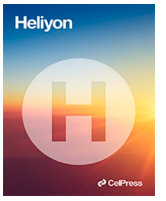
Heliyon 10 (2024) e37286

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

A novel multi-objective dung beetle optimizer for Multi-UAV cooperative path planning

Qianwen Shen, Damin Zhang *, Qing He, Yunfei Ban, Fengqin Zuo

School of Big Data and Information Engineering, Guizhou University, Guiyang, 550000, People's Republic of China

ARTICLE INFO

Keywords:
Cooperative path planning
Multiple UAVs
Directional evolutionary strategy
Adaptive stochastic ranking mechanism

ABSTRACT

Path planning for multiple unmanned aerial vehicles (UAVs) is crucial in collaborative operations and is commonly regarded as a complicated, multi-objective optimization problem. However, traditional approaches have difficulty balancing convergence and diversity, as well as effectively handling constraints. In this study, a directional evolutionary non-dominated sorting dung beetle optimizer with adaptive stochastic ranking (DENSDBO-ASR) is developed to address these issues in collaborative multi-UAV path planning. Two objectives are initially formulated: the first one represents the total cost of length and altitude, while the second represents the total cost of threat and time. Additionally, an improved multi-objective dung beetle optimizer is introduced, which integrates a directional evolutionary strategy including directional mutation and crossover, thereby accelerating convergence and enhancing global search capability. Furthermore, an adaptive stochastic ranking mechanism is proposed to successfully handle different constraints by dynamically adjusting the comparison probability. The effectiveness and superiority of DENSDBO-ASR are demonstrated by the constrained problem functions (CF) test, the Wilcoxon rank sum test, and the Friedman test. Finally, three sets of simulated tests are carried out, each including different numbers of UAVs. In the most challenging scenario, DENSDBO-ASR successfully identifies feasible paths with average values of the two objective functions as low as 637.26 and 0. The comparative results demonstrate that DENSDBO-ASR outperforms the other five algorithms in terms of convergence accuracy and population diversity, making it an exceptional optimization approach to path planning challenges.

1. Introduction

Unmanned aerial vehicles (UAVs) have been widely used in various fields, such as aerial photography, inspection, and emergency rescue, owing to their real-time performance, flexible deployment, and cost-effective maintenance. However, for complex tasks, the limitations of the payload capacity and energy storage of a single UAV highlight the benefits of collaborative efforts among multiple UAVs. Path planning is a crucial part of planning missions for multiple UAVs, which involves finding the best routes between specified starting and ending sites to decrease the length of the path and reduce the chances of collisions [1]. The trajectory optimization problem has evolved into a large-scale combinatorial optimization challenge due to the increasing number of UAV swarms [2]. It is essential to optimize the trajectory of each UAV at the same time to achieve the overall mission objectives. Additionally, environmental factors and interactions among UAVs should be taken into account. Therefore, it is of great interest to investigate efficient techniques

* Corresponding author.
E-mail address: dmzhang@gzu.edu.cn (D. Zhang).

https://doi.org/10.1016/j.heliyon.2024.e37286
Received 15 February 2024; Received in revised form 22 August 2024; Accepted 30 August 2024
Available online 4 September 2024

for determining the optimal routes in collaborative path planning for multiple UAVs.

In recent years, extensive research has been carried out on cooperative path-planning methods. Path optimization is typically addressed using classical approaches and metaheuristic algorithms (MA) [3]. Classical approaches such as artificial potential field [4] and rapidly exploring random tree (RRT) [5] have been widely used. Rao et al. [4] introduced a hybrid artificial potential field-A* algorithm (APF-A*), which utilizes repulsive force to optimize the A* algorithm's heuristic function. The results show that the algorithm can minimize flight collisions while enhancing safety. Guo et al. [5] considered the limitations of multiple UAVs in terms of both spatial and temporal constraints and proposed the RRT* algorithm based on heuristic decentralized prioritized planning (HDP-TSRRT*). This method designs the cooperative paths of all UAVs in a hierarchical manner, which enhances the efficiency of collaborative path planning. However, these techniques can be time-consuming and may not adequately adjust to uncertainty in complex environments. Metaheuristic algorithms have gained attention and have proven to be effective in addressing these drawbacks in the last two decades [6]. Inspired by real biological communities, metaheuristic algorithms (MA) with self-learning and adaptive properties, such as particle swarm optimization (PSO) [7] and genetic algorithm (GA) [8], can develop practical solutions in difficult contexts. Shao et al. [9] proposed a hierarchical optimization approach that combines improved particle swarm optimization and the Gaussian pseudo-spectral method (GPM), taking into account the flight time and the dynamic model of the UAV during the evolutionary process. The results indicate that the suggested method has better obstacle avoidance and a faster runtime. Yan et al. [10] utilized an improved genetic algorithm to solve the path planning problem for multiple UAVs. They also presented an unlocking technique to avoid deadlocks. Karthik et al. [11] proposed an improved green anaconda optimization system (IGAOS) to address the issue of coverage path planning for UAVs. The results demonstrate that the algorithm expedites the cooperative searching process. Despite the approved benefits of MAs, their drawbacks, such as slow convergence and local optima vulnerability, should be considered [12,13]. As an increasingly common optimization approach, these issues can be addressed by adopting effective strategies, resulting in satisfactory results [14,15].

The dung beetle optimizer (DBO) [16] is a recently developed swarm intelligence algorithm that imitates the behaviors of dung beetles, including ball-rolling, dancing, foraging, stealing, and breeding. Although the method is efficient and easy to implement, it has specific obstacles. For example, the performance of the system depends on how its parameters are set and adjusted, yet there are no clear criteria for identifying the best parameter values. Furthermore, the DBO does not typically guarantee the identification of the global optimal solutions, particularly in situations with several objectives and various local optima [17]. Moreover, since DBO was originally designed for single-objective optimization, it requires modifications to be applicable to multi-objective optimization situations. The improvement works of DBO have been the subject of extensive systematic research [18,19].

Collaborative multi-UAV path planning plays a crucial role in improving mission execution and reducing costs and energy consumption in aviation. An ideal model should give priority to safety and cost considerations, ensuring smooth operation of the system by minimizing risks and efficiently managing task allocation. Over the past decades, researchers have successfully developed various path planners using metaheuristic algorithms to address trajectory planning problems for UAVs. Ren and Zhang [20] proposed an adaptive evolutionary multi-objective estimation distribution algorithm (AEMO-EDA) for high-dimensional multi-objective path planning. Path length, threat level, flight height, and spatial coordination cost are the four goals that this method optimizes. However, simultaneously optimizing these functions greatly increases the complexity of the model. Ge et al. [21] enhanced the NSGA-III algorithm by incorporating the local fruit fly optimization algorithm to tackle path planning for multiple UAVs with various constraints. Nevertheless, there is potential to reduce the time complexity of the approach. Xu et al. [22] improved the PSO algorithm by using dynamic multi-swarm and deep learning strategies to address path planning in the presence of communication limitations. While this approach is practical as a single UAV solver for cooperative trajectory planning, it reduces the solution's complexity at the expense of global optimization. Ghambari et al. [23] designed an enhanced non-dominated sorting genetic algorithm-II (ENSGA-II) for multi-objective path planning of UAVs. The tests with real data showed that the algorithm has faster convergence speed and better scalability in finding solutions. However, the limited path model may reduce flying safety and effectiveness. Wan et al. [24] focused on path planning in disaster emergency response and proposed a precise three-dimensional path planning method that reduces both the trajectory length and terrain threat level. However, the model should take into account the time cost, which is essential for the efficiency of mission rescue.

