Agent Evaluation in Deployment of Multi-SUAVs for Communication Recovery

Qian Jiang, *Graduate Student Member, IEEE*, Haibin Zhu, *Senior Member, IEEE*, Yan Qiao, *Senior Member, IEEE*, Zhiwei He, Dongning Liu, *Member, IEEE*, and Baoying Huang, *Graduate Student Member, IEEE*

***Abstract*—When earthquakes occur, solar-powered unmanned aerial vehicles (SUAVs), deployed as communication relay points, can construct a signal relay network to assist the ground mobile communication vehicles in resuming communication. Considering the urgency of disaster relief, a practical, accurate, and robust modeling method for multiple SUAVs deployments is vital. For this concern, this article first formalizes the deployment problem of multiple solar-powered UAVs in communication recovery by extending Group MultiRole Assignment (GMRA) (UGRA). In the second step, the success in this assignment process depends on the choice of the agent evaluation method. The evaluation benchmark in UGRA is SUAV path planning in a complex environment with uncertain subpaths and accumulative attitude errors. In response to this issue, we propose two innovative algorithms: 1) dynamic curve path-planning algorithm (DCPPA) and 2) greedy curved straight path-planning algorithm (GCSPPA). Moreover, with the time requirement in mind, one sufficient condition and one necessary condition are established to help the DCPPA achieve fast convergence. With these two novel agent evaluation algorithms, UGRA can rapidly deploy multiple SUAVs to establish a collaborative relay network within an acceptable time. Finally, simulation experiments at different scales are carried out to demonstrate the accuracy and effectiveness of the proposed solution.**

***Index Terms*—Agent evaluation, disaster relief, dynamic curve path-planning algorithm (DCPPA), greedy curved straight path-planning algorithm (GCSPPA), group multi-role assignment (GMRA), human–machine cooperation, the deployment of multiple solar-powered unmanned aerial vehicles (SUAVs) in communication recovery.**

I. Introduction

SOLAR-POWERED unmanned aerial vehicles (SUAVs) have a better endurance performance than the traditional unmanned aerial vehicles (UAVs) and SUAVs have been widespread in recent years [1], [2]. When the multiple SUAVs are appropriately equipped with communication devices [3], [4], the successful deployment of them from different SUAV base stations can construct a signal relay network to help ground communication vehicles recovery communication in disaster relief. Such an aerial relay network requires unified modeling that is accurate, robust, and practically executed. This article attempts to satisfy these requirements by formalizing and modeling the deployment of multiple solar-powered UAVs via Group MultiRole Assignment (UGRA).

Essentially, UGRA is an extension of Group MultiRole Assignment (GMRA) [5], [6], while GMRA is an extended form of group role assignment (GRA) [7]–[9], which is a significant step of role-based collaboration (RBC). Collaboration has been revealed as a complex process throughout the life cycle of RBC (Fig. 1) [10], [11]. RBC and GMRA have been developed into a practical centralized modeling methodology allowing decision makers to achieve distributed collaborative task assignments.

In our research, the whole process of multiple SUAVs deployments for communication recovery can be taken as an RBC process. First, in the role negotiation part, we take the signal relay points as roles because a signal relay network consists of them. Then, we need to know who the participants are, i.e., agents, in the collaboration. Superficially, it is easy to take SUAVs as agents because an SUAV carries a signal reply device. More pertinently, we take base stations as agents in our approach because a base station can play the signal relay's roles by sending SUAVs to the relay points. That is, it is the base stations that play the roles of signal relay.

To conduct optimized role assignment in RBC, agent evaluation is a required step, which is highly domain oriented and

Manuscript received 22 July 2021; accepted 16 November 2021. Date of publication 7 December 2021; date of current version 17 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62072120 and Grant 61972102; in part by the Research and Development Program of Guangdong Province under Grant 2020B010166006; in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN2018-04818; and in part by the Innovation for Defence Excellence and Security (IDEaS) Program from the Canadian Department of National Defence (DND) under Grant CFPMN2-051. This article was recommended by Associate Editor Z. Peng. *(Corresponding author: Dongning Liu.)*

Qian Jiang, Yan Qiao, and Baoying Huang are with the Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems, Macau University of Science and Technology, Macau, China (e-mail: jiangqian@mail2.gdut.edu.cn).

Haibin Zhu is with the Collaborative Systems Laboratory, Nipissing University, North Bay, ON P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

Zhiwei He is with the Network Operations Center, China Telecom Corporation Limited Guangdong Telecom Company, Guangzhou 510062, China (e-mail: zhiwei.he96@gmail.com).

Dongning Liu is with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, Guangdong, China (e-mail: liudn@gdut.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2021.3130161.

Digital Object Identifier 10.1109/TSMC.2021.3130161

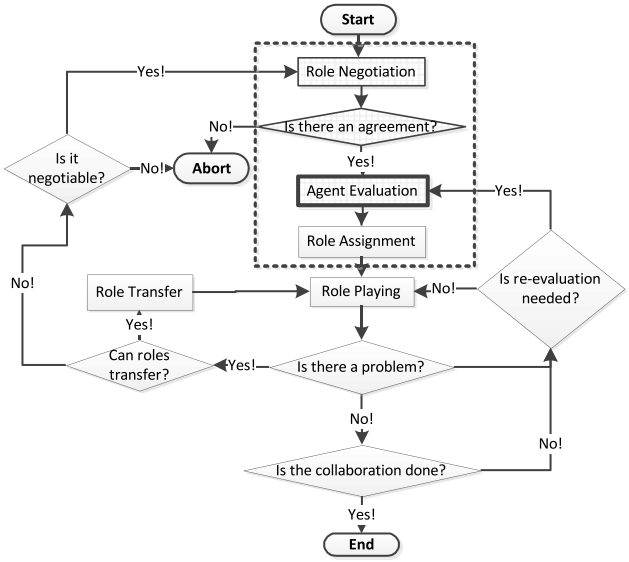


Fig. 1. Relationship of this research work to the RBC process [7]. *Note:* The dashed black box is the core part of GMRA, while the solid black box is this article's concentration.

dependent on the application scenario. At this step, agent evaluation can be conducted to determine the appropriateness of different bases (agents) for specific relay points (roles). This is also what our work focuses on in the life cycle of RBC (the solid black box in Fig. 1). We assume that all the SUAVs from different base stations are the solar-powered fixed-wing drones of [12] and the stored energy of all SUAVs can keep the communication work at night. Since every second counts for disaster relief tasks, the agent evaluation method is mainly based on the flight trajectory length of an SUAV from its take-off base station (agent) to a proper relay point (role).

The first intuitive thought is using straight-line distances between an SUAV take-off base station and a relay point to evaluate the flight trajectory length of an SUAV. However, due to SUAVs' flights are affected by environmental factors, such as wind, magnetism, and signal interruption, straight-line distances are not good indicators. The actual SUAV flight path requires continuously dynamical refinement for precise positioning [13]. Also, the SUAV airframe's mechanical characteristics require a curvature flight path because it cannot instantly turn. These constraints complicate the path-planning problem, and existing path-planning algorithms cannot properly solve this problem. Consequently, we design two innovative algorithms: 1) dynamic curve path-planning algorithm (DCPPA) and 2) greedy curved straight path-planning algorithm (GCSPPA) to tackle this issue. Additionally, we find one sufficient condition and one necessary condition to accelerate the DCPPA algorithm for meeting the urgency of disaster relief. These two algorithms cooperatively form the agent evaluation method, swiftly calculating the proper path length between a base station and a relay point under a complex environment with multiple constraints.

After agent evaluation, role assignment is straightforward by using a similar solution to GMRA: SUAV operators will fly the SUAVs to the relay points based on such an assignment. Because our proposed process is instant (in a minute), the environmental change can be ignored in practice. Role playing takes place when SUAVs start to work at their designated relay points for the signal relay. With our proposed solution, it is possible to establish a signal relay network within an acceptable time. Please note that we are concentrating on SUAV assignment and deployment but not on communication establishment.

The contributions of this article include the following.

1) This article formalizes the problem of deploying SUAVs as signal relay nodes, in disaster relief, via GMRA, i.e., UGRA.
2) To rapidly construct the qualification matrix required by the agent evaluation part in UGRA, two original curve path-planning algorithms DCPPA and GCSPPA are proposed.
3) Finally, by UGRA, we provide a swift way to assign multiple SUAVs to establish a signal relay network in a disaster relief situation.

This article is organized as follows. It first discusses related research efforts in Section II; then in Section III, it describes a real-world scenario related to the proposed problem. Section IV formally defines the problem via RBC and its environments—classes, agents, roles, groups, and objects (E-CARGO) model. Section V illustrates the agent evaluation method in UGRA. Section VI discusses the improvement of the agent evaluation method proposed in Section V. Results of large-scale simulation experiments are provided in Section VII. This article concludes and points out future work in Section VIII.

II. Related Work

A. Deployment and Multi-SUAVs

UAVs, including SUAVs, can overcome the constraints of harsh geographical environments or obstacles [14]–[16] and play important roles in the fields of communication establishment [17]–[21], target tracking [22]–[25], and collaborative deployment of multiple UAVs [3], [26]. This article focuses on the third field, which is essentially a many-to-many assignment problem [27], [28].

To solve the multi-SUAVs assignment problem in disaster relief, related research works focus on mathematical assignment algorithms [26], [28]. For example, Geng *et al.* [26] used a modified centralized algorithm based on particle swarm optimization (MCPSO) to solve the static UAV deployment's task allocation problem in the search and rescue domain since it has all the available information (like the UAV path) at hand. But in reality, the randomness and uncertainty of environmental changes will make it hard to collect all the available information. Besides, these greedy strategies are also related to the distribution and density of data, making it difficult to obtain an optimal value.

Some researchers have recently concentrated on designing a unified mathematical modeling method [29]–[31] to solve the multi-SUAVs assignment problem because it can help build a robust model regardless of the distribution of the tasks, i.e., RBC [3], [10], [11], [32] and RBAC [33].

For example, Liu *et al.* [3] solved the signal relay problem of multi-SUAVs deployment in a disaster area by RBC and its

extended GRA model. Thousands of random simulation experiments show that their solution can quickly find relay points' locations, assign multi-SUAVs from different base stations to these relay points, and finally construct a collaborative signal relay network. The model of our work is different from that of [3] because the variable's domain is changed from {0, 1} to {0, 1, ..., $\Upsilon[i, j]$}, i.e., $0 \leq T[i, j] \leq \Upsilon[i, j]$ ($0 <= i < m$, $0 <= j < n$, and $m, n \in \mathbb{N}^+$, which expresses the set of non-negative numbers), where Υ is a domain-oriented matrix.

Although [3] has demonstrated that GRA is a practical method for the assignment of multiple SUAVs, two problems remain to be improved. First, each relay point may require different numbers of SUAVs to work simultaneously since different relay points on the signal relay network have different connectivities. Hence, the problem of the deployment of multi-SUAVs is an extended GMRA problem instead of a GRA one. Second, their agent evaluation method is based on straight-line path lengths between the SUAV take-off base stations and relay points. Due to attitude errors and mechanical characteristics, the actual SUAV trajectory needs to pass through calibration points to correct SUAV attitude errors dynamically [13]. Also, the trajectory of SUAV may be a curvature flight path since SUAV cannot instantly turn.

