Knowledge-Based Systems 260 (2023) 110164

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Cooperative path planning optimization for multiple UAVs with communication constraints

Liang Xu [a,b], Xianbin Cao [a], Wenbo Du [a], Yumeng Li [a,*]

[a] School of Electronic and Information Engineering, Beihang University, Beijing 100191, PR China
[b] Shen Yuan Honors College, Beihang University, Beijing 100191, PR China

ARTICLE INFO

*Article history:*
Received 27 July 2022
Received in revised form 8 November 2022
Accepted 26 November 2022
Available online 30 November 2022

*Keywords:*
Multiple UAVs
Cooperative path planning
Communication constraints
Particle swarm optimization

ABSTRACT

Path planning is a complicated optimization problem that is crucial for the safe flight of unmanned aerial vehicles (UAVs). Especially in the scenarios involving multiple UAVs, this problem is highly challenging due to the constraints of complex environments, various tasks and inherent UAV maneuverability. In this paper, a cooperative path planning model for multiple UAVs is presented. In addition to common limitations such as the path length minimization, UAV maneuverability limitation and collision avoidance, the communication requirements between UAVs and the impact of obstacles in the flight environment on the quality of communication are also taken into account in the presented path planning model. On this basis, the corresponding objective function is designed. Then, an improved particle swarm optimization (PSO) algorithm is proposed to solve the above path planning problem. Utilizing the ideas of the dynamic multi-swarm PSO (DMSPSO) algorithm and the comprehensive learning PSO (CLPSO) algorithm, the proposed algorithm, denoted as CL-DMSPSO, further improves the performance of both algorithms. The effectiveness and superiority of the novel CL-DMSPSO algorithm is verified on benchmark functions, especially for complex multimodal functions. Finally, we present an effective path planning method using CL-DMSPSO to generate optimized flyable paths for multiple UAVs. And simulation and comparison results on the designed scenario indicate the proposed UAV path planning method can efficiently plan high-quality paths for UAVs and demonstrate the advantages of the proposed CL-DMSPSO algorithm compared with other PSO algorithms in UAV path planning.


## 1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been widely used in civil aviation and military affairs, including express transportation, search and rescue missions, and target surveillance [1–4]. Compared with a single UAV, the UAV group has the unique capability to carry out more complicated tasks, and shows significant advantages of lower cost and better robustness [5,6].

Path planning is one of the most important techniques for the successful flight of multiple UAVs. It aims to find the optimal path for each UAV to fly from the starting point to the desired destination under some specified constraints. Various requirements for UAVs have been presented to obtain suitable paths in different missions. For example, the minimization of the path length is a common criterion to save the cost, and the inherent maneuverability of a UAV has to be considered, such as the restrictions on the turning angle, the climbing angle, etc. In addition, the safety of each UAV is supposed to be guaranteed during the whole flight [7,8]. Specifically, multiple UAVs are often applied for pretty complex missions. Sometimes they have to fly over areas with the existence of many obstacles, where they are exposed to a high risk of collisions. In this case, each UAV is expected to pass the desired waypoints without collisions with obstacles or other UAVs [9–12]. In particular, the requirements for communication between UAVs are necessary, so that data and information can be transmitted and shared quickly, which is significant to the efficient and collaborative completion of tasks for multiple UAVs [13,14]. Moreover, in many real-world path planning problems, the existence of obstacles may affect the quality of communication between UAVs, and thus, this makes communication more difficult. However, these are seldom involved in existing studies. Taking the search and rescue missions for multiple UAVs as an example, suppose one UAV finds the rescue site, and then, others can come to reinforce it as soon as possible owing to the existence of communication connections between UAVs. In addition, the assault mission requests UAVs to take off from the base and arrive at the designated location simultaneously [13]. And multiple UAVs can collaboratively execute the attack task against the specific target with the communication connections between them. Therefore, it is significant to

* Corresponding author.
*E-mail addresses:* xuliang@buaa.edu.cn (L. Xu), xbcao@buaa.edu.cn (X. Cao), wenbodu@buaa.edu.cn (W. Du), liyumeng@buaa.edu.cn (Y. Li).

https://doi.org/10.1016/j.knosys.2022.110164
0950-7051/

explore effective algorithms in solving cooperative path planning problems for multiple UAVs under above considerations.

In the past decades, scholars have performed a large number of valuable studies and proposed various types of path planning methods. Here four types of methods in common use are summarized as follows. First, many mathematical programming methods are introduced to solve path planning problems, including mixed integer linear programming (MILP), nonlinear programming (NP) and dynamic programming (DP) [15–17]. These methods are developed on the basis of the strict mathematical theory, and suitable for solving relatively small-scale path planning problems. However, with the increase of the scale in the planning problem, the computing time of the mathematical programming methods increases exponentially. Second, artificial potential field (APF) method is also a type of commonly used path planning method, which has the advantage of good real-time performance. Unfortunately, in the cases where the gravitation generated by the target and the repulsion caused by obstacles are equal in some potential points, such a method is easily trapped into local optima, and sometimes it even fails to obtain a feasible solution [18–20]. Third, some methods have been proposed based on the graph theory to solve path planning problems, including the random path graph algorithm, Voronoi diagram algorithm and A* algorithm. This kind of algorithms can effectively find the optimal path, while the performance depends on the division of the search space. Therefore, such methods always take a long execution time in path planning when the search space is pretty complex [21–23]. In addition, various computational intelligence algorithms have been introduced in path planning problems; for instance, the genetic algorithm (GA), the ant colony optimization (ACO) algorithm, the particle swarm optimization (PSO) algorithm, and differential evolution (DE) algorithm, etc. [24–27]. Such algorithms can plan flyable paths for UAVs that meet the specified requirements in a limited computing time. Owing to the advantages of simplicity, low computational cost and great performance, computational intelligence algorithms have been widely applied to complex path planning problems in recent years [28–32].

As a typical computational intelligence algorithm, PSO shows great performance in various optimization problems due to its population-based characteristics, easy implementation and rapid convergence. However, the particles are easily trapped into local optima in standard PSO (FPSO). Thus, in order to overcome this deficiency, researchers have put forward numerous PSO variants from the perspectives of the selection of parameters, population structures, learning strategies, the hybridization with other algorithms, etc. In recent years, PSO algorithms have been widely utilized in the UAV path planning and other optimization tasks, demonstrating satisfactory performance. For example, with the introduction of a novel neighborhood structure, the authors of Ref. [33] propose a dynamic multi-swarm PSO (DMSPSO) algorithm and it is applied to the UAV path planning problems and demonstrates great advantages [34,35]. Moreover, Ref. [36] proposes a comprehensive learning particle swarm optimization (CLPSO) algorithm, utilizing a new learning strategy. The test functions and the experimental results in path planning scenarios prove the effectiveness of CLPSO [37,38]. Ref. [27] presents a hybrid differential evolution (DE) with quantum-behaved particle swarm optimization (QPSO), denoted as DEQPSO, which aims to further improve the performance of these two algorithms, and on this basis, a new route planning method for UAV on the sea is developed. More recently, Ref. [39] introduces the comprehensively improved particle swarm optimization (CIPSO) based on FPSO, involving several improvements of the parameters selection, learning strategy and initial particle distribution, and applies it to the path planning for UAV formation to enhance the convergence speed and the solution optimality. In addition to path planning, PSO is often used in other optimization tasks. For example, to solve the UAV communication coverage problem, the heterogeneous network PSO (NHPSO) algorithm is introduced in Ref. [40]. Furthermore, Refs. [41,42] present a discrete PSO for complex network clustering, the experimental results demonstrate that the proposed method is effective and promising compared with several state-of-the-art network clustering methods. Ref. [43] uses the discrete PSO for identifying community structures in signed social networks. Experiments on both synthetic and real-world signed networks exhibit a good performance. To solve the UAV vision-based surface inspection problem, the authors of Ref. [44] introduce an enhanced discrete PSO and verify the validity and effectiveness of the proposed technique in successful experiments.

Motivated by the above considerations, our work focuses on the cooperative path planning problem for multiple UAVs with quite a few constraints. In particular, communication constraints between UAVs are considered as an important factor, which largely benefits the coordination and cooperation of all UAVs to complete missions efficiently. And the possible impact of obstacles on the quality of communication is also taken into account, which is common and significant in real world problems. On this basis, the corresponding objective function is designed to find the feasible path with the shortest length for each UAV. Then, inspired by the DMSPSO and CLPSO algorithms, we propose a novel CL-DMSPSO algorithm to solve the above path planning problem, further enhancing the performance of both two algorithms. And we conduct experiments on given benchmark functions to verify the effectiveness of our proposed CL-DMSPSO. Finally, we introduce the detailed implementation of the presented UAV path planning method using the CL-DMSPSO algorithm. And simulations and comparisons are carried out on the real scenario of path planning for multiple UAVs to explore the performance of our proposed method.

The remaining part of this paper is organized as follows. First, Section 2 presents a cooperative path planning model for multiple UAVs with necessary constraints, and then the corresponding objective function is designed. In Section 3, after a brief review or several existing PSO algorithms, we develop a novel CL-DMSPSO algorithm and verify its effectiveness on benchmark functions, and on this basis, an effective path planning method using CL-DMSPSO is proposed to solve the presented model. Next, Section 4 includes simulation and comparison results on the designed scenario of the cooperative path planning for multiple UAVs using the CL-DMSPSO algorithm and other PSO algorithms. Finally, our work in the paper is concluded in Section 5.

## 2. UAV path planning model

In this section, we first depict a cooperative path planning problem for multiple UAVs with several necessary constraints in Section 2.1. Then, we introduce the detailed path representation in Section 2.2. On this basis, we design the objective function of the described path planning problem in Section 2.3.

### 2.1. Problem description

The cooperative path planning problem plays a significant role in determining the flyable paths for multiple UAVs in a variety of complex missions. It is a well-known optimization problem, based on the criterion of minimizing the path length and quite a few necessary constraints. For example, suppose there is a path planning problem involving $M$ UAVs. Specifically, these UAVs are required to fly from different starting points to the same destination in the specified flight space, where a number of obstacles exist and are known in prior. The path planning

problem aims to find the flyable path with the shortest length for each UAV to minimize the path cost, with several significant factors taken into account. First, the generated path must be sufficiently smooth, subject to the maneuverability of a UAV. Second, each UAV is supposed to avoid collisions with known obstacles and teammates to guarantee the safety. In addition to the above general requirements studied in most of existing work, the communication constraints are considered as essential factors in this paper. Specifically, all UAVs are required to maintain communication connections during the whole flight, that is, any two UAVs can communicate through at least one route at any time [14,45]. This benefits the efficient and collaborative completion of tasks a lot for multiple UAVs, since information can be transmitted and shared quickly owing to the existence of communication connections between UAVs. Furthermore, Given the fact that the blocking of obstacles makes the communication between two UAVs more difficult, the impact of obstacles on the quality of communication must be considered in the communication constraints.

Overall, our work concentrates on the cooperative path planning problem for multiple UAVs which aims to obtain the shortest path for each UAV with the following constraints satisfied:

(1) The turning angle at any waypoint should be smaller than a given threshold value during flight.

(2) Each UAV is required to avoid collisions with obstacles during flight.