In previous research, multi-UAV cooperative path planning problems have typically been modeled as multi-objective optimization problems. These problems involve balancing different objectives, such as path length, energy consumption, and threat cost, which are in competition with each other. However, these models often overlook the extensive effects of temporal and spatial constraints on collaboration, resulting in a disadvantage in practical applications. Moreover, path-planning algorithms based on swarm intelligence tend to converge on local optimal solutions, thus reducing their robustness. The objective of this study is to develop a comprehensive model that can consider both temporal and spatial synergy constraints to generate more reliable and efficient path-planning solutions. In addition, improved multi-objective optimization techniques are utilized to enhance the ability to identify global optimal solutions within complex constraints.

However, since multiple conflicting objectives with different constraints need to be optimized simultaneously, dealing with constrained multi-objective optimization problems is extremely challenging. To effectively solve these problems, it is crucial to use the constrained handling technique (CHT). One of the most popular approaches in the present research is the penalty function method [25]. This method transforms constrained optimization problems into unconstrained ones by incorporating penalty terms into the objective function. However, the efficiency of the algorithm is greatly influenced by the selection of penalty coefficients. Finding an appropriate parameter setting is challenging. To avoid adjusting the penalty coefficients, scholars have suggested a novel approach that involves separating and comparing objectives and constraints, including the constrained dominance principle (CDP) [26], the ε constrained method [27], and stochastic ranking (SR) [28]. The SR method is a widely used approach in constrained optimization that

effectively balances feasible solutions with multiple objectives. For example, Balanda et al. [29] proposed a hybrid algorithm called SRIFA, which combines the improved firefly algorithm with stochastic ranking. Gu et al. [30] developed an enhanced stochastic ranking technique that introduced a fitness mechanism and a dynamic probability operator to achieve a balance between solution convergence and diversity. Due to its simplicity and efficiency, stochastic ranking is suitable to be combined with multi-objective algorithms for collaborative constrained path planning optimization. It can significantly enhance computing speed while maintaining the quality of candidate solutions. However, when facing intricate challenges like multi-UAV cooperative path planning, it increases selection pressure and reduces diversity, leading to poor candidate solutions and path generation.

Motivated by the above considerations, a directional evolutionary non-dominated sorting dung beetle optimizer combined with adaptive stochastic ranking (DENSDBO-ASR) is presented for solving multi-UAV path planning. The main contributions are summarized as follows:

- A directional evolutionary multi-objective dung beetle optimizer based on non-dominated sorting (DENSDBO) is developed. A directional evolutionary strategy is introduced, including directional crossover and mutation operators, in which directional crossover helps to improve convergence speed and accuracy and directional mutation helps to enhance global search capability.
- An adaptive stochastic ranking mechanism is proposed to handle the multiple constraints, which dynamically adjusts the comparison probability according to the current evolutionary stage and constraint violation degree. By utilizing information from infeasible solutions, this mechanism can improve the quality of the candidate solutions, thus obtaining effective paths for UAVs.
- The DENSDBO algorithm is incorporated with the adaptive stochastic ranking mechanism to guide the search toward optimal solutions, thereby generating safe and feasible flight paths for multi-UAV path planning. The evaluation of DENSDBO-ASR through benchmark functions and UAV simulation experiments demonstrates its effectiveness.

The rest of this paper is organized as follows: Section 2 illustrates the related research background. In Section 3, the multi-UAV path planning model is described in detail. Section 4 presents the novel DENSDBO-ASR algorithm. In Section 5, the experimental results of DENSDBO-ASR are analyzed. Section 4.8 describes the implementation of DENSDBO-ASR in UAV path planning. Section 6 conducts the UAV planning simulation experiments. Section 7 gives the documented discussion. Finally, Section 8 provides a summary of the paper.

2. Related research background

2.1. Constrained multi-objective optimization problem

Generally, the multi-objective optimization problem with constraints can be formulated by equation (1).

$$\begin{aligned} &\min\ F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\ &s.t.\ \begin{cases} g_j(x) \le 0, j = 1, \ldots, p \\ h_j(x) = 0, j = p+1, \ldots, q \\ x = (x_1, x_2, \ldots, x_D)^T \in S \end{cases} \end{aligned} \tag{1}$$

where S is decision space, x is the D-dimensional decision vector, $lb_k \le x_k \le ub_k$; m is the dimension of the objectives; lb_k and ub_k are the lower and upper bounds of the kth dimensional decision variable. $g_j(x)$ and $h_j(x)$ represent the jth inequality and equality constraints, respectively. p and (q-p) denote the number of inequality and equality constraints, respectively.

For equality constraints, the following operations can be performed according to equation (2).

$$|h(x)| - \delta \le 0 \tag{2}$$

where, δ is a positive tolerance value to relax the equality constraints.

In general, the constraint violation degree of a decision vector x on the jth constraint can be expressed by equation (3).

$$cv_j(x) = \begin{cases} \max\left(0, g_j(x)\right), 1 \le j \le p \\ \max\left(0, \left|h_j(x)\right| - \delta\right), p+1 \le j \le q \end{cases} \tag{3}$$

Then, the total constraint violation degree can be calculated by equation (4).

$$CV(x) = \sum_{j=1}^{k} cv_j(x) \tag{4}$$

The CV reflects the infeasibility of individual x in the population. If the CV of a solution is 0, then it is a feasible solution; otherwise, it is an infeasible solution.

When two feasible solutions x_1 and x_2 satisfy equation (5), the solution individual x_1 Pareto dominates x_2.

$$\begin{aligned} &\forall i = 1, 2, \ldots, m, f_i(x_1) \le f_i(x_2) \\ &\wedge \exists j = 1, 2, \ldots, m, f_j(x_1) \le f_j(x_2) \end{aligned} \tag{5}$$

If there is no other solution that can Pareto dominate x, then solution x is called a non-dominated solution. The set composed of non-dominated solutions is the Pareto set (PS), and the mapping of PS in objective space is the Pareto front (PF).

2.2. Multi-UAV path planning algorithms

In the existing research, multi-UAV path planning methods are divided into three categories: 1) deterministic algorithms, such as the A-star (A*) algorithm and the Dijkstra algorithm [31]; 2) stochastic algorithms, such as particle swarm optimization (PSO) and the ant colony algorithm (ACO) [7,32]; 3) learning-based algorithms, such as the neural networks and deep learning [33]. Deterministic algorithms, based on mathematical principles, provide UAVs with precise routes and are suitable for static environments. Zhang et al. [31] proposed an improved A-star algorithm that includes bidirectional sector extension and variable step-size search techniques to avoid static radar threats, thereby increasing the efficiency of multi-UAV path planning. Mandloi et al. [34] presented two variants of the A* algorithm: Theta* (Theta-star) and Lazy Theta* (Lazy Theta-star), which effectively address the constraints of the A* algorithm in practical scenarios. However, when dealing with extensive settings and a significant number of UAVs, the computational cost of deterministic methods can be extremely high. For tasks that demand real-time response, such as military surveillance or disaster management, deterministic algorithms may face difficulties in meeting the specified time constraints.