In summary, the previous work cannot be applied to solving the UGRA problem without paying significant effort.

B. UGRA and Dynamic Path Planning of SUAV

For these concerns, this article first formalizes this multi-SUAVs deployment problem by extending GMRA (UGRA). As for the agent evaluation method of UGRA, it needs to find the flight trajectory length of SUAV in a complex environment. There exist many research works on trajectory planning of SUAV. For example, Wang *et al.* optimized the trajectory of the SUAV for solar energy optimization and obstacle avoidance [1], [2], [20] while Demiane *et al.* [23] and Uluskan [24] planned the path of UAV for ground target tracking. The aforementioned studies have achieved substantial results for SUAV path planning and have laid the foundation for SUAV path planning. However, disaster relief tasks race against time; hence time and accuracy are the most critical metrics. Therefore, the agent evaluation method in UGRA has to find the shortest curve flight trajectory for SUAV while dynamically correcting its attitude errors through error correction points. However, there is no literature in finding the trajectories of SUAVs in such a complex environment as far as we know.

Regarding the shortest path algorithm, the first thing that comes to mind is traditional path-planning algorithms like the Dijkstra algorithm [34], A* algorithm [35], and their derived algorithms like Lifelong Planning A* and D* Lite. However, affected by the attitude errors of an SUAV, the infeasible subpath set of an SUAV dynamically changes. Besides, since an SUAV cannot turn instantly, its speed direction may cause the collection of subpaths making up the shortest path to change frequently. Fig. 2 helps illustrate this feature. We assume that there are two available paths for SUAV, ℓ_1 and ℓ_2, from the beginning point to the last correction point and ℓ_2 is the shortest subpath. Due to the different velocities' direction at the last correction point for different paths, two different subpaths will be obtained, namely, ℓ_4 and ℓ_3. If $\ell_1 - \ell_2 > \varepsilon$, $\varepsilon > 0$, and $\ell_3 \gg \ell_4$, then it may result in the optimal path being $\ell_1 + \ell_4$ instead of $\ell_2 + \ell_3$. But these above-mentioned planning algorithms obtain the path through ℓ_2 instead of the shortest path through ℓ_1.

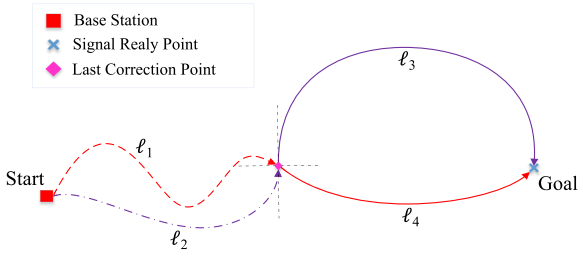


Fig. 2. Example of the SUAV flight trajectory.

Another option is to apply intelligent algorithms like ant colony optimization [36] and particle swarm optimization. However, due to the urgency of disaster relief time, these intelligent algorithms' solution time may not meet the requirements. Moreover, these intelligent algorithms also may not obtain the optimal solution due to the impact of the velocity's direction.

In other words, existing algorithms cannot accomplish the tasks of solving our formalized problem.

To meet this challenge, we propose two algorithms: 1) DCPPA and 2) GCSPPA. We also find one sufficient condition and one necessary condition to help improve DCPPA convergence time. With these two agent evaluation algorithms, UGRA can swiftly assign multiple SUAVs from the different SUAV take-off base stations to establish a collaborative aerial signal relay network. Our work on these two algorithms is unique and innovative.

III. Real-World Problem

To better describing the modeling process, we first introduce a real-world problem here. The County of Jiuzhaigou is situated in mountainous terrain bordering the three counties of Nanping, Songpan, and Pingwu. These lands lie within the Aba Tibetan and Qiang Autonomous Prefecture, Northwestern Sichuan, China. The remoteness of this county has allowed it to remain undisturbed until recent years. Transportation services are provided via horse trails or mountain paths.

In August 8, 2017, a 7.0-magnitude earthquake jolted Jiuzhaigou County. The communication system was affected by the quake. Although decision makers have deployed mobile communication vehicles in specific locations (the green circle shapes in Fig. 3), their coverage was limited by terrain conditions. To illustrate the situation, we randomly generated seven communication vehicles' coordinates based on Jiuzhaigou's partial elevation data obtained in [12].

To restore communication over the disaster area, appropriately equipped SUAVs should be deployed to the signal relay points (the blue fork shapes in Fig. 3) to form a signal network.

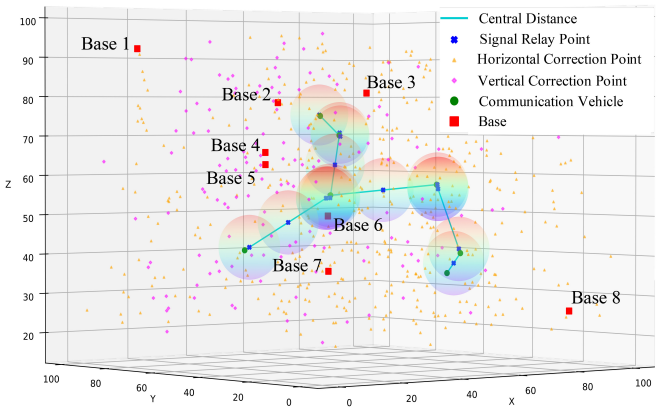


Fig. 3. Real-world scenario. *Note:* The transparent rainbow spheres represent the communication coverage of the signal relay points. The depth of the rainbow-colored sphere's shadow represents the required number of SUAVs for the relay point.

TABLE I
LIMITED NUMBERS OF SUAV

Base Station	1	2	3	4	5	6	7	8
The Limited Numbers of SUAV	13	11	6	7	8	10	9	12

These SUAVs are stored at different high-altitude base stations in Jiuzhaigou. In this example, the locations of eight SUAV base stations were randomly generated. The communication between SUAVs and the communication between SUAVs and communication vehicles is the same as [3]. Due to the load limitations, all the SUAVs from different base stations can maintain stable communication coverage over a spherical area with a limited 8-km radius while the communication vehicle can maintain stable communication in a spherical area with a 2-km radius.

Each base station is equipped with a limited number of fixed-wing SUAVs [12]. It is understandable because the capacity of each base station is limited (see Table I). Without loss of generality, we will conduct large-scale simulation experiments in Section VII for the values in Table I.

From the above scenario, the assignment of multi-SUAVs can be dealt with by following the initial steps of RBC and can be considered as a UGRA problem, which is a variant of GMRA. The most vital step in UGRA is agent evaluation. The main agent evaluation benchmark for this step is the SUAV flight path length. However, natural events, such as wind and limitations of the SUAV positioning system, will impact location accuracy. Also, because the SUAV cannot turn instantaneously, its flight path must maintain a curvature degree during the calibration process. These constraints complicate the path-planning environment.

Inspired by [13], the decision makers decided to use correction points to timely correct SUAVs attitude errors. They used buildings or landmarks whose horizontal or vertical positions are not affected as horizontal or vertical calibration points. Fig. 3 shows the distribution of collected calibration points. Then, the decision makers summarized the constraints during SUAV calibration (Table II lists the symbols used in Table III, and Table III depicts all the constraints during an SUAV's flight). The method of generating the calibration point data and the range of the parameters in Table II will be described in Section VII.

TABLE II
PARAMETERS OF THE SUAV USED IN TABLE III

δ	A coefficient represents the proportional relationship between the errors and flight distance. For each meter of an SUAV's flight, the vertical and horizontal errors will increase by $1 \times \delta$ meter(s).
α_1	It represents the threshold of vertical error (in meters) when vertical correction is allowed.
α_2	It represents the threshold of horizontal (in meters) error when vertical correction is allowed.
β_1	It represents the threshold of vertical error (in meters) when horizontal correction is allowed.
β_2	It represents the threshold of horizontal error (in meters) when horizontal correction is allowed.
θ	Maximum tolerable error during the flight (in meters).
r_{min}	The minimum turning radius of the SUAV (in meters).

TABLE III
CONSTRAINTS DURING THE FLIGHT OF SUAVS

0	The correction points of an SUAV are divided into vertical correction points and horizontal correction points. The vertical correction point is used for vertical error correction of the SUAV while the horizontal correction point is for horizontal error correction. Besides, there are no priority rules between horizontal error correction and vertical error correction.
1	The vertical error and the horizontal error at the take-off point are zero.
2	After an SUAV performs vertical error correction at the vertical error correction point, its vertical error will become 0, and the horizontal error will remain unchanged.
3	After an SUAV performs horizontal error correction at the horizontal error correction point, its horizontal error will become 0, and the vertical error will remain unchanged.
4	The vertical error correction can only be performed when the vertical error of the SUAV is less than α_1, the horizontal error is not greater than α_2, and $\alpha_1 > \alpha_2$.
5	The horizontal error correction can only be performed when the vertical error of an SUAV is less than β_1, the horizontal error is not greater than β_2, and $\beta_1 < \beta_2$. Besides, α_i and β_i are independently and identically distributed ($i \in \{1, 2\}$).
6	The turning radius of an SUAV should not be less than r_{min}.
7	The cumulative error of an SUAV during the flight (including the destination) with δ units is not higher than θ.
8	In order to ensure reaching the destination as successfully as possible, SUAV error correction is performed alternately. That is, vertical or horizontal errors cannot be accumulated.
9	To ensure that the path does not return, the included angle between the initial velocity's direction of each subpath and the direction from the start point to the end point should be less than 90 degrees. The angle between the end velocity's direction of the subpath and direction from the start point to the end point should also be less than 90 degrees.
10	It is assumed that an SUAV makes a uniform circular motion or a uniform linear motion at each subpath, and there are no collisions between drones during the flight. Besides, the maximum altitude of the UAV is enough to complete the signal relay mission.

IV. PROBLEM FORMALIZATIONS

To solve the UGRA problem, we first formalize it in a similar way to GMRA. With the E-CARGO model, a system $\sum$ can be described as a 9-tuple $\sum ::= < \mathcal{C}, O, \mathcal{A}, \mathcal{M}, \mathcal{R}, \mathcal{E}, \mathcal{G}, s_0, \mathcal{H} >$, where $\mathcal{C}$ is a set of classes, O is a set of objects, $\mathcal{A}$ is

a set of agents, $\mathcal{M}$ is a set of messages, $\mathcal{R}$ is a set of roles, $\mathcal{E}$ is a set of environments, $\mathcal{G}$ is a set of groups, s_0 is the initial state of the system, and $\mathcal{H}$ is a set of users. In such a system, $\mathcal{A}$ and $\mathcal{H}$, $\mathcal{E}$, and $\mathcal{G}$ are tightly coupled sets. A human user and his/her agent perform a role together. Every group should work in an environment. An environment regulates a group.

In discussing role assignment problems, environments (e) and groups (g) are simplified into vectors and matrices, respectively. Furthermore, we use non-negative integers m ($= |\mathcal{A}|$, where $|\mathcal{A}|$ is the cardinality of set $\mathcal{A}$) to express the size of the agent set $\mathcal{A}$, n ($= |\mathcal{R}|$) the size of the role set $\mathcal{R}$, i, i_1, i_2, ... the indices of agents, and j, j_1, j_2, ... the indices of roles.

