similarity that an efficient TSP solver like the LKH can be used to provide a high-quality solution [38]. In the unsupervised learning methods, the solution of the sequencing part of the problem is combined with the online sampling of the suitable positions and headings [39], [40]. The unsupervised learning method can be regarded as a kind of constructive algorithm, which can find a solution with high efficiency. However, the initialization and competition of neurons have a great influence on the solution in different cases. In the direct search methods, the mixed variables are optimized simultaneously and many optimization algorithms can be used as search engine [26], [32], [41]–[46]. In this method, the model has no sacrifice in accuracy, but the solution space is complex, which makes it difficult to find the optimal solution.

Many scholars devote themselves to the research of the DTSPN. Obermeyer proposed a genetic algorithm (GA) that uses a hybrid encoding scheme for mixed variables [20]. In addition, the authors also proposed the other two methods to solve a polygon visiting Dubins traveling salesman problem (PVDTSP), and both of them belong to the transformation method [36], [37]. Under certain technical assumptions, the algorithms are resolution-complete, and can converge to a global optimum as the number of samples grows but consumes more computing time. Macharet proposed a three-stage evolutionary method that optimizes mixed variables independently [44]. The authors also proposed a heuristic method for optimizing visiting positions within the target's neighborhood and connecting them with Dubins paths [47], [48]. Pěnička proposed a variable neighborhood search (VNS) algorithm for solving the Dubins Orienteering Problem with Neighborhoods (DOPN), which can be regarded as a variant of the DTSPN [45]. A set of neighborhood operators are designed to perform a well combinatorial optimization. Zhang proposed a memetic algorithm in which a relaxed DTSPN model is used to obtain an approximated optimal solution, and a local search based on an approximated gradient is adopted to improve the quality of the solution [32].

B. MAIN CONTRIBUTIONS

In view of two typical high-tech means in forest fire monitoring, this paper combines their advantages, and regards the UAV data collection task in forest fire monitoring as a DTSPN. Because the DTSPN is a mixed-variable optimization problem with extremely complex solution space, this paper deliberates the algorithm design to reduce the complexity of problem solving, and makes the following innovations and contributions:

1) In the proposed bi-level hybridization-based metaheuristic algorithm (BLHMA), differential evolution (DE) is adopted to evolve a group of continuous communication positions and UAV headings which are metaphorize as multiple individuals. Besides, a constructive heuristic based on self-organized multi-agent competition (SOMAC) is proposed to generate the discrete collection sequence by solving an asymmetric traveling salesman problem (ATSP) corresponding to the combination of the positions and headings generated by DE. In such a cooperative way, a data collection tour can be generated. Because SOMAC is a deterministic constructive heuristic, it avoids a large amount of blind search in the mixed-variable space and leads to better convergence. Compared with the transformation method [36], the huge amount of computation caused by a large number of samplings is avoided.

2) A local search based on a multistage approximate gradient is developed to improve the communication positions and UAV headings during the iteration of DE, which can lead to a better trade-off between exploration and exploitation in the solution space. Compared with the strategy proposed in the literature which does not optimize UAV headings [32], the proposed local search can achieve precise adjustment of the communication positions and UAV headings. This is helpful to accelerate the convergence of the BLHMA.

3) This paper provides an effective method for the UAV data collection task planning in forest fire monitoring, which is conducive to the efficient execution of the task and the energy saving of UAV.

This papaer organized as follows. Section II presents the formulation of the UAV data collection task planning. Section III gives a detailed description of the proposed algorithm. Section IV evaluates the proposed algorithm through a series of computational experiments and analyses. Section V concludes the paper.

II. PROBLEM FORMULATION

In this paper, it is assumed that multiple wireless-based, low-power, and self-sustaining DNs are deployed in high-risk areas of the forest. Because of their poor signal transmission condition in the mountain forest, DNs can only

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

TABLE 1. Parameters in the formulation of the UAV data collection task planning

Symbol	Description
n	Number of DNs.
Ω_i	The neighborhood of the i^{th} DN, $i = 1, 2, \cdots, n$.
Ω	$\Omega = \{\Omega_1, \Omega_2, \cdots, \Omega_n\}$ is the set of neighborhoods of DNs.
s_i	The index of the i^{th} DN to be inspected, $\forall i \neq j$: $s_i \neq s_j$, $i, j = 1, 2, \cdots, n$.
$\mathbf{S}$	$\mathbf{S} = [s_1, s_2, \cdots, s_n]$ is the collection sequence of all DNs.
Y_i	$Y_i = [x_i, y_i, \theta_i]$ is the Dubins state of UAV when approaching to the i^{th} DN, $i = 1, 2, \cdots, n$, $[x_i, y_i] \in \Omega_i$, $\theta_i \in [0, 2\pi)$.
$\mathbf{Y}$	$\mathbf{Y} = [Y_1, Y_2, \cdots, Y_n]$ is the vector of Dubins states of UAV when approaching to all DNs.
$d(\cdot, \cdot)$	Dubins distance between two Dubins states.
$J(\cdot, \cdot)$	Objective function.

perform short-range communication. Each DN can continuously monitor the temperature, humidity, and harmful gas of the surrounding environment, and record environmental data. Then, a fixed-wing UAV circularly inspects each DN to obtain the monitoring data.

In this situation, neighborhoods of DNs should be defined to fit the effective communication range between UAV and the DN. For the sake of clarity, the neighborhood of each DN is specified as a disk centered at the DN. For the UAV, the Dubins vehicle model is used to approximate its kinematic model [12], and the model is described as follows:

$$\begin{cases} \dot{x} = \upsilon \cdot \cos\theta, \\ \dot{y} = \upsilon \cdot \sin\theta, \\ \dot{\theta} = \frac{\upsilon}{r} \cdot u, u \in \{-1, 0, 1\}. \end{cases} \tag{1}$$

where $[x, y]$ and θ are the planar coordinates and the heading of the Dubins vehicle, respectively; $[x, y, \theta]$ is called Dubins state; υ and r represent the speed and the minimum turning radius of the Dubins vehicle, respectively; u denotes the control input that controls the movement direction of the Dubins vehicle.

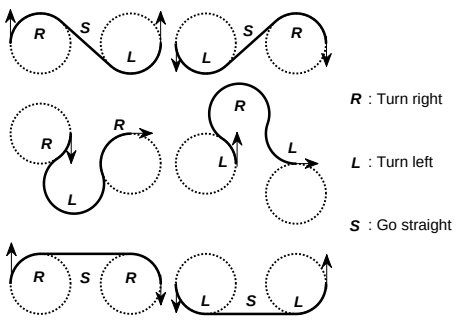


FIGURE 2. The six possible shortest Dubins paths

Based on the model, Dubins gave an important conclusion that there are at most six possible shortest paths for any transition from one Dubins state to another [12], as shown in Fig. 2.

Based on the above description, the optimization model of the UAV data collection task planning is shown as (2). Detailed descriptions of parameters and variables used in the model are shown in Table 1.

$$min\ J(\mathbf{S}, \mathbf{Y}) = \sum_{i=1}^{n-1} d(Y_{s_i}, Y_{s_{i+1}}) + d(Y_{s_n}, Y_{s_1}) \tag{2}$$

By using the boundary-based encoding scheme [32], a Dubins state can be further simplified from $[x_i, y_i, \theta_i]$ to $[\varphi_i, \theta_i]$, φ_i and $[x_i, y_i]$ can be calculated from each other through coordinate transformation, as shown as follow:

$$\begin{aligned} x_i &= c_{x_i} + R \cdot \cos\varphi_i \\ y_i &= c_{y_i} + R \cdot \sin\varphi_i \end{aligned} \tag{3}$$

where $[c_{x_i}, c_{y_i}]$ is the position of the i^{th} DN, and R is the radius of its neighborhood.

In this way, the optimization scale is effectively reduced from $4n$ to $3n$. However, mixed-variable optimization with a scale of $3n$ is still a great challenge.

Remark 1: In order to simulate the UAV data collection planning in forest fire monitoring with different scales and scenarios, instances with different minimum distance constraints D_K are considered in this paper [49], including D_4 and D_1-constraint instances, and more intensive ones with overlapping situations, denoted by D^*. In D_4-constraint instances, the distribution of high-risk areas is relatively sparse, and all DNs are far apart. In D_1-constraint instances and D^*-constraint instances, the distribution of high-risk areas is relatively dense, and DNs are relatively close. The definition of the minimum distance constraint is that: for all $i, j \in \{1, 2, \cdots, n\}$, $i \neq j$, $\forall p_i \in \Omega_i$, $\forall p_j \in \Omega_j$, and $\|p_i - p_j\| > K \cdot r$, where r is the minimum turning radius of UAV. The bigger the value of K is, the closer the UAV data collection planning problem is to the common ATSP. In this sense, the decoupling strategy is more suitable for instances in which DNs are far away from each other. On the contrary, it is not suitable for instances with dense DNs distribution.

III. BI-LEVEL HYBRIDIZATION-BASED METAHEURISTIC ALGORITHM

In this section, a bi-level hybridization-based metaheuristic algorithm (BLHMA) is proposed to solve the UAV data collection task planning in forest fire monitoring. An outline of the BLHMA is illustrated in Fig. 3. First of all, only the communication positions and UAV headings are encoded

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

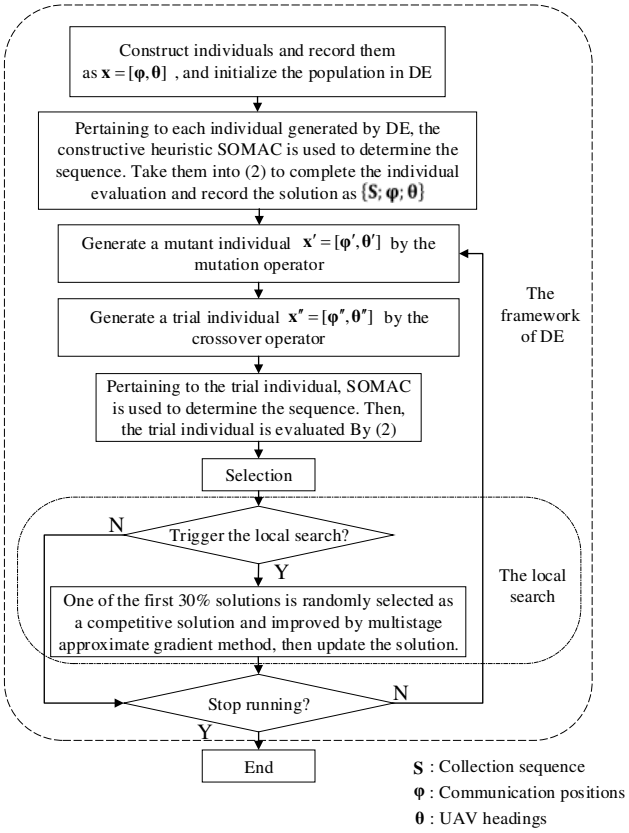


FIGURE 3. The outline of the BLHMA

to form multiple individuals in the initialization of DE. By using the information of an individual, including communication positions and UAV headings, a Dubins distance matrix that represents Dubins distances between all DNs can be calculated to form an ATSP. While the ATSP is addressed by SOMAC to obtain the collection sequence. Then a data collection tour is obtained by introducing the collection sequence, communication positions, and UAV headings into (2). With the iteration of the BLHMA, communication positions and UAV headings are constantly updated in the process of DE, and new ATSPs are constantly constructed and handled, which leads to new data collection tours. Then, a high-quality data collection tour can be obtained by the way of iterative competition.

Moreover, to achieve a desirable trade-off between exploration and exploitation of the algorithm, a modified local search based on the previous work [32] is integrated after the selection operator of DE, which includes three stages: the full-dimensional and local-dimensional approximate gradient descents, and the diversification mechanism. As mentioned above, the so-called BLHMA is constructed, and the details are described as follows.

A. THE OPTIMIZATION OF COMMUNICATION POSITIONS AND UAV HEADINGS BY DIFFERENTIAL EVOLUTION

In this paper, DE is used to optimize communication positions and UAV headings, and the following gives a detailed introduction from the aspects of the encoding, decoding, mutation, crossover, and selection.

Remark 2: Generally, GA is good at dealing with discrete search domain. DE is an efficient and effective global optimizer in the continuous search domain, and its variants also have excellent performance [50]. Compared with classical population-based optimizers, such as GA, particle swarm optimization (PSO), etc., DE has outstanding performance in many studies [51], [52]. Besides, DE and its variants also have been applied to the international contest on evolutionary optimization (ICEO) many times, and they all have excellent performance.