Stochastic algorithms exhibit flexibility and adaptability when it comes to path planning in complex or dynamic environments. The ACO algorithm emulates the foraging behavior of ants to find the optimal path through pheromone transmission. For the static path planning problem, Wu et al. [32] proposed an adaptive ant colony optimization algorithm. A heuristic factor with directional information is introduced to accelerate the convergence speed, while the state transfer rule is improved to enhance the efficiency of path search. The PSO algorithm optimizes paths by modeling the social behavior of bird swarms. Yu et al. [7] employed the simulated annealing algorithm to improve the updating strategy of the particle and used the quantum learning approach to accelerate the convergence of the process. The results indicate that the improved particle swarm optimization algorithm has superior robustness in complex situations. However, stochastic algorithms sometimes tend to converge in the local search area rather than finding a globally optimal solution. These algorithms often require significant parameter adjustment to achieve a balance between exploration and exploitation.

Learning-based algorithms use machine learning techniques to predict and plan paths with adaptivity and flexibility. Neural networks create a model to predict paths by learning a large amount of data. Sanna et al. [35] used artificial neural networks to establish the relationship between the UAV's current states and its subsequent actions for the path planning problem in complex urban networks. The multi-UAV path search is accomplished by first learning the search strategy of a single UAV and then replicating it to multiple UAVs. Chen et al. [36] proposed an architecture that simplifies the path planning of multi-UAVs in unknown environments. The proposed architecture involves centralized training and decentralized execution. However, learning-based algorithms typically require substantial quantities of data to effectively train the model, which can be challenging to get in real-world applications. Moreover, the process of training complex neural networks or deep learning models sometimes needs significant computational resources.

Inspired by real biological communities, metaheuristic algorithms have been extensively used in collaborative multi-UAV path planning because of their capacity to handle environmental uncertainties and their low computational cost [37]. Recent studies have shown that bio-inspired methods are a primary way for achieving high-quality solutions in complex and dynamic environments with multiple constraints [6]. Jiaqi et al. [38] presented an adaptive grey wolf optimization algorithm for multi-UAV path planning. The enhancement incorporates the spiral update position method and an adaptive mechanism to adjust the number of leaders. The results demonstrate that the proposed method can effectively improve the convergence speed of the algorithm and shorten the UAV flight time. Kumar et al. [39] developed two variants of the grey wolf optimization algorithm: the improved grey wolf optimizer and the variable-weighted grey wolf optimizer. By integrating these two variants, an enhanced variable weight grey wolf optimization algorithm based on reinforcement learning (RLV-GWO) is proposed to solve the multi-UAV path planning problem. Duan et al. [40] proposed a dynamic discrete pigeon-inspired optimization to address the path planning problem in a multi-UAV cooperative mission. The original pigeon-inspired optimization is enhanced by incorporating a sigmoid model with response thresholds, and a probabilistic graph is established to guide the UAV navigation. Yu et al. [41] combined the simulated annealing algorithm and whale optimization algorithm to propose a hybrid approach called SA-WOA for the multi-UAV cooperative detection problem. The method includes adaptive weights with Levy flight characteristics to optimize the Metropolis criterion and integrates an adaptive tempering mechanism into the optimization process. The results confirm the effectiveness of SA-WOA in realizing multi-UAV cooperative path planning with strong stability and search accuracy.

Previous research has focused on improving specific algorithms, ignoring the diversity and complexity of multi-UAV cooperative scenarios. It is impossible to develop an algorithm that can be applied to any scenario. Furthermore, according to the "No Free Lunch" theorem [42], a determined metaheuristic cannot perform well in all types of optimization problems. This fact motivates researchers to improve current algorithms or develop new ones, which is the primary motivation for this study.

2.3. Dung beetle optimize

As a typical swarm intelligence algorithm, DBO simulates the ball-rolling, dancing, foraging, stealing, and reproduction behaviors of dung beetles. The DBO classifies the population into four search agents: ball-rolling dung beetle, brood ball, small dung beetle, and thief. Here is a brief introduction.

Ball-rolling dung beetles rely on celestial cues to roll their dung balls in a straight line. The position of the beetle can be expressed by

equations (6) and (7). When encountering obstacles, the beetle uses a tangent function to determine a new rolling direction, as shown in equation (8).

$$x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \tag{6}$$

$$\Delta x = |x_i(t) - X^w| \tag{7}$$

$$x_i(t+1) = x_i(t) + \tan(\theta)|x_i(t) - x_i(t-1)| \tag{8}$$

where t is the current iteration, $x_i(t)$ represents the position of the ith dung beetle after the tth iteration, $k \in (0, 0.2]$ represents the deflection coefficient, b is a constant value, α is a natural coefficient assigned to 1 or -1, Δx simulates changes in light intensity, X^w is the global worst position, θ is the deflection angle.

To choose a suitable spawning site, female dung beetles use a boundary selection strategy defined in equation (9). The position of the egg ball changes dynamically during the iteration process, as shown in equation (10).

$$\begin{aligned} Lb^* &= \max(X^* \times (1-R), Lb) \\ Ub^* &= \min(X^* \times (1-R), Ub) \end{aligned} \tag{9}$$

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \tag{10}$$

where Lb and Ub are the lower and upper bounds of the search space, Lb^* and Ub^* represent the lower and upper bounds of the spawning area, $R = 1 - t/T_{\max}$, and $T_{\max}$ denotes the maximum number of iterations. $B_i(t)$ is the position of the ith egg at the tth iteration, b_1 and b_2 are two random vectors with a size of $1 \times D$, and D is the dimension of the search space.

Small dung beetles burrow out of the ground to search for food within optimal foraging areas, as defined in equation (11). The position of the small dung beetle is updated using equation (12).

$$\begin{aligned} Lb^b &= \max\left(X^b \times (1-R), Lb\right) \\ Ub^b &= \min\left(X^b \times (1-R), Ub\right) \end{aligned} \tag{11}$$

$$x_i(t+1) = x_i(t) + C_1 \times \left(x_i(t) - Lb^b\right) + C_2 \times \left(x_i(t) - Ub^b\right) \tag{12}$$

where X^b represents the global optimal position, Lb^b and Ub^b are the lower and upper bounds of the optimal foraging area, respectively. $x_i(t)$ represents the position of the ith dung beetle at the tth iteration, C_1 is a random number subject to a normal distribution, and C_2 is a random vector within the range of (0,1).

Thieves steal dung balls from other beetles, and their location is updated using equation (13).

$$x_i(t+1) = X^b + S \times g \times \left(|x_i(t) - X^*| + |x_i(t) - X^b|\right) \tag{13}$$

where g is a random vector with the size of $1 \times D$, and S is a constant value.

3. UAV path planning model

A brief introduction to the collaborative multi-UAV path planning problem is given in Section 3.1. In Section 3.2, we define the path in three-dimensional space. Finally, the detailed objective functions and constraints are depicted in Section 3.3.