(3) Each UAV needs to avoid conflicts with other UAVs during flight.

(4) Any two UAVs must maintain the communication connection during flight.

### 2.2. Path representation

Consider the presented path planning problem in Section 2.1 involving $M$ UAVs. We assume that all UAVs keep a constant altitude during flight and any a path of each UAV consists of $N$ waypoints, including the starting point and the destination. Suppose $(x_{ij}, y_{ij})$ represents the coordinate of the $j$th waypoint in a path of the $i$th UAV, $i = 1, 2, \ldots, M$; $j = 1, 2, \ldots, N$. Thus, a complete path for UAV $i$ can be expressed as $(x_{i1}, y_{i1}, x_{i2}, y_{i2}, \ldots, x_{iN}, y_{iN})$.

In general, the requirements on a flyable UAV path can be described in the geometric form and divided into two types. For the first type of requirements, only the information of waypoints in the path are needed to calculate the cost. That is, these requirements can be evaluated utilizing the coordinates of the waypoints and the geometrical relationship between them; for instance, the minimization of path length and the maximum limitation of turning angle. As concerns the other type of constraints, the waypoints are not sufficient to the provide the comprehensive characterization of the UAV path. As a result, the segments of the path may be not flyable though the corresponding waypoints are located in suitable positions. Such constraints include the minimal risk of collisions with obstacles and other UAVs, and the existence of communication connections between UAVs. In this work, for the second type of requirements, following Ref. [24], we divide each segment into shorter piecewise parts and then calculate the cost of each dividing point, as the approximation of the behavior for each segment of the path.

### 2.3. Objective function

As discussed in Section 2.1, our objective is to find the shortest feasible path for each UAV in the cooperative path planning problem for multiple UAVs. Here, a feasible path refers to a flight path with the four constraints satisfied for a UAV in the given flight space, including the turning angle limitation, the obstacle-collision avoidance, the separation maintenance and the communication maintenance. To solve this complicated constrained optimization problem, the penalty function method is applied in our work to obtain an unconstrained optimization problem, and the corresponding overall objective function can be constructed as follows.

(1) Cost of path length

First, using the notation in Section 2.2, the cost function associated with the path length can be expressed as follows [2]:

$$f_L = 1 - \frac{\sqrt{(x_{iN} - x_{i1})^2 + (y_{iN} - y_{i1})^2}}{\Sigma_{j=2}^{N} \sqrt{\left(x_{ij} - x_{ij-1}\right)^2 + \left(y_{ij} - y_{ij-1}\right)^2}} \tag{1}$$

where the numerator represents the length of the straight line connecting the starting point and the destination for UAV $i$, and the denominator is the actual path length. Obviously, the value of $f_L$ ranges from 0 to 1, and a smaller value of $f_L$ means less cost of the corresponding actual path length.

Thus, the aim of the presented cooperative path planning problem is to minimize the cost function in Eq. (1) under the four constraints above. Using the penalty function method, this constrained optimization problem can be transformed into an unconstrained optimization problem, while the following criteria are taken into consideration to construct the corresponding objective function. Firstly, the objective function value (OFV) for any feasible path is smaller than that for any infeasible path for each UAV. Secondly, for two feasible paths, the one with less cost of path length has smaller OFV. Thirdly, for two infeasible paths, the less the constraint violations, the smaller the OFV. From the three criteria above, we can conclude that the designed objective function depends on not only the cost of the path length, but also the number of constraint violations among the four constraints for a UAV along this path, which can be constructed by adding the number of constraint violations for a UAV along this path to the cost of this path length. Motivated by the discussions above, for each of the four constraints, a corresponding penalty function is introduced. More specifically, the penalty function value is set to one if this constraint is violated, otherwise it is set to zero. Then, the overall objective function is constructed as the linear combination of the cost function in Eq. (1) and the four introduced penalty functions. Using the path representation shown in Section 2.2, the four penalty functions and the overall objective function can be expressed in the following.

(2) Turning angle limitation

Considering the limitation of UAV maneuverability, the planned path is supposed to remain smooth. Specifically, the turning angle of each UAV at any waypoint should be relatively small to guarantee the smoothness. Here, the turning angle is defined as the angle between the previous direction of the UAV and the current direction in the horizontal direction. Thus, we define the penalty function with respect to the turning angle for UAV $i$ as follows [9]:

$$f_T = \begin{cases} 1, & \exists\, \theta_{ij} > \theta_{max} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\theta_{ij} = \arccos\left(\frac{(x_{ij} - x_{ij-1}, y_{ij} - y_{ij-1}) \cdot (x_{ij+1} - x_{ij}, y_{ij+1} - y_{ij})^T}{\left\|(x_{ij} - x_{ij-1}, y_{ij} - y_{ij-1}) \cdot (x_{ij+1} - x_{ij}, y_{ij+1} - y_{ij})\right\|}\right) \tag{3}$$

where $\theta_{max}$ is the maximal turning angle limitation, and $\theta_{ij}$ represents the turning angle at the $j$th waypoint in the path of the $i$th UAV, $j = 2, 3, \ldots, N-1$.

(3) Obstacle-collision avoidance

Each UAV should avoid collisions with obstacles in the specified environment to guarantee the safety. In many applications,

the larger sizes of obstacles generally reduce the feasible space. In this case, a relatively short path generally implies that the UAV has to take higher obstacle-collision risk along this path, while a relatively safe path may need a longer length to avoid collisions with obstacles. Assume that the positions and sizes of all obstacles are known in advance. We divide each path of one UAV into $N_r - 1$ piecewise parts with equal length and then evaluate the risk of $N_r$ dividing points, called risk evaluation points, with the starting point and destination included. The number $N_r$ is usually determined by the detailed situation, which characterizes the tradeoff between the computing cost and the accuracy of planned paths. Suppose $(x_{ik}, y_{ik})$ denotes the coordinate of the $k$th obstacle-collision risk evaluation point of the $i$th UAV, $i = 1, 2, \ldots, M; k = 1, 2, \ldots, N_r$, and it can be calculated from the coordinates of the $N$ waypoints. To ensure obstacle-collision avoidance, those paths with some segments located inside the obstacles should be penalized in the overall objective function. Specifically, if one or more risk evaluation points exist inside some obstacles, the corresponding path is obviously not safe, where the obstacle-collision avoidance constraint is violated, and thus, the associated penalty function $f_R$ is set to one. Otherwise, the generated flight path is safe and $f_R$ is equal to zero. That is, the penalty function concerning the obstacle-collision avoidance constraint for UAV $i$ can be presented as follows [2]:

$$f_R = \begin{cases} 1, & \exists\ (x_{ik}, y_{ik}) \text{ in an obstacle} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

(4) Separation maintenance

When planning paths for multiple UAVs, it is significant to maintain the separation between UAVs for safety. In this paper, we assume that all UAV arrive at next waypoints at the same time. Suppose each UAV has a large speed range, and the flight speed keeps constant between two adjacent waypoints. Each segment between the $j$th waypoint and the $(j+1)$th waypoint in the path for each UAV is evenly divided into piecewise parts. Then $N_s$ dividing points are obtained for this path, with the starting point and destination included, and are called separation evaluation points. The coordinates of the $u$th separation evaluation point of the $i$th UAV are denoted as $(x_{iu}, y_{iu})$, where $i = 1, 2, \ldots, M; u = 1, 2, \ldots, N_s$, which can be calculated from the coordinates of the $N$ waypoints.

Suppose $d_{min}$ represents the minimal distance between two UAVs permitted for safety to avoid the conflict risk. Given the paths of UAV $i$ and UAV $v$, where $v = 1, 2, \ldots, M, v \neq i$, if the distance between the $u$th separation evaluation points in the paths of the $i$th UAV and the $v$th UAV, denoted as $d_u^{iv}$, is smaller than the value of $d_{min}$, it is considered that the two UAVs lose separation and there exists a conflict risk between them. Therefore, to avoid the conflict risk between UAVs, the corresponding paths should be penalized if any two UAVs are getting too close during flight. Thus, the penalty function associated with the separation maintenance constraint for UAV $i$ is defined as follows [2,9]:

$$f_S = \begin{cases} 1, & \exists d_u^{iv} < d_{min} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

(5) Communication maintenance

The UAVs need to maintain communication connections during flight, that is, there is at least one route between any two UAVs at any time during flight [45]. In the flight environment, the line-of-sight (LoS) link between UAVs may be occasionally blocked by obstacles, which influences the quality of communication. Therefore, we need to distinguish the different propagation environment between LoS and non-line-of-sight (NLoS) scenarios. The large-scale channel coefficient $\beta(d)$ is modeled as follows [46]:

$$\beta(d) = \begin{cases} \beta_0 d^{-\alpha}, & \text{LoS environment} \\ k\beta_0 d^{-\alpha}, & \text{NLoS environment} \end{cases} \tag{6}$$

$$\beta_0 = \left(\frac{c}{4\pi f_c}\right)^2 \tag{7}$$

where $\beta_0$ is the path loss at the reference distance of 1 m under LoS conditions and $k < 1$ is the additional attenuation factor due to the NLoS propagation. $c$ represents the light speed, and $f_c$ is the carrier frequency. We judge whether there is LoS propagation according to whether there are obstacles between two UAVs, and then determine the channel coefficient between them. Then we can calculate the signal-to-noise ratio $SNR$ as follows:

$$SNR = \beta(d)\, 10^{\frac{P_t - P_n}{10}} \tag{8}$$

where $P_t$ denotes the UAV transmission power and $P_n$ represents the noise power. If the value of $SNR$ is greater than a threshold value $SNR_t$, it is considered that there exists a link between the two UAVs. In our model, if there is a link between UAV 1 and UAV 2, and one between UAV 2 and UAV 3, then UAV 1 and UAV 3 can communicate through the relay function of UAV 2. Consequently, to guarantee communication connection between each UAV and any other UAV, the path where there exist two UAVs losing communication connections at some point should be penalized. Specifically, the penalty function associated with the communication maintenance constraint for UAV $i$ is equal to zero if there is a communication connection between UAV $i$ and any other UAV at any time during flight, otherwise it is equal to one, which can be expressed as:

$$f_C = \begin{cases} 0, & \exists \text{ communication connection} \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

(6) Overall objective function

As discussed above, by means of the penalty function method, solving the presented cooperative path planning problem, which can be regarded as a constrained optimization problem, is equivalent to solving an unconstrained optimization problem. And the overall objective function $F_{obj}$ for a candidate path for UAV $i$ can be formulated as a linear combination of the cost function in Eq. (1) and the four penalty functions in Eqs. (2) (4) (5) (9). That is,

$$F_{obj} = f_L + f_T + f_R + f_S + f_C \tag{10}$$

From Eq. (10), we can observe that, for a feasible path where the four constraints are satisfied, each penalty function is equal to 0, and thus, its objective function value is equal to its cost function value and ranges from 0 to 1. However, for an infeasible path, the OFV depends on both the cost of the path length and the number of the constraint violations for the UAV along this path, and consequently, it is larger than 1. From these discussions, we can conclude that the presented three criteria above are satisfied. Overall, to find the shortest feasible path for each UAV under the four constraints in the presented cooperative path planning problem, we aim to minimize the overall objective function shown in Eq. (10) without any constraints. In the next section, an improved PSO algorithm is introduced to solve the unconstrained optimization problem above.