1) Encoding and decoding

Because when UAV flies to a DN to collect data, it must pass through the boundary of the DN's neighborhood, so only the communication positions and headings of the UAV at each DN's neighborhood need to be determined. Based on the boundary-based encoding scheme described in Fig. 4, any position and heading on the boundary can be expressed by the polar angle, and the coordinates of the communication position can be transformed by using (3).

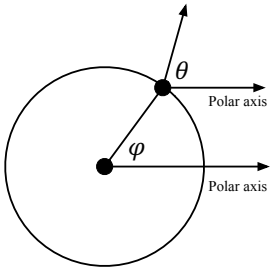


FIGURE 4. Schematic of the encoding scheme

Based on the encoding scheme, an individual can be described as (4), a population is formed by randomly generating multiple individuals.

$$\mathbf{x}_i = [\varphi_i^1, \varphi_i^2, \cdots, \varphi_i^n, \theta_i^1, \theta_i^2, \cdots, \theta_i^n] \tag{4}$$

where $i = 1, 2, \cdots, NP$, and NP is the population size. φ and θ represents the encoding of the communication positions and UAV headings respectively, and they are generated randomly in the range of $[0, 2\pi)$.

As for the decoding, in addition to communication positions and UAV headings, the collection sequence is also needed, and all of them form a solution, as (5) shows. Take them together into (2) to obtain the data collection tour and complete the individual evaluation. The next section will introduce how to use SOMAC to get the collection sequence.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

$$SOL_{\mathbf{x}_i} = \left\{ \begin{array}{c} \mathbf{S}_{\mathbf{x}_i} \\ \boldsymbol{\varphi}_{\mathbf{x}_i} \\ \boldsymbol{\theta}_{\mathbf{x}_i} \end{array} \right\} = \left\{ \begin{array}{c} s^1_{\mathbf{x}_i}, s^2_{\mathbf{x}_i}, \cdots, s^n_{\mathbf{x}_i} \\ \varphi^1_{\mathbf{x}_i}, \varphi^2_{\mathbf{x}_i}, \cdots, \varphi^n_{\mathbf{x}_i} \\ \theta^1_{\mathbf{x}_i}, \theta^2_{\mathbf{x}_i}, \cdots, \theta^n_{\mathbf{x}_i} \end{array} \right\} \quad (5)$$

2) Mutation operator

After the decoding, all individuals have been evaluated. Then DE employs the differential mutation to produce a mutant vector $\mathbf{u}_i$ with respect to different individuals in the current population. In this paper, an advanced DE variant, called SADE [50], is adopted, and four kinds of mutation operators are selected.

1) DE / rand / 1

$$\mathbf{u}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (6)$$

2) DE / current-to-rand / 1

$$\mathbf{u}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (7)$$

3) DE / best / 1

$$\mathbf{u}_i = \mathbf{x}_{best} + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}) \quad (8)$$

4) DE / best / 2

$$\mathbf{u}_i = \mathbf{x}_{best} + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}) + F \cdot (\mathbf{x}_{r3} - \mathbf{x}_{r4}) \quad (9)$$

where $r1, r2, r3$ and $r4$ are mutually exclusive integers that are randomly generated from the set $\{1, 2, \cdots, NP\}$, and all of them are different from i. F is the scaling factor. x_{best} is the best individual found so far.

3) Crossover operator

After the mutation, a crossover operator, shown as (10), is developed to produce a trial vector $\mathbf{v}_i$.

$$v_{i,d} = \left\{ \begin{array}{l} u_{i,d}, \ if \ rand_i^d \leq CR \ or \ d = rn_i \\ x_{i,d}, \ otherwise \end{array} \right. \quad (10)$$

where $d = 1, 2, \cdots, 2n$. CR is the crossover probability, and it is usually set as a fixed value or dynamically adjusted within the interval $(0, 1)$. $rand_i^d$ is a random number within the interval $(0, 1)$. rn_i is a number randomly selected from the set $\{1, 2, \cdots, 2n\}$, and it is used for ensuring that $\mathbf{v}_i$ and $\mathbf{x}_i$ are different.

4) Selection operator

After the evaluation of the trial individual $\mathbf{v}_i$, the selection operation will be performed. If the data collection tour calculated by the trial individual is not worse than that of the original individual $\mathbf{x}_i$, the original individual will be replaced and the trial individual will survive to the next generation. Otherwise, the original individual will enter the next generation. The selection operation is shown as (11). In addition, the solution also needs to be updated, as shown as (12). It is noted that, the evaluation of the trial individual need the sequence $(\mathbf{S}_{\mathbf{v}_i})$ obtained by SOMAC which will be introduced in the next section.

$$\mathbf{x}_i = \left\{ \begin{array}{l} \mathbf{v}_i, \ if \ J(\mathbf{S}_{\mathbf{v}_i}, \mathbf{v}_i) \leq J(\mathbf{S}_{\mathbf{x}_i}, \mathbf{x}_i) \\ \mathbf{x}_i, \ otherwise \end{array} \right. \quad (11)$$

$$SOL_{\mathbf{x}_i} = \left\{ \begin{array}{l} SOL_{\mathbf{v}_i}, \ if \ J(\mathbf{S}_{\mathbf{v}_i}, \mathbf{v}_i) \leq J(\mathbf{S}_{\mathbf{x}_i}, \mathbf{x}_i) \\ SOL_{\mathbf{x}_i}, \ otherwise \end{array} \right. \quad (12)$$

where $J(\cdot)$ is the objective function of the data collection planning problem, as shown in (2).

B. COLLECTION SEQUENCE OBTAINED BY SOMAC

Solving the collection sequence based on the communication positions and UAV headings generated by DE is essentially an ATSP problem. Black arrows on the boundary of each DN's neighborhood shown in Fig. 5 represents the communication positions and UAV headings at all DNs. The positions and headings are used to calculate a Dubins distance matrix, as (13) shows, which leads to an ATSP. Then, the ATSP is solved by SOMAC.

$$\mathbf{D} = \left[\begin{array}{ccccc} inf & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & inf & d_{23} & \cdots & d_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & inf \end{array} \right] \quad (13)$$

where $d_{ij} = d(Y_i, Y_j)$, $i, j = 1, 2, \cdots, n$, $i \neq j$. d_{ij} is the Dubins distance from the i^{th} DN to j^{th} DN, which is calculated by the positions and headings at the i^{th} DN and j^{th} DN.

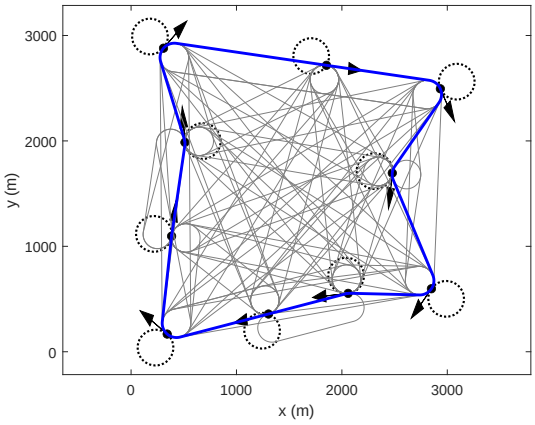


FIGURE 5. Solving the collection sequence based on the communication positions and UAV headings

For solving the ATSP, a directed graph $G = (V, A)$ is constructed, where V is the vertex set and A is the arc set. The mathematical model of the ATSP is shown as follows [53]:

$$\min \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} d_{ij} \cdot y_{ij}, \quad (14)$$

$$s.t. \sum_{i=1, i \neq j}^{n} y_{ij} = 1, i = 1, 2, \cdots, n, \quad (15)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

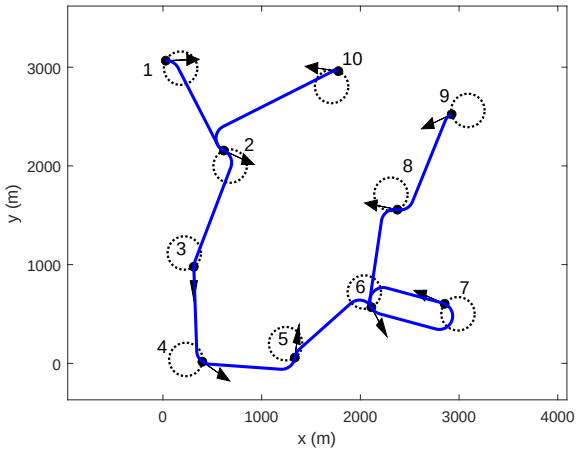


FIGURE 6. Connection of agents after selection

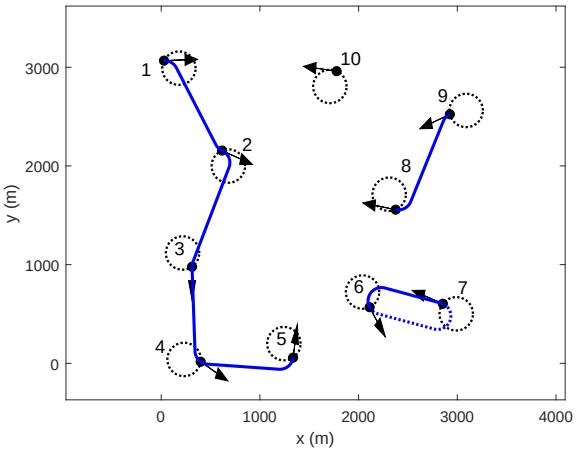


FIGURE 7. Connection of agents after conflict resolution

$$\sum_{j=1,j\neq i}^{n} y_{ij} = 1, j = 1, 2, \cdots, n, \tag{16}$$

$$\sum_{i\in K}\sum_{j\in K, j\neq i} y_{ij} \leq |K| - 1, K \subsetneqq V, K \neq \varnothing \tag{17}$$

$$y_{ij} \in \{0, 1\}, i, j = 1, 2, \cdots, n, i \neq j. \tag{18}$$

where d_{ij} is associated with $arc_{i,j} \in A$. y_{ij} is equal to 1 if $arc_{i,j}$ is selected. Equations (15)-(16) guarantee that every DN is inspected exactly once. K is a subset of V and (17) makes sure that there is no local loop.

Then, the solution of the data collection planning can be obtained by solving the ATSP. Figure 5 also shows the solution of the simulation scenario of the UAV data collection in forest fire monitoring. The ATSP can be regarded as an inner-loop subproblem embedded in DE for determining the collection sequence. Here, an efficient constructive heuristic is expected to address it, rather than an iterative algorithm with a large amount of computations. Thus, SOMAC is proposed to handle the ATSP.

In SOMAC, the Dubins state of UAV at each DN's neighborhood shown in Fig. 5 is regarded as an independent agent. Those agents connect with each other through self-organized competition. It is noted that each agent can only connect to other agents once, and it can only be connected once. Because the connection among all agents is simultaneous, it is possible that a certain agent is connected by many other agents. In addition, some agents may connect with each other, which leads to a local loop. A local objective function is designed to eliminate the above conflicts. The details are shown as follows.

1) Initialization of all agents

Because each agent can only connect to other agents once, and it can only be connected once, an outgoing edge and an incoming edge are defined for each agent. At the beginning, all agents have neither outgoing edges nor incoming edges.

2) Local objective function and selection operation

Each agent will gather information from other agents, and select the one with the shortest Dubins distance. The local objective function of the connected agent can be denoted as follows:

$$f_{a_j}(C^j, a_j) = \begin{cases} inf, & C^j = \varnothing \\ d_{c_q^j a_j}, & C^j \neq \varnothing \end{cases} \tag{19}$$

where a_j stands for the j^{th} agent, $C^j = \{c_1^j, c_2^j, \cdots, c_{m_j}^j\}$ is the set of agents that can connect to a_j. m_j is the number of members in C^j. $c_q^j \in C^j, q = 1, 2, \cdots, m_j$.

In determining C^j of a_j, three kinds of agents need to be excluded, including itself, the agents with incoming edges, and the agents that can cause a local loop.

3) Conflict resolution operation

Because the connection among all agents is simultaneous, a certain agent may be connected by multiple agents. The agent will be disconnected proactively from its connected agents one by one according to the local objective function value until the conflict is resolved.

An example is shown in Fig. 6. Agent 2 is connected by agents 1 and 10, and agent 6 is connected by agents 5, 7, and 8. According to (19), for agent 2, two candidate incoming edges can be obtained. Then, agent 1 leads to the best local objective function value of agent 2. Thus, the connection between agents 1 and 2 is retained, and agents 2 and 10 are disconnected. The conflict resolution of agents 5⌣8 is similar. Fig. 7 shows the connection of agents after conflict resolution.