To better describe RBC and its E-CARGO model, we use the real-world scenario in Section III as an example. Without loss of generality, and to better illustrate the problem-solving process, we randomly generate parameters' values as follows. The number of relay points is 14, and the number of base stations is 8. The area covered by an SUAV has a radius of 8 km. That covered by a communication vehicle has a radius of 2 km. We also set the seven distributed communication vehicles' locations to range from 0 to 100 km. It is a suitable range for the partial real-world elevation data of Jiuzhaigou. By referring to the parameters in [13], we randomly set the parameters in Table II as follows: $\alpha_1 = 55$, $\alpha_2 = 30$, $\beta_1 = 40$, $\beta_2 = 55$, $\theta = 55$, $\delta = 0.0009$, and $r_{min} = 200$. And we will conduct large-scale simulation experiments in Section VII.

Based on the analysis above, we define the relay points on the signal relay network as roles and the different SUAV take-off base stations as agents.

Definition 1: A *role range vector* L is a vector of the lower bound of the ranges of roles in the environment e of group g.

Note: L is a valuable component of the E-CARGO model. It indicates the minimum number of agents required for each role within a properly functioning group in UGRA. Since each relay point in Fig. 3 has different connectivities, different relay points require different numbers of SUAVs. Therefore, the L vector based on the relay points in Fig. 3 is $L = [2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 1, 2, 3, 1]$.

Definition 2: An *ability limit vector* [6] L^a is an m-vector, where $L^a[i]$ $(0 \le i < m)$ indicates, at most, the number of roles that can be assigned to agent i. The superscript of L^a indicates that L^a is a definition for the agents.

From Definition 2, we introduce a new constraint, i.e., $|a_i.R_p| + 1 \le L^a[i] (0 \le i < m)$, where $a_i.R_p$ means the set of potential roles. The number of roles assigned to agent i should be less than the number of roles allowed for agent i. For example, as Table I illustrated, we randomly generated the limited number of drones for different base stations in our scenario, so $L^a = [13, 11, 6, 7, 8, 10, 9, 12]$.

Definition 3: A *vector* $L^{a\prime}$ is an m-vector. It satisfies the equation $L^{a\prime}[i] = \lceil [(L^a[i])/(2)] \rceil$ $(0 \le i < m)$. This equation depicts that each base station needs to keep at least half of the limited number of SUAVs for cruising or replacing those assigned drones accidentally damaged. In our scenario, $L^{a\prime}$ is [7, 6, 3, 4, 4, 5, 5, 6]. For example, $L^a[0] = 13$, we have $L^{a\prime}[0] = 7$ and L^a $[1] = 11$, then $L^{a\prime}[1] = 6$.

$$\begin{pmatrix} 0.99 & 1.00 & 0.90 & 0.73 & \dots & 0.49 & 0.55 & 0.58 & 0.59 \\ 0.35 & 0.36 & 0.30 & 0.15 & \dots & 0.23 & 0.16 & 0.03 & 0.32 \\ 0.60 & 0.59 & 0.45 & 0.39 & \dots & 0.79 & 0.27 & 0.38 & 0.21 \\ 0.59 & 0.57 & 0.44 & 0.40 & \dots & 0.76 & 0.32 & 0.40 & 0.30 \\ 0.62 & 0.61 & 0.50 & 0.33 & \dots & 0.12 & 0.28 & 0.19 & 0.40 \\ 0.21 & 0.21 & 0.35 & 0.47 & \dots & 0.91 & 0.68 & 0.65 & 0.79 \\ 0.53 & 0.49 & 0.35 & 0.26 & \dots & 0.10 & 0.40 & 0.24 & 0.57 \\ 0.38 & 0.39 & 0.41 & 0.41 & \dots & 0.98 & 0.32 & 0.44 & 0.25 \end{pmatrix}$$

Fig. 4. Q matrix in our scenario. *Note:* For display purposes, only the first and last four columns are shown.

Definition 4: A *qualification matrix* Q is an $m \times n$ matrix, where $Q[i, j] \in [0, 1] \cup \{+\infty\}$ expresses the qualification value of agent i $(0 \le i < m)$ for role j$(0 \le j < n)$. $Q[i, j] = 0$ indicates the lowest value, $Q[i, j] = 1$ represents the highest, and $Q[i, j] = +\infty$ represents that role i cannot act by agent j.

Note: A Q matrix is the result of the agent evaluation step of RBC. It can be obtained by comparing all the qualifications of agents with all the requirements of roles. As the assignment method with PuLP [37] has been ready to use in this article, creating Q is our major concern. Please note that a pertinent Q matrix in GMRA is not trivial and needs a significant effort, especially in a complex environment as stated in the scenario. In [3], the Q matrix's value is directly using the normalized central distance between a signal relay point and a base station. However, due to attitude errors and mechanical characteristics, the path trajectory is a continuous curve not only passing through some calibration points to timely correct its attitude errors but also by considering the SUAV's initial velocity and flight direction. For these specific requirements, we propose a more pertinent method to create Q with two algorithms in Sections VI and VII. Fig. 4 depicts the normalized shortest trajectory distances of the SUAVs obtained by our proposed algorithms for the scenario.

Definition 5: A *role assignment matrix* T is defined as an $m \times n$ matrix, where $T[i, j] \in \mathbb{N}$ $(0 \le i < m, 0 \le j < n)$ indicates whether or not agent i is assigned to role j. $T[i, j] \in \mathbb{N}^+$ means yes and 0 no.

Definition 6: A *matrix* Υ is an $m \times n$ matrix. It satisfies the equation $\Upsilon[i, j] = min\{L[j], L^{a\prime}[i]\}$. This equation represents that $\Upsilon[i, j]$ is the maximum value for the control variable $T[i, j]$ to take.

Note: In our scenario, $L^{a\prime}[i]$ represents the maximum number of SUAVs that agent (base station) i can provide while $L[j]$ represents the number of SUAVs that role (replay point) j required. Therefore, the value of $\Upsilon[i, j]$ is $min\{L[j], L^{a\prime}[i]\}$.

Definition 7: The *group performance* σ of group g is defined as the sum of the assigned agents' qualifications, i.e.,

$$\sigma = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i, j] \times T[i, j].$$

Definition 8: Role j is *workable* in group g if it has been assigned enough agents, i.e.,

$$\sum_{i=0}^{m-1} T[i, j] \ge L[j].$$

Definition 9: T is *workable* if each role j is workable, i.e., $\sum_{i=0}^{m-1} T[i, j] = L[j]$ $(0 \le j < n)$. Group g is *workable* if

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 2 & \dots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 2 & 0 \\ 2 & 2 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \end{pmatrix}$$

Fig. 5. T matrix in our scenario. *Note:* For display purposes, only the first and last four columns are shown.

T is workable. From the above definitions, group g can be expressed by a Q, L, T, $L^{a'}$, and Υ.

Definition 10: The SUAVs assignment problem via GMRA (UGRA) is to find a workable T to

$$\min\sigma = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} Q[i,j] \times T[i,j]$$

subject to

$$0 \le T[i,j] \le \Upsilon[i,j] \ (0 \le i < m, 0 \le j < n) \tag{1}$$

$$\sum_{i=0}^{m-1} T[i,j] = L[j] \ (0 \le j < n) \tag{2}$$

$$\sum_{j=0}^{n-1} T[i,j] \le L^{a'}[i] \ (0 \le i < m) \tag{3}$$

where

1) Expression (1) is an integer constraint, and it depicts the threshold of the control variable $T[i, j]$.
2) Expression (2) indicates that each role must be assigned to enough agents.
3) Expression (3) depicts that each agent can only be assigned to a limited number of roles.

Note that the UGRA problem is different from the GMRA [5] or GRA problem [3] due to the changes in Expressions (1) and (3). Now, the UGRA problem is expressed as a linear programming problem if we transfer the Q and T matrices into vectors. Then, it can be solved using an industry-standard optimization tool like the PuLP linear programming toolkit [37], which is an open-source package of Python. Fig. 5 is a T matrix, following the constraints $L = [2, 2, 2, 2, 3, 2, 3, 2, 2, 1, 2, 3, 1]$ and $L^{a'} = [7, 6, 3, 4, 4, 5, 5, 6]$. The total cost of the drones from eight bases is 5.185, which is an optimal result via UGRA.

From the above modelization, agent evaluation (obtaining the Q matrix in Definition 4) is the most significant part of GMRA and UGRA. It is also the foundation for the proposed problem to be solved. With an SUAV from one base station to a corresponding relay point, the main agent evaluation benchmark concerning qualification matrix Q is the length of the trajectory from the base station to the corresponding relay point.

V. Agent Evaluation

In this section, we mainly discuss the method to create the Q matrix. In a simple format, we can state that $Q[i, j] = AE$ (Base Station i, Relay Point j) ($0 <= i < m$, $0 <= j < n$). Agent evaluation is to create the qualification matrix Q and the key issue is to compute the matching degree between an agent and a role. Conventionally, one may think that it is just a simple computation by comparing two vectors of values. However, from the RBC's point of view, we also need to consider the environmental factors (e.g., fly trajectories, the instant situations of the sites) in the evaluation. Our major idea is to compute the shortest path from each base station to each relay point for an SUAV to fly and obtains the total path length of each corresponding path. The major challenge is to find a practical way to obtain each element of the qualification matrix, i.e., the $m \times n$ real path lengths correspond to each tuple of <base station, relay point>.

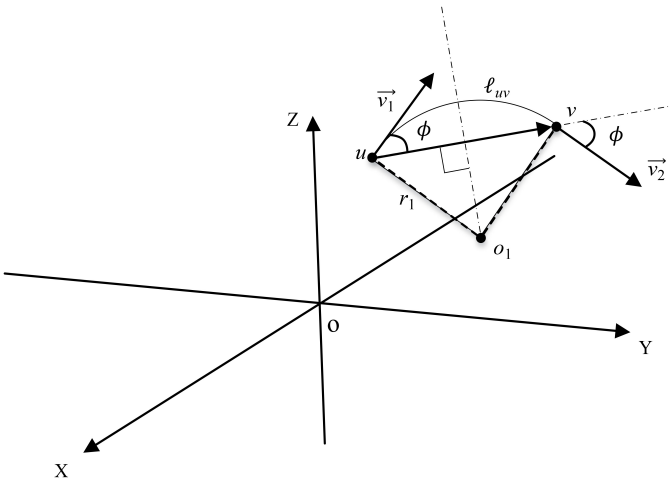


Fig. 6. Example of computing the subpath of an SUAV.

A. Dynamic Curve Path-Planning Algorithm

When calculating $Q[i, j]$, it is unavoidable to consider the error correction, velocity's direction, and multiple constraints (Table III) during an SUAV's flight. To meet this challenge, we design a DCPPA to dynamically correct attitude errors of the SUAV as well as find the shortest curve path from a base station to a relay point. To better describe DCPPA, we randomly choose one base station (Base station 3 in Fig. 3) as the start vertex s and one of its allocated signal relay points in Fig. 3 as the end vertex t. Here, we define some special symbols as follows for a better description.

1) G is the fully connected directed graph after being pruned by Constraint 7 in Table III.
2) V is the set of vertices in G. It consists of the start vertex s (one base station), the end vertex t (one signal relay point), and the calibration points.
3) $d(v)$ collects the tentative distance from the start s to some vertex v in G. $d(v) = +\infty$ represents an unreachable case.
4) Γ represents the close set that collects the vertices in V that have been checked.
5) $P(v)$ records the available path from the start s to some vertex v in G, using only vertices in Γ as intermediates. $P(v) = \{\}$ represents an unreachable path.
6) $updateError(P(v))$ is a function that removes those paths in $P(v)$ that do not meet the requirements of error correction. The flowchart of this function is illustrated in Fig. 7.