## 3. UAV path planning method

In this part, we first review the standard PSO and its variants including DMSPSO, CLPSO, DEQPSO, CIPSO and NHPSO in Sections 3.1 and 3.2. Then we introduce a new CL-DMSPSO algorithm in Section 3.3 and then verify its effectiveness on benchmark functions in Section 3.4. Finally, the detailed implementation of the path planning method using CL-DMSPSO is proposed in Section 3.5.

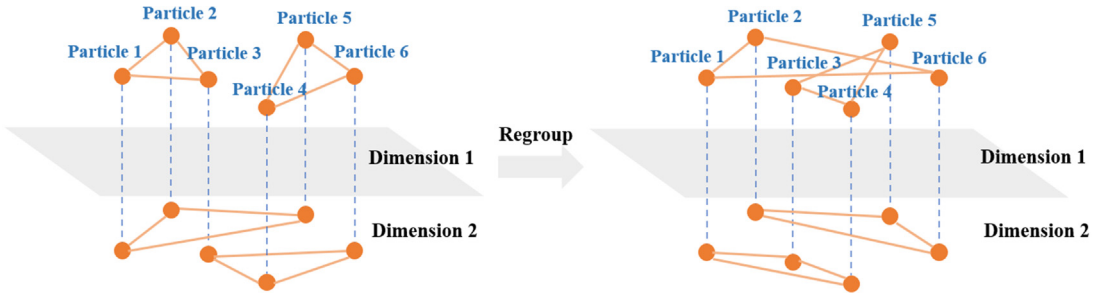



**Fig. 1.** CL-DMSPSO regrouping process.

### 3.1. The standard PSO algorithm

Particle swarm optimization is a stochastic optimization algorithm based on population characteristics [47,48]. In PSO framework, each particle $l$ in the search space of $D$ dimensions is characterized by its position, $\mathbf{x}_l = (x_l^1, x_l^2, \ldots, x_l^D)$, and velocity, $\mathbf{v}_l = (v_l^1, v_l^2, \ldots, v_l^D)$. The particle searches for the optimal solution by utilizing its own experience reflected via the personal best position, $\mathbf{P}_l = [pbest_l^1, pbest_l^2, \ldots, pbest_l^D]$, and the swarm experience reflected via the global best position, $\mathbf{G} = [gbest^1, gbest^2, \ldots, gbest^D]$. For a swarm consisting of $N_p$ particles, the velocities and positions are updated by the following equations:

$$v_l^d \leftarrow w * v_l^d + c_1 * rand1_l^d * \left(pbest_l^d - x_l^d\right) + c_2 * rand2_l^d * \left(gbest^d - x_l^d\right) \tag{11}$$

$$x_l^d \leftarrow x_l^d + v_l^d \tag{12}$$

where $w$ is the inertia weight coefficient, that is used to balance the global and local search abilities. A relatively large inertia weight coefficient enhances the global search ability, while a relatively small one facilitates local search. $c_1$ and $c_2$ are acceleration coefficients, and $rand1_l^d$ and $rand2_l^d$ are two independent random numbers in the interval $[0, 1]$. During the whole optimization process, each particle has a dynamic fitness value corresponding to its current position. The $\mathbf{p}_l$ of particle $l$ and $\mathbf{G}$ of the whole population are also updated during each iteration if a better fitness value is available. The algorithm stops until a certain number of iterations is reached or a certain goal for the fitness value is met.

### 3.2. Some improved PSO algorithms

To overcome the deficiency that the standard PSO is easily trapped into local optima, various improved PSO algorithms have been proposed considering improvements with respect to parameter selection, population structure, learning strategy, hybridization with other algorithms, etc. Examples are as follows.

DMSPSO utilizes a new population structure. The whole population is divided into some small sized sub-swarms. All particles search for better positions in the search space using the information of corresponding sub-swarms. In addition, by means of a regrouping schedule, the sub-swarms are regrouped frequently so that the particles can learn information from different neighbors. Owing to the learning of particles in sub-swarms and the regrouping between sub-swarms, DMSPSO algorithm has the strong ability of global search. For a more detailed introduction, see Ref. [33].

In addition, CLPSO is one of the most promising variants. In CLPSO, each particle has a certain probability to learn from different neighbors in different dimensions, rather than selecting the same best neighbor for all the dimensions. Therefore, the whole population can maintain the diversity to avoid falling into local optimization. For more details, refer to Ref. [36].

Based on DE and QPSO, the authors of Ref. [27] propose a DEQPSO algorithm to further strength the global search ability of the two algorithms. The proposed DEQPSO, combining the ideas of the DE and the QPSO, includes two calculation steps. In each iteration, first, all particles update their velocities and positions by the QPSO algorithm, and then the DE algorithm is implemented.

To search for the global optimal solution more quickly and accurately, Ref. [39] proposes CIPSO, which comprehensively improves the standard PSO from several aspects, including using chaos-based particle initialization strategy, adaptive parameter adjustment strategy, maximum velocity adjustment strategy and a new position updating strategy.

In Ref. [40], the authors introduce a heterogeneity population structure into CLPSO, proposing NHPSO. In NHPSO, the scale-free network model is used as the population structure of particle information interaction, to improve the efficiency of information transmission. Combining the heterogeneity of population structure, the authors design heterogeneous particle behavior, so that the central particles can learn from other particles more frequently, and non-central particles can learn from themselves more often. The NHPSO has a good performance through the division and cooperation of different particles in the population.

### 3.3. The proposed CL-DMSPSO algorithm

In view of the advantages of the DMSPSO and CLPSO algorithms, we propose a new algorithm that combines the ideas of both algorithms, called CL-DMSPSO, to solve the cooperative path planning problem for multiple UAVs. We assume that particles can learn from different neighbors in different dimensions on the basis of the DMSPSO algorithm.

In our CL-DMSPSO algorithm, the particles are divided into some small sized sub-swarms in each dimension. To better illustrate the proposed algorithm, a detailed description is presented as follows. Take the population size $N_p = 6$ and dimension $D_p = 2$ as an example. First, six particles are divided into two sub-swarms in each dimension. Then, each dimension of each particle works within the sub-swarm where it stays. The velocity and position of the particle $l$ is updated using Eqs. (13) and (12).

$$v_l^d \leftarrow w * v_l^d + c_1 * rand1_l^d * \left(pbest_l^d - x_l^d\right) + c_2 * rand2_l^d * \left(lbest_l^d - x_l^d\right) \tag{13}$$

where $lbest_l^d$ refers to the best position that the two neighbors of particle $l$ in the $d$th dimension have reached ever before, $\mathbf{L}_l = [lbest_l^1, lbest_l^2, \ldots, lbest_l^D]$. After one regrouping period, the pattern is broken, and the particles are regrouped in each dimension, as shown in Fig. 1. Based on the new sub-swarms, all the particles update their velocities and positions again. The process is repeated until the number of iterations reaches a set value. The last ten percent of all iterations run the standard PSO. The pseudo-code of CL-DMSPSO is given below in Algorithm 1.

**Algorithm 1.** The pseudo-code of CL-DMSPSO.

**Initialize:**

$pz$: population size; $m$: number of particles in a sub-swarm; $R$: regrouping period; $w$: weight coefficient; $c_1$,$c_2$: acceleration coefficients; $iter\_max$: max iterations; Initialize the position $x_l^d$ and velocity $v_l^d$ for the particle $l$ and the population structure;

**Iterate:**

1: **for** $iter$ = 1 to 0.9 * $iter\_max$ **do**

2: Calculate $\mathbf{P}_l$ and $\mathbf{L}_l$ for each particle $l$ and $\mathbf{G}$ for the whole population;

3: Update the position and velocity for each particle using Eqs. (13) and (12);

4: **if** mod($iter$, $R$) = 0 **then**

5: Regroup the sub-swarms in each dimension randomly;

6: **end if**

7: **end for**

8: **for** $iter$ = 0.9 * $iter\_max$ to $iter\_max$ **do**

9: Calculate $\mathbf{P}_l$ for each particle $l$ and $\mathbf{G}$ for the whole population;

10: Update the position and velocity for each particle using Eqs. (11) and (12);

11:**end for**

**Table 1**
Specified parameters for different PSOs.

| Algorithm | Main Parameters |
|---|---|
| FPSO | $w = 0.729$, $c_1 = c_2 = 1.494$ |
| DMSPSO | $w_0 = 0.9$, $w_1 = 0.4$, $c_1 = c_2 = 1.494$, $m = 3$, $R = 5$ |
| CLPSO | $w_0 = 0.9$, $w_1 = 0.4$, $c = 1.494$, $n = 7$ |
| DEQPSO | $c = 1.494$, $\varphi_0 = 1.0$, $\varphi_1 = 0.5$, $CR = 0.85$ |
| CIPSO | $w_0 = 0.9$, $w_1 = 0.4$, $c_1 = 3.5$, $c_2 = 0.5$, $V_1 = 0.5$, $V_2 = 0.1$, $u = 4$, $a = 2$ |
| NHPSO | $w_0 = 0.9$, $w_1 = 0.4$, $c = 1.494$, $n = 7$, $m_0 = 4$, $I = 4$ |
| CL-DMSPSO | $w_0 = 0.9$, $w_1 = 0.4$, $c_1 = c_2 = 1.494$, $m = 3$, $R = 5$ |

### 3.4. Experimental results on benchmark functions

To verify the effectiveness of the proposed CL-DMSPSO, 20 benchmark functions are selected in this paper, including 5 unimodal functions (f1–f5), 8 multimodal functions (f6–f13) and 7 rotated multimodal functions (f14–f20) [36,49–51], as shown in Table A.1 in the Appendix. More details about these test functions are available in the Appendix. For each benchmark function, 50 Monte Carlo simulation experiments are repeated.

We choose the standard PSO (FPSO) [47] and several variants introduced in Section 3.2 as competitors, including DMSPSO [33], CLPSO [36], DEQPSO [27], CIPSO [39] and NHPSO [40], and compare the performance of different algorithms.

Referring to the experimental parameters set in other literatures, the parameters in the subsequent experiments in this subsection are determined as follows. The dimension of each benchmark function is 30, and the population size is 50. And for each algorithm, the final objective function value is obtained after 5000 iterations. Furthermore, the main parameters for different PSOs are specified as shown in Table 1.

Table 2 presents the results regarding the solution quality for each algorithm, which is calculated by averaging the final objective function values of 50 repetitions, and the value below represents the corresponding standard deviation. As all selected benchmark functions are minimization problems, smaller values reflect better performance. Here the best result of each function is marked in bold. We can observe that FPSO performs best on 2 of the 5 unimodal functions due to its high convergence speed. Our CL-DMSPSO shows the best result on the unimodal function $f_3$, and presents pretty competitive performance for the other four unimodal functions. In addition, the proposed CL-DMSPSO performs best on 9 of the 15 multimodal and rotated multimodal problems, showing a remarkable advantage compared with the other algorithms. All these results can be attributed to the flexibility of learning from neighbors and the random regrouping schedule, which may avoid the algorithm falling into local optima via the effects of high-quality information in different dimensions of different particles. Furthermore, Fig. 2 depicts the number of times that each algorithm performs top-Z among all the 20 benchmark functions [52]. It can be observed that the curve of the proposed CL-DMSPSO is never beneath that of any other PSO algorithm, and it first reaches 20 at $Z = 4$, which further shows its outstanding performance.