It can be observed that there is a local loop between agents 6 and 7, then the longest Dubins path (dotted line) in the loop is disconnected.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

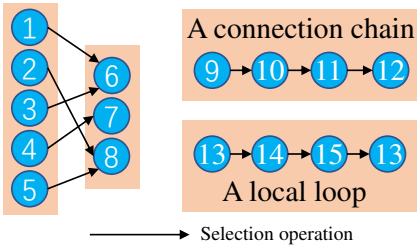
4) Termination criteria

When all the agents have selected two agents without conflict, a complete directed graph emerges, and the complete data collection tour can be obtained by sequentially connecting the Dubins path in the digraph.

Based on the above, the agents can link with each other to form a feasible solution, and the collection sequence is obtained. Then the sequence is combined with the positions and headings generated by DE in the previous section, and an individual $\mathbf{x}_i$ can be evaluated.

5) Pseudo-code of the SOMAC

In order to understand SOMAC more intuitively, some key sets are defined to store agents with different connection states and some concepts are explained in Fig. 8.



S_{NI} = {1,2,3,4,5,9} represents the set of agents that are not connected
S_{NO} = {6,7,8,12} represents the set of agents that are not connected to others
S_{con} = {6,8} represents the set of agents that are connected many times
S_{lp}={10,11,13,14,15} represents the set of agents that are neither in S_{NI} nor in S_{NO}

FIGURE 8. Connection of agents after selection

The pseudo-code of the SOMAC is presented in Algorithm 1. In line 1, the sets S_{NI} and S_{NO} reflect the connection status of all agents. Every agent in S_{NI} is not connected by any other agent, and every agent in S_{NO} is not connected to any other agent. In line 5, L_i is the i^{th} connection chain which reflects the connection between agents. Take Fig. 7 as an example, $\{a_1, a_7, a_9, a_{10}\} \in S_{NI}$, there are four connection chains. In the first chain, the agent 1 serve as a starting of a connection chain, and the connection is completed when the next connected agent is in S_{NO}. Then, a connection chain L_1 is formed, like $L_1 = \{a_1, a_2, a_3, a_4, a_5\}$. In this way, each agent in S_{NI} corresponds to a connection chain. The other three chains are $L_2 = \{a_7, a_6\}$, $L_3 = \{a_9, a_8\}$, and $L_4 = \{a_{10}\}$. It is noted that, all agents are in S_{NI} at the beginning of the SOMAC, and a single agent can be regarded as a special connection chain.

In line 9, for an agent a_i in S_{NO}, three kinds of agents cannot be connected to, which is mentioned in the selection operation. It should be noted that when the selection operation shown in lines 8∽10 is executed for the first time, all agents are in S_{NI} and S_{NO}. Because they connect to others simultaneously, they can connect to any other agents except for themselves. However, from the second execution of lines 8-10, some agents are already connected together. By analyzing these connections, when an agent without an outgoing edge performing the selection operation, the agents that have

Algorithm 1 Heuristic based on self-organized multi-agent competition (SOMAC) for solving the collection sequence

1: Two sets S_{NI} and S_{NO} are initialized, and both of them include all agents;
2: **while** S_{NI} and S_{NO} are not empty **do**
3: Find all agents in S_{NI}, and record the number of agents in S_{NI} as M.
4: **for** $i = 1 : M$ **do**
5: Built the connection chain L_i among agents, and the chain starts with the agent in S_{NI} and ends with the agent in S_{NO}.
6: **end for**
7: Find all agents in S_{NO}, and record the number of agents in S_{NO} as K.
8: **for** $i = 1 : K$ **do**
9: For the agent a_i, find the agents that a_i cannot connect to, and from the rest of agents, find the agent with the shortest Dubins distance to connect. Then, remove a_i from S_{NO}.
10: **end for**
11: Remove the connected agents from S_{NI}
12: Find the agents that have been connected by multiple agents, and form the set S_{con}. The size of the set is J.
13: **for** $i = 1 : J$ **do**
14: For the agent a_i in S_{con}, among multiple connections, find the one leading to the shortest Dubins distance. Then, disconnect the other connections, and put the corresponding agents back into S_{NO}.
15: **end for**
16: Find the agents that are neither in S_{NI} nor in S_{NO}, and form the set S_{lp}.
17: **while** S_{lp} is not empty **do**
18: For an agent in S_{lp}, starting from it, the connection chain is gradually built.
19: **if** the connection chain is looped and does not contain all agents **then**
20: Disconnect the two agents in the loop with the longest Dubins Distance and put them back into S_{NI} and S_{NO}.
21: **end if**
22: Find agents that exist in both the connection chain and S_{lp}, and remove them from S_{lp}.
23: **end while**
24: **end while**
25: Get the connection chain of all agents, and output the collection sequence.

incoming edges or cause local loops can be excluded. In line 16, because the agents that are neither in S_{NI} nor in S_{NO} are either in a chain or in a loop, local loops can be judged by the connection chains constructed by these agents.

The time complexity of the SOMAC is analyzed as follows. Denote the number of agents by n. In the selection operation, when constructing all connection chains shown in lines 4∽6, all agents will be retrieved once. And for the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

operation shown in lines 8∽10, the selection operation can be executed n times at most. Thus, the time complexity of the selection operation is $O(n)$. In the conflict resolution operation, the maximum size of the set S_{con} is $n-1$, so the time complexity of this operation is $O(n)$. In the local loop resolution operation, the maximum size of S_{lp} is n, and with the construction of the connection chain, the size of S_{lp} is gradually reduced to zero. Thus, the time complexity of this operation is also $O(n)$.

The time complexity of the SOMAC mainly depends on the number of loop execution shown in lines 2∽24. There are two possibly worst cases. In the first case, all agents connect to the same agent when executing the selection operation. After the conflict and local loop resolution, at least one connection can be determined. In this case, the number of loop execution is $n-1$, and the time complexity of the SOMAC is $O(n^2)$. In the second case, agents are always connected with each other and lead to the occurrence of local loop. In this case, the maximum value of the number of loop execution is not more than n, and the time complexity of the SOMAC is also $O(n^2)$. It should be noted that in either case, the probability of its occurrence is very small.

C. LOCAL SEARCH

In the process of population evolution, the local search is triggered at a certain interval T, and a competitive tour is selected to be improved [32]. Then, the full-dimensional approximate gradient descent, the diversification mechanism, and the local-dimensional approximate gradient descent act on this tour sequentially. In this paper, the communication positions and UAV headings are optimized simultaneously. First, a full-dimensional approximate gradient descent is proposed to improve all the communication positions and UAV headings, which can make a full dimensional improvement.

When the local optimal Dubins path shown in Fig. 9 appears, the diversification mechanism and the local approximate gradient descent are applied to each component of these variables. In this way, the communication positions and UAV headings are further improved, and the quality of the data collection tour is further enhanced.

1) Full-dimensional approximate gradient descent

An approximate gradient on a certain dimension of continuous variables can be calculated by two solutions, as shown in the following example:

$$SOL_{\hat{\mathbf{x}}} = \left\{ \begin{array}{c} \mathbf{S}_{\hat{\mathbf{x}}} \\ \boldsymbol{\varphi}_{\hat{\mathbf{x}}} \\ \boldsymbol{\theta}_{\hat{\mathbf{x}}} \end{array} \right\} = \left\{ \begin{array}{l} s^1_{\hat{\mathbf{x}}}, s^2_{\hat{\mathbf{x}}}, \cdots, s^n_{\hat{\mathbf{x}}} \\ \varphi^1_{\hat{\mathbf{x}}}, \varphi^2_{\hat{\mathbf{x}}}, \cdots, \varphi^n_{\hat{\mathbf{x}}} \\ \theta^1_{\hat{\mathbf{x}}}, \theta^2_{\hat{\mathbf{x}}}, \cdots, \theta^n_{\hat{\mathbf{x}}} \end{array} \right\} \tag{20}$$

$${}^*SOL_{\hat{\mathbf{x}}} = \left\{ \begin{array}{c} \mathbf{S}_{\hat{\mathbf{x}}} \\ {}^*\boldsymbol{\varphi}_{\hat{\mathbf{x}}} \\ \boldsymbol{\theta}_{\hat{\mathbf{x}}} \end{array} \right\} = \left\{ \begin{array}{l} s^1_{\hat{\mathbf{x}}}, s^2_{\hat{\mathbf{x}}}, \cdots, s^n_{\hat{\mathbf{x}}} \\ \varphi^1_{\hat{\mathbf{x}}}, \cdots, \varphi^q_{\hat{\mathbf{x}}} + \Delta\sigma, \cdots, \varphi^n_{\hat{\mathbf{x}}} \\ \theta^1_{\hat{\mathbf{x}}}, \theta^2_{\hat{\mathbf{x}}}, \cdots, \theta^n_{\hat{\mathbf{x}}} \end{array} \right\} \tag{21}$$

where $SOL_{\hat{\mathbf{x}}}$ is a competitive solution, and ${}^*SOL_{\hat{\mathbf{x}}}$ is formed by adding a perturbation to $SOL_{\hat{\mathbf{x}}}$. ${}^*\boldsymbol{\varphi}_{\hat{\mathbf{x}}}$ is the position vector after the perturbation. The two solutions have different communication positions. $\Delta\sigma$ is a small and fixed real number which is used as a small perturbation to calculate the approximate gradient of each dimension of continuous variables.

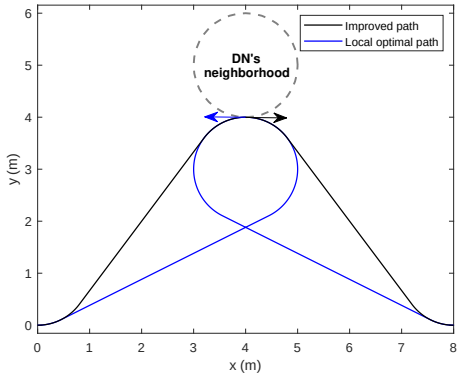


FIGURE 9. Local optimal Dubins path

Then, an approximate gradient of the q^{th} dimension of positions can be calculated by (22).

$$g_q \approx \frac{J(\mathbf{S}_{\hat{\mathbf{x}}}, (\boldsymbol{\varphi}_{\hat{\mathbf{x}}}, \boldsymbol{\theta}_{\hat{\mathbf{x}}})) - J(\mathbf{S}_{\hat{\mathbf{x}}}, ({}^*\boldsymbol{\varphi}_{\hat{\mathbf{x}}}, \boldsymbol{\theta}_{\hat{\mathbf{x}}}))}{\Delta\sigma},\ q \in \{1, 2, \cdots, n\} \tag{22}$$

Based on (20)-(22), the gradient vector of the $2n$ continuous variables, as shown in (23), can be obtained by calculating the approximate gradient on the n positions and n headings gradually.

$$\mathbf{g} = [g_1, g_2, \cdots, g_{2n}] \tag{23}$$

After deriving the approximate gradient vector, the corresponding update equation of the $2n$ continuous variables is shown as follow:

$$[\boldsymbol{\varphi}_{k+1}, \boldsymbol{\theta}_{k+1}] = [\boldsymbol{\varphi}_k, \boldsymbol{\theta}_k] - \rho \cdot \frac{\mathbf{g}}{|\mathbf{g}|} \tag{24}$$

where ρ is the step size, and k is the iteration number. The iteration goes on until the tour no longer turns better.

2) Local-dimensional approximate gradient descent

Due to the full-dimensional approximate gradient descent has strong directionality, the local optimal Dubins path caused by an improper heading shown in Fig. 9 may exist. The diversification mechanism is adopted firstly, and the reverse operation is performed for each dimension of the UAV headings to jump out of the local optimal Dubins path. Then the updated $SOL_{\hat{\mathbf{x}}}$ is obtained, and the local-dimensional approximate gradient descent is adopted to it. Because the position and heading at a certain DN have the greatest influence on the Dubins paths connected to the adjacent DNs, only the adjacent DNs are considered to reduce the computational cost.

Thus, a local solution $SOL^{local}_{\hat{\mathbf{x}},}$ shown as (25) is considered, which can lead to a local data collection path by using (2). For a pair of position and heading $[\varphi^j_{\hat{\mathbf{x}},}, \theta^j_{\hat{\mathbf{x}},}]$ at the j^{th} DN, an approximate gradient vector can be obtained by the following.