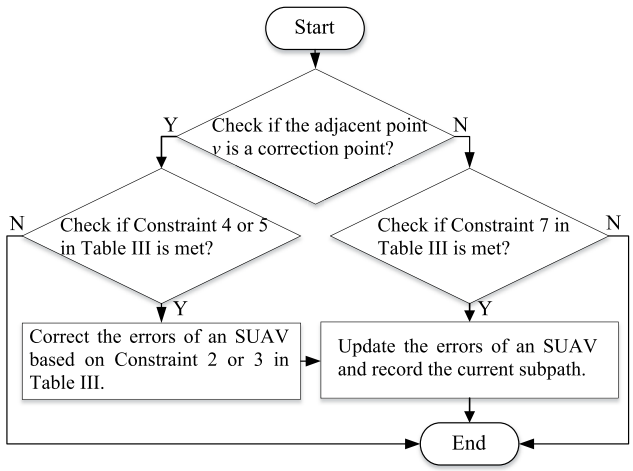


Fig. 7. Flowchart of checking and updating the errors of an SUAV.

Algorithm 1 Pseudocode of DCPPA

Input: s, t, V
Output: $d(t)$
begin
for each $u \in V-\{s\}$ **do**
$d(u) \leftarrow +\infty$;
$P(u) \leftarrow \{\}$;
end for
$u \leftarrow s$;
$d(s) \leftarrow 0$;
$P(s) \leftarrow \{s\}$;
$\Gamma \leftarrow \{\ \}$;
while $u \neq t$ **do**
$\Gamma \leftarrow \Gamma \cup \{u\}$;
for each vertex v adjacent to u **do**
computePathSet(u, v, Γ, $P(u)$);
updateErrors($P(v)$);
if $d(v) > min\{P(v)\}$ **then**
$d(v) \leftarrow min\{P(v)\}$;
end if
end for
select $u \in V-\Gamma$ with smallest $d(u)$;
end while
return $d(t)$; /*return the shortest path length from s to t*/
end

Algorithm 1 illustrates the pseudocode of the DCPPA. The main difference between DCPPA and traditional algorithms like A* is *computePathSet*(u, v, Γ, $P(u)$). This function will help $P(v)$ collect all the available candidate paths to find the shortest path. The pseudocode of this function is as follows.

With this function, DCPPA can solve the problem of finding the optimal path in Fig. 2.

There are two vital steps for DCPPA: 1) computing $P'(v)$ with $P(u)$ and adjacent vertex v and 2) checking SUAV attitude errors. Here, we use Fig. 6 to illustrate the main process of computing $P'(v)$. We assume that the start and end of the subpath are vertex u and vertex v, respectively. The initial velocity at u is $\overrightarrow{v_1}$, that at v is $\overrightarrow{v_2}$, and name the circle center and the

Pseudocode of *computePathSet*(u, v, Γ, $P(u)$)

Input: u, v, Γ, $P(u)$
Output: $P(v)$
begin
compute $P'(v)$ with $P(u)$ and u's adjacent vertex v; /*$P'(v)$ is a temporary variable for $P(v)$*/
if v not in Γ **then**
$P(v) \leftarrow P(v) \cup P'(v)$;
return $P(v)$;
else
for each vertex w adjacent to v **do**
computePathSet(v, w, Γ, $P'(v)$)
end for
end if
end

turning radius as o_1 and r_1, respectively. The main problem of computing the $P'(v)$ is to calculate r_1, o_1, and $\overrightarrow{v_2}$.

We first solve the turning radius r_1. According to the spatial geometric position relationship and the nature of the dot product of a vector, r_1 can be expressed as follows:

$$r_1 = \frac{\|\overrightarrow{uv}\|_2}{2\times\sin\left(\arccos\left(\frac{\overrightarrow{uv}\cdot\overrightarrow{v_1}}{\|\overrightarrow{uv}\|_2\times\|\overrightarrow{v_1}\|_2}\right)\right)} \tag{4}$$

where $\|\overrightarrow{uv}\|_2$ and $\|\overrightarrow{v_1}\|_2$ are the L_2 norm of $\overrightarrow{uv}$ and $\overrightarrow{v_1}$ separately. After getting the turning radius r_1, the next step is to calculate the circle center o_1. Based on the parallelogram law of vector, we can formalize the coordinate of the circle center o_1 as

$$o_1 = \overrightarrow{ou} + r_1 \times \frac{\overrightarrow{h}}{\left\|\overrightarrow{h}\right\|_2} \tag{5}$$

where

$$\overrightarrow{h} = \frac{\overrightarrow{uv}\cdot\overrightarrow{v_1}}{\left|\overrightarrow{uv}\cdot\overrightarrow{v_1}\right|} \times \left(\frac{\overrightarrow{v_1}\cdot\overrightarrow{v_1}}{\overrightarrow{uv}\cdot\overrightarrow{v_1}} \times \overrightarrow{uv} - \overrightarrow{v_1}\right) \tag{6}$$

and o is the coordinate origin.

Proof: We take Fig. 6 as an example to illustrate Expression (5). According to the nature of the vector, we can express o_1 as $\overrightarrow{ou} + \overrightarrow{uo_1}$. Since $\left|\overrightarrow{uo_1}\right| = r_1$, $\overrightarrow{uo_1}$ can be expressed as $r_1 \times \overrightarrow{e}$, where $\overrightarrow{e}$ is the unit vector of $\overrightarrow{uo_1}$. The problem is transformed into solving the unit vector $\overrightarrow{e}$. With the help of the Parallelogram rule, we assume that there is a vector $\overrightarrow{h}$ in the $\overrightarrow{uo_1}$ direction. As $\overrightarrow{h} \cdot \overrightarrow{v_1} = 0$, it can be synthesized by $\overrightarrow{uv}$ and $\overrightarrow{v_1}$ as in

$$\overrightarrow{h} = \frac{\overrightarrow{uv}\cdot\overrightarrow{v_1}}{\left|\overrightarrow{uv}\cdot\overrightarrow{v_1}\right|} \times \left(\frac{\overrightarrow{v_1}\cdot\overrightarrow{v_1}}{\overrightarrow{uv}\cdot\overrightarrow{v_1}} \times \overrightarrow{uv} - \overrightarrow{v_1}\right).$$

The left part of the equation above is to unify the expression of $\overrightarrow{h}$ because the angle between $\overrightarrow{uv}$ and $\overrightarrow{v_1}$ may be acute or obtuse. Therefore, o_1 can be expressed as

$$o_1 = \overrightarrow{ou} + r_1 \times \frac{\overrightarrow{h}}{\left\|\overrightarrow{h}\right\|_2}.$$

Hence, we can conclude the proof. ■

After determining r_1 and o_1, the final step is calculating $\overrightarrow{v_2}$. As u, $\overrightarrow{v_1}$ and v are known, the angle ϕ between $\overrightarrow{v_1}$ and

$\overrightarrow{uv}$ can be easily calculated. Due to the symmetry of the arc, the center angle of the subpath ℓ_{uv} is 2ϕ. Besides, the unit normal vector $\overrightarrow{j}$ of the plane consisting of u, o_1, and v can be represented as

$$\overrightarrow{j} = \frac{\left(-\overrightarrow{v_1}\right) \times \overrightarrow{uv}}{\left\|\left(-\overrightarrow{v_1}\right) \times \overrightarrow{uv}\right\|_2}.$$

Here, we take the opposite direction of $\overrightarrow{v_1}$ to fix the direction of the normal vector $\overrightarrow{j}$. With the help of the rotation matrix [38], it is easy to calculate $\overrightarrow{v_2}$ as follows:

$$\overrightarrow{v_2} = M \cdot \overrightarrow{v_1} \tag{7}$$

where

$$M = \begin{bmatrix} \cos\vartheta+(1-\cos\vartheta)x^2 & (1-\cos\vartheta)xy-(\sin\vartheta)z & (1-\cos\vartheta)xz+(\sin\vartheta)y \\ (1-\cos\vartheta)yx+(\sin\vartheta)z & \cos\vartheta+(1-\cos\vartheta)y^2 & (1-\cos\vartheta)yz-(\sin\vartheta)x \\ (1-\cos\vartheta)zx-(\sin\vartheta)y & (1-\cos\vartheta)zy+(\sin\vartheta)x & \cos\vartheta+(1-\cos\vartheta)z^2 \end{bmatrix}$$

$\vartheta = -2\phi$ and unit normal vector $\overrightarrow{j} = (x, y, z)$. We set ϑ as -2ϕ because $\overrightarrow{v_1}$ rotates -2ϕ clockwise around $\overrightarrow{j}$ and can obtain the vector of $\overrightarrow{v_2}$. Once r_1, o_1, and $\overrightarrow{v_2}$ are determined, $P'(v)$ can be easily calculated.

Another vital step for DCPPA is to immediately check and update an SUAV's attitude errors. Fig. 7 shows the flowchart of checking and updating the attitude errors of an SUAV.

B. Theoretical Proof of the Optimality of DCPPA

To demonstrate our algorithm's feasibility, we now prove its optimality using mathematical induction. Here, for clarity, we introduce several additional symbols.

1) $\tau(v)$ collects the shortest path from the start vertex s to some vertex v in G.
2) $\text{Set}_{\text{DCPPA}}(s, t)$ is a set to collect all the vertices of the shortest path from s to t obtained by the DCPPA.
3) D_{uv} is a value to record subpath length from vertex u to some vertex v and u, $v \in \text{Set}_{\text{DCPPA}}(s, t)$.

Lemma 1: $\forall\ v \in V-\Gamma$ and v is adjacent to Γ, if $d(v) \neq +\infty$, then $d(v) = min\{P(v)\}$.

Proof: Let $|\Gamma| = N_1$ when v is the first time to be the adjacent vertex to Γ, and $|\Gamma| = N_1 + \rho$ ($\rho \in \mathbb{N}^+$) when v is the last time to be the adjacent vertex to Γ.

Base Case: Initially, when $|\Gamma| = N_1$, $d^{N_1}(v) = min\{+\infty, P(v)\}$, so Lemma 1 is trivially true.

Induction Hypothesis: When $|\Gamma| = N_1 + \rho - 1$, we denote the current $P(v)$ as $P^{N_1+\rho-1}$ and have $d^{N_1+\rho-1}(v) = min\{P^{N_1+\rho-1}\}$.

Induction Step: To complete the proof, we only need to demonstrate that Lemma 1 still holds when $|\Gamma| = N_1 + \rho$. We assume that the added path when $|\Gamma| = N_1 + \rho$ is $P'(v)$. Therefore, the possible path set is $P(v) = P^{N_1+\rho-1} \cup P'(v)$. Clearly, the current shortest path length from s to v of current Γ is

$$\begin{aligned} d^{N_1+\rho}(v) &= min\{P(v)\} \\ &= min\left\{min\left\{P^{N_1+\rho-1}\right\}, min\{P'(v)\}\right\} \\ &= min\left\{d^{N_1+\rho-1}(v), min\{P'(v)\}\right\}. \end{aligned}$$

Consequently, when $|\Gamma| = N_1 + \rho$, $d(v)$ still holds the shortest path from s to v in Γ. Hence, we can conclude the proof. ■

Note: Lemma 1 demonstrates that $d(v)$ in Algorithm 1 collects the shortest path from the start vertex s to vertex v in the close set Γ.