To further compare the effectiveness of different PSOs, we calculate and provide the results of success rate for each algorithm on each benchmark function; refer to Table A.3 in the Appendix. Here, the success rate for each algorithm is defined as the percentage of the repetitions where the final objective function value obtained is less than the acceptance value presented in Table A.1 in the 50 repetitions. We can see that our CL-DMSPSO ranks first on 18 functions, presenting great competitiveness among these algorithms. Overall, the proposed CL-DMSPSO performs best with respect to the solution quality and success rate.

### 3.5. Implementation of the UAV path planning method using CL-DMSPSO

In this subsection, the path planning method for multiple UAVs based on the CL-DMSPSO algorithm is introduced. In this framework, CL-DMSPSO serves as the solver for a single UAV. An individual solution in CL-DMSPSO for the $i$th UAV has the form of $\mathbf{x}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, \ldots, x_{iN}, y_{iN})$. Therefore, the dimension

**Table 2**
The performance of different PSOs regarding solution quality on benchmark functions.

| Function | FPSO | DMSPSO | CLPSO | DEQPSO | CIPSO | NHPSO | CL-DMSPSO |
|---|---|---|---|---|---|---|---|
| $f_1$ | **7.2E−103** ±2.2E−102 | 1.09E−61 ±1.82E−61 | 1.92E−14 ±9.37E−15 | 6.06E+02 ±8.75E+02 | 3.97E−13 ±6.09E−13 | 2.61E−23 ±5.03E−23 | 3.32E−71 ±4.72E−71 |
| $f_2$ | 1.41E+01 ±1.69E+01 | 2.11E+01 ±1.67E+00 | 2.49E+01 ±1.73E+00 | 9.23E+01 ±4.98E+01 | **1.40E+01** ±2.04E+01 | 2.21E+01 ±1.74E+00 | 2.11E+01 ±8.80E−01 |
| $f_3$ | 2.00E+00 ±4.00E+00 | 5.00E−36 ±4.15E−36 | 2.33E−09 ±6.19E−10 | 8.84E+00 ±6.34E+00 | 2.96E−10 ±3.28E−10 | 1.12E−15 ±1.39E−15 | **3.98E−42** ±3.87E−42 |
| $f_4$ | 2.89E−03 ±1.31E−03 | 1.50E−03 ±6.31E−04 | 4.97E−03 ±1.08E−03 | 9.57E−03 ±4.21E−03 | 1.59E−02 ±5.05E−03 | **1.49E−03** ±4.74E−04 | 2.11E−03 ±8.02E−04 |
| $f_5$ | **4.0E−165** ±6.8E−166 | 2.29E−92 ±1.57E−91 | 1.18E−28 ±1.10E−28 | 4.88E−02 ±7.05E−02 | 1.03E−19 ±3.34E−19 | 6.24E−36 ±1.49E−35 | 1.70E−114 ±5.00E−114 |
| $f_6$ | 1.68E−14 ±4.22E−14 | 1.25E−15 ±8.32E−16 | 6.88E−04 ±1.86E−04 | 1.58E+00 ±1.37E+00 | 2.22E−09 ±3.41E−09 | 7.93E−12 ±2.52E−11 | **8.78E−16** ±9.15E−16 |
| $f_7$ | 1.02E+00 ±8.47E−01 | 6.70E−15 ±1.52E−15 | 7.87E−08 ±1.87E−08 | 1.11E+01 ±4.81E+00 | 2.87E−07 ±2.71E−07 | 8.88E−13 ±7.09E−13 | **4.85E−15** ±1.52E−15 |
| $f_8$ | 4.09E+03 ±6.68E+02 | 2.22E+03 ±5.17E+02 | **3.82E−04** ±8.45E−08 | 6.12E+03 ±9.97E+02 | 3.25E+03 ±8.75E+02 | 1.18E+01 ±3.55E+01 | 1.70E+03 ±4.33E+02 |
| $f_9$ | 6.54E+01 ±2.20E+01 | 2.84E+01 ±6.85E+00 | 4.82E−05 ±5.35E−05 | 6.62E+01 ±1.85E+01 | 4.22E+01 ±1.46E+01 | **8.93E−08** ±5.01E−07 | 2.43E+01 ±6.92E+00 |
| $f_{10}$ | 3.82E+01 ±1.96E+01 | 1.72E+01 ±5.61E+00 | 1.88E−04 ±1.19E−04 | 7.55E+01 ±2.11E+01 | 3.23E+01 ±1.61E+01 | **8.05E−07** ±2.17E−06 | 1.13E+01 ±4.27E+00 |
| $f_{11}$ | 4.43E+00 ±1.93E+00 | **0** ±0.00E+00 | 4.68E−08 ±1.55E−08 | 1.32E+01 ±3.45E+00 | 3.10E−02 ±6.80E−02 | **0** ±0.00E+00 | **0** ±0.00E+00 |
| $f_{12}$ | 1.43E−01 ±3.44E−01 | 1.24E−02 ±4.47E−02 | 1.65E−15 ±8.56E−16 | 3.28E+02 ±1.14E+03 | 2.90E−02 ±5.50E−02 | 1.05E−24 ±1.65E−24 | **1.57E−32** ±5.47E−48 |
| $f_{13}$ | 1.74E−02 ±5.91E−02 | **1.35E−32** ±1.09E−47 | 2.32E−14 ±1.09E−14 | 4.47E+04 ±9.38E+04 | 4.61E−03 ±8.80E−03 | 1.47E−23 ±2.68E−23 | **1.35E−32** ±1.09E−47 |
| $f_{14}$ | 2.10E+00 ±7.93E−01 | 6.48E−02 ±2.58E−01 | 5.99E−05 ±1.24E−04 | 8.30E+00 ±1.98E+00 | 1.68E+00 ±6.94E−01 | 1.18E−11 ±2.38E−11 | **4.85E−15** ±1.52E−15 |
| $f_{15}$ | 6.26E+03 ±7.06E+02 | 4.62E+03 ±1.50E+03 | 6.14E+03 ±7.94E+02 | 4.41E+03 ±1.06E+03 | 4.34E+03 ±1.53E+03 | 4.46E+03 ±2.11E+03 | **4.30E+03** ±1.38E+03 |
| $f_{16}$ | 7.36E+01 ±2.25E+01 | **3.42E+01** ±9.44E+00 | 9.49E+01 ±1.53E+01 | 7.09E+01 ±1.59E+01 | 5.21E+01 ±1.39E+01 | 7.89E+01 ±1.27E+01 | 4.24E+01 ±1.21E+01 |
| $f_{17}$ | 7.88E+01 ±1.51E+01 | **4.65E+01** ±1.54E+01 | 9.67E+01 ±1.39E+01 | 7.86E+01 ±2.42E+01 | 7.12E+01 ±2.29E+01 | 7.74E+01 ±1.80E+01 | 5.45E+01 ±1.65E+01 |
| $f_{18}$ | 1.03E+01 ±3.48E+00 | 1.60E+00 ±1.04E+00 | 1.16E+01 ±1.50E+00 | 1.79E+01 ±3.09E+00 | 7.29E+00 ±4.02E+00 | **4.80E−01** ±3.09E−01 | 1.22E+00 ±8.86E−01 |
| $f_{19}$ | 4.04E−01 ±7.08E−01 | 2.49E−02 ±4.89E−02 | 3.26E−02 ±4.05E−02 | 4.40E+02 ±1.56E+03 | 1.06E+00 ±1.34E+00 | 2.07E−02 ±7.47E−02 | **1.24E−02** ±4.92E−02 |
| $f_{20}$ | 5.90E−03 ±8.21E−03 | 3.45E−32 ±2.09E−32 | 2.40E−05 ±1.31E−05 | 4.76E+04 ±1.84E+05 | 4.33E−03 ±7.26E−03 | 7.34E−11 ±2.98E−10 | **2.67E−32** ±2.97E−32 |

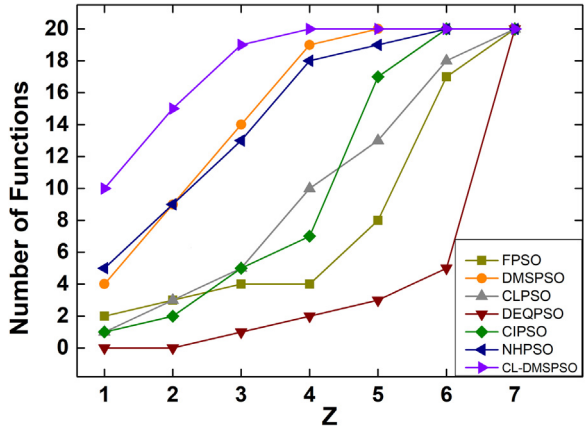



**Fig. 2.** The top-Z curves of different algorithms, where $Z = 1, 2, \ldots, 7$.

of a particle in CL-DMSPSO is $2 * N$, where $N$ is the number of waypoints for each UAV, with the starting point and destination included. To implement the computation of the communication maintenance cost and separation maintenance cost shown in Section 2.2, the best candidate solutions from the other UAVs are acquired by UAV $i$ in each iteration. Then, with the iterative update of $\mathbf{x}_i$ by CL-DMSPSO, it is easy to find the flyable path for UAV $i$ that satisfies the safety and cooperation requirements. The specific procedure is summarized as follows

Step 1: Initialize the parameters of cooperative path planning scenario for multiple UAVs, including the number of UAVs, the number of waypoints, the starting point and destination for each UAV, obstacle positions, and flight constraints for the UAVs.

Step 2: For the $i$th UAV, initialize the parameters and population structure of CL-DMSPSO. The candidate solutions have the form of $\mathbf{x}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, \ldots, x_{iN}, y_{iN})$ that need to be updated.

Step 3: For the $i$th UAV, in each iteration, calculate the cost of each selected candidate solution with Eqs. (1)–(4). Simultaneously, obtain the best candidate solutions at the current iteration from the other UAVs and calculate the communication maintenance and separation maintenance cost with Eqs. (5)–(9). Then, the overall cost $F_{obj}$ for this selected candidate solution can be calculated by sum the costs.

Step 4: For the $i$th UAV, in each iteration, implement CL-DMSPSO according to Algorithm 1 to update the candidate solutions. If the algorithm has reached the maximal number of iterations, the search process is finished and the optimal candidate solution is output as the flight path of UAV $i$. Otherwise, return to Step 3.