Taking the first dimension of $[\varphi^j_{\hat{\mathbf{x}},}, \theta^j_{\hat{\mathbf{x}},}]$ as an example, a perturbation is added, as shown in (25)-(26).

$$SOL^{local}_{\hat{\mathbf{x}},} = \left\{ \begin{array}{l} \mathbf{S}^{local}_{\hat{\mathbf{x}},} \\ \boldsymbol{\varphi}^{local}_{\hat{\mathbf{x}},} \\ \boldsymbol{\theta}^{local}_{\hat{\mathbf{x}},} \end{array} \right\} = \left\{ \begin{array}{l} s^{j-1}_{\hat{\mathbf{x}},}, s^{j}_{\hat{\mathbf{x}},}, s^{j+1}_{\hat{\mathbf{x}},} \\ \varphi^{j-1}_{\hat{\mathbf{x}},}, \varphi^{j}_{\hat{\mathbf{x}},}, \varphi^{j+1}_{\hat{\mathbf{x}},} \\ \theta^{j-1}_{\hat{\mathbf{x}},}, \theta^{j}_{\hat{\mathbf{x}},}, \theta^{j+1}_{\hat{\mathbf{x}},} \end{array} \right\} \quad (25)$$

$$^*SOL^{local}_{\hat{\mathbf{x}},} = \left\{ \begin{array}{l} \mathbf{S}^{local}_{\hat{\mathbf{x}},} \\ {}^*\boldsymbol{\varphi}^{local}_{\hat{\mathbf{x}},} \\ \boldsymbol{\theta}^{local}_{\hat{\mathbf{x}},} \end{array} \right\} = \left\{ \begin{array}{l} s^{j-1}_{\hat{\mathbf{x}},}, s^{j}_{\hat{\mathbf{x}},}, s^{j+1}_{\hat{\mathbf{x}},} \\ \varphi^{j-1}_{\hat{\mathbf{x}},}, \varphi^{j}_{\hat{\mathbf{x}},} + \Delta\sigma, \varphi^{j+1}_{\hat{\mathbf{x}},} \\ \theta^{j-1}_{\hat{\mathbf{x}},}, \theta^{j}_{\hat{\mathbf{x}},}, \theta^{j+1}_{\hat{\mathbf{x}},} \end{array} \right\} \quad (26)$$

where $^*SOL^{local}_{\hat{\mathbf{x}},}$ is formed by adding a perturbation to $SOL^{local}_{\hat{\mathbf{x}},}$, and $^*\boldsymbol{\varphi}^{local}_{\hat{\mathbf{x}},}$ is the position vector after the perturbation.

Then, the gradient of the first dimension of $[\varphi^j_{\hat{\mathbf{x}},}, \theta^j_{\hat{\mathbf{x}},}]$ can be calculated by (27).

$$g_{\varphi^j_{\hat{\mathbf{x}},}} \approx \frac{J(\mathbf{S}^{local}_{\hat{\mathbf{x}},}, (\boldsymbol{\varphi}^{local}_{\hat{\mathbf{x}},}, \boldsymbol{\theta}^{local}_{\hat{\mathbf{x}},})) - J(\mathbf{S}^{local}_{\hat{\mathbf{x}},}, ({}^*\boldsymbol{\varphi}^{local}_{\hat{\mathbf{x}},}, \boldsymbol{\theta}^{local}_{\hat{\mathbf{x}},}))}{\Delta\sigma} \quad (27)$$

Based on the above, the gradient of $\theta^j_{\hat{\mathbf{x}},}$ can be calculated in the same way. The gradient vector $\mathbf{g}^{local} = [g_{\varphi^j_{\hat{\mathbf{x}},}}, g_{\theta^j_{\hat{\mathbf{x}},}}]$ of position and heading on the j^{th} DN is formed, and $[\varphi^j_{\hat{\mathbf{x}},}, \theta^j_{\hat{\mathbf{x}},}]$ can be updated by (28).

$$[\varphi^j_{\hat{\mathbf{x}},}(k+1), \theta^j_{\hat{\mathbf{x}},}(k+1)] = [\varphi^j_{\hat{\mathbf{x}},}(k), \theta^j_{\hat{\mathbf{x}},}(k)] - \rho \cdot \frac{\mathbf{g}^{local}}{|\mathbf{g}^{local}|} \quad (28)$$

Remark 3: It is easy to see from (24) and (28) that the upper bound of the iteration number is $\lfloor 2\pi/\rho \rfloor$, where $\lfloor \cdot \rfloor$ is the downrounding function.

The time complexity of the local search is analyzed below. For the full-dimensional approximate gradient, the gradients on $2n$ continuous variables need to be sequentially calculated, so the time complexity of this part is $O(2n\lfloor 2\pi/\rho \rfloor)$. For the diversification mechanism, the reverse operations are executed n times for the n-dimension headings, so the time complexity of this part is $O(n)$. For the local-dimensional approximate gradient, for each DN, it needs to calculate the gradients on φ and θ. Thus, the time complexity of this part is $O(2n\lfloor 2\pi/\rho \rfloor)$. Based on the above, because the three parts are executed sequentially, the time complexity of the local search can be regarded as $O(n)$.

D. COMPLEXITY ANALYSIS OF THE BLHMA

The pseudo-code of the BLHMA is presented in Algorithm 2, and the time complexity and space complexity are analyzed in Table 2. The evaluation operation has the highest time complexity because the time complexity of the construction of the Dubins distance matrix is $O(n^2)$. G is the number of iterations, and $mod(\cdot)$ is the modulo operation. When the algorithm satisfies the criterion $mod(G, T) = 0$, the local search is triggered, and the number of runs of the local search is G/T. Based on the above, the time complexity and the space complexity of the BLHMA are $O(G \cdot NP \cdot n^2)$ and $O(n \cdot NP + n^2)$, respectively.

Algorithm 2 Bi-level hybridization-based metaheuristic algorithm (BLHMA) for the UAV data collection planning

1: Initialize a population based on (4);
2: Pertaining to each individual generated by DE, SOMAC is used to determine the sequence which mentioned in subsection III-B. Take them into (2) to complete the individual evaluation, and record the solution as (5);
3: Set the number of iterations $G = 1$
4: **while** the stopping criterion is not satisfied **do**
5: **for** $i = 1 : NP$ **do**
6: According to the adaptive selection strategy mentioned in the literature [50], mutate $\mathbf{x}_i$ by (6)-(9), then a trial vector $\mathbf{v}_i$ is generated by the crossover based on (10);
7: Evaluate the trial vector $\mathbf{v}_i$ by using the methods in line 2;
8: Update the population based on the selection operator shown as (11), and update the solution by (12);
9: **end for**
10: **if** $mod(G, T) == 0$ (T is a constant) **then**
11: All solutions are sorted, and one of the first 30% solutions is randomly selected as the competitive solution. Then the multistage local search mentioned in subsection III-C is conducted to improve its parts of positions and headings;
12: Re-evaluate the improved positions and headings by using the methods in line 2, and update the corresponding solution;
13: **end if**
14: $G = G + 1$
15: **end while**
16: Output the best solution and the corresponded data collection tour.

TABLE 2. Complexity of the BLHMA

Operation	Time Complexity	Space Complexity
Initialization	$O(1)$	$O(NP \cdot n + n^2)$
Evaluating	$O(G \cdot NP \cdot n^2)$	$O(n)$
Mutation	$O(G \cdot NP)$	$O(n)$
Crossover	$O(G \cdot NP \cdot n)$	$O(n)$
Selection	$O(G \cdot NP)$	$O(1)$
Local search	$O(G \cdot n)$	$O(n)$

IV. COMPUTATIONAL EXPERIMENTS

For the UAV data collection planning in forest fire monitoring, it is assumed that the installation height of each DN is 5m, and the maximum communication distance of DN is 200m. A small and low-altitude UAV cruises 110m

above the ground at a speed of 40m/s, and its maximum communication distance is more than 200m. UAV circularly flies to all DNs to obtain their data, as Fig. 10 shows, and the effective communication radius (R) between the UAV and a DN can be calculated by simple geometric relation, which is approximately equal to 170m. The schematic diagram is shown in Fig. 11. Besides, assume that the maximum roll angle of the UAV is 50°. According to the force analysis about the UAV turning, (29) can be obtained, then by using (30), the turning radius (r) of the UAV is approximately equal to 137m.

$$\frac{\mathcal{F}_{hori}}{\mathcal{F}_{vert}} = \frac{\mathcal{L} \cdot sin(\beta)}{\mathcal{L} \cdot cos(\beta)} = \frac{M \cdot \frac{V^2}{r}}{M \cdot \mathfrak{g}} = tan(\beta) \tag{29}$$

$$r = \frac{V^2}{\mathfrak{g} \cdot tan(\beta)} \tag{30}$$

where $\mathcal{F}_{hori}$ and $\mathcal{F}_{vert}$ are the force of the UAV in the horizontal and vertical direction. $\mathcal{L}$ is the lift force of the UAV. V and M are the speed and weight of the UAV, $\mathfrak{g}$ is the gravitational acceleration, and β is the maximum roll angle.

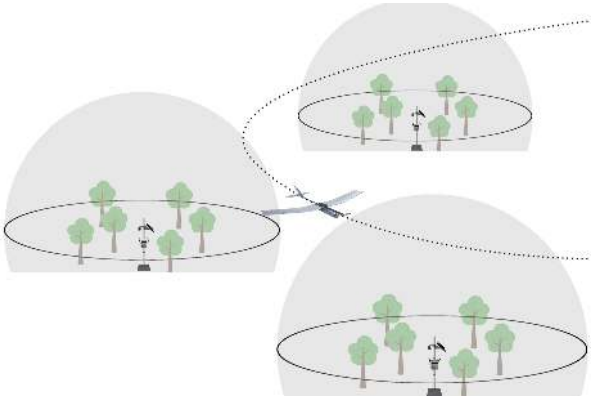


FIGURE 10. UAV obtains data from multiple DNs

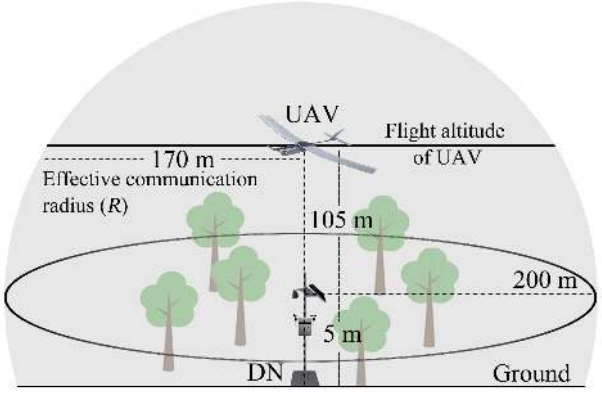


FIGURE 11. Transmission diagram of UAV and a DN

Based on the above assumptions, to verify the performance of the BLHMA in solving the UAV data collection planning, two-part computational experiments are conducted in this section. In addition, three state-of-the-art algorithms identified in the literature are involved in the computational experiments (denoted by transformation method [36], VNS [45], and MA [32], respectively).

Firstly, general analyses on a series of random instances with different minimum distance constraints are given, including D_4, D_1, and D^*-constraint instances. Secondly, the convergence performance of different algorithms is carried out, and the mechanisms of the BLHMA are analyzed.

All algorithms were implemented on a personal computer with Intel(R) Xeon(R) Silver 4114 CPU 2.20GHz, 32GB RAM. BLHMA, VNS, and MA were implemented by the MATLAB R2019b. For the transformation method, a GTSP was formed by the uniform sampling of communication positions and UAV headings firstly, and then the GTSP was converted to an ATSP by the Noon-Bean transformation [36], which was implemented by the MATLAB. At the last, LKH Ver.2.0.9 was applied for solving the ATSP to obtain a data collection tour, which was implemented by the Visual Studio 2019 (C++). Thus, for the fairness of comparison, the solution obtained by the transformation method is only regarded as a reference value. The code of LKH can be downloaded from the website: http://akira.ruc.dk/~keld/research/LKH/.

The parameter settings of the BLHMA are shown in Table 3, and the selection of some key parameters is analyzed. Here, the framework of the SADE is adopted [50], and the four mutation operators shown in (6)-(9) are selected. In order to ensure the best performance of the SADE, the setting of all parameters in SADE was consistent with the original literature, except for population size NP. NP was left as a user-specified parameter because it highly depends on the complexity of a given problem.