3.1. Problem description

In the scenario of collaborative path planning, multiple UAVs work together to accomplish specific tasks in a complex environment. Multi-UAV path planning involves determining optimal paths for UAVs to minimize the overall flight path and avoid any collisions. Several factors need to be considered, such as cooperative constraints among UAVs (temporal and spatial synergy constraints) and UAV maneuvering constraints (flight speed, distance, altitude, and turning angle). Therefore, this paper proposes a multi-UAV path planning model in a three-dimensional task space to realize optimal path planning for multiple UAVs.

3.2. Path representation

In this paper, the rasterized map is used to represent the flight space. The rasterization method divides the x-axis, y-axis, and z-axis into a series of sub-areas with a certain step. Then, the flight space of the UAV can be expressed by equation (14).

$$\Omega = \left\{ \left(x_i, y_j, z_k\right) \middle| \; i = 1, 2, \ldots, n_x + 1, j = 1, 2, \ldots, m_y + 1, k = 1, 2, \ldots, l_z + 1 \right\} \tag{14}$$

where, n_x, m_y, l_z represent the total number of sub-regions divided by the x-axis, y-axis, and z-axis, respectively.

Assuming that the UAV's starting point is $S = (x_0, y_0, z_0)$ and the endpoint is $E = (x_{n+1}, y_{n+1}, z_{n+1})$. The flight path of the UAV can be represented as a combination of a series of discrete points, where the two adjacent points are linked by a straight-line segment. For

convenience, we use P to denote the flight path of the UAV, then the discrete path can be described by equation (15).

$$P = \{p_0, p_1, \ldots, p_k, \ldots p_{n+1}\} \tag{15}$$

where, $p_k = (x_k, y_k, z_k)$, n is the number of path points.

In order to ensure the smoothness of the UAV, B-spline curves are adopted to define the UAV's desired path, which guarantees the curvature continuity of the whole path curve. The UAV path includes a start point, N waypoints, and a target point as $Path = \{p_0, \ldots, p_i, p_{i+1}, p_{i+2}, \ldots, p_{n+1}\}$, which is the solid black line in Fig. 1. The path can be determined by n control points $w_i = (x_i, y_i, z_i)$ and a pre-defined path smooth strategy. The control points can be any point in space, enabling the UAV to avoid danger zones by moving backwards. The B-spline curve requires several control points to define complex path curves. It allows the UAV to move flexibly and avoid obstacles, which is essential for the stable flight of the UAV.

3.3. Cost function and performance constraints

3.3.1. Length cost

The path length is a significant factor in determining the optimal path. Generally speaking, the shorter the path length, the less the flight time and fuel consumption. The function f_L is the length cost of the UAV, which can be calculated by equations (16) and (17).

$$f_L = \sum_{i=1}^{M} \sum_{k=1}^{N} l_{i,k} \tag{16}$$

$$l_{i,k} = \sqrt{\left(x_{i,k+1} - x_{i,k}\right)^2 + \left(y_{i,k+1} - y_{i,k}\right)^2 + \left(z_{i,k+1} - z_{i,k}\right)^2} \tag{17}$$

where, $l_{i,k}$ represents the distance cost of the ith UAV on the kth path segment, $\left(x_{i,k}, y_{i,k}, z_{i,k}\right)$ is the coordinate of the kth UAV at the kth point.

3.3.2. Altitude cost

A reasonable flight altitude can avoid ground obstacles and guarantee flight safety. The flight altitude cost f_H can be expressed by equations (18) and (19).

$$f_H = \sum_{i=1}^{M} \sum_{k=1}^{N} u_{i,k} \tag{18}$$

$$u_{i,k} = \begin{cases} z_{i,k} - h_{\max}, if \ \ z_{i,k} > h_{\max} \\ h_{\min} - z_{i,k}, if \ \ z_{i,k} < h_{\min} \\ 0, otherwise \end{cases} \tag{19}$$

where, u is the penalty function for the flight altitude, $h_{\max}$ and $h_{\min}$ are the maximum and minimum altitude for UAV flight, respectively.

3.3.3. Threat cost

This paper mainly considers the threat cost from the terrain and the environmental obstacles, which is calculated by equations (20)–(22).

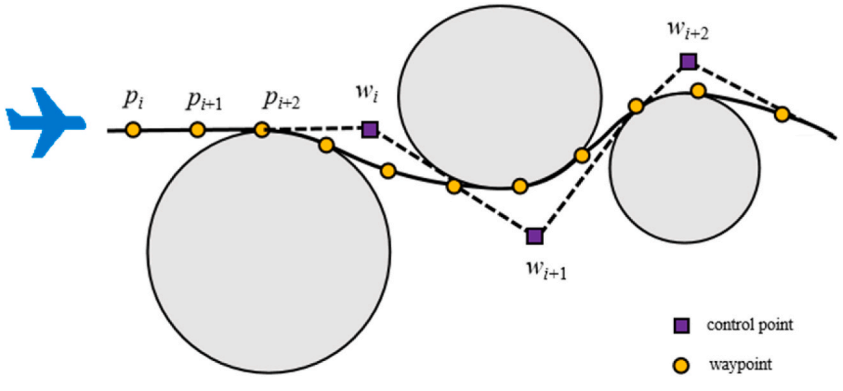


Fig. 1. Examples of the B-spline curve for smoothing paths.

$$f_O = \sum_{i=1}^{M}\sum_{k=1}^{N} s_{i,k}^{terrain} + \sum_{i=1}^{M}\sum_{k=1}^{N}\sum_{m=1}^{O} s_{i,k,m}^{obstacle} \tag{20}$$

with

$$s_{i,k}^{terrain} = \begin{cases} 0, if\ z_{i,k} - z_{\text{interp}}\left(x_{i,k}, y_{i,k}\right) < h_{safe} \\ 1, otherwise \end{cases} \tag{21}$$

$$s_{i,k,m}^{obstacle} = \begin{cases} 0, if\ \ d_{i,k} > d_s + R_m \\ \left|d_{safe} + R_m - d_{i,k,m}\right|, otherwise \end{cases} \tag{22}$$

where, $s_{i,k}^{terrain}$ and $s_{i,k,m}^{obstacle}$ are the penalty functions for the terrain and obstacle threat, $z_{i,k}$ is the fight height of the UAV to the ground, $z_{\text{interp}}\left(x_{i,k}, y_{i,k}\right)$ is the height of the ground at $\left(x_{i,k}, y_{i,k}\right)$, h_{safe} is the minimum safe fight height, $d_{i,k,m}$ is the distance from the ith UAV to the center of the mth obstacle, d_s is the danger zone radius, R_m is the radius of the mth obstacle.

3.3.4. Time cost

In multi-UAV path planning, the flight speeds and mission execution times of different UAVs need to be synchronized, which can ensure that each UAV successfully completes tasks within the corresponding time, hence improving overall efficiency. Assume that all UAVs start from different starting points at the same time. By setting a commanded arrival time t_C, the multiple UAVs can work together to complete the tasks on time. Based on the speed and flight length of the UAV, the actual time range t_r for the UAV to reach the endpoint can be determined, which is defined by equation (23).

$$\begin{aligned} t_r &= \left[t_{\min}^i, t_{\max}^i\right], \\ t_{\min}^i &= \frac{L_i}{v_{\max}^i}, \\ t_{\max}^i &= \frac{L_i}{v_{\min}^i} \end{aligned} \tag{23}$$

where, L_i represents the total length of the trajectory of the ith UAV, $v_{\max}^i$ and $v_{\min}^i$ are the maximum and minimum flight speeds of the UAV, respectively.