**Table 3**
Environmental parameters.

| Type | Location (km) | Radius (km) |
|---|---|---|
| Obstacle 1 | (23.4, 55.0) | 7.5 |
| Obstacle 2 | (33.1, 25.7) | 7.0 |
| Obstacle 3 | (38.4, 88.4) | 9.0 |
| Obstacle 4 | (71.1, 31.8) | 11.5 |

**Table 4**
UAV coordinates of starting point $(x_s, y_s)$ and destination $(x_d, y_d)$ (km).

| Coordinate | UAV 1 | UAV 2 | UAV 3 | UAV 4 | UAV 5 |
|---|---|---|---|---|---|
| $x_s$ | 6.3 | 1.5 | 6.0 | 3.1 | 27.0 |
| $y_s$ | 81.3 | 46.7 | 32.9 | 23.5 | 2.6 |
| $x_d$ | 92.1 | 92.1 | 92.1 | 92.1 | 92.1 |
| $y_d$ | 53.1 | 53.1 | 53.1 | 53.1 | 53.1 |

## 4. UAV path planning results

In this part, Section 4.1 presents a designed scenario for UAV path planning. Then, we show the simulation results on the scenario obtained by the proposed path planning method using CL-DMSPSO in Section 4.2. Subsequently, the comparison results of different PSOs with respect to the planned paths on the designed scenario are presented and discussed in Section 4.3.

### 4.1. Designed scenario

In the designed scenario, we consider a path planning problem including five UAVs that fly with a constant flight height, and the other two dimensions of their flying space are defined in [0, 100]. Suppose that four obstacles (no fly zones) exist in the above flying space, and Table 3 presents their detailed locations and radii. All five UAVs are required to fly simultaneously from their starting points to their destinations that are specified in Table 4.

Then, we construct the cooperative path planning model for multiple UAVs presented in Section 2.3 with the following specified parameters. The number of waypoints $N$ reflects the tradeoff between the computational cost and the accuracy of approximation. Generally, the larger $N$ is, the higher the accuracy of the approximation will be, while the efficiency will fall. In this paper, we set $N$ for each UAV to a suitable value of 12, with the starting point and destination included; thus, the dimension of each particle is $D = 24$. The number of risk evaluation points is $N_r = 20$, and the number of separation evaluation points is $N_s = 5$. In addition, the maximal turning angle $\theta_{max}$ is specified as 45°. The value of $d_{min}$ is affected by the flight speed of UAV, and in this paper, we set $d_{min}$ to a suitable value of 0.02 km. As for the parameters related to communication constraints, suppose $c = 3 \times 10^8$ m/s, $f_c = 2.4$ GHz, $P_t = 26$ dBm, $P_n = -110$ dBm, $k = 0.1$, $\alpha = 2$, and $SNR_t = 2.5$.

Next, we apply the proposed CL-DMSPSO to solve the constructed model in the designed scenario and make a comparison with the competing algorithms used in Section 3.4. For each algorithm, the maximal number of iterations is set as 300, and the swarm size is considered as 120. The detailed parameters for different PSO algorithms provided in Table 1 are still applied.

### 4.2. Path planning results obtained by CL-DMSPSO

In this section, we present the cooperative path planning results obtained by CL-DMSPSO. Fig. 3(a)–(f) are time series diagrams showing the flight process of the UAVs.

For the convenience of subsequent analysis, four waypoints are omitted for each UAV in Fig. 3, which does not affect the experimental results. We can observe that feasible flying paths are generated by CL-DMSPSO and that there is no conflict with the obstacles. In addition, according to the calculation results, the turning angle at each waypoint is smaller than $\theta_{max}$, and the distance between UAVs is always larger than $d_{min}$ during the flying process.

Moreover, the flying paths shown in Fig. 3 guarantee the communication connections between UAVs. In Fig. 3, if any two UAVs can communicate directly, we use a dotted line to connect them. Here, direct communication means there is a link between the two UAVs, where no other UAV is required as a communication relay.

Specifically, all UAVs are at their starting points in Fig. 3(a). At this time, LoS propagation is applied between all UAVs. We can observe that UAV 2, UAV 3 and UAV 4 can communicate directly with each other, while UAV 1 and UAV 5 can communicate through a UAV relay due to the limitation of distance. Next, when all UAVs fly to the second waypoints as shown in Fig. 3(b), the propagation mode between UAV 1, UAV 3 and UAV 5 changes to NLoS propagation on account of being blocked by obstacles. Compared with Fig. 3(a), the link between UAV 3 and UAV 5 breaks because of the decline of communication quality. UAV 3 communicates with UAV 1 and UAV 5 through a UAV relay. It is worth noting that the flight speed of UAV 2 is relatively slow at this time, and it plays a role of communication relay that connects UAV 1 and the other UAVs. In Fig. 3(c), all UAVs fly to their third waypoints. We can see that UAV 3, with a fast flight speed, becomes the communication relay to keep other four UAVs connected. Then, when the UAVs fly to the fourth waypoints as shown in Fig. 3(d), LoS propagation applies between all UAVs again. Here any two UAVs can communicate directly, expected for UAV 1 and UAV 5, which can only communicate through a communication relay. In Fig. 3(e), all UAVs can communicate directly with each other. Finally, all the UAVs fly to the destination and complete the mission as presented in Fig. 3(f).

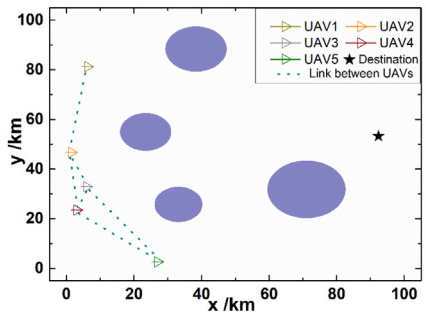
Therefore, the proposed CL-DMSPSO algorithm can find a relatively short flight path from the starting point to the destination for each UAV while satisfying all the constraints.

In order to further investigate the influence of communication constraints on path planning, we present the cooperative path planning results without considering communication constraints as a comparison. Fig. 4 shows part of the flight process of the UAVs obtained by CL-DMSPSO.

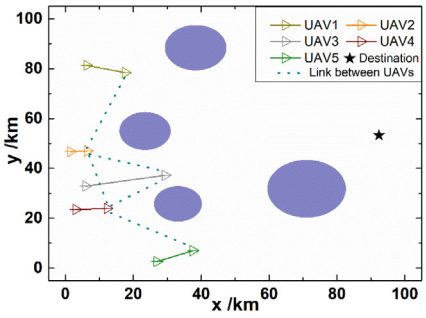
Specifically, when all UAVs fly to the second waypoints as shown in Fig. 4(a), UAV 3 communicates with UAV 1 and UAV 5 through a UAV relay, which is similar to Fig. 3(b). However, when all UAVs fly to their third waypoints in Fig. 4(b), the propagation mode between UAV 1 and UAV 3, UAV 3 and UAV 5 changes to NLoS propagation on account of being blocked by obstacles. And the link between UAV 1 and UAV 3, UAV 3 and UAV 5 breaks because of the decline of communication quality. Meanwhile, since no other UAV acts as a communication relay, UAV 1 and UAV 5 cannot communicate with other UAVs. Therefore, although all UAVs can fly to their destinations similar to Fig. 3(f), the requirements for communication cannot be satisfied. As concerns the path planning results with communication constraints, as presented in Fig. 3, the UAVs can maintain communication connections during the whole flight, which is significant to the efficient and collaborative completion of tasks for multiple UAVs.

### 4.3. Comparisons of different PSOs

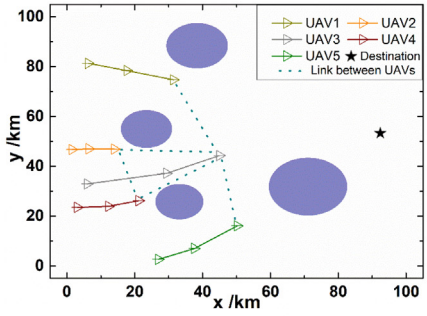
In this section, we present the comparison results of different PSOs on the designed scenario shown in Section 4.1. Here, two critical indicators, including the solution optimality and success rate, are utilized to evaluate different PSO algorithms used in the cooperative path planning problems. And the simulation results



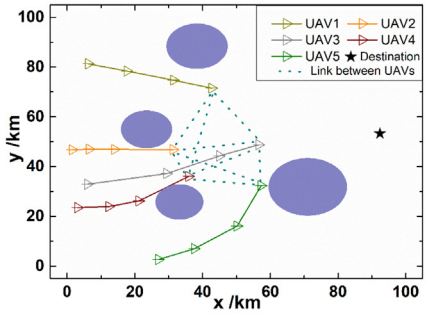

(a) Initial time of flight



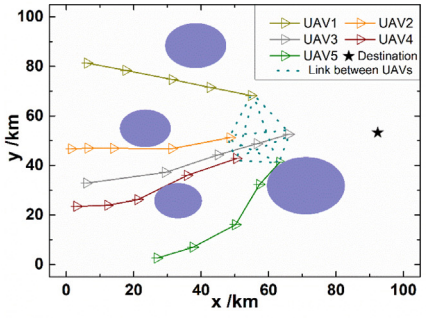

(b) UAVs fly to the second waypoints



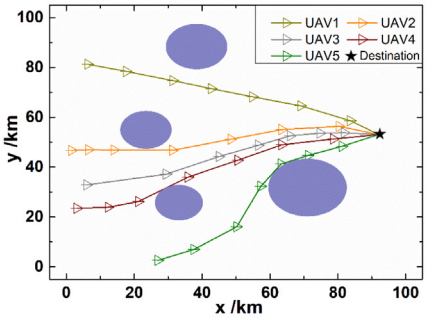

(c) UAVs fly to the third waypoints




(d) UAVs fly to the fourth waypoints

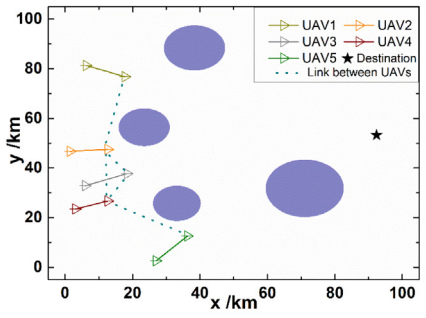



(e) UAVs fly to the fifth waypoints

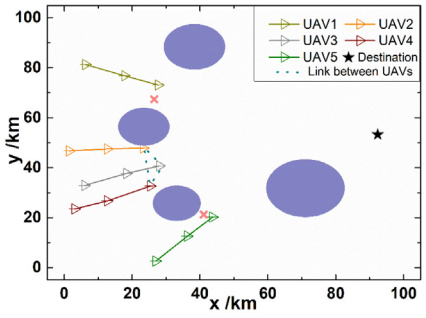



(f) UAVs fly to the destination

**Fig. 3.** Planning results with communication constraints by CL-DMSPSO.




(a) UAVs fly to the second waypoints




(b) UAVs fly to the third waypoints

**Fig. 4.** Planning results without communication constraints by CL-DMSPSO.

**Table 5**
Comparisons of different PSOs for the UAV group.

| Indicator | FPSO | DMSPSO | CLPSO | DEQPSO | CIPSO | NHPSO | CL-DMSPSO |
|---|---|---|---|---|---|---|---|
| AOFV | 2.57 | 0.95 | 1.07 | 1.50 | 1.69 | 0.58 | 0.32 |
| IP | N/A | 62.92% | 58.40% | 41.70% | 34.26% | 77.32% | 87.60% |
| Std | 2.41 | 1.54 | 1.40 | 1.65 | 2.23 | 1.12 | 0.95 |
| OOFV | 0.08 | 0.04 | 0.12 | 0.04 | 0.05 | 0.14 | 0.02 |
| WOFV | 7.48 | 6.24 | 5.32 | 6.15 | 6.49 | 5.18 | 4.18 |
| Success rate | 0.33 | 0.63 | 0.63 | 0.47 | 0.57 | 0.73 | 0.93 |