Theorem 1: $\forall\ v \in \Gamma$, if $d(v) \neq +\infty$, then $d(v) = \tau(v)$.

Proof:

Base Case: The only time when $|\Gamma| = 1$ is $\Gamma = \{s\}$ and $d(s) = \tau(s) = 0$, hence Theorem 1 is trivially correct.

Induction Hypothesis: When $|\Gamma| = \rho$ ($\rho \in \mathbb{N}^+$), Theorem 1 still holds. As Γ has been expanded, we name the current Γ as Γ'.

Induction Step: We select $v \in V - \Gamma$ with the smallest $d(v)$, add it to Γ, and $\Gamma = \Gamma' \cup \{v\}$. All the possible paths from s to v are composed of two types of paths: 1) those that only use the vertices in Γ as intermediates and 2) the other is those using vertices in both Γ and $V - \Gamma$ as intermediates. Lemma 1 has shown that $d(v)$ is the shortest distance for the first type of path. We only need to prove that $d(v)$ is also the shortest path length for the second type of path to complete the proof. Suppose that $d(v)$ is not the shortest path length for the second type of path, which means

$$\tau(v) < d(v).$$

We assume that this type of path starts from Γ, leaves Γ at vertex w, and goes through vertices of $V - \Gamma$. Based on Lemma 1, $\tau(v)$ can be expressed as

$$\tau(v) = d(w) + \varepsilon$$

where ε ($\varepsilon > 0$) represents the path length from vertex w to the end vertex t via the vertices in $V-\Gamma$. Based on Algorithm 1, we select $v \in V-\Gamma$ with the smallest $d(v)$. Hence, combining these inequalities gives us the contradiction that $d(v) > d(w)$. Consequently, $d(v)$ is the shortest path in Γ and $d(v) = \tau(v)$. Hence, we can conclude the proof. ■

Note: Theorem 1 illustrates that $d(v)$ in Algorithm 1 collects the shortest path from the start vertex s to vertex v in G.

Corollary 1: $\forall s, t \in G$, if $v \in \text{Set}_{\text{DCPPA}}(s, t)$, then $d(v) + D_{vt} \leq d(t)$.

Proof: If the shortest path from s to v is subpath of the shortest path from s to t, we can deduce $d(v) = D_{sv}$ and $d(v) + D_{vt} = D_{sv} + D_{vt} = d(t)$. Otherwise, the shortest path from s to v is not the same as the subpath of the shortest path from s to t, i.e., $d(v) < D_{sv}$. Therefore, we can deduce $d(v) + D_{vt} < D_{sv} + D_{vt} = d(t)$. Hence, we can conclude the proof. ■

Note: Corollary 1 depicts that for any vertex v in G, if vertex v is on the shortest path obtained by the DCPPA (named $\mathcal{P}_{\text{DCPPA}}(s, t)$) from the start s to the end t, the value of $d(v)$ will be less than or equal to the subpath length of $\mathcal{P}_{\text{DCPPA}}(s, t)$ from s to v. It is applied to demonstrate Theorem 3.

VI. Improvement

Since the worst condition of DCPPA will traverse all the feasible paths from the start vertex s to the end vertex t, the worst-case time complexity of this process is

Algorithm 2 Pseudocode of GDCPPA

Input: s, t, V
Output: $d(t)$
begin
for each $u \in V-\{s\}$ **do**
$d(u) \leftarrow +\infty$;
end for
$u \leftarrow s$;
$d(s) \leftarrow 0$;
$R \leftarrow \{\}$;
while $u \neq t$ **do**
$\Gamma' \leftarrow \Gamma \cup \{u\}$
for each vertex $v \in V-\Gamma$ and v adjacent to u **do**
compute $Test(u, v)$;
if $Test(u, v) =$ True **then**
compute $\Im(u, v)$;
for each vertex $w \in V-\Gamma'$ and w adjacent to v **do**
compute $Test(v, w)$;
if $Test(v, w) =$ True **then**
if $d(v) > \Im(u, v)+d(u)$ **then**
$d(v) \leftarrow \Im(u, v)+d(u)$;
end if
$\Gamma \leftarrow \Gamma'$;
end if
end for
end if
end for
select $u \in V-\Gamma$ with smallest $d(u)$;
end while
return $d(t)$
end

$O(\sum_{z=0}^{k} k!/(k-z)!) = O(\lfloor e \times k! \rfloor)$, where k represents the number of correction points. This high time complexity of the DCPPA may be intolerable for disaster relief tasks. To help the UGRA model meet the time requirement of disaster relief, we need to improve the algorithm by proposing one sufficient condition and one necessary condition to accelerate DCPPA for decreasing the time of calculating $Q[i, j]$.

A. Sufficient Condition

Due to Constraint 6 in Table III, when computing the minimum subpath, we should consider both vertex position and velocity's direction. That means we may need to compare all the available paths from the beginning point s to a calibration point in G. This process is quite time consuming. As the distribution of correction points is sparse in the actual disaster area, the effect of the initial velocity direction on the path length is relatively small. Thus, we empirically ignore the impact of the velocity and merely focus on comparing the path length. With this premise, we propose a new algorithm called the greedy DCPPA (GDCPPA). Algorithm 2 illustrates the pseudocode of the GDCPPA. $\mathcal{T}(u, v)$ denotes path length from u to v. Test(u, v) checks if the current path from u to v satisfies the constraints in Table III (including errors correction).

Inspired by Constraint 8 in Table III, a vital feature of GDCPPA is that it needs to judge whether the SUAV can reach the next two nearby calibration points before updating the path distance. The complexity of GDCPPA is $O(k^3)$ where $k = |V|$. With the path length result obtained by GDCPPA as a heuristic condition, it is able to greatly reduce the solution space of DCPPA (see Theorem 3). Here, we introduce several additional symbols for better describing the theorem.

1) Ξ_{vt} is a list to record the minimum straight-line distance obtained by the Dijkstra algorithm from some point v in G to the end t.
2) $\mathcal{P}_{\mathrm{GDCPPA}}(s, t)$ is a value to record the path length from vertex s to t obtained by the GDCPPA. $\mathcal{P}_{\mathrm{GDCPPA}}(s, t) = +\infty$ represents no solution.

Theorem 2: $\forall s, t \in G$, if $\mathcal{P}_{\mathrm{GDCPPA}}(s, t) \neq +\infty$, then $D_{st} \neq +\infty$ and $D_{st} <= \mathcal{P}_{\mathrm{GDCPPA}}(s, t)$.

Proof: If there exists $\mathcal{P}_{\mathrm{GDCPPA}}(s, t) \neq +\infty$, the GDCPPA can obtain a path, named ℓ_1, from the start vertex to the end vertex in the global space that satisfies the constraints of Table III. If ℓ_1 is the shortest path, we can deduce $\mathcal{P}_{\mathrm{GDCPPA}}(s, t) <= D_{st}$ based on Theorem 1. Otherwise, our proposed DCPPA will obtain a new path, assumed ℓ_2, that is shorter than ℓ_1 and $D_{st} < \mathcal{P}_{\mathrm{GDCPPA}}(s, t)$. Therefore, we can conclude the proof. ■

Note: Theorem 2 demonstrates that if the GDCPPA has a solution, the DCPPA must have a solution.

Based on Theorem 2, we can accelerate the solving time of the DCPPA with Theorem 3.

Theorem 3: $\forall s, t \in G$, if vertex $v \in \mathrm{Set}_{\mathrm{DCPPA}}(s, t)$, then $D_{sv} + \Xi_{vt} < \mathcal{P}_{\mathrm{GDCPPA}}(s, t)$.

Proof: Corollary 1 illustrates $\Xi_{vt} < D_{vt}$ and $D_{sv}+D_{vt} \leq D_{st}$. Thereby, we can deduce $D_{sv} + \Xi_{vt} < D_{st}$. Also, Theorem 2 tells that $D_{st} = \mathcal{P}_{\mathrm{GDCPPA}}$. Consequently, we can deduce $D_{sv} + \Xi_{vt} < \mathcal{P}_{\mathrm{GDCPPA}}(s, t)$. Hence, we can conclude the proof. ■

Note: Theorem 3 depicts a heuristic condition that for any vertex v on the shortest path obtained by the DCPPA, the path length from the origin s to v plus the straight trajectory length from v to the end t is shorter than the path length from s to t obtained by the GDCPPA.

B. Necessary Condition

Now, we consider an ideal situation. We assume that an SUAV can maintain a straight-line flight trajectory throughout the calibration process. Hence, we named this flight strategy as a dynamic straight-line path-planning algorithm (DSPPA). It is a straight-line instance of the DCPPA and its optimality proof process has already been stated. Here, we propose Theorem 4 to depict the necessary condition of the DCPPA. We first introduce additional symbols for clarity.

1) $\mathcal{P}_{\mathrm{DSPPA}}(s, t)$ is a value to record the path length from vertex s to t obtained by the DSPPA. $\mathcal{P}_{\mathrm{DSPPA}}(s, t) = +\infty$ represents no solution.

Theorem 4: $\forall s, t \in G$, if $\mathcal{P}_{\mathrm{DSPPA}}(s, t) = +\infty$, then $D_{st} = +\infty$.

Proof: To simplify the proof, we consider the converse-negative proposition. If there exists $D_{st} \neq +\infty$, the DCPPA can obtain the shortest curve path ℓ_1 from s to t. This path ℓ_1 is also a candidate curve path ℓ_2 for the DSPPA. As the shortest distance between two points is always a straight

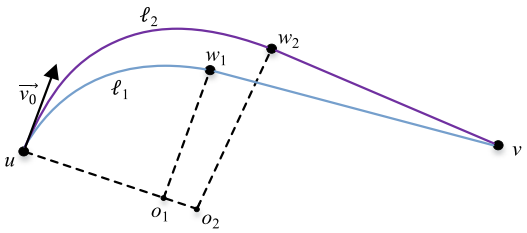


Fig. 8. Feasible extreme flight trajectory. *Note:* o_1 and o_2 are the centers of arc parts, and w_1 and w_2 represent the tangent points for paths ℓ_1 and ℓ_2, respectively.

line, the distance d'_i between two adjacent vertices of ℓ_2 is shorter than the counterpart d_i of ℓ_1. Therefore, we can deduce $\mathcal{P}_{\mathrm{DSPPA}}(s,\mathrm{t}) = \sum_i d'_i < D_{st} = \sum_i d_i \neq +\infty$. Hence, we can conclude the proof. ■

Note: Theorem 4 demonstrates that if the DSPPA has no solution, the DCPPA must have no solution.

With Theorem 4, we can speed up the DCPPA when the DSPPA has no solution. Based on Theorems 3 and 4, we can preprune the tree-shaped solution space in advance to speed up the DCPPA convergence (see the accelerated DCPPA part in Fig. 9).