Then, the time cost f_T is expressed by equation (24).

$$f_T = \begin{cases} 0, if\ \ t_{\min}^i \le t_C \le t_{\max}^i \\ |t_i - t_C|, otherwise \end{cases} \tag{24}$$

where, t_i is the actual time for the ith UAV to reach the endpoint.

3.3.5. Turning angle constraint

The turning angle $\varphi_{i,k}$ of the ith UAV is the angle between two adjacent path segments, given by equation (25).

$$\varphi_{i,k} = \arctan\left(\frac{y_{i,k+2} - y_{i,k+1}}{x_{i,k+2} - x_{i,k+1}}\right) - \arctan\left(\frac{y_{i,k+1} - y_{i,k}}{x_{i,k+1} - x_{i,k}}\right) \tag{25}$$

The maximum turning angle $\varphi_{i,k}^{\max}$ is applied to obtain feasible flight paths. Then, the turning angle constraint C_1 is calculated by equations (26) and (27).

$$C_1 = \sum_{i=1}^{M}\sum_{k=1}^{N} q_{i,k} \tag{26}$$

with

$$q_{i,k} = \begin{cases} 0, if\ \ \left|\varphi_{i,k}\right| > \varphi_{i,k}^{\max} \\ 1, otherwise \end{cases} \tag{27}$$

where, $q_{i,k}$ is the penalty function for the turning angle $\varphi_{i,k}$.

3.3.6. Climbing slope constraint

The climbing angle $\theta_{i,k}$ of the ith UAV refers to the angle between two adjacent path segments onto the horizontal plane, which is calculated by equations (28) and (29).

$$\theta_{i,k} = \arctan\left(\frac{z_{i,k+2} - z_{i,k+1}}{l_{i,k+2,k+1}}\right) - \arctan\left(\frac{z_{i,k+1} - z_{i,k}}{l_{i,k+1,k}}\right) \tag{28}$$

with

$$l_{i,k+1,k} = \sqrt{\left(x_{i,k+1} - x_{i,k}\right)^2 + \left(y_{i,k+1} - y_{i,k}\right)^2} \tag{29}$$

The climbing slope constraint C_2 is expressed by equations (30) and (31).

$$C_2 = \sum_{i=1}^{M} \sum_{k=1}^{N} q_{i,k} \tag{30}$$

with

$$q_{i,k} = \begin{cases} 0, if \ \ |\theta_{i,k}| > \theta_{i,k}^{\max} \\ 1, otherwise \end{cases} \tag{31}$$

where, $q_{i,k}$ is the penalty function for the climbing angle $\theta_{i,k}$, $\theta_{i,k}^{\max}$ is the defined maximum climbing angle.

3.3.7. Flight distance constraint

There is a minimum straight-line distance limit for UAVs, which means that the UAV cannot change direction and turn frequently in the course. Setting the minimum straight-line flight distance constraint can reduce the UAV's flight time and fuel consumption. The flight distance constraint C_3 is calculated by equations (32) and (33).

$$C_3 = \sum_{i=1}^{M} \sum_{k=1}^{N} q_{i,k} \tag{32}$$

with

$$q_{i,k} = \begin{cases} 0, if \ \ l_{i,k} > l_{\min} \\ 1, otherwise \end{cases} \tag{33}$$

where, $q_{i,k}$ is the penalty function for the flight distance, $l_{i,k}$ represents the distance of the ith UAV on the kth path segment, $l_{\min}$ is the minimum straight flight distance.

3.3.8. Spatial collision constraint

The spatial collision constraint C_4 is to prevent airborne collisions between multiple UAVs and is defined by equations (34) and (35).

$$C_4 = \sum_{i=1}^{M} \sum_{k=1}^{N} q_{i,k} \tag{34}$$

with

$$q_{i,k} = \begin{cases} 0, if \ \ d_{i-j,k} > d_{us} \\ 1, otherwise \end{cases} \tag{35}$$

where, $q_{i,k}$ is the penalty function for the spatial collision, $d_{i-j,k}$ is the minimum distance between the trajectory points of the ith UAV and the jth UAV during flight, d_{us} is the minimum safe flight distance between UAVs.

In summary, the overall objective optimization model can be expressed by equations (36) and (37).

$$\begin{aligned} &\min \ \ F(x) = (f_1(x), f_2(x))^T \\ &s.t. \ \ CV(x) = \sum_{k=1}^{4} C_k = 0 \end{aligned} \tag{36}$$

with

$$\begin{aligned} f_1(x) &= f_L + f_H \\ f_2(x) &= f_O + f_T \end{aligned} \tag{37}$$

where, f_1 and f_2 are the two objective functions of the optimization, CV represents the constraint violation degree.

4. The proposed method

In this part, we first introduce the basic idea of the DENSDBO-ASR algorithm in Section 4.1 and then propose a novel non-dominated sorting dung beetle optimizer in Section 4.2. Then, in Section 4.3, a directional evolutionary strategy is introduced to improve the search performance of the algorithm. In Section 4.4, an adaptive stochastic ranking mechanism is designed to handle multiple constraints. Then we analyze the settings and influence of the parameters of DENSDBO-ASR in Section 4.5. Section 4.6 provides the steps for DENSDBO-ASR. The time complexity is analyzed in Section 4.7. Finally, the UAV path planning based on DENSDBO-ASR is introduced in Section 4.8.

4.1. Basic idea of DENSDBO-ASR

The objective functions of multi-objective optimization problems are frequently incompatible or not directly connected. Due to inherent conflicts between objectives, optimizing one objective is at the expense of the others. As a result, finding a unique optimal solution becomes difficult, necessitating a compromise between numerous solutions to achieve the overall optimization. In order to improve reliability and robustness, multi-objective algorithms seek a set of Pareto-optimal solutions compared to single-objective algorithms. In general, an algorithm's global and local search capabilities are two fundamental elements. The algorithm is expected to exhibit faster convergence in the early iterations and maintain diversity in the subsequent ones. In addition, path planning problems in the real world are often subject to various constraints, including spatial limitations, environmental constraints, and energy consumption constraints, making the problem of path planning for multi-UAVs even more challenging. Therefore, a directional evolutionary non-dominated sorting dung beetle optimizer combined with adaptive stochastic ranking (DENSDBO-ASR) is proposed to address these challenges in practical multi-UAV path planning scenarios.