**Table 6**
Comparisons of different PSOs for the UAV group in the model without communication constraints.

| Indicator | FPSO | DMSPSO | CLPSO | DEQPSO | CIPSO | NHPSO | CL-DMSPSO |
|---|---|---|---|---|---|---|---|
| AOFV | 2.07 | 0.41 | 1.02 | 1.43 | 1.47 | 0.15 | 0.05 |
| Std | 1.14 | 0.50 | 0.72 | 0.76 | 1.06 | 0.02 | 0.02 |
| OOFV | 0.06 | 0.02 | 0.06 | 0.03 | 0.02 | 0.12 | 0.01 |
| WOFV | 4.34 | 1.87 | 2.58 | 3.40 | 4.47 | 0.21 | 0.10 |

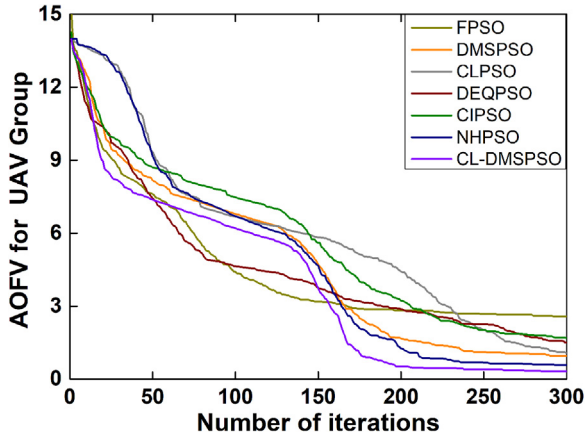



**Fig. 5.** Convergence curves of the AOFV.

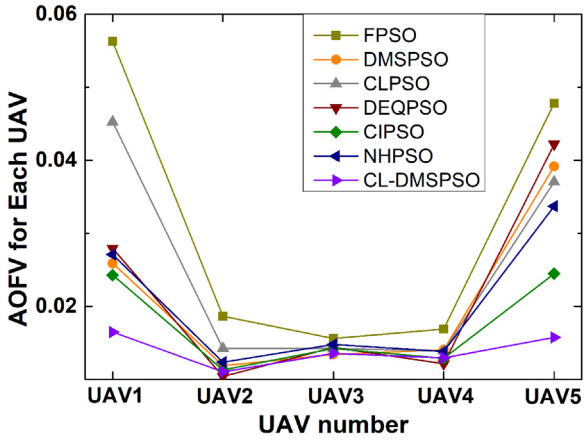



**Fig. 6.** AOFV comparison for each UAV.

exhibit the advantages of the novel CL-DMSPSO algorithm in generating optimal paths for multiple UAVs.

Considering the randomness of the search process in the PSO algorithm, 30 Monte-Carlo simulations are repeated to evaluate different algorithms and make fair and convincing comparisons. Specifically, at each repetition, we apply the proposed CL-DMSPSO and other six PSO related algorithms presented in Section 3.4 to solve the path planning problem. And then, we calculate the objective function values for the UAVs for each algorithm.

The average objective function value (AOFV), optimal objective function value (OOFV), worst objective function value (WOFV) and standard deviation (Std) for the UAV group are vital indicators that characterize the solution optimality of an algorithm. To further demonstrate the improvements of PSO variants, the improvement percentage (IP), compared with FPSO, with respect to the AOFV for the UAV group is calculated for each PSO. These results are presented in Table 5. In addition, Fig. 5 exhibits the evolution process of the AOFV for each PSO. From these we can observe that the performance of six PSO variants are significantly improved compared with FPSO. Among them, CL-DMSPSO performs best, showing the smallest AOFV, OOFV, WOFV, Std and the greatest improvement, which demonstrates the advantages of our proposed algorithm for UAV path planning.

Moreover, we also carry out path planning experiments without communication constraints. As the statistical results shown in Table 6, the proposed CL-DMSPSO algorithm still provides the best results in AOFV, OOFV, WOFV, and Std. These results further prove that CL-DMSPSO has excellent capability for UAV path planning.

In addition to the solution optimality, the success rate is also a crucial index in evaluating the performance of PSO. Here, a successful flight means that the obtained paths for all UAVs meet all constraints, that is, the only cost for each UAV is the path length. As shown in Table 5, the CL-DMSPSO algorithm has the highest success rate among these PSOs, which is approximately three times that of FPSO. This further indicates the significant advantages of the proposed path planning method based on CL-DMSPSO.

We further carry out comparisons among different PSOs with respect to AOFVs for each UAV and the UAV group under successful flight, as shown in Fig. 6 and Table 7. Table 7 shows that the CL-DMSPSO algorithm has the lowest AOFV for the UAV group, which reflects the advantages of the proposed algorithm. In addition, we observe that the AOFVs for UAV 1 and UAV 5 vary to a large extent, compared with those of UAVs 2–4. We can see that the AOFVs for UAVs 2–4 are very close with different PSOs. As for UAV 1 and UAV 5, FPSO and CL-DMSPSO maintain the maximum and minimum AOFVs, respectively, with those for other methods in the middle. This is mainly due to environmental and communication constraints and will be analyzed in the following.

We find that in the case of a successful flight, the flight paths for UAVs obtained by the other PSO algorithms appear more frequently like Fig. 7 than those by CL-DMSPSO, to satisfy the communication constraints. In contrast to Fig. 3, since there is no UAV playing a role of communication relay, UAV 1 and UAV 5 have to bypass the obstacles and fly on the same side as the other UAVs in Fig. 7. In this case, LoS propagation is applied between all UAVs during the whole flight to make it easily to achieve

**Table 7**
AOFV comparisons for each UAV and the UAV group.

| Indicator | FPSO | DMSPSO | CLPSO | DEQPSO | CIPSO | NHPSO | CL-DMSPSO |
|---|---|---|---|---|---|---|---|
| UAV 1 | 0.056 | 0.026 | 0.045 | 0.028 | 0.024 | 0.027 | 0.016 |
| UAV 2 | 0.019 | 0.012 | 0.014 | 0.010 | 0.011 | 0.012 | 0.011 |
| UAV 3 | 0.016 | 0.013 | 0.014 | 0.014 | 0.014 | 0.015 | 0.014 |
| UAV 4 | 0.017 | 0.014 | 0.014 | 0.012 | 0.013 | 0.014 | 0.013 |
| UAV 5 | 0.048 | 0.039 | 0.037 | 0.042 | 0.024 | 0.034 | 0.016 |
| UAV group | 0.156 | 0.104 | 0.124 | 0.106 | 0.086 | 0.102 | 0.070 |

**Table A.1**
Benchmark functions.

| Test function | Dimension | Search space | $f_{min}$ | Acceptance | Name |
|---|---|---|---|---|---|
| Unimodal functions | | | | | |
| $f_1$ | 30 | $[-100, 100]^D$ | 0 | 0.01 | Sphere [49] |
| $f_2$ | 30 | $[-2.048, 2.048]^D$ | 0 | 100 | Rosenbrock [49] |
| $f_3$ | 30 | $[-10, 10]^D$ | 0 | 0.01 | Schwefel P2.22 [49] |
| $f_4$ | 30 | $[-1.28, 1.28]^D$ | 0 | 0.05 | Quartic noise [49] |
| $f_5$ | 30 | $[-1.28, 1.28]^D$ | 0 | 0.05 | De Jong [50] |
| Multimodal functions | | | | | |
| $f_6$ | 30 | $[-10, 10]^D$ | 0 | 0.01 | Alpine [51] |
| $f_7$ | 30 | $[-32, 32]^D$ | 0 | 0.01 | Ackley [49] |
| $f_8$ | 30 | $[-500, 500]^D$ | 0 | 2000 | Schwefel [49] |
| $f_9$ | 30 | $[-5.12, 5.12]^D$ | 0 | 100 | Rastrigin [49] |
| $f_{10}$ | 30 | $[-5.12, 5.12]^D$ | 0 | 100 | Noncontinuous Rastrigin [49] |
| $f_{11}$ | 30 | $[-0.5, 0.5]^D$ | 0 | 0.01 | Weierstrass [49] |
| $f_{12}$ | 30 | $[-50, 50]^D$ | 0 | 0.01 | Penalized 1 [49] |
| $f_{13}$ | 30 | $[-50, 50]^D$ | 0 | 0.01 | Penalized 2 [49] |
| Rotated multimodal functions | | | | | |
| $f_{14}$ | 30 | $[-32, 32]^D$ | 0 | 0.01 | Rotated Ackley [36] |
| $f_{15}$ | 30 | $[-500, 500]^D$ | 0 | 2000 | Rotated Schwefel [36] |
| $f_{16}$ | 30 | $[-5.12, 5.12]^D$ | 0 | 100 | Rotated Rastrigin [36] |
| $f_{17}$ | 30 | $[-5.12, 5.12]^D$ | 0 | 100 | Rotated Noncontinuous Rastrigin [36] |
| $f_{18}$ | 30 | $[-0.5, 0.5]^D$ | 0 | 0.01 | Rotated Weierstrass [36] |
| $f_{19}$ | 30 | $[-50, 50]^D$ | 0 | 0.01 | Rotated Penalized 1 [49] |
| $f_{20}$ | 30 | $[-50, 50]^D$ | 0 | 0.01 | Rotated Penalized 2 [49] |

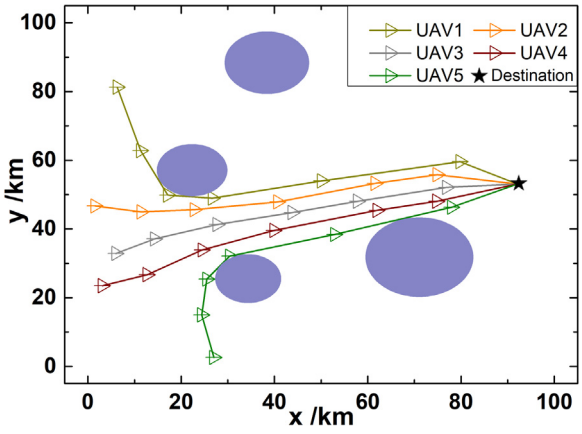



**Fig. 7.** Planning results compared with Fig. 3.

communication. However, the path length increases, especially for UAV 1 and UAV 5. As concerns our proposed CL-DMSPSO algorithm, it provides the planning path shown in Fig. 3 in most cases rather than in Fig. 7. In this case, through the relay function of UAVs as shown in Fig. 3, the UAVs can meet the communication constraints with a smaller fitness value by cooperating with each other. And the long path is avoided as much as possible.

Overall, as concerns the solution optimality and success rate, CL-DMSPSO shows better performance than the other PSOs, which illustrates the advantages of the proposed CL-DMSPSO algorithm in solving path planning problems for multiple UAVs.