C. Theoretically Feasible Extreme Flight Trajectory

Although a curving flight path is airframe-friendly, when the calibration points are sparse, this flight mode may cause no solution when the distribution of calibration points is extremely sparse. Therefore, here we propose an alternative flight method that meets the constraints in Table III and has a shorter subpath flight. Reference [39, Proposition 5.6], where Crofton's formula [40] is utilized, has demonstrated that between two nested, convex, closed curves on the plane, the inner one is shorter. Consequently, When an SUAV flies at a minimum turning radius (r_{min}in Constraint 6) before transitioning to linear motion, it has the shortest subpath (see Fig. 8).

Although Algorithms 1 and 2 are applicable for this flight trajectory, we use Algorithm 2 because it is faster than Algorithm 1. For convenience, we name this algorithm as greedy curved straight path-planning algorithm (GCSPPA). Based on the above analysis, the flowchart of the agent evaluation method for UGRA is illustrated by Fig. 9.

Fig. 10 shows the path-planning result for an allocated SUAV from Base 3 to a designated signal relay point. The red points on the curve of GCSPPA are the tangent points (similar to the w_1, w_2 in Fig. 8). The results of these path-planning algorithms in this example are DCPPA-107.36 km, Dijkstra and A*-110.10 km, Improved ACO-111.68 km, and GCSPPA-104.83 km.

VII. Experiment

To verify our proposed agent evaluation method, we performed simulation experiments on a laptop configured as Table IV, and all the simulation datasets utilized in this article are accessible in the public Github project of the authors: https://github.com/jiangqian1997/E-CARGO-Codes/tree/main/UGRA_datasets. The descriptions of these datasets are in a file named "README.txt" of this project list.

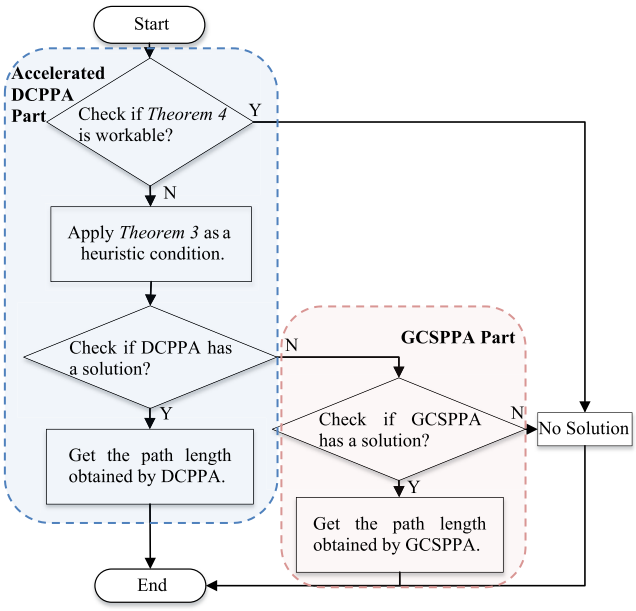


Fig. 9. Flowchart of the agent evaluation method for UGRA.

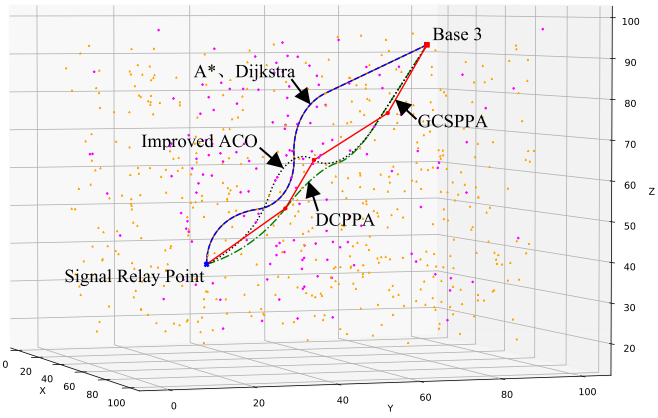


Fig. 10. Example of an SUAV's path-planning result.

TABLE IV
Configuration of the Experimental Platform

Hardware	
CPU	2.6 GHz Intel Core i7
Memory	16GB 2400 MHz DDR4
Software	
OS	macOS Big Sur Version 11.1
Editor	Visual Studio Code Version 1.52.1
Python	Python 3.8.5

The 16th China Post-Graduate Mathematical Contest in Modeling [13] provides two types of calibration point data sets, one sparse data set and one dense data set. Here, we use these two data sets as the original data sets to carry out our experiments. For the predefined parameters in Table II, [41] and [42] point out that the position errors parameters, such as α_1, α_2, β_1, β_2, and θ, are positively related to $\mathfrak{L}\times\delta$, where $\mathfrak{L}$ is the flight trajectory length of an SUAV. By referring the UAV parameters in [41] and [43], we empirically set the range of δ from 0 to 0.002 and the value of $\mathfrak{L}$ as 50 000 m. Besides, based

TABLE V
DISTRIBUTIONS OF THE PREDEFINED PARAMETERS

Parameters	Distributions
δ	$U[0, 0.002]$
θ	$U[0, 50000\times\delta]$
$\alpha_1, \alpha_2, \beta_1, \beta_2$	$U[0, \theta]$
r_{min}	$U[0, 100000\times\delta]$

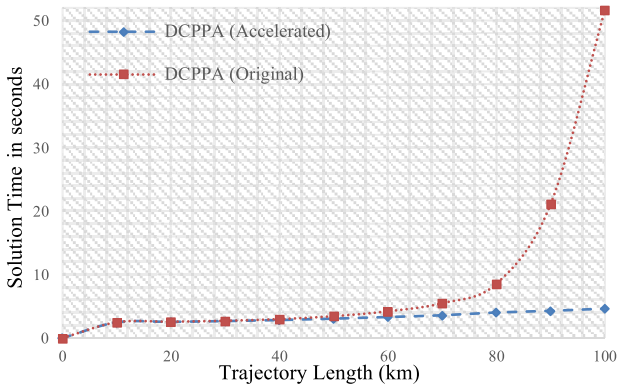


Fig. 11. Comparison of average solution times between original DCPPA and accelerate DCPPA.

on the data of the shortest turning radius in [44] and [45], we set the value of r_{min} from 0 to 200 m. For convenience, here we empirically judge r_{min} is positively related δ so we can use δ to represent the range of r_{min}, i.e., [0, 100 000 $\times\delta$]. Table V shows the distributions of the predefined parameters in Table II. We conduct our experiments with randomly generated parameters in Table V for thousands of times.

We first focus on the validity of one sufficient condition and one necessary condition proposed in Section VI for the DCPPA. Fig. 11 shows the average solution time of the original DCPPA and the accelerated DCPPA at different trajectory lengths of SUAVs. It demonstrates that the proposed acceleration method can greatly reduce the solution time of DCPPA when the flight trajectory length increases.

Then, we compare the performance of our accelerated DCPPA and GCSPPA with some traditional algorithms. Fig. 12 depicts the average solution time of all algorithms under the two original data sets, and Table VI shows the average solution rate and average flight path distance of these algorithms. Here, we improve the ACO algorithm by adding a heuristic condition for acceleration. Fig. 12 illustrates that our proposed algorithms can solve the complex path-planning problem in an acceptable time. Although the A* and Dijkstra algorithm can also swiftly solve the problem, Table VI shows that their solution rates, especially in a relatively sparse data set, are too low and their trajectory lengths are longer than the accelerated DCPPA. This result also confirms the issue we mentioned in Fig. 2.

Without loss of generality, we test our proposed algorithms under different scales of correction point sets. Here, we design Equation (8) to extend two original data sets. Equation (8) can extend a small data set to a newly designed one by adding random interferences. Here, we name the original data set as $S = \{s_i | s_i \in \mathbb{R}^3,\ 1 \le i \le \rho\}$ where $\rho \in \mathbb{N}^+$. We denote

TABLE VI
COMPARISON OF THE ALGORITHMS

Sparse Data Set					
	DCPPA (Accelerated)	Improved ACO	Dijkstra	A*	GCSPPA
Solution Rate (In average) (%)	69.80%	69.80%	25.80%	27.30%	99.30%
Trajectory Length (In average) (km)	70.59	70.75	74.41	75.26	63.54
Dense Data Set					
	DCPPA (Accelerated)	Improved ACO	Dijkstra	A*	GCSPPA
Solution Rate (In average) (%)	99.10%	99.10%	67.60%	74.50%	100%
Trajectory Length (In average) (km)	66.92	68.22	73.37	73.70	63.06

$s_i = (x_i, y_i, z_i)$. Besides, we denote F as $max\{x_i\}-min\{x_i\}$, W as $max\{y_i\}-min\{y_i\}$, H as $max\{z_i\}-min\{z_i\}$ to, respectively, represent the length, width, and the height of the minimal cuboid containing all the original points. To automatically generate large-scale point data, we extend S along three axes with $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$ ($\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z} \in \mathbb{N}^*$) times, respectively. Then, the new large-scale point set is defined as

$$S' = \left\{ s_i + \Delta d + \xi \;\middle|\; \begin{array}{c} 0 \le \sigma \le \mathcal{X},\ 0 \le \mathcal{P} \le \mathcal{Y},\ 0 \le \mathcal{Q} \le \mathcal{Z}, \\ \Delta d = (\sigma \times F, \mathcal{P} \times W, \mathcal{Q} \times H), \\ \xi = (r\cos\varphi\sin\omega, r\cos\varphi\cos\omega, r\sin\varphi), \\ s_i \in S,\ 1 \le i \le \rho \end{array} \right\} \tag{8}$$

where $\sigma, \mathcal{P}, \mathcal{Q} \in \mathbb{N}^*$, ρ represents the number of correction points of the original point set S. ξ is stochastic disturbance and

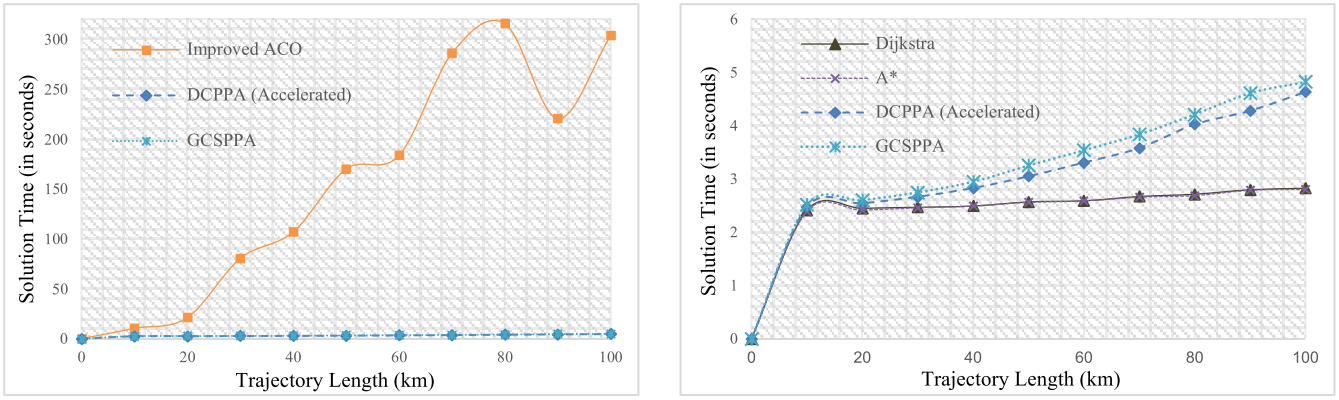
$$\varphi,\ \omega \in U(0,\ 2p),\ r \sim p(\chi;\eta) = \frac{\sqrt{2}}{\eta\sqrt{\pi}}\exp\left(\frac{-\chi^2}{2\eta^2}\right),\ \chi \ge 0.$$

Here, we empirically fix $r \sim p(\chi; 1)$ for maintaining the data distribution of two original data sets. Fig. 13 illustrates the average solution time of the proposed algorithms when the number of correction points changes.