4.2. Non-dominated sorting dung beetle optimizer (NSDBO)

The standard dung beetle optimizer (DBO) is originally designed for single-objective optimization tasks. Thus, it lacks the ability to handle multi-objective optimization problems, where the goal is to achieve a balance between multiple objective functions when searching for Pareto-optimal solutions. Therefore, we need to modify the DBO algorithm for multi-objective optimization. Non-dominated sorting is a highly popular and effective technique in the field of multi-objective optimization, as demonstrated by the success of NSGA-II. Inspired by this, we designed a non-dominated sorting dung beetle optimizer (NSDBO) using this powerful technique. The non-dominated sorting technique classifies solutions into different domination levels based on the comparison of their Pareto dominance relationship, ensuring that solutions within each level do not dominate each other. An illustration of non-dominated sorting is given in Fig. 2. Firstly, each individual p has two parameters, n_p and s_p, where n_p is the number of solutions that dominate p, and s_p is the set of solutions that are dominated by p. Secondly, find all individuals in the population with $n_p = 0$, give them a domination level of 1, and store them in Pareto rank 1. Thirdly, for the individuals in the Pareto rank 1, visit the solutions in the s_p of each individual and decrease the n_p value of each solution by one. Then, store the solutions with $n_p = 0$ into Pareto rank 2. Finally, repeat until all individuals have been assigned a rank.

Basic working of NSDBO is as follows:

Step 1: Initialize the population. Randomly generate dung beetles' positions and calculate the fitness of each dung beetle. Assuming that the population size is N.

Step 2: Sort out the initial population. Each dung beetle is filled in ascending order of the dominance level by using the non-dominated sorting method. The crowding distance can be used to distinguish the optimal degree of solutions in the same rank. The larger the crowding distance, the better the solution. The crowding distance is calculated by equation (38).

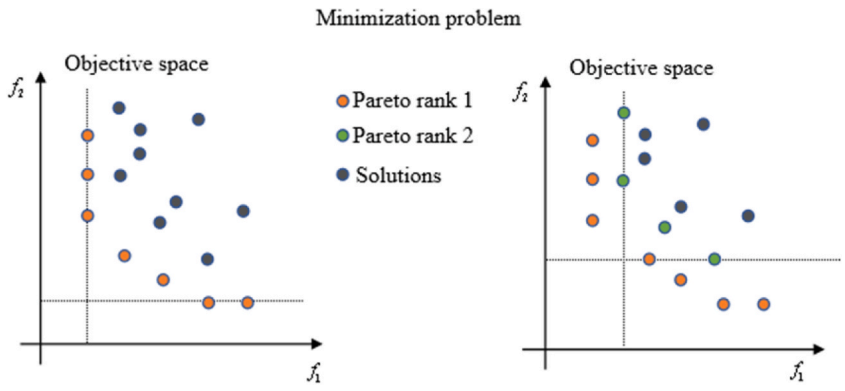


Fig. 2. Illustration of non-dominated sorting.

$$CD_i = \sum_{k=1}^{m} \left(\left| f_k^{i+1} - f_k^{i-1} \right| \right) \tag{38}$$

where, f_k^{i+1} and f_k^{i-1} are the neighbors of individual i on the kth objective function.

Then, the best solution is selected and stored in a collection set. Note that the size of the collection set is equal to the population size.

Step 3: Update the position of each dung beetle, and then calculate the new fitness.
Step 4: The initial population generated in **Step 1** is merged with the offspring generated in **Step 3** to generate a new population with a population size of 2*N*, and then non-dominated sorting is performed again.
Step 5: Then, select the best solutions and store them in the collection set until it reaches the predefined size. Lower-ranked individuals and those with higher crowding distances in the same rank are preferred to enter the next generation.
Step 6: Repeat **Steps 3** to **5** until the maximum number of iterations is reached, and output the Pareto-optimal set once the termination condition is met.

4.3. Directional evolutionary strategy

The NSDBO algorithm is an enhanced version of the original DBO algorithm. It is capable of addressing multi-objective optimization problems, but it also inherits the drawbacks of being susceptible to local optimums and having slow convergence speed. To address these issues and enhance the algorithm's search performance, a directional evolutionary technique is implemented. Specifically, the directional information of parents is used to guide the search, which can help maintain population diversity and improve convergence accuracy. Das et al. [43] proposed an improved real-coded genetic algorithm (IRGA) that primarily used directional mutation (DM) and directional crossover (DX) to enhance the search performance. The experimental results have shown that the algorithm outperforms the other five RGAs in terms of convergence speed and solution accuracy. Inspired by this, this paper introduces DX and DM into NSDBO to enhance its search and exploitation capabilities.

4.3.1. Directional crossover (DX)

The DX operator plays a crucial role in the DE strategy by utilizing the prior knowledge about the problem region that produces better results in space. This differs from traditional crossover operators, which ignore any prior information about the features of the search space. DX determines the crossover site by comparing the parent's position to the current optimal solution's position. During the search, the DX operator can adaptively adjust the crossover direction to explore the superior regions, thus avoiding the algorithm from converging to a local optimum. DX produces two new solutions by crossing the dimensions of two different individuals. The principle of DX is as follows:

The four key parameters in the DX are crossover probability (p_c), variable-wise crossover probability (p_{cv}), multiplying factor (α), and directional probability (p_d). Assuming that the jth dimensions of the parents p_1 and p_2 perform DX. p_{mean}^j is the mean value of the two parents, p_{best}^j is the value of the current optimal solution in the jth dimension. Then, the crossover update formula can be defined by equations (39) and (40).

$$val = \begin{cases} 1 - (0.5)^{e^{\left[\frac{|p_1^j - p_2^j|}{(Ub^j - Lb^j)}\right]}}, & if \ p_1^j \neq p_2^j \\ 1 - (0.5)^{e^{\left[\frac{|p_{best}^j - p_{mean}^j|}{(Ub^j - Lb^j)}\right]}}, & if \ p_1^j = p_2^j \ and \ p_{best}^j \neq p_{mean}^j \end{cases} \tag{39}$$

$$\beta = \frac{r}{\alpha^2} \tag{40}$$

where, val and β are two intermediate parameters. Ub^j and Lb^j represent the upper and lower bounds of the variable in the jth dimension, r is a random number between (0,1).

In the first case, the values of the two mating parents (p_1^j and p_2^j) are different. The children (c_1^j and c_2^j) can be determined by equations (41) and (42).

$$c_1^j = val \times \left(p_1^j + p_2^j \right) \pm \alpha^{r_1} \times e^{(1-\beta)} \times (1 - val) \times \left| p_1^j - p_2^j \right| \tag{41}$$

$$c_2^j = (1 - val) \times \left(p_1^j + p_2^j \right) \pm \alpha^{(1-r_1)} \times e^{(-\beta)} \times val \times \left| p_1^j - p_2^j \right| \tag{42}$$

In the second case, the values of the two parent individuals (p_1^j and p_2^j) are found to be the same and p_{mean}^j is not equal to p_{best}^j. The updated formula for the offspring is expressed by equations (43) and (44).

$$c_1^j = val \times \left(p_{best}^j + p_{mean}^j\right) \pm \alpha^{r_1} \times e^{(1-\beta)} \times (1 - val) \times \left(p_{best}^j - p_{mean}^j\right) \tag{43}$$

$$c_2^j = (1 - val) \times \left(p_{best}^j + p_{mean}^j\right) \pm \alpha^{(1-r_1)} \times e^{(-\beta)} \times val \times \left(p_{best}^j - p_{mean}^j\right) \tag{44}$$

where, the symbol $\pm$ in equations (41)–(44) is determined by p_d and $r_1 \in (0,1)$.