## 5. Conclusions

In this paper, we focus on the cooperative path planning problem for multiple UAVs with quite a few essential factors taken into account, including general limitations regarding UAV maneuverability and collision avoidance, and especially the communication constraints between UAVs, aiming to find the feasible path with the shortest length for each UAV. Then, we further design the corresponding overall objective function, based on the criterion of minimizing the path length and necessary constraints formulated by several penalty functions. Subsequently, a novel CL-DMSPSO algorithm is proposed to solve the above path planning problem, which enhances the performance of CLPSO and DMSPSO and significantly improves the particle search accuracy. The experimental results on benchmark functions demonstrate the effectiveness and advantages of the proposed algorithm compared with other PSOs, especially for multimodal functions. In addition, we provide the detailed implementation of the presented cooperative path planning method for multiple UAVs using the CL-DMSPSO algorithm. Finally, the simulation and comparison results on the specified scenario of path planning for multiple UAVs indicate that the proposed CL-DMSPSO outperforms other PSOs with respect to the effective planning of high-quality flyable paths for multiple UAVs.

## CRediT authorship contribution statement

**Liang Xu:** Conceptualization, Methodology, Software, Writing – original draft. **Xianbin Cao:** Resources, Supervision, Writing – review & editing. **Wenbo Du:** Investigation, Project administration, Funding acquisition. **Yumeng Li:** Data curation, Validation, Visualization, Formal analysis.

**Table A.2**
Benchmark functions.

| Test function | Formula | Name |
|---|---|---|
| Unimodal functions | | |
| $f_1$ | $f_1(\boldsymbol{x}) = \Sigma_{i=1}^{D} x_i^2$ | Sphere [49] |
| $f_2$ | $f_2(\boldsymbol{x}) = \Sigma_{i=1}^{D}(100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2)$ | Rosenbrock [49] |
| $f_3$ | $f_3(\boldsymbol{x}) = \Sigma_{i=1}^{D}\lvert x_i\rvert + \Pi_{i=1}^{D}\lvert x_i\rvert$ | Schwefel P2.22 [49] |
| $f_4$ | $f_4(\boldsymbol{x}) = \Sigma_{i=1}^{D} i x_i^4 + random[0, 1)$ | Quartic Noise [49] |
| $f_5$ | $f_5(\boldsymbol{x}) = \Sigma_{i=1}^{D} i x_i^4$ | De Jong [50] |
| Multimodal functions | | |
| $f_6$ | $f_6(\boldsymbol{x}) = \Sigma_{i=1}^{D}\lvert x_i \sin(x_i) + 0.1x_i\rvert$ | Alpine [51] |
| $f_7$ | $f_7(\boldsymbol{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\Sigma_{i=1}^{D}x_i^2}) - \exp(\frac{1}{D}\Sigma_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$ | Ackley [49] |
| $f_8$ | $f_8(\boldsymbol{x}) = 418.9829 \times D - \Sigma_{i=1}^{D} x_i \sin(\lvert x_i\rvert^{\frac{1}{2}})$ | Schwefel [49] |
| $f_9$ | $f_9(\boldsymbol{x}) = \Sigma_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$ | Rastrigin [49] |
| $f_{10}$ | $f_{10}(\boldsymbol{x}) = \Sigma_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right), y_i = \begin{cases} x_i, \lvert x_i\rvert < 0.5 \\ round(2x_i)/2, \lvert x_i\rvert \geq 0.5 \end{cases}$ | Noncontinuous Rastrigin [49] |
| $f_{11}$ | $f_{11}(\boldsymbol{x}) = \Sigma_{i=1}^{D}\left(\Sigma_{k=0}^{kmax}\left[a^k\cos\left(2\pi b^k(x_i + 0.5)\right)\right]\right) - D\Sigma_{k=0}^{kmax}\left[a^k\cos\left(\pi b^k\right)\right], a = 0.5, b = 3, kmax = 20$ | Weierstrass [49] |
| $f_{12}$ | $f_{12}(\boldsymbol{x}) = \frac{\pi}{D}[10\sin^2(\pi y_1) + \Sigma_{i=1}^{D-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1})) + (y_D - 1)^2] + \Sigma_{i=1}^{D} u(x_i, 10, 100, 4), y_i = 1 + 1/4(x_i + 1), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \leq x_i \leq a \\ k(-x_i - a)^m, x_i < -a \end{cases}$ | Penalized 1 [49] |
| $f_{13}$ | $f_{13}(\boldsymbol{x}) = 0.1[\sin^2(\pi 3x_1) + \Sigma_{i=1}^{D-1}(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2(1 + \sin^2(2\pi x_D))] + \Sigma_{i=1}^{D} u(x_i, 5, 100, 4), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \leq x_i \leq a \\ k(-x_i - a)^m, x_i < -a \end{cases}$ | Penalized 2 [49] |
| Rotated multimodal functions | | |
| $f_{14}$ | $f_{14}(\boldsymbol{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\Sigma_{i=1}^{D}y_i^2}\right) - \exp\left(\frac{1}{D}\Sigma_{i=1}^{D}\cos(2\pi y_i)\right) + 20 + e, \boldsymbol{y} = \boldsymbol{M} * \boldsymbol{x}$ | Rotated Ackley [36] |
| $f_{15}$ | $f_{15}(\boldsymbol{x}) = 418.9829 \times D - \Sigma_{i=1}^{D} z_i \sin(\lvert z_i\rvert^{\frac{1}{2}}), z_i = \begin{cases} y_i \sin(\lvert y_i\rvert^{\frac{1}{2}}), \lvert y_i\rvert \leq 500 \\ 0.001(\lvert y_i\rvert - 500)^2, \lvert y_i\rvert > 500 \end{cases}, \boldsymbol{y} = \boldsymbol{y}' + 420.96, \boldsymbol{y}' = \boldsymbol{M} * (\boldsymbol{x} - 420.96)$ | Rotated Schwefel [36] |
| $f_{16}$ | $f_{16}(\boldsymbol{x}) = \Sigma_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right), \boldsymbol{y} = \boldsymbol{M} * \boldsymbol{x}$ | Rotated Rastrigin [36] |
| $f_{17}$ | $f_{17}(\boldsymbol{x}) = \Sigma_{i=1}^{D}\left(z_i^2 - 10\cos(2\pi z_i) + 10\right), z_i = \begin{cases} y_i, \lvert y_i\rvert < 0.5 \\ \frac{round(2y_i)}{2}, \lvert y_i\rvert \geq 0.5 \end{cases}, \boldsymbol{y} = \boldsymbol{M} * \boldsymbol{x}$ | Rotated Noncontinuous Rastrigin [36] |

(*continued on next page*)

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

This research is funded by the National Natural Science Foundation of China (NSFC) under Grant 61722102 and the Postdoctoral Science Foundation 2021M700332, School of Electronic Information Engineering, Beihang University, Beijing, China.

## Appendix

Function $f_1$ is easily to be solved by most algorithms. $f_2$ is relatively difficult. When $f_2$ has a high dimension, it is regarded

**Table A.2** (*continued*).

| | | |
|---|---|---|
| $f_{18}$ | $f_{18}(\boldsymbol{x}) = \Sigma_{i=1}^{D}\left(\Sigma_{k=0}^{kmax}\left[a^k \cos\left(2\pi b^k (y_i + 0.5)\right)\right]\right) - D\Sigma_{k=0}^{kmax}\left[a^k \cos\left(\pi b^k\right)\right],$ $a = 0.5, b = 3, kmax = 20, \boldsymbol{y} = \boldsymbol{M} * \boldsymbol{x}$ | Rotated Weierstrass [36] |
| $f_{19}$ | $f_{19}(\boldsymbol{x}) = \frac{\pi}{D}[10\sin^2(\pi y_1) + \Sigma_{i=1}^{D-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1}))$ $+ (y_D - 1)^2] + \Sigma_{i=1}^{D} u(z_i, 10, 100, 4), y_i = 1 + 1/4(z_i + 1),$ $u(z_i, a, k, m) = \begin{cases} k(z_i - a)^m, z_i > a \\ 0, -a \le z_i \le a \\ k(-z_i - a)^m, z_i < -a \end{cases}, \boldsymbol{z} = \boldsymbol{M} * \boldsymbol{x}$ | Rotated Penalized 1 [49] |
| $f_{20}$ | $f_{20}(\boldsymbol{x}) = 0.1[\sin^2(\pi 3y_1) + \Sigma_{i=1}^{D-1}(y_i - 1)^2(1 + \sin^2(3\pi y_{i+1}))+$ $(y_D - 1)^2(1 + \sin^2(2\pi y_D))] + \Sigma_{i=1}^{D} u(y_i, 5, 100, 4)$ $u(y_i, a, k, m) = \begin{cases} k(y_i - a)^m, y_i > a \\ 0, -a \le y_i \le a \\ k(-y_i - a)^m, y_i < -a \end{cases}, \boldsymbol{y} = \boldsymbol{M} * \boldsymbol{x}$ | Rotated Penalized 2 [49] |

**Table A.3**
Success rate after 5000 iterations.

| Function | FPSO | DMSPSO | CLPSO | DEQPSO | CIPSO | NHPSO | CL-DMSPSO |
|---|---|---|---|---|---|---|---|
| $f_1$ | **1.00** | **1.00** | **1.00** | 0.00 | **1.00** | **1.00** | **1.00** |
| $f_2$ | 0.98 | **1.00** | **1.00** | 0.54 | **1.00** | **1.00** | **1.00** |
| $f_3$ | 0.80 | **1.00** | **1.00** | 0.00 | **1.00** | **1.00** | **1.00** |
| $f_4$ | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** | **1.00** |
| $f_5$ | **1.00** | **1.00** | **1.00** | 0.72 | **1.00** | **1.00** | **1.00** |
| $f_6$ | **1.00** | **1.00** | **1.00** | 0.00 | **1.00** | **1.00** | **1.00** |
| $f_7$ | 0.36 | **1.00** | **1.00** | 0.00 | **1.00** | **1.00** | **1.00** |
| $f_8$ | 0.00 | 0.34 | **1.00** | 0.00 | 0.08 | **1.00** | 0.72 |
| $f_9$ | 0.96 | **1.00** | **1.00** | 0.98 | **1.00** | **1.00** | **1.00** |
| $f_{10}$ | 0.98 | **1.00** | **1.00** | 0.84 | **1.00** | **1.00** | **1.00** |
| $f_{11}$ | 0.00 | **1.00** | **1.00** | 0.00 | 0.70 | **1.00** | **1.00** |
| $f_{12}$ | 0.66 | 0.92 | **1.00** | 0.00 | 0.76 | **1.00** | **1.00** |
| $f_{13}$ | 0.56 | **1.00** | **1.00** | 0.00 | 0.66 | **1.00** | **1.00** |
| $f_{14}$ | 0.04 | 0.94 | **1.00** | 0.00 | 0.10 | **1.00** | **1.00** |
| $f_{15}$ | 0.00 | 0.08 | 0.00 | 0.00 | 0.08 | **0.12** | 0.06 |
| $f_{16}$ | 0.88 | **1.00** | 0.66 | 0.94 | 0.98 | **1.00** | **1.00** |
| $f_{17}$ | 0.92 | **1.00** | 0.54 | 0.90 | 0.88 | 0.92 | **1.00** |
| $f_{18}$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.20** |
| $f_{19}$ | 0.32 | 0.78 | 0.28 | 0.00 | 0.26 | 0.90 | **0.92** |
| $f_{20}$ | 0.56 | **1.00** | **1.00** | 0.00 | 0.66 | **1.00** | **1.00** |

as a multimodal function sometimes. $f_3$ is similar to $f_1$ but with discontinuous points of the first derivative. $f_4$ adds a random number based on $f_5$. Multimodal functions are much more complex. $f_6$ produces a large number of differentiable local extremes along the direction of the independent variables, and it is difficult to optimize. Except in the global minimum region, $f_7$ has larger function values at the local minima in other regions, so it is a less difficult function among the multimodal functions. The global minimum area of $f_8$ is far from the center of the search space, so it is extremely difficult to optimize. $f_9$ has a large number of deep local minimum regions, so it is difficult for particles to jump out once they fall into local optima. $f_{10}$ adds discontinuous points on the basis of $f_9$. $f_{11}$ has some non-derivable points and has more than one global minimum point. $f_{12}$ and $f_{13}$ also contain a large number of local optima. The rotated multimodal functions rotate the independent variable on the basis of ordinary multimodal functions. The general method of the rotation operation is to obtain the rotation vector **y** by multiplying the rotation matrix **M** to the left for the input vector **x**, and then use **y** as the input vector to calculate the objective function value [53]. After the rotation operation, the objective function cannot decompose the different dimensions, and the relationship between the dimensional variables is more complex, so the difficulty of optimization is greatly increased (see Table A.2).