Finally, we test the feasibility of the UGRA. Here, we analyze the average solution time of UGRA when the number of the relay points increases. The solution time of UGRA consists of the time of role negotiation, the time for the agent evaluation method computing the qualification matrix Q, and the time of the role assignment for the multi-SUAVs systems in UGRA. Fig. 14 shows that the agent evaluation part is the most vital factor affecting the solution time of UGRA, while Fig. 9 illustrated that the accelerated DCPPA and GCSPPA cooperatively form our agent evaluation method. Since each element in Q is independent, Q can be processed in parallel, and the theoretical solution time of UGRA is within a minute.

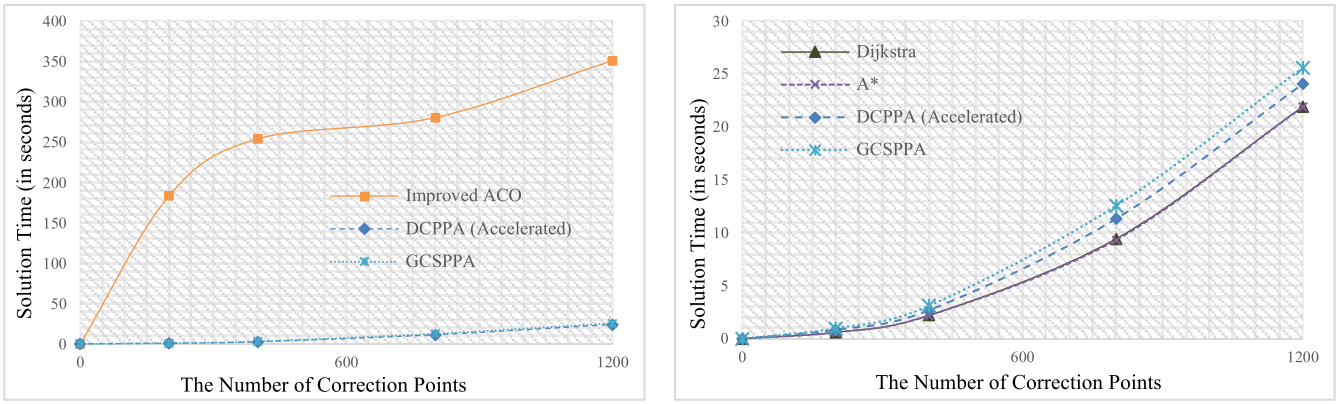
From the experiment above, we can conclude the following.

1) Fig. 11 illustrates that accelerated by one sufficient condition and one necessary condition proposed in Section VI, the solution time of DCPPA is greatly improved.
2) Figs. 12 and 13 and Table VI illustrate that the traditional algorithms and intelligent algorithms are not suitable for solving this complex path-planning problem.



(a) (b)

Fig. 12. Comparison of average solution times of the proposed algorithms with traditional algorithms. (a) Compared with the improved ACO algorithm. (b) Compared with the Dijkstra and A* algorithms. **Note:* Due to the large time span of the ACO algorithm, we compare it separately with the Dijkstra and A* algorithms, namely, Fig. 12(a) and (b).



(a) (b)

Fig. 13. Comparison of average solution times of the proposed algorithms with traditional algorithms when the number of correction points increases. (a) Compared with the improved ACO algorithm. (b) Compared with the Dijkstra and A* algorithms. **Note:* Due to the large time span of the ACO algorithm, we compare it separately with the Dijkstra and A* algorithms, namely, Fig. 13(a) and (b).

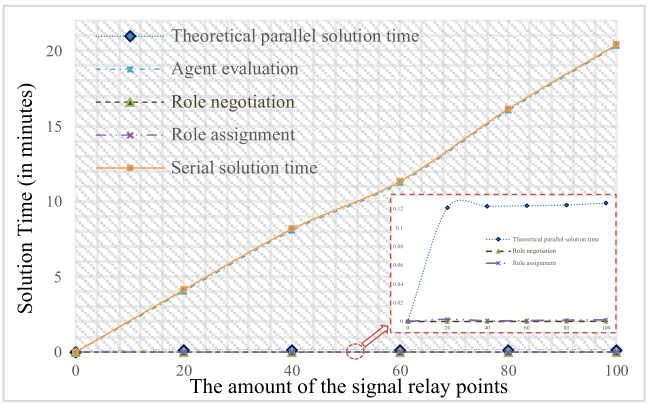


Fig. 14. Average solution time of UGRA. **Note:* Due to the large time span of the agent evaluation method and the serial solution process, we independently compare the time costs of the other three processes. See the small figure corresponding to the wine red dotted circle for details. Please note that the axes of this small figure have the same meaning as those in Fig. 14.

3) Although the continuous curve flight is airframe-friendly, Table V shows that the accelerated DCPPA still has some unsolved cases for a curve flight. Therefore, we use the collaborative combination of the accelerated DCPPA and GCSPPA to form the agent evaluation method in UGRA (see Fig. 9).
4) With the agent evaluation method consisting of the accelerated DCPPA and GCSPPA, the qualification matrix Q in UGRA can be acquired. Once the qualification matrix Q is determined, UGRA can rapidly deploy multiple SUAVs from different SUAV take-off base stations to the corresponding relay points within a minute (see Fig. 14).

VIII. Conclusion

This article proposes an innovative agent evaluation method for SUAV dispatching under the unified GMRA model, i.e., the formalization of multi-SUAVs deployment for signal relays via GMRA (UGRA). This agent evaluation method aids in SUAV path planning under a complex environment in disaster relief. With this method, the assignment of multi-SUAVs to different relay points for communication recovery can be undertaken quickly via the unified modeling method UGRA with the PuLP

package of Python. Thousands of varying scale simulation experiments based on real-world data are carried out to test the proposed assignment method's accuracy and effectiveness.

The agent evaluation method in UGRA consists of two algorithms: 1) DCPPA and 2) GCSPPA. These two algorithms can swiftly plan the appropriate path for the SUAV in a complex environment with uncertain subpaths and accumulative attitude errors. Note that existing algorithms, such as Dijkstra, A* and intelligent algorithms like Improved ACO cannot properly accomplish this path-planning task. Besides, we accelerate DCPPA by proposing one sufficient condition and one necessary condition.

Our proposed agent evaluation method above can be used to compute the qualification matrix Q, which can help UGRA assign multiple SUAVs from different base stations to corresponding relay points to establish collaborative relay networks within an acceptable time.

From this article, further investigations may be required in the following directions.

1) The proposed accelerated DCPPA algorithm is still time consuming on large and high-density data sets. More acceleration conditions are needed to reduce time complexity.
2) In an urban environment, obstacle avoidance should be considered. Hence, our proposed algorithms need to be further studied soon to meet these unique requirements.
3) More curve types, such as the Bezier curve, can be considered relative to an SUAV flight trajectory.

ACKNOWLEDGMENT

The authors appreciate the assistance of Mike Brewes in proofreading this article. Any opinions and conclusions in this work are strictly those of the author(s) and do not reflect the views, positions, or policies of—and are not endorsed by—IDEaS, DND, or the Government of Canada.

REFERENCES

[1] J. Wu *et al.*, "Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by adaptive grasshopper optimization algorithm," *Aerosp. Sci. Technol.*, vol. 70, pp. 497–510, Nov. 2017.

[2] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, and H. Yang, "Path planning for solar-powered UAV in urban environment," *Neurocomputing*, vol. 275, pp. 2055–2065, Jan. 2018.

[3] D. Liu, Q. Jiang, H. Zhu, and B. Huang, "Distributing UAVs as wireless repeaters in disaster relief via group role assignment," *Int. J. Cooper. Inf. Syst.*, vol. 29, nos. 1–2, 2020, Art. no. 2040002.

[4] X. Liu and N. Ansari, "Resource allocation in UAV-assisted M2M communications for disaster rescue," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 580–583, Apr. 2019.

[5] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, and S. Teng, "Solving the many to many assignment problem by improving the Kuhn-Munkres algorithm with backtracking," *Theor. Comput. Sci.*, vol. 618, pp. 30–41, Mar. 2016.

[6] H. Zhu, D. Liu, S. Zhang, S. Teng, and Y. Zhu, "Solving the group multi-role assignment problem by improving the ILOG approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 12, pp. 3418–3424, Dec. 2017.

[7] H. Zhu and M. C. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 578–589, Jul. 2006.

[8] X. Zhu, H. Zhu, D. Liu, and X. Zhou, "Criteria making in role negotiation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3731–3740, Oct. 2020.

[9] H. Zhu and R. Alkins, "Improvement to rated group role assignment algorithms," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (ICSMC)*, 2009, pp. 4716–4721.

[10] D. Liu, Y. Yuan, H. Zhu, S. Teng, and C. Huang, "Balance preferences with performance in group role assignment," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1800–1813, Jun. 2018.

[11] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 535–547, Apr. 2016.

[12] China Academic Degrees and Graduate Education Development Center, "Optimal usage of UAV in rescue and disaster relief," in *Proc. 14th China Post Graduate Math. Contest Model.*, 2017, pp. 1–3.

[13] China Academic Degrees and Graduate Education Development Center, "Rapid path planning of intelligent aircraft under multiple constraints," in *Proc. 16th China Post Graduate Math. Contest Model.*, 2019, pp. 1–4.

[14] Y. Lin and S. Saripalli, "Sampling-based path planning for UAV collision avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3179–3192, Nov. 2017.

[15] S. Wang, F. Jiang, B. Zhang, R. Ma, and Q. Hao, "Development of UAV-based target tracking and recognition systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3409–3422, Aug. 2020.

[16] K. Harikumar, J. Senthilnath, and S. Sundaram, "Mission aware motion planning (MAP) framework with physical and geographical constraints for a swarm of mobile stations," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1209–1219, Mar. 2020.

[17] Z. Wang and C. Shen, "Small cell transmit power assignment based on correlated bandit learning," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1030–1045, May 2017.

[18] X. Zhou, Q. Wu, S. Yan, F. Shu, and J. Li, "UAV-enabled secure communications: Joint trajectory and transmit power optimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4069–4073, Apr. 2019.

[19] B. Ji, Y. Li, D. Cao, C. Li, S. Mumtaz, and D. Wang, "Secrecy performance analysis of UAV assisted relay transmission for cognitive network with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7404–7415, Jul. 2020.

[20] H. Wang, J. Wang, G. Ding, J. Chen, F. Gao, and Z. Han, "Completion time minimization with path planning for fixed-wing UAV communications," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3485–3499, Jul. 2019.

[21] A. Mardani, M. Chiaberge, and P. Giaccone, "Communication-aware UAV path planning," *IEEE Access*, vol. 7, pp. 52609–52621, 2019.

[22] P. Yao, H. Wang, and Z. Su, "Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment," *Aerosp. Sci. Technol.*, vol. 47, pp. 269–279, Dec. 2015.

[23] F. Demiane, S. Sharafeddine, and O. Farhat, "An optimized UAV trajectory planning for localization in disaster scenarios," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107378.