4.3.2. Directional mutation (DM)

The DM operator uses the optimization problem's directional information to guide the search to the optimal region. The DM achieves a balance between population diversity and selection pressure by adjusting the directional parameter. The mutation strength of the DM operator is determined by a random number, enabling it to efficiently balance the search performance and diversity in various conditions. This enhances the probability of finding the global optimal solution. DM randomly mutates the positions of individuals in any direction. Specifically, the DM adjusts the direction of the mutation according to the fitness of the parent solution and the current optimal solution. The description of DM is as follows:

Assuming that the nth dimension of the parent x_i performs DM. x_{best}^n is the value of the current best solution in the nth dimension. When $x_{best}^n \geq x_i^n$, the mutated solution (c_i^n) can be created by equations (45)–(47).

$$\beta_1 = e^{\left(2r - \frac{2}{r}\right)} \tag{45}$$

$$\beta_2 = e^{\left(r - \frac{2}{r}\right)} \tag{46}$$

$$c_i^n = \begin{cases} x_i^n + \beta_1 \times \left(Ub^n - x_i^n\right), if\ r_2 \leq p_d \\ x_i^n - \beta_2 \times \left(x_i^n - Lb^n\right), otherwise \end{cases} \tag{47}$$

When $x_{best}^n < x_i^n$, then the formula for the DM is as shown in equation (48).

$$c_i^n = \begin{cases} x_i^n - \beta_1 \times \left(x_i^n - Lb^n\right), if\ r_2 \leq p_d \\ x_i^n + \beta_2 \times \left(Ub^n - x_i^n\right), otherwise \end{cases} \tag{48}$$

where, r and r_2 are two different random numbers between 0 and 1. Ub^j and Lb^j represent the upper and lower limits of the variable in the nth dimension, respectively. p_d is the directional probability between 0.5 and 1.

4.4. Adaptive stochastic ranking mechanism

To effectively handle the numerous constraints in the UAV path planning model, an adaptive stochastic ranking approach is incorporated into the NSDBO algorithm. The strategy is integrated into the non-dominated sorting mechanism to adaptively adjust the sorting process based on the constraint violation degree and objective function value. The stochastic ranking method (SR) proposed by Runarsson is based on the probability *Pf* and a random number *r*. When comparing two individuals, the probability represented by *Pf* only compares their objective function values, while the probability represented by (1-*Pf*) compares the degree of constraint violation. As shown in Fig. 3, the SR is illustrated. *x* represents an individual, and *f* and *CV* are the objective value and constraint violation degree, respectively. When other individuals are compared with *x*, if they are located in the shaded region, it means they are superior to *x*.

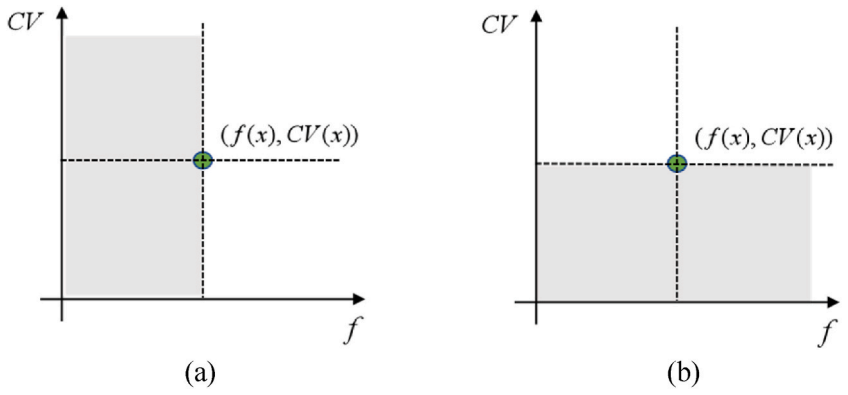


Fig. 3. Illustration of SR. (a) Compared by *f*; (b) Compared by *CV*.

Fig. 3 (a) represents the comparison using only the objective function values, while Fig. 3 (b) shows the comparison using the constraint violation degree.

Runarsson's experiment results indicated that *Pf*, taken as 0.475, had better optimization effects on most test functions. However, it is worth noting that a fixed *Pf* needs to be predetermined and may not always ensure optimal optimization results. For instance, Fig. 4 illustrates the number of infeasible solutions in the population after 300 iterations using the CF1 test function. It can be seen from Fig. 4 that there are still infeasible solutions in the final optimized population, which is because the stochastic ranking algorithm cannot adaptively adjust during iteration.

Therefore, this paper designs a *Pf* that can adaptively adjust according to the current iteration and the degree of constraint violation. In order to improve the efficiency of the search process, a large number of infeasible solutions are permitted during the early iterations. However, as the number of iterations increases, the algorithm needs to prioritize minimizing the violation of constraints to ensure the feasibility of the final solution. If too many infeasible solutions exist in the population, *Pf* is reduced to retain more feasible solutions. Based on the above consideration, the adaptive *Pf* is as shown in equation (49).

$$Pf = 0.475 * \left(\frac{0.5 * \cos\left(0.5 * \left(\frac{t}{T}\right) * 2 * \pi\right) + 0.5}{1 + \gamma * P_{ifea}} \right) \tag{49}$$

where, 0.475 is the optimal initial value determined by the SR method, t is the current number of iterations, T is the maximum number of iterations, γ is an influence factor between (0,1), and P_{ifea} is the proportion of infeasible solutions. The parameter settings for γ are detailed in Section 4.5.

Additionally, the CF1 test function is employed to evaluate the adaptive *Pf*. The maximum number of iterations is 300. The results are shown in Fig. 5. From Fig. 5, it can be seen that as the number of iterations increases, the number of infeasible solutions decreases. After about 220 iterations, the number of infeasible solutions in the population is 0, which indicates that the adaptive stochastic ranking has a good optimization performance.

4.5. Settings and influence of the parameters of DENSDBO-ASR

There are six user-defined parameters in the proposed algorithm: crossover probability (p_c), variable-wise crossover probability (p_{cv}), mutation probability (p_m), multiplying factor (α), directional probability (p_d), and influence factor (γ).

Crossover probability (p_c): Theoretically, the value of p_c varies from 0 to 1. Generally, the selection of the value is based on the nature of the problem. In this paper, p_c takes the same value presented in Reference [44]. DAS has demonstrated that the performance of DX is robust when $p_c = 0.9$.

Variable-wise crossover probability (p_{cv}): p_{cv} is used to determine whether a crossover occurs at a variable location. Similarly to p_c, the range of values varies between 0 and 1. The value of p_{cv} is kept the same with p_c.

Mutation probability (p_m): Theoretically, p_m varies from 0 to 1. However, the value of p_m is usually set to a minimal value (close to 0.1) to ensure an efficient search.

Multiplying factor (α): α affects the distance between offspring and parents. As the value of α increases, offspring are created far away from their parents, thus increasing the diversity of the algorithm. On the contrary, a smaller value of α will enhance the searching ability of the algorithm. Literature [43] suggests an optimal range of 0.5–2.5. Here, α = 0.95 yields stable optimization results, as evidenced by Qi [45] in image segmentation.