## References


[1] M. Erdelj, E. Natalizio, K.R. Chowdhury, I.F. Akyildiz, Help from the sky: leveraging UAVs for disaster management, IEEE Pervasive Comput. 16 (2017) 24–32.

[2] E. Besada-Portas, L.d.l. Torre, J.M.d.l. Cruz, B.d. Andrés-Toro, Evolutionary trajectory planner for multiple UAVs in realistic scenarios, IEEE Trans. Robot. 26 (2010) 619–634.

[3] H. Shakhatreh, A.H. Sawalmeh, A. Al-Fuqaha, Z.C. Dou, E. Almaita, I. Khalil, N.S. Othman, A. Khreishah, M. Guizani, Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges, IEEE Access 7 (2019) 48572–48634.

[4] H. Shen, D.F. Lin, T. Song, A real-time siamese tracker deployed on UAVs, J. Real-Time Image Process 19 (2022) 463–473.

[5] E. Besada-Portas, L.d.l. Torre, A. Moreno, J.L. Risco-Martin, On the performance comparison of multi-objective evolutionary UAV path planners, Inform. Sci. 238 (2013) 111–125.

[6] G. Skorobogatov, C. Barrado, E. Salami, Multiple UAV systems: a survey, Unmann. Syst. 8 (2020) 149–169.

[7] X.Y. Zhang, H.B. Duan, An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, Appl. Soft Comput. 26 (2015) 270–284.

[8] H.R. Zhu, Y.H. Wang, X.T. Li, UCAV path planning for avoiding obstacles using cooperative co-evolution spider monkey optimization, Knowl.-Based Syst. 246 (2022) 108713.

[9] C.W. Zheng, L. Li, F.J. Xu, F.H. Sun, M.Y. Ding, Evolutionary route planner for unmanned air vehicles, IEEE Trans. Robot. 21 (2005) 609–620.

[10] M. Shanmugave, A. Tsourdos, B. White, R. Zbikowski, Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs, Control Eng. Pract. 18 (2010) 1084–1092.
[11] Y.J. Zhao, Z. Zheng, Y. Liu, Survey on computational-intelligence-based UAV path planning, Knowl.-Based Syst. 158 (2018) 54–64.
[12] P. Yao, H.L. Wang, H.X. Ji, Multi-UAVs tracking target in urban environment by model predictive control and improved grey wolf optimizer, Aerosp. Sci. Technol. 55 (2016) 131–143.
[13] D.F. Zhang, H.B. Duan, Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning, Neurocomputing 313 (2018) 229–246.
[14] S. Perez-Carabaza, J. Scherer, B. Rinner, J.A. Lopez-Orozco, E. Besada-Portas, UAV trajectory optimization for minimum time search with communication constraints and collision avoidance, Eng. Appl. Artif. Intell. 85 (2019) 357–371.
[15] Y. Kuwata, J.P. How, Cooperative distributed robust trajectory optimization using receding horizon MILP, IEEE Trans. Control Syst. Technol. 19 (2011) 423–431.
[16] M.G. Earl, R. D'andre, Iterative MILP methods for vehicle control problems, IEEE Trans. Robot. 21 (2005) 1158–1167.
[17] T.R. Jorris, R.G. Cobb, Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints, J. Guid. Control Dyn. 32 (2009) 551–572.
[18] U. Orozco-Rosas, O. Montiel, R. Sepulveda, Mobile robot path planning using membrane evolutionary artificial potential field, Appl. Soft Comput. 77 (2019) 236–251.
[19] Y.B. Chen, G.C. Luo, Y.S. Mei, J.Q. Yu, X.L. Su, UAV path planning using artificial potential field method updated by optimal control theory, Internat. J. Systems Sci. 47 (2016) 1407–1420.
[20] Y.B. Chen, J.Q. Yu, X.L. Su, G.C. Luo, Path planning for multi-UAV formation, J. Intell. Robot. Syst. 77 (2015) 229–246.
[21] S. Bayili, F. Polat, Limited-Damage A*: A path search algorithm that considers damage as a feasibility criterion, Knowl.-Based Syst. 24 (2011) 501–512.
[22] Z. Nilforoushan, A. Mohades, M.M. Rezaii, A. Laleh, 3D hyperbolic voronoi diagrams, Comput. Aided Des. 42 (2010) 759–767.
[23] A. Autere, Hierarchical A* based path planning - a case study, Knowl.-Based Syst. 15 (2002) 53–66.
[24] P. Yang, K. Tang, J.A. Lozano, X.B. Cao, Path planning for single unmanned aerial vehicle by separately evolving waypoints, IEEE Trans. Robot. 31 (2015) 1130–1146.
[25] V. Roberge, M. Tarbouchi, G. Labonte, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, IEEE Trans. Ind. Inform. 9 (2013) 132–141.
[26] D. Pandit, L. Zhang, S. Chattopadhyay, C.P. Lim, C.Y. Liu, A scattering and repulsive swarm intelligence algorithm for solving global optimization problems, Knowl.-Based Syst. 156 (2018) 12–42.
[27] Y.G. Fu, M.Y. Ding, C.P. Zhou, H.P. Hu, Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization, IEEE Trans. Syst. Man Cybern. Syst. 43 (2013) 1451–1465.
[28] P.K. Das, H.S. Behera, S. Das, H.K. Tripathy, B.K. Panigrahi, S.K. Pradhan, A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment, Neurocomputing 207 (2016) 735–753.
[29] P. Yang, G.Z. Lu, K. Tang, X. Yao, A multi-modal optimization approach to single path planning for unmanned aerial vehicle, in: Proceedings of the 2016 IEEE Congress on Evolutionary Computation, 2016, pp. 1735–1742.
[30] C.Z. Qu, W.D. Gai, J. Zhang, M.Y. Zhong, A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning, Knowl.-Based Syst. 194 (2020) 105530.
[31] X.B. Yu, C.L. Li, J.F. Zhou, A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios, Knowl.-Based Syst. 204 (2020) 106209.
[32] X.Y. Zhang, S. Xia, T. Zhang, et al., Hybrid FWPS cooperation algorithm based unmanned aerial vehicle constrained path planning, Aerosp. Sci. Technol. 118 (2021) 107004.
[33] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: Proceedings of the IEEE Congress on Swarm Intelligence Symposium, 2005, pp. 124–129.
[34] J.J. Liang, H. Song, B.Y. Qu, X.B. Mao, Path planning based on dynamic multi-swarm particle swarm optimizer with crossover, in: 8th International Conference on Intelligent Computing, ICIC, vol. 7390, 2013 pp. 159–166.
[35] J.J. Liang, H. Song, B.Y. Qu, Performance evaluation of dynamic multi-swarm particle swarm optimizer with different constraint handling methods on path planning problems, in: IEEE 2nd Workshop on Memetic Computing, MC, 2014, pp. 65–71.
[36] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (2006) 281–295.
[37] E. Lu, L.Z. Xu, Y.M. Li, Z. Ma, Z. Tang, C.M. Luo, A novel particle swarm optimization with improved learning strategies and its application to vehicle path planning, Math. Probl. Eng. 2019 (2019).
[38] J. Liu, S. Anavatti, M.G.H. Abbass, Comprehensive learning particle swarm optimisation with limited local search for UAV path planning, in: IEEE Symposium Series on Computational Intelligence, SSCI, 2019 pp. 2287–2294.
[39] S.K. Shao, Y. Peng, C.L. He, Y. Du, Efficient path planning for UAV formation via comprehensively improved particle swarm optimization, ISA Trans. 97 (2020) 415–430.
[40] W.B. Du, W. Ying, P. Yang, X.B. Cao, G. Yan, K. Tang, D.P. Wu, Network-based heterogeneous particle swarm optimization and its application in UAV communication coverage, IEEE Trans. Emerg. Top. Comput. Intell. 4 (2020) 312–323.
[41] Q. Cai, M.G. Gong, L.J. Ma, S.S. Ruan, F.Y. Yuan, L.C. Jiao, Greedy discrete particle swarm optimization for large-scale social network clustering, Inform. Sci. 316 (2015) 503–516.
[42] M.G. Gong, Q. Cai, X.W. Chen, L.J. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, IEEE Trans. Evol. Comput. 18 (2014) 82–97.
[43] Q. Cai, M.G. Gong, B. Shen, L.J. Ma, L.C. Jiao, Discrete particle swarm optimization for identifying community structures in signed social networks, Neural Netw. 58 (2014) 4–13.
[44] M.D. Phung, C.H. Quach, T.H. Dinh, Q. Ha, Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection, Autom. Constr. 81 (2017) 25–33.
[45] H.T. Zhao, H.J. Wang, W.Y. Wu, J.B. Wei, Deployment algorithms for UAV airborne networks toward on-demand coverage, IEEE J. Sel. Areas Commun. 36 (2018) 2015–2031.
[46] Y. Zeng, Q.Q. Wu, R. Zhang, Accessing from the sky: a tutorial on UAV communications for 5G and beyond, Proc. IEEE 107 (2019) 2327–2375.
[47] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, 1995 pp. 1942–1948.
[48] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intell. 1 (2007) 33–57.
[49] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
[50] J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in: Proceedings of the 2004 Congress on Evolutionary Computation, 2004, pp. 325–331.
[51] W. Li, X. Meng, Y. Huang, Z.H. Fu, Multipopulation cooperative particle swarm optimization with a mixed mutation strategy, Inform. Sci. 529 (2020) 179–196.
[52] K. Tang, P. Yang, X. Yao, Negatively correlated search, IEEE J. Sel. Areas Commun. 34 (2016) 542–550.
[53] R. Salomon, Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms, Biosystems 39 (1996) 263–278.