[24] S. Uluskan, "Noncausal trajectory optimization for real-time range-only target localization by multiple UAVs," *Aerosp. Sci. Technol.*, vol. 99, Apr. 2020, Art. no. 105558.

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

[26] N. Geng, Q. Meng, D. Gong, and P. W. H. Chung, "How good are distributed allocation algorithms for solving urban search and rescue problems? A comparative study with centralized algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 478–485, Jan. 2019.

[27] D. Liu, B. Huang, and H. Zhu, "Solving the tree-structured task allocation problem via group multirole assignment," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 41–55, Jan. 2020.

[28] J. Schwarzrock, I. Zacarias, A. L. C. Bazzan, R. Q. de Araujo Fernandes, L. H. Moreira, and E. P. de Freitas, "Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 10–20, Jun. 2018.

[29] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems (Revised Reprint)*. Philadelphia, PA, USA: SIAM, 2012.

[30] R. Wang, G. Xiao, and P. Wang, "Hybrid centralized-decentralized (HCD) charging control of electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6728–6741, Aug. 2017.

[31] D. Liu *et al.*, "Task-driven relay assignment in distributed UAV communication networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11003–11017, Nov. 2019.

[32] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a Kuhn-Munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 3, pp. 739–750, May 2012.

[33] D. Xu, M. Kent, L. Thomas, T. Mouelhi, and Y. Le Traon, "Automated model-based testing of role-based access control using predicate/transition nets," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2490–2505, Sep. 2015.
[34] Z. Li, I. V. Kolmanovsky, E. M. Atkins, J. Lu, D. P. Filev, and Y. Bai, "Road disturbance estimation and cloud-aided comfort-based route planning," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3879–3891, Nov. 2017.
[35] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9585–9596, Dec. 2016.
[36] M. Mavrovouniotis, F. M. Müller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1743–1756, Jul. 2017.
[37] S. Mitchell, M. O'Sullivan, and I. Dunning, *PuLP: A Linear Programming Toolkit for Python*, Dept. Eng. Sci., Univ. Auckland, Auckland, New Zealand, 2011.
[38] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, nos. 15–16, pp. 1–35, 2006.
[39] D. Bao, L. R. Bryant, S.-S. Chern, and Z. Shen, "A sampler of Riemann–Finsler geometry," in *Mathematical Sciences Research Institute 50*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[40] L. A. Santalo, M. Kac, L. A. Santalo, and M. Kac, "Integral geometry and geometric probability," in *Integral Geometry and Geometric Probability*. Cambridge, U.K.: Cambridge Univ. Press, 2010, pp. 19–20.
[41] Y. H. Qu and F. Liang, "A method of wind field estimation and trajectory correction online in DR/GPS/RP integrated navigation of UAV," in *Proc. IEEE Int. Conf. Autom. Logist.*, Qing'dao, China, 2008, pp. 1402–1407.
[42] J. Yi, L. Zhang, J. Deng, R. Shu, and J. Wang, "GPS/SINS/BARO integrated navigation system for UAV," in *Proc. Int. Forum Inf. Technol. Appl.*, Kun'ming, China 2010, pp. 19–25.
[43] D. Weibel, D. Lawrence, and S. Palo, "Small unmanned aerial system attitude estimation for flight in wind," *J. Guid. Control Dyn.*, vol. 38, no. 7, pp. 1300–1305, 2015.
[44] D. A. John, *Aircraft Performance and Design*. Boston, MA, USA: WCB/McGraw-Hill, 1999.
[45] P. H. Chung, D. M. Ma, and J. K. Shiau, "Design, manufacturing, and flight testing of an experimental flying wing UAV," *Appl. Sci.*, vol. 9, no. 15, p. 3043, 2019.

Qian Jiang (Graduate Student Member, IEEE) was born in January 1997. He received the B.S. degree in network engineering and the M.S. degree in computer science and technology from the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree in intelligent science and system with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, China.

He has published a paper on *International Journal of Cooperative Information Systems* in 2020 and currently he has also published a paper on IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS. His research interests include human–machine systems, computational social simulation, and scheduling and Optimization.

Mr. Jiang has reported a conference paper as the first author in the Chinese CSCW 2019. He has served as a Reviewer for IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS.

Haibin Zhu (Senior Member, IEEE) received the B.S. degree in computer engineering from the Institute of Engineering and Technology, Zhengzhou, China, in 1983, and the M.S. and Ph.D. degrees in computer science from the National University of Defense Technology (NUDT), Changsha, China, in 1988 and 1997, respectively.

He is a Full Professor and the Coordinator of the Computer Science Program, the Founding Director of the Collaborative Systems Laboratory, Nipissing University, North Bay, ON, Canada. He was a Visiting Professor and a Special Lecturer with the College of Computing Sciences, New Jersey Institute of Technology, Newark, NJ, USA, from 1999 to 2002. He is a Lecturer, an Associate Professor, and a Full Professor with NUDT from 1988 to 2000. He has accomplished (published or in press) over 200 research works, including 30+ IEEE TRANSACTIONS articles, six books, five book chapters, three journal issues, and three conference proceedings. He has offered over 70 invited talks, including keynote and plenary speeches on related topics internationally, e.g., Australia, Canada, China, Germany, Hong Kong, Macau, Singapore, Turkey, U.K., and USA. His research has been being sponsored by NSERC, IBM, DNDC, DRDC, and OPIC. His research interests include collaboration theory, technologies, systems, and applications, human–machine systems, computational social simulation, collective intelligence, multiagent systems, software engineering, and distributed intelligent systems.

Prof. Zhu was a receipt of the Meritorious Service Award from IEEE SMC Society in 2018, the Chancellor's Award for Excellence in Research in 2011 and two Research Achievement Awards from Nipissing University in 2006 and 2012, the IBM Eclipse Innovation Grant Awards in 2004 and 2005, the Best Paper Award from the 11th ISPE International Conference on Concurrent Engineering (ISPE/CE2004), the Educator's Fellowship of OOPSLA'03, a 2nd Class National Award for Education Achievement in 1997, and three 1st Class Ministerial Research Achievement Awards from China in 1997, 1994, and 1991. He is serving as an Associate Vice President, Systems Science and Engineering (SSE), the Co-Chair of the Technical Committee of Distributed Intelligent Systems, a member of the SSE Technical Activity Committee, the Conferences and Meetings Committee, and the Electronic Communications Subcommittee of IEEE Systems, Man and Cybernetics (SMC) Society, an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, *IEEE SMC Magazine*, and *IEEE Canada Review*. He has been an Active Organizer for the Annual IEEE International Conference on SMC since 2003, as the Registration Co-Chair in 2021, the Co-Chair in 2020, the Poster Co-Chair in 2020, the Special Session Chair in 2019, the Tutorial Chair in 2018, the Area Co-Chair in 2017, the Social Media Co-Chair in 2015, the Web Co-Chair in 2015, the Session Chair in 2003–2020, and a Special Session Organizer in 2003–2020. He was the Program Co-Chair for the 13th International Conference on Computer Science and Information Technology, October 14–16, 2020, Online (ICCSIT2020) and the 10th International Conference on Pervasive and Parallel Computing, Communication, and Sensors, November 3–5, 2020, Online (PECCS2020), the Publication Chair for the 1st IEEE International Conference of Human–Machine Systems, September 7–9, 2020 (online), and the Program Chair for 16th IEEE International Conference on Networking, Sensing and Control, Banff, AB, Canada, May 8–11, 2019. He is the PC Chair for 24th IEEE International Conference on Computer Supported Cooperative Work in Design, Dalian, China, May 6–8, 2020, and was the PC Chair for 17th IEEE International Conference on Computer Supported Cooperative Work in Design, Whistler, BC, Canada, June 27–29, 2013. He also served as a PC member for 100+ academic conferences. He is a Fellow of the Institute of Cognitive Informatics and Cognitive Computing, a Senior Member of ACM, a Full Member of Sigma Xi, and a Life Member of the Chinese Association of Science and Technology, USA.

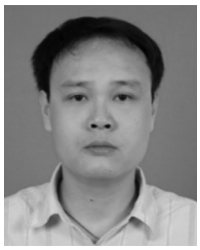
Yan Qiao (Senior Member, IEEE) received the B.S. and Ph.D. degrees in industrial engineering and mechanical engineering from the Guangdong University of Technology, Guangzhou, China, in 2009 and 2015, respectively.

From September 2014 to September 2015, he was a visiting student with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. From January 2016 to December 2017, he was a Postdoctoral Research Associate with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, China, where he has been an Assistant Professor since January 2018. He has over 80 publications, including one book chapter and 30+ regular papers in IEEE TRANSACTIONS. His research interests include scheduling and optimization, semiconductor manufacturing systems, and smart manufacturing.

Dr. Qiao was a recipient of the QSI Best Application Paper Award Finalist of 2011 IEEE International Conference on Automation Science and Engineering, the Best Student Paper Award of 2012 IEEE International Conference on Networking, Sensing and Control, the Best Conference Paper Award Finalist of 2016 IEEE International Conference on Automation Science and Engineering, the Best Student Paper Award Finalist of 2020 IEEE International Conference on Automation Science and Engineering, and the 2021 Hsue-Shen Tsien Paper Award of IEEE/CAA JOURNAL OF AUTOMATICA SINICA. He has served as a reviewer for a number of journals.

Zhiwei He received the B.S. degree in automation from the School of Automation, Guangdong University of Technology, Guangzhou, China, in 2018, and the M.S. degree in signal and information processing from the School of Electronics and Information, South China University of Technology, Guangzhou, in 2021.

He is currently working as an Operation Support System Engineer with China Telecom Corporation Ltd. Guangdong Telecom Company, Guangzhou. He has published one workshop paper in 23rd ACM International Conference on Multimodal Interaction. His research interests include sentiment analysis and network traffic analysis.



Dongning Liu (Member, IEEE) was born in January 1979. He received the Ph.D. degree in logic from Sun Yat-sen University, Guangzhou, China, in 2007.

He is a Full Professor with the Guangdong University of Technology, Guangzhou, since 2009. He is responsible for teaching artificial intelligent logic and discrete math in the School of Computer Science and Technology. He is engaged in education and technology transfer on collaborative computing. He is the Vice Dean of the School of Computer Science and Technology, Guangdong University of Technology. He was a Postdoctoral Fellow of Mathematics with Sun Yat-sen University from 2007 to 2009. He was a Visiting Professor with Nipissing University, North Bay, North Bay, ON, Canada, from 2015 to 2016. He has published more than 50 papers on computer magazines and international conferences.

Prof. Liu is a Reviewer of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is a member of IEEE SMC Society and serving as a TC Member of the technical committee of Distributed Intelligent Systems, a Senior Member of CCF Society, and serving as a Vice Secretary-General and a Standing Committee Member of Technical Committee on Cooperative Computing of China Computer Federation.

Baoying Huang (Graduate Student Member, IEEE) received the B.S. degree in network engineering and the M.S. degree in computer science and technology from the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China, in 2016 and 2019, respectively. She is currently pursuing the Ph.D. degree in intelligent science and system with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, China.

She has published two regular papers in IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, respectively. Her research interests include scheduling and optimization, human–machine systems, computational social simulation, and semiconductor manufacturing systems.

Ms. Huang has served as a reviewer for a number of journals and conferences.