TABLE 3. The main parameters of the BLHMA

Symbols	Description	Setting
NP	Population size	n
ρ	Step size of (24) and (28)	0.2
$\Delta\sigma$	The value of the perturbation of (21) and (26)	0.05
T	The trigger frequency of the local search	5

To find a suitable value of NP, three typical values were tried. NP with the settings of n, $10n$, and $20n$ were tried in the instances with different scales. BLHMA was run 20 times, and the length of the so-far-best tour is stored every 5s. Then the mean value at each time is calculated. The results are shown in Figs. 12-14. In the 10-DNs instance, when NP is set to n, although the ability of the algorithm to jump out of the local optimal solution is not strong, the algorithm has a good balance between convergence and performance, as shown in the displayed curve. Similarly, good results are achieved in the 20 or 30-DNs instances when NP is set to n.

In the local search, $\Delta\sigma$ is used to construct approximate gradients, and it can usually be set to a small and fixed positive real number. We tested many different values of $\Delta\sigma$ ranging from 0.01% of 2π to 2π, and found that when its

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

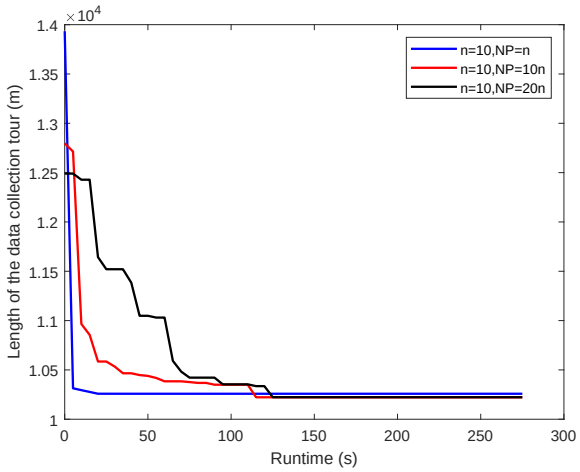


FIGURE 12. Convergence curves of the BLHMA under different population sizes on 10-DNs instance

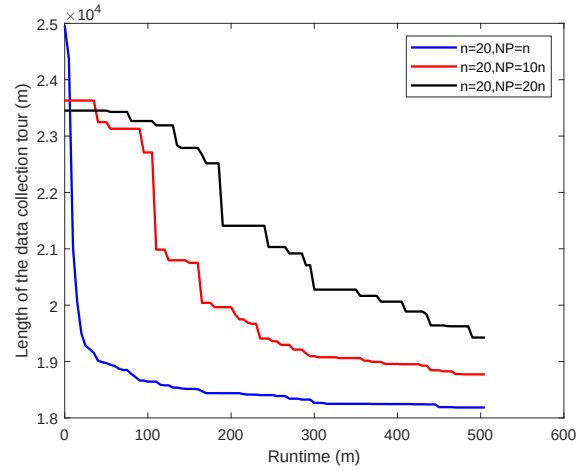


FIGURE 13. Convergence curves of the BLHMA under different population sizes on 20-DNs instance

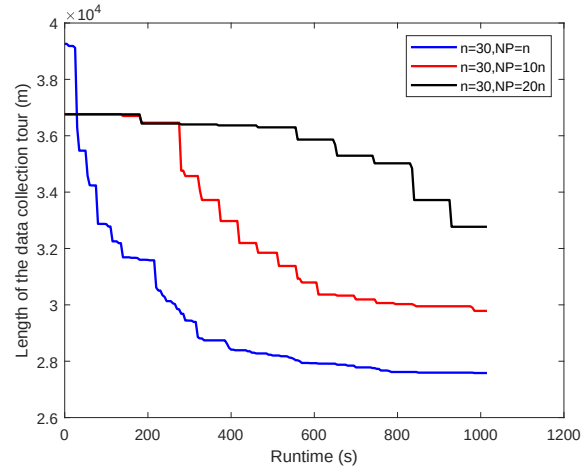


FIGURE 14. Convergence curves of the BLHMA under different population sizes on 30-DNs instance

value is less than 1% of 2π, its value has little effect on the performance of the local search. ρ and T usually have a great impact on the performance of the BLHMA, and both of them were determined as follows.

In order to determine the step size (ρ) in the local search, different step sizes were used to improve a poor data collection tour on the 10-DNs instance. The local search was run for each step value, and Fig. 15 shows the values of the length of the data collection tour and the runtime. When the value of ρ is small, the improvement effect of the local search is remarkable, but it takes a long time. On the contrary, the runtime is very short, but the search capability of the local search is poor. It can be seen that when ρ is set as 0.2, the local search has a good tradeoff between the runtime and performance.

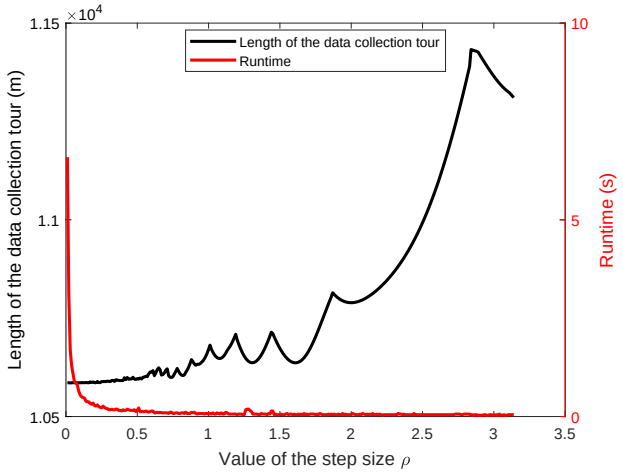


FIGURE 15. Influence of the step size of the local search

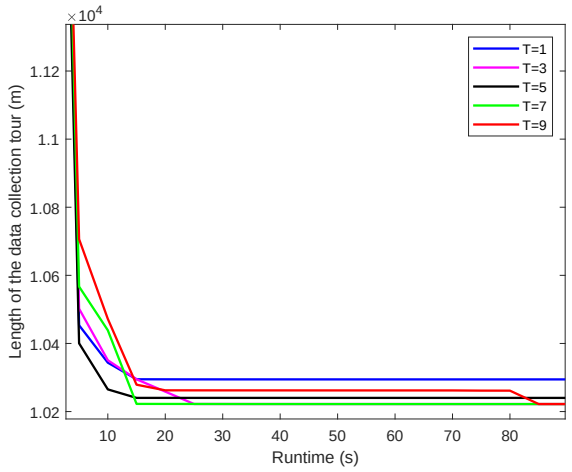


FIGURE 16. Comparison of different execution frequencies of the local search strategy in BLHMA

After determining the step size, the execution frequency of the local search was analyzed, and it also affects the convergence of the BLHMA. It can be adjusted to achieve

a better tradeoff between exploration and exploitation in the solution space by changing the value of T. A comparison of different T of the BLHMA on the 10-DNs instance is shown in Fig. 16. Similarly, BLHMA was run 20 times, and the length of the so-far-best tour is stored every 5s. Then the mean value at each time is calculated. When the value of T is small, the local search is frequently triggered, and the algorithm tends to exploitation, which makes it easy to fall into a local optimal solution. When the value of T is large, the trigger frequency of the local search becomes lower, the algorithm tends to explore, and the ability to jump out of the local optimal solution becomes stronger, but it takes more time to converge. As Fig. 16 shows, when T takes the value of 5, the comprehensive performance of the algorithm is better.

After analyzing the parameter setting of the BLHMA, the parameter setting of the competitors is explained as follows. For the transformation method, in order to get high-quality solutions, the highest sample counts are used according to the numerical study of Obermeyer [36]. The sample counts of the transformation method are set as 1100, 1600, and 2100 corresponding to the 10, 20, and 30-DNs instances, respectively. The mean runtimes on these three kinds of instances are 210s, 420s, and 850s, respectively. In addition, in order to get a better balance between position samples and heading samples in the designed instances in this paper, the weighting parameter which determines how many heading samples there will be per position is set to 68. For the other algorithms, the parameter settings are consistent with the original works of literature, and the maximum runtime is defined as the stop condition. Those algorithms will be terminated when the time of the transformation method is exhausted. It is worth noting that in order to make MA generate the cyclic data collection tour, the optimization of the position and UAV heading at the first DN is considered without changing the key framework and the idea of the algorithm.

In addition, to verify the performance of the SOMAC algorithm, the nearest neighbor rule (NNR) is used which can also solve the ATSP quickly and stably [54], and the bi-level hybridization-based metaheuristic algorithm with the nearest neighbor rule (BLHMA-NNR) is formed. Because the time complexity of the SOMAC and NNR is less than that of calculating the Dubins distance matrix, the influence of replacing SOMAC with NNR on the time complexity of the collection sequence obtaining can be ignored.

A. COMPARISON IN RANDOM INSTANCES WITH DIFFERENT MINIMUM DISTANCE CONSTRAINTS AND SCALES

A series of random simulated UAV data collection instances with different minimum distance constraints and scales are designed and used for analyzing the performance of all algorithms. The tested instances are labelled as $n - i$, where n is the number of DNs and i is the identifier of instances. Similar to the study [49], all DNs are randomly generated inside a bounding box with the side $7\sqrt{n}((R + r)/2)$, which can provide a relatively high density. The five algorithms were applied to solve the 27 instances with different minimum distance constraints and scales, and the results were compared. For each instance, the statistical results of all algorithms are presented in Table 4.

BLHMA, BLHMA-NNR, VNS, and MA were run 20 times, and the processing time of each algorithm is the same among all 20 attempts. Then, the length of the data collection tours can be calculated by the objective function, as shown in (2). The results were evaluated from four aspects including the mean value (avg), the maximum value (max), the minimum value (min), and the standard deviation (std). The Wilcoxon rank-sum test is also used for analyzing the significance of the difference of results [55], and the confidence level is set as 0.95. Besides, the data collection tours obtained by the transformation method are regarded as references.

From the statistical results, for most instances, BLHMA is the best one as it can find high-quality data collection tours in all aspects as compared with MA, VNS, and even the reference value. BLHMA-NNR also performed well in most instances. For D_4-constraint instances, when the number of DNs is small, BLHMA, BLHMA-NNR, and VNS can find better tours than that of the reference, and BLHMA and BLHMA-NNR outperform other algorithms with minor advantages. By comparing BLHMA-NNR and BLHMA, it can be seen that their performances are comparable in most D_4 instances, and the minimum length values of tours obtained by BLHMA-NNR is only slightly inferior to that of the BLHMA.

With the increase of the number of DNs and the enhancement of the constraint, in most cases, data collection tours found by MA are better than that of the VNS. In addition, tours obtained by BLHMA are obviously better than those of its competitors in all aspects, including BLHMA-NNR and the other three algorithms. It can be observed that the difference between BLHMA and BLHMA-NNR is not significant from the Wilcoxon rank-sum test results, but BLHMA can find better tours than BLHMA-NNR in most cases.

By summarizing the results of the experiments on the simulated instances of the UAV data collection task in forest fire monitoring, it can be obviously observed that MA is hard to find a satisfactory data collection tour in most cases. Because the Dubins path is relaxed, resulting in incomplete solution space. VNS can achieve a good balance between exploration and exploitation by using a neighborhood structure composed of different actions to the mixed variables. However, due to the large amount of optimization variables, the solution space is still very complex. Supported by an efficient solver for the ATSP, the sampling-based transformation method can find a better approximate optimal data collection tour. However, the extensive calculation caused by a large number of samplings usually consumes a large amount of time, and uniform sampling is not conducive to finding the optimal tour. In contrast, BLHMA proposed in this paper can decompose the solution space and search cooperatively in different subspaces, which significantly reduces the difficulty of searching, and it can ob-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

TABLE 4. Statistical results about different algorithms in solving randomly generated instances

Instance	BLHMA	BLHMA-NNR	MA [32]	VNS [45]	Reference [36]
	D_4-constraint instances: $(avg\ (m), max\ (m), min\ (m), std)$				(m)
10-1	(**10247.7** 10710.5 **10223.3 108.9**)	(10295.8 **10585.7 10223.3** 148.7)$^{=}$	(10697.1 11208.4 10645.1 153.8)$^{\Downarrow}$	(10900.2 12557.8 10247.4 666.9)$^{\Downarrow}$	10468.8
10-2	(10781.0 11114.2 10593.8 183.8)	(**10682.6 11059.0 10593.0 162.8**)$^{\Uparrow}$	(11131.1 11408.5 10995.1 165.8)$^{\Downarrow}$	(11270.7 12097.9 10616.2 514.7)$^{\Downarrow}$	10606.8
10-3	(10544.4 **10988.6** 10267.1 246.4)	(**10542.7** 11125.4 **10266.9** 285.1)$^{=}$	(10889.8 11175.9 10745.2 **129.0**)$^{\Downarrow}$	(10961.0 12099.6 10278.5 480.5)$^{\Downarrow}$	10316.1
20-1	(**19162.5 20179.1 18511.7** 423.2)	(19463.8 20338.7 18512.9 **375.4**)$^{\Downarrow}$	(22456.8 23401.0 21259.3 617.2)$^{\Downarrow}$	(22999.0 25228.4 20975.2 1199.0)$^{\Downarrow}$	18556.9
20-2	(**19766.9 20530.9 19490.0** 322.2)	(19843.3 20578.8 19490.1 **308.9**)$^{=}$	(22045.7 23320.1 20761.0 584.5)$^{\Downarrow}$	(25337.9 27362.2 22301.0 1423.9)$^{\Downarrow}$	19524.5
20-3	(20210.8 21439.6 **19559.4** 617.5)	(**20106.9 21436.8** 19560.0 **530.6**)$^{=}$	(22816.4 24517.0 21689.5 835.1)$^{\Downarrow}$	(24597.7 27524.7 21997.3 1631.7)$^{\Downarrow}$	19762.2
30-1	(**28767.0** 31075.6 **27592.9** 911.0)	(29028.0 **30409.1** 28107.4 **760.2**)$^{=}$	(38310.8 43590.3 34140.6 2578.3)$^{\Downarrow}$	(38047.7 42417.1 35213.5 1946.1)$^{\Downarrow}$	27600.3
30-2	(**29376.5** 31094.8 **28054.0** 882.3)	(29567.7 **30549.0** 28472.5 **607.7**)$^{=}$	(37493.6 41643.2 33711.6 2194.5)$^{\Downarrow}$	(39171.6 43823.8 35318.6 2587.7)$^{\Downarrow}$	28126.0
30-3	(27073.4 29010.7 **25868.9** 958.3)	(**27064.9 28839.5** 25871.4 **784.8**)$^{=}$	(37043.4 39698.4 33150.8 1716.3)$^{\Downarrow}$	(35755.0 38480.2 32616.0 1630.4)$^{\Downarrow}$	25896.9
	D_1-constraint instances: $(avg\ (m), max\ (m), min\ (m), std)$				(m)
10-1	(**6250.8 6254.5 6249.2 2.4**)	(6276.4 6767.5 **6249.2** 115.6)$^{=}$	(7151.6 7896.8 6423.4 436.8)$^{\Downarrow}$	(7023.8 7545.5 6362.3 386.7)$^{\Downarrow}$	6279.8
10-2	(**6925.7** 7423.4 **6823.5** 178.6)	(6877.8 **7284.8** 6823.5 **122.1**)$^{=}$	(7494.3 7791.1 7348.4 182.9)$^{\Downarrow}$	(7533.2 8150.5 7154.7 254.4)$^{\Downarrow}$	6859.8
10-3	(**6631.8 6678.1 6623.6 19.9**)	(6654.0 7122.1 6623.6 111.4)$^{=}$	(6939.7 7035.7 6858.5 87.5)$^{\Downarrow}$	(7493.2 8168.5 6750.5 467.3)$^{\Downarrow}$	6658.1
20-1	(13461.7 15050.5 13207.6 472.3)	(**13348.2 14289.2 13207.3 271.8**)$^{=}$	(15284.8 16432.8 14465.7 394.0)$^{\Downarrow}$	(17866.4 19568.7 16597.0 806.7)$^{\Downarrow}$	13233.5
20-2	(**15201.5 16095.9 14282.2** 550.5)	(15230.7 16125.3 14473.9 **426.4**)$^{=}$	(16343.3 17299.8 14900.9 659.1)$^{\Downarrow}$	(19038.9 21007.8 17608.3 906.3)$^{\Downarrow}$	14321.2
20-3	(**15363.0 16136.6 14877.4 314.2**)	(15407.2 16180.6 14967.9 359.4)$^{=}$	(17975.4 19360.7 16140.9 749.7)$^{\Downarrow}$	(19475.7 21262.0 17370.8 1029.5)$^{\Downarrow}$	14915.1
30-1	(**23590.6 24930.0** 22287.1 **616.9**)	(23737.1 25526.7 22893.1 787.1)$^{=}$	(29779.6 32434.1 27701.2 1342.2)$^{\Downarrow}$	(32060.2 34576.5 27256.0 1768.4)$^{\Downarrow}$	**22254.5**
30-2	(**22405.9** 24322.3 21099.9 865.9)	(22413.9 **24052.6 21066.5 842.2**)$^{=}$	(27482.7 29711.2 25813.8 935.8)$^{\Downarrow}$	(31717.5 35274.5 28052.7 1779.8)$^{\Downarrow}$	21105.3
30-3	(**21008.9 21935.5 20168.4 520.3**)	(22413.9 24052.6 21066.5 842.2)$^{=}$	(29574.4 33389.2 25659.1 2145.4)$^{\Downarrow}$	(31066.8 35044.2 28034.1 1788.3)$^{\Downarrow}$	20233.0
	D^*-constraint instances: $(avg\ (m), max\ (m), min\ (m), std)$				(m)
10-1	(**5987.5 6277.4** 5916.0 **98.1**)	(6116.8 6687.3 **5914.3** 250.1)$^{\Downarrow}$	(6188.0 6379.7 6145.0 65.7)$^{\Downarrow}$	(6854.2 7322.0 6346.7 328.5)$^{\Downarrow}$	6710.5
10-2	(**6835.8 7208.2 6740.3 153.4**)	(7012.9 7541.9 6771.5 160.3)$^{\Downarrow}$	(7574.7 8042.9 6990.6 312.9)$^{\Downarrow}$	(7623.6 8389.1 6952.8 448.9)$^{\Downarrow}$	6779.8
10-3	(7592.8 8232.9 **7502.6** 167.3)	(**7577.2 8216.3** 7504.9 **160.6**)$^{=}$	(7654.7 7826.1 7570.8 68.4)$^{\Downarrow}$	(8503.5 9186.1 7937.6 398.9)$^{\Downarrow}$	7934.0
20-1	(**12086.0 12553.3 11536.6 282.7**)	(12264.1 13460.8 11573.9 527.8)$^{=}$	(14383.3 16199.1 13484.9 962.0)$^{\Downarrow}$	(16928.9 19978.5 15190.8 1420.7)$^{\Downarrow}$	11584.1
20-2	(**13619.2 14310.1 13384.1 281.5**)	(13892.6 15651.2 13386.7 575.3)$^{=}$	(17350.2 18402.1 16438.3 401.3)$^{\Downarrow}$	(19115.5 22214.4 17105.1 1225.1)$^{\Downarrow}$	13891.4
20-3	(**11075.1** 12347.3 **10321.2** 477.5)	(11281.4 **12093.2** 10352.2 **390.1**)$^{=}$	(11672.4 13561.0 11249.5 706.6)$^{\Downarrow}$	(14130.6 15780.8 12412.0 1011.7)$^{\Downarrow}$	10397.2
30-1	(**20750.9** 21805.9 **19501.9** 626.0)	(20929.9 **21656.5** 20122.2 **516.7**)$^{=}$	(26499.9 29962.6 22876.9 1622.8)$^{\Downarrow}$	(28248.0 31104.6 25527.2 1439.3)$^{\Downarrow}$	21584.2
30-2	(**23372.1 24425.7** 22266.5 **658.8**)	(23562.8 24850.8 22237.3 786.3)$^{=}$	(34380.8 37601.0 26221.5 2921.8)$^{\Downarrow}$	(31034.1 34349.3 28743.0 1422.4)$^{\Downarrow}$	**21434.8**
30-3	(**22474.1 23761.9** 21067.5 805.3)	(23414.2 24484.6 22035.6 **715.3**)$^{\Downarrow}$	(29571.3 34463.1 25308.4 2921.8)$^{\Downarrow}$	(30679.5 34141.7 27533.3 1709.5)$^{\Downarrow}$	**20911.9**

" $\Uparrow$ " indicates that the competitor performs significantly better than BLHMA in the Wilcoxon rank-sum test with a confidence level of 0.95.
" $=$ " indicates that the competitor have no significant difference with BLHMA in the Wilcoxon rank-sum test with a confidence level of 0.95.
" $\Downarrow$ " indicates that BLHMA performs significantly better than the competitor in the Wilcoxon rank-sum test with a confidence level of 0.95.

tain significantly shorter data collection tours in most cases. The data collection tours generated by BLHMA, BLHMA-NNR, MA, VNS, and the transformation method in different instances are presented in Fig. 17. It can be seen intuitively that the tours generated by BLHMA, BLHMA-NNR, and the transformation method are almost coincident in most cases, and BLHMA wins BLHMA-NNR and the reference tours with a slight advantage.

In addition to the simulated UAV data collection instances, this paper also considers an instance in a real-world case. According to historical experience, there are several high-risk areas in the Xishuangbanna forest, and it is necessary to implement the fire monitoring mission in these areas. As Fig. 18 shows, 25 DNs are distributed in the high-risk areas of the forest around the scenic spot, and a UAV is used to perform a data collection task in forest fire monitoring. BLHMA, BLHMA-NNR, VNS, and MA were run 20 times. The mean runtime of the transformation method is 718s, and BLHMA, BLHMA-NNR, VNS, and MA stop running at 718s as termination conditions. The best data collection tours obtained by the four algorithms are 13906.2m (BLHMA), 14038.9m (BLHMA-NNR), 17986.3m (VNS), 17406.2m (MA), and 13930.3m (Reference). The tours generated by BLHMA, BLHMA-NNR, and the transformation method are also almost coincident. By observing the regions marked by the blue dot line in Fig. 18, it can be observed that BLHMA also wins BLHMA-NNR and the reference tours with a slight advantage, so the data collection tour obtained by BLHMA is more conducive to the efficient mission execution and the energy saving of the UAV.

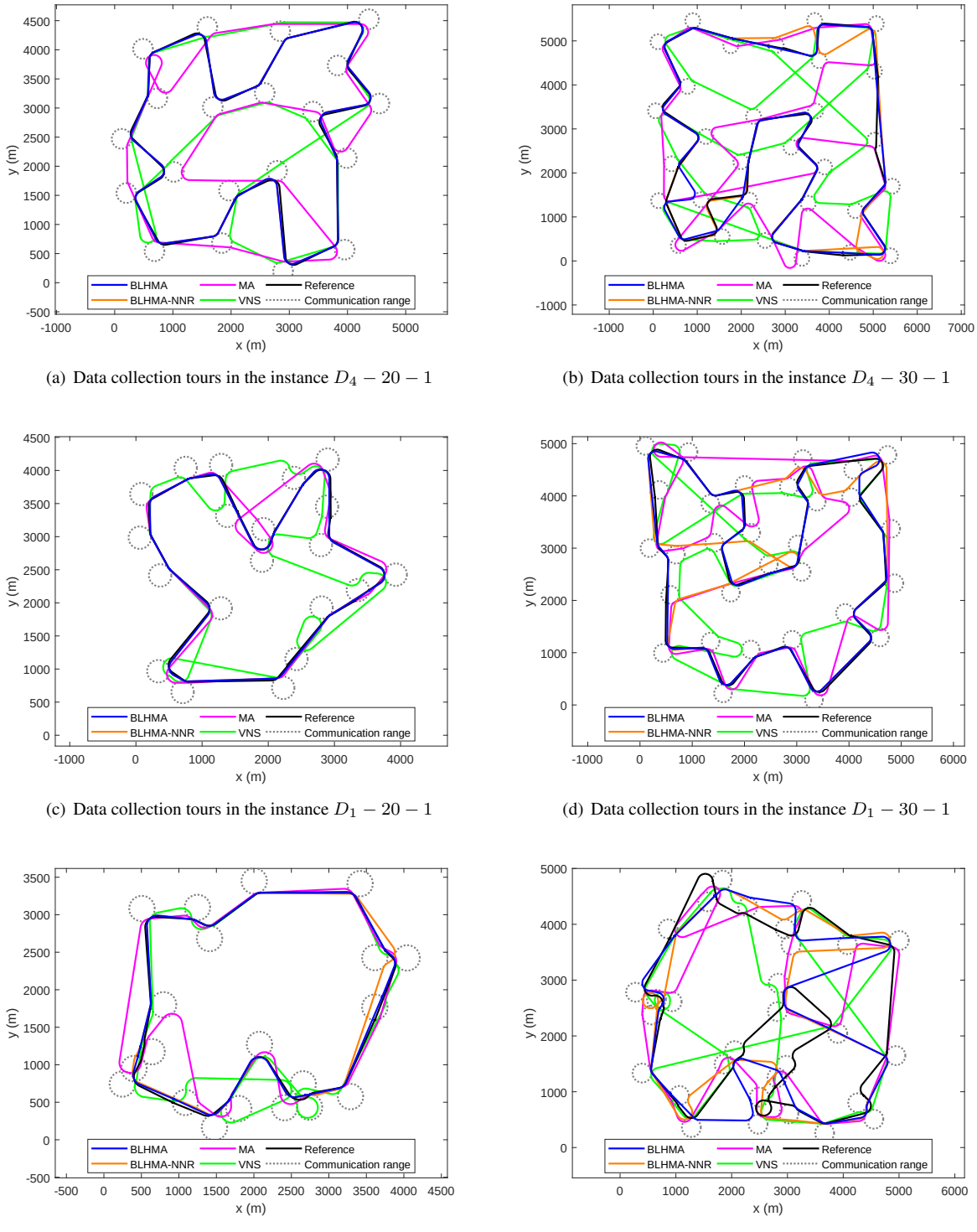
B. CONVERGENCE AND MECHANISM ANALYSIS

In this subsection, the convergence of the BLHMA is discussed and compared with its competitors. Each algorithm runs 20 times on a 10-DNs instance repeatedly and independently. During the runtime, the length of the so-far-best tour is stored every 5s, then the mean value at each time is calculated, as shown in Fig. 19. The mean runtime of the transformation method is 210s, and the data collection tour obtained by the transformation method is regarded as a reference. All the other algorithms run the same time.

It can be observed intuitively from Fig. 19 that BLHMA converges in a very short time, and the obtained data collection tour is better than that of the transformation method. MA also converges quickly, but due to the incomplete solution space, it cannot converge to a better solution. It can be seen from the mean curve that the convergence rate of the BLHMA is the fastest. BLHMA can obtain a comparable tour to the transformation method in a very short time.

To verify the contribution of the local search to BLHMA, Fig. 20 shows the comparison of the BLHMA and its version without the local search (BLHMAxLS). Each algorithm runs 20 times on a 10-DNs instance repeatedly and independently, the length of the so-far-best tour is stored every 5s, then the mean value at each time is calculated. It can be seen that the mean curve of the BLHMA tends to be flat in about 15

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access



(a) Data collection tours in the instance $D_4 - 20 - 1$

(b) Data collection tours in the instance $D_4 - 30 - 1$

(c) Data collection tours in the instance $D_1 - 20 - 1$

(d) Data collection tours in the instance $D_1 - 30 - 1$

(e) Data collection tours in the instance $D^* - 20 - 1$

(f) Data collection tours in the instance $D^* - 30 - 1$

FIGURE 17. The data collection tours generated by different algorithms in different instances

seconds, while BLHMAxLS reaches the same level in about 70 seconds.

For further verifying the performance of the local search, different search strategies were used to improve a poor data collection tour shown in Fig. 21 (black line). Fig. 21 and Table 5 show the improvement of the tour by different search strategies. Although multistage approximate gradient descent strategy proposed in this paper consumes a little more time,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3102317, IEEE Access

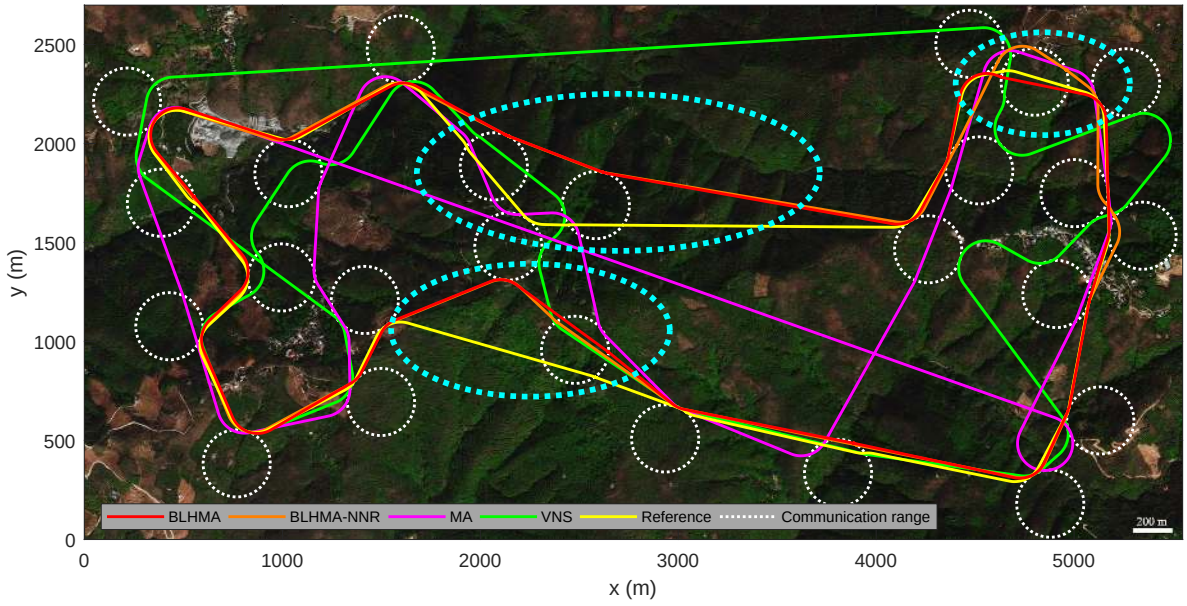


FIGURE 18. UAV data collection instance in a real world case.

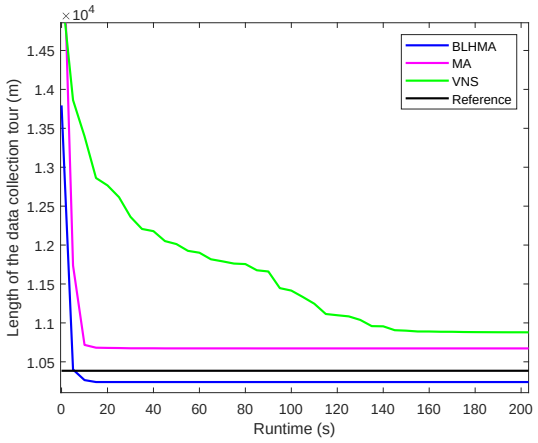


FIGURE 19. Convergence curves of different algorithms on the 10-DNs instance

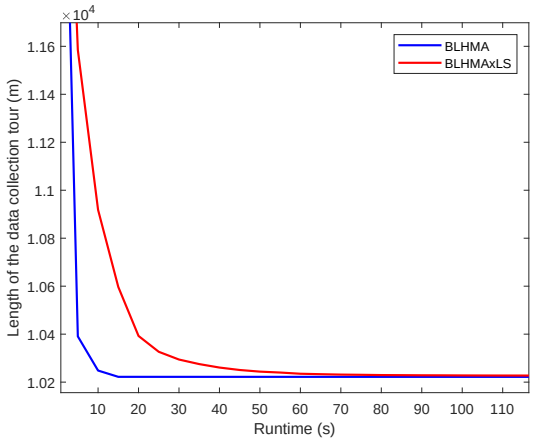


FIGURE 20. Comparison of the BLHMA and BLHMAxLS on the 10-DNs instance

TABLE 5. Local search performance of different strategies

Local search strategy	Time (s)	Tour length (m)
Multistage strategy	0.41	10586.07
Full-dimensional strategy	0.29	11360.93
Local-dimensional strategy	0.14	11369.77

its improvement to the tour is better than that of the two sub-strategies.

V. CONCLUSIONS

In this paper, a bi-level hybridization-based metaheuristic algorithm is proposed for the UAV data collection task planning in forest fire monitoring. At the first hybridization level, DE and the proposed SOMAC act on continuous and discrete variables respectively to produce data collection tours in a cooperative way, which avoids a large amount of blind search in complex solution space. To enhance the quality of the tours and accelerate the convergence of the algorithm, at the second level, a local search strategy based on a multistage approximate gradient is incorporated to improve the communication position and UAV heading at each DN. Finally, numerical results based on the simulated and real-world UAV data collection instances demonstrate that the algorithm proposed in this paper, compared with the state-of-the-art algorithms, can generate shorter data collection tours in most cases. It is conducive to the efficient execution of a UAV data

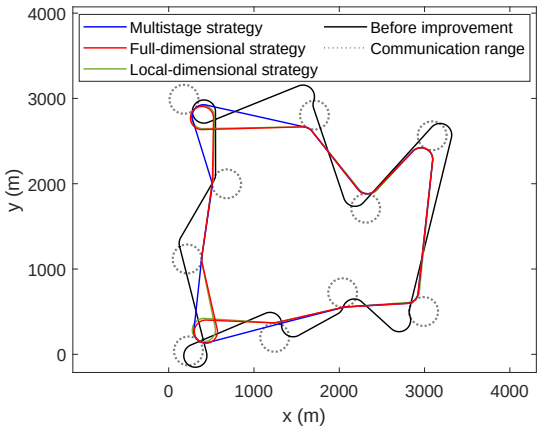


FIGURE 21. A poor data collection tour improved by different local search strategies

collection task in the fire monitoring, and effectively reduces the energy consumption of the UAV. This paper provides favorable technical support for the UAV data collection in forest fire monitoring, and the idea can be extended to many other applications, such as electric power inspection, pipeline detection, mobile edge computing, etc.

REFERENCES

[1] D. Cao, C. D. Ramirez, "Air pollution, government pollution regulation, and industrial production in china," *Journal of Systems Science and Complexity*, vol. 33, no. 2, pp. 1064-1079, May. 2020, 10.1007/s11424-020-9128-6.

[2] G. Skorobogatov, C. Barrado, and E. Salamí, "Multiple UAV Systems: A Survey," *Unmanned Systems*, vol. 8, no. 2, pp. 149-169, Apr. 2020, 10.1142/S2301385020500090.

[3] H. Zhang, B. Xin, L. H. Dou, J. Chen, and K. Hirota, "A review of cooperative path planning of an unmanned aerial vehicle group," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 12, pp. 1671-1694, Dec. 2020, 10.1631/FITEE.2000228.

[4] Q. Li, J. P. Queralta, T. N. Gia, Z. Zou, and T. Westerlund, "Multi-Sensor Fusion for Navigation and Mapping in Autonomous Vehicles: Accurate Localization in Urban Environments," *Unmanned Systems*, vol. 8, no. 3, pp. 229-237, Jul. 2020, 10.1142/S2301385020500168.

[5] F. Khoshnoud, I. I. Esat, C. W. D. Silva, J. D. Rhodes, A. A. Kiessling, and M. B. Quadrelli, "Self-Powered Solar Aerial Vehicles: Towards Infinite Endurance UAVs," *Unmanned Systems*, vol. 8, no. 2, pp. 95-117, Apr. 2020, 10.1142/S2301385020500077.

[6] J. Xie, L. R. Garcia Carrillo and L. Jin, "Path Planning for UAV to Cover Multiple Separated Convex Polygonal Regions," *IEEE Access*, vol. 8, pp. 51770-51785, Mar. 2020, 10.1109/ACCESS.2020.2980203.

[7] M. Mnich and T. Mömke, "Improved integrality gap upper bounds for traveling salesperson problems with distances one and two," *European Journal of Operational Research*, vol. 266, no. 2, pp. 436-457, Apr. 2018, 10.1016/j.ejor.2017.09.036.

[8] Y. Cui, J. Zhong, F. Yang, S. Li and P. Li, "Multi-Subdomain Grouping-Based Particle Swarm Optimization for the Traveling Salesman Problem," *IEEE Access*, vol. 8, no. 99, pp. 227497-227510, Dec. 2020, 10.1109/ACCESS.2020.3045765.

[9] P. Guo, M. Hou and L. Ye, "MEATSP: A Membrane Evolutionary Algorithm for Solving TSP," *IEEE Access*, vol. 8, pp. 199081-199096, Oct. 2020, 10.1109/ACCESS.2020.3035058.

[10] C. Olsen, K. Kalyanam, W. Baker, and D. Kunz, "Maximal distance discounted & weighted revisit period: A utility approach to persistent unmanned surveillance," *Unmanned Systems*, vol. 7, no. 4, pp. 215-232, Sep. 2019, 10.1142/S2301385019500079.

[11] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Transactions on Automation Science and Engineering* vol. 4, no. 1, pp. 98-104, Jan. 2007, 10.1109/TASE.2006.872110.

[12] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497-516, 1957, 10.2307/2372560.

[13] L. Babel, "Curvature-constrained traveling salesman tours for aerial surveillance in scenarios with obstacles," *European Journal of Operational Research*, vol. 262, no. 1, pp. 335-346, Oct. 2017, 10.1016/j.ejor.2017.03.067.

[14] J. Wilhelm, J. Rojas, G. Eberhart, and M. Perhinschi, "Heterogeneous aerial platform adaptive mission planning using genetic algorithms," *Unmanned Systems*, vol. 5, no. 1, pp. 19-30, Mar. 2017, 10.1142/S2301385017500029.

[15] P. Skorput, S. Mandzuka , and H. Vojvodic. "The use of Unmanned Aerial Vehicles for forest fire monitoring," presented at the *IEEE International Symposium Elmar Conference*, Zadar, Croatia. Sep 12-14, 2016.

[16] I. G. Hendel, and G. M. Ross, "Efficacy of Remote Sensing in Early Forest Fire Detection: A Thermal Sensor Comparison," *Canadian Journal of Remote Sensing*, vol. 46, no. 4, pp. 414-428, Nov. 2019, 10.1080/07038992.2020.1776597.

[17] S. Sudhakar, V. Vijayakumar, C. Sathiya Kumar, V. Priya, Logesh Ravi and V, "Subramaniyaswamy. Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires," *Computer Communications*, vol. 149, pp. 1-16, Jan. 2020, 10.1016/j.comcom.2019.10.007.

[18] C. Yuan, Z. Liu, Y. Zhang, "Learning-Based Smoke Detection for Unmanned Aerial Vehicles Applied to Forest Fire Surveillance," *Journal of Intelligent & Robotic Systems*, vol. 39, no. 12, pp. 337-349, Feb. 2019, 10.1007/s10846-018-0803-y.

[19] J. Guaman, W. Chamba-Zaragocin, E. Guaman-Quinche, A. P. Francisco, H. T. Carrion, "Study of routing protocols used in sensor networks for forest fire detection," presented at the *15th Iberian Conference on Information Systems and Technologies (CISTI)*, Seville, Spain. Jun 24-27, 2020.

[20] K. J. Obermeyer, "Path planning for a UAV performing reconnaissance of static ground targets in terrain," presented at the *AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois. Aug 10-13, 2009.

[21] J. Isaacs, D. Klein, and J. Hespanha, "Algorithms for the traveling salesman problem with neighborhoods involving a Dubins vehicle," presented at the *American Control Conference*, San Francisco, USA. Jun 29-Jul 1, 2011.

[22] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the Dubins vehicle," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1378-1391, Jul. 2008, 10.1109/TAC.2008.925814.

[23] J. Le Ny, E. Feron, and E. Frazzoli, "On the Dubins traveling salesman problem," *IEEE Transactions on Automatic Control* vol. 57, no. 1, pp. 265-270, Jan. 2012, DOI:10.1109/TAC.2011.2166311.

[24] A. Dumitrescu and J. S. B. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane," *Journal of Algorithms*, vol. 48, no. 1, pp. 135-159, Jul. 2001, 10.1016/S0196-6774(03)00047-6.

[25] B. Yuan, M. Orlowska, and S. Sadiq, "On the optimal robot routing problem in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 9, pp. 1252-1261, Oct. 2007, 10.1109/TKDE.2007.1062.

[26] Z. Yang, M.-Q. Xiao, Y.-W. Ge, D. Feng, L. Zhang, H.-F. Song, and X.-L. Tang, "A double-loop hybrid algorithm for the traveling salesman problem with arbitrary neighbourhoods," *European Journal of Operational Research*, vol. 265, no. 1, pp. 65-80, Jul.2017, 10.1016/j.ejor.2017.07.024.

[27] J. Isaacs and J. Hespanha, "Dubins traveling salesman problem with neighborhoods: A graph-based approach," *Algorithms*, vol. 6, no. 1, pp. 84-99, Feb. 2013, 10.3390/a6010084.

[28] K. Savla, E. Frazzoli, and F. Bullo, "On the point-to-point and traveling salesperson problems for Dubins' vehicle," presented at the *American Control Conference*, Portland, USA, Jun 8-10, 2005.

[29] S. Yadlapalli, W. Malik, S. Darbha, and M. Pachter, "A lagrangian-based algorithm for a multiple depot, multiple travelling salesmen problem," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 4, pp. 1990-1999, Aug. 2009, 10.1016/j.nonrwa.2008.03.014.

[30] P. Oberlin, S. Rathinam, and S. Darbha, "Combinatorial motion planning for a Dubins vehicle with precedence constraints," presented at the *ASME 2009 Dynamic Systems and Control Conference*, California, USA. Oct 12-14, 2009.

[31] S. G. Manyam, S. Rathinam, D. Casbeer, and E. Garcia, "Tightly bounding the shortest Dubins paths through a sequence of points," *Journal of*

Intelligent & Robotic Systems, vol. 88, no. 2-4, pp. 495-511, Dec. 2017, 10.1007/s10846-016-0459-4.

[32] X. Zhang, J. Chen, B. Xin, and Z.-H. Peng, "A memetic algorithm for path planning of curvature-constrained UAVs performing surveillance of multiple ground targets," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 622-633, May. 2014, 10.1016/j.cja.2014.04.024.

[33] P. Váňa, J. Faigl, "Optimal solution of the Generalized Dubins Interval Problem: finding the shortest curvature-constrained path through a set of regions," *Autonomous Robots*, vol. 44, no. 2, pp. 1359-1376, Sep. 2020, 10.1007/s10514-020-09932-x.

[34] J. Le Ny and E. Feron, "An approximation algorithm for the curvature-constrained traveling salesman problem," presented at the *43rd Annual Allerton Conference on Communications, Control and Computing*, Monticello, IL. Sep, 2005.

[35] J. Le Ny, E. Frazzoli, and E. Feron, "The curvature-constrained traveling salesman problem for high point densities," presented at the *46th IEEE Conference on Decision and Control*," New Orleans, USA. Dec 12-14, 2007.

[36] K. J. Obermeyer, P. Oberlin, and S. Darbha, "Sampling-based path planning for a visual reconnaissance unmanned air vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 619-631, Mar. 2012, 10.2514/1.48949.

[37] K. J. Obermeyer, P. Oberlin, and S. Darbha, "Sampling-based roadmap methods for a visual reconnaissance UAV," presented at the *AIAA Conference on Guidance, Navigation, and Control*, Toronto, Ontario Canada. Aug 2-5, 2010.

[38] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106-130, Oct. 2020, 10.1016/S0377-2217(99)00284-2.

[39] J. Faigl and P. Váňa, "Unsupervised learning for surveillance planning with team of aerial vehicles," presented at the *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, Alaska. May 14-19, 2017.

[40] J. Faigl, P. Váňa, R. Pěnička, and M. Saska, "Unsupervised learning-based flexible framework for surveillance planning with aerial vehicles," *Journal of Field Robotics*, vol. 36, no. 4, pp. 270-301, Oct. 2018, 10.1002/rob.21823.

[41] Z. wang, L. Liu, T. Long, and Y. Wen, "Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding," *Chinese Journal of Aeronautics*, vol. 31, no. 2, pp. 339-350, Oct. 2017, 10.1016/j.cja.2017.09.005.

[42] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Transactions on Systems Man & Cybernetics Part C*, vol. 42, no. 5, pp. 744-767, Sep. 2012, 10.1109/TSMCC.2011.2160941.

[43] Z. Chen, C. Sun, X. Shao, W. Zhao, "A descent method for the Dubins traveling salesman problem with neighborhoods," *Front Inform Technol Electron Eng*, Aug. 2020, 10.1631/FITEE.2000041, [Online].

[44] D. G. Macharet, A. Neto, V. Da Camara Neto, and M. Campos, "An evolutionary approach for the Dubins' traveling salesman problem with neighborhoods," presented at the *14th International Conference on Genetic and Evolutionary Computation*, Kitakyushu, Japan. Aug 25-28, 2012.

[45] R. Pěnička, J. Faigl, M. Saska, et al. "Data collection planning with non-zero sensing distance for a budget and curvature constrained unmanned aerial vehicle," *Auton Robot*, vol. 43, no. 4, pp. 1937-1956, Dec. 2019, 10.1007/s10514-019-09844-5.

[46] 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 & Applications*, vol. 31, no. 7, pp. 941-954, Jul. 2014, 10.7641/CTA.2014.40280.

[47] V. Da Camara Neto, D. G. Macharet, A. Neto, and M. Campos, "Dynamic region visit routing problem for vehicles with minimum turning radius," *Journal of Heuristics*, vol. 24, no. 2, pp. 83-109, Feb. 2018, 10.1007/s10732-017-9359-4.

[48] D. G. Macharet, A. Neto, V. Da Camara Neto, and M. Campos, "Data gathering tour optimization for Dubins' vehicles," presented at the *IEEE Congress on Evolutionary Computation*, Brisbane, Australia. Jun 10-15, 2012.

[49] P. Váňa, J. Faigl, "On the Dubins Traveling Salesman Problem with Neighborhoods," presented at the *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sept 28-Oct 2, 2015.

[50] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398-417, May. 2009, 10.1109/TEVC.2008.927706.

[51] K. Chandrasekar, N. V. Ramana, "Performance Comparison of GA, DE, PSO and SA Approaches in Enhancement of Total Transfer Capability using FACTS Devices," *Journal of Electrical Engineering and Technology*, vol. 7, no. 4, pp. 493-500, Jul. 2012, 10.5370/JEET.2012.7.4.493.

[52] A. Semnani, M. Kamyab, "Comparison of differential evolution and particle swarm optimization in one-dimensional reconstruction problems," presented at the *Asia Pacific Microwave Conference*, Hong Kong, China. Dec 16-20, 2008.

[53] G. B. Dantzig, D. R. Fulkerson, S. M. Johnson, "Solutions of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, no. 4, pp. 393-410, Nov. 1954, 10.1287/opre.2.4.393.

[54] C. A. J. Hurkens and G. J. Woeginger, "On the nearest neighbor rule for the traveling salesman problem," *Operations Research Letters*, vol. 32, no. 1, pp. 1-4, Jan. 2004, 10.1016/S0167-6377(03)00093-2.

[55] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80-83, Nov. 1945, 10.2307/3001968.

HAO ZHANG received his Master's degree in 2016 from Beijing Information Science and Technology University. He is currently working towards his Ph.D. degree in the School of Automation, Beijing Institute of Technology, Beijing, China.

His current research interests include intelligent optimization, combinatorial optimization, Multi-UAV cooperative path planning.

LIHUA DOU received her 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 at the Control Science and Engineering at Key Laboratory of Complex System Intelligent Control and Decision, School of Automation, Beijing Institute of Technology.

Her research interests include multi-objective optimization and decision, pattern recognition, and image processing.

BIN XIN (S'09-M'10) 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 an Associate Editor of the Journal of Advanced Computational Intelligence and Intelligent Informatics and the journal Unmanned Systems.

JIE CHEN (M'09-SM'12-F'19) 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 in the California State University, U.S.A. From 1996 to 1997, he was a research fellow in the School of E&E, University of Birmingham, U.K. He is currently a professor of control science and engineering, the Beijing Institute of Technology and the Tongji University, China. He is also an academician of the Chinese Academy of Engineering and an IFAC Fellow. He serves as a managing editor for the Journal of Systems Science and Complexity (2014-2022) and an associate editor for the IEEE Transactions on Cybernetics (2016-2022) and many other international journals.

His main research interests include intelligent control and decision in complex systems, multi-agent systems, and optimization methods. He has co-authored 5 books and more than 200 research papers.

MINGGANG GAN received the B.E. and Ph.D. degrees in Control Science and Engineering from Beijing Institute of Technology, Beijing, China, in 2001 and 2007, respectively. During 2015-2016, he was a visiting scholar in New York University. He is currently a Professor with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology.

His main research interest covers intelligent information processing and intelligent control.

YULONG DING received the Bachelor's degree from Qilu University of Technology, Jinan, China, in 2012, and Master's degree from Xiamen University, Xiamen, China, in 2015. He received the Ph.D. degree from Beijing Institute of Technology, Beijing, China, in 2020. He is currently working in the Peng Cheng Laboratory postdoctoral workstation, Shenzhen, China.

His current research interests include human-agent collaborative systems, heterogeneous multi-agent systems, and task planning of multi-robot systems.