Directional probability (p_d): p_d decides children's search direction. Theoretically, the value ranges from 0 to 1, but is usually set

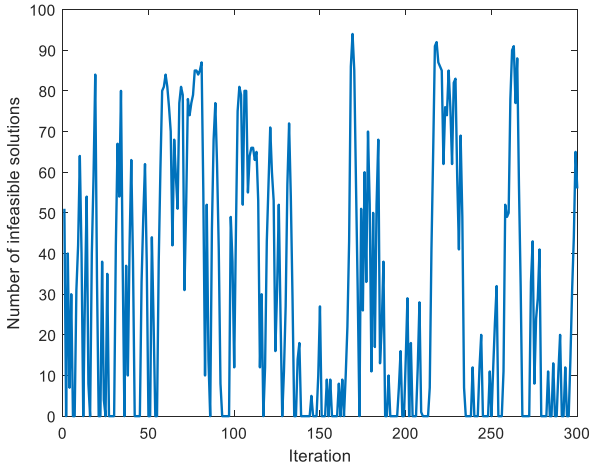


Fig. 4. The curve of the number of infeasible solutions with fixed *Pf*.

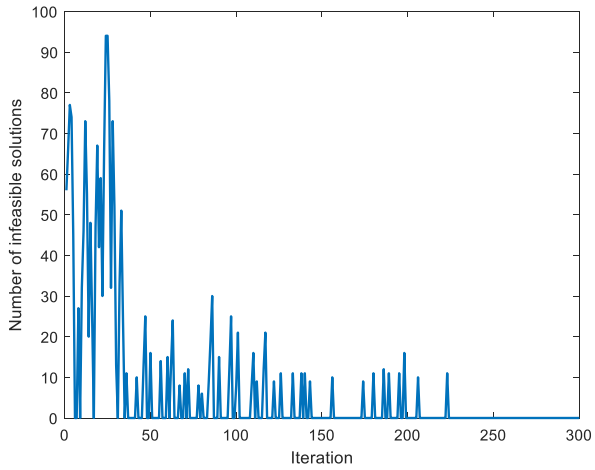


Fig. 5. The curve of the number of infeasible solutions with adaptive *Pf*.

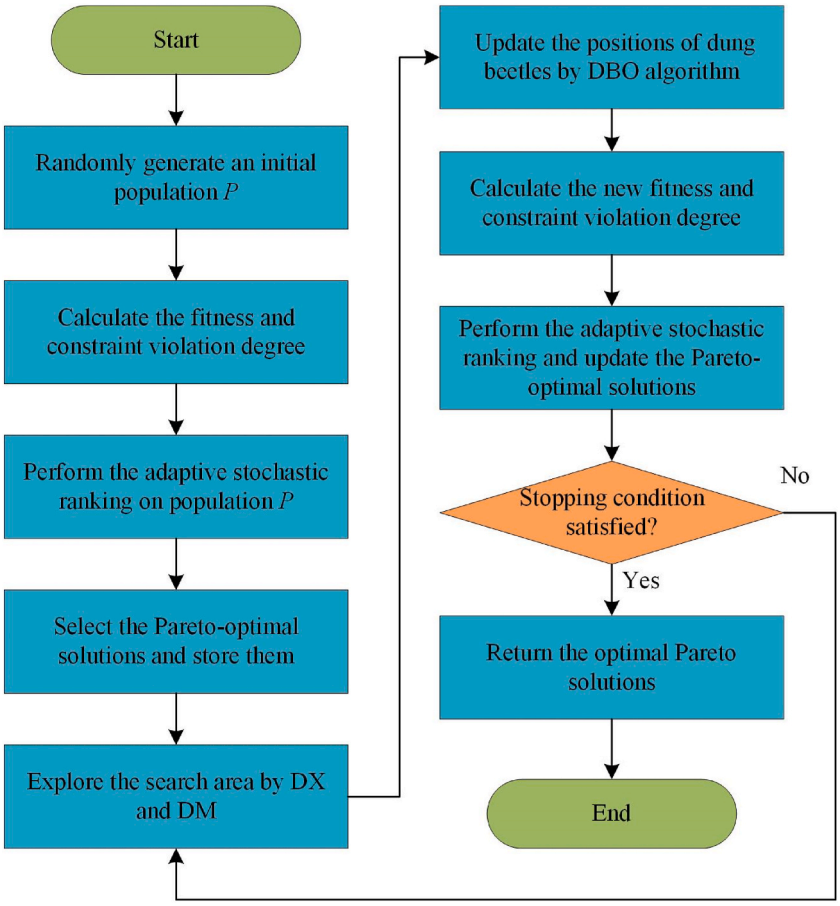


Fig. 6. Flowchart of DENSDBO-ASR

between 0.5 and 1. When p_d takes a higher value (close to 1), the algorithm will be guided toward the optimal solutions, increasing exploitation capability. On the contrary, when p_d takes a lower value (close to 0.5), the children will be generated in a random direction, enhancing their exploration abilities. The empirical determination of p_d is detailed in Section 5.2.

Influence factor (γ): γ controls infeasible solutions' impact on the population. A smaller γ implies lighter penalties for infeasible solutions, potentially retaining more infeasible solutions. And a larger γ eliminates infeasible solutions, preserving more ideal

solutions. Usually, it takes a value between 0 and 1. The exact value of γ is set in Section 5.2.

4.6. Steps of DENSDBO-ASR

The DENSDBO-ASR algorithm combines the directional evolutionary strategy (DE) and adaptive stochastic ranking mechanism (ASR) with the NSDBO algorithm. The following are the detailed steps:

Step 1: Initialize the positions of the dung beetles.
Step 2: Calculate the fitness and constraint violation degree of the population, then perform the adaptive stochastic ranking mechanism and select the Pareto-optimal solutions.
Step 3: Implement the directional evolutionary strategy and update the dung beetles' positions.
Step 4: Update the positions of the dung beetles using the DBO algorithm.
Step 5: Calculate the new fitness and constraint violation degrees and record them.
Step 6: Perform the adaptive stochastic ranking on the population and update the Pareto-optimal solutions.
Step 7: If the maximum number of iterations is reached, then output the Pareto-optimal solution; otherwise, skip to **Step 3** and continue to the next iteration.

The pseudo-code of DENSDBO-ASR is given below in Algorithm 1.

Algorithm 1: The pseudo-code of DENSDBO-ASR

```
Input: The number of the particle's population N, the maximum iterations T, and the dimensions of each particle D
Output: The Pareto-optimal set P.
Initialize the population P and calculate the fitness values
  Perform the adaptive stochastic ranking on each individual
  Select the Pareto-optimal solutions
  while t < T do
    Calculate Pf based on the current iterations and the proportion of infeasible solutions using Equation (49)
    for i = 1 to N do
      Perform DX on the ith individual using Equations (39)–(44)
      Perform DM on the ith individual using Equations (45)–(48)
      Update the position of ith dung beetle
      Calculate the fitness and the constraint violation degree of each individual
    end for
    Perform the adaptive stochastic ranking on the population
    Select N optimal solutions from the population as the next generation
  end while
```

The flowchart of DENSDBO-ASR is in Fig. 6

4.7. Computing complexity

This section analyzes the computing complexity of the proposed DENSDBO-ASR, including the initialization, directional evolutionary strategy, adaptive stochastic ranking, and updating the position of dung beetles. For convenience, we assume that the N individuals and T iterations are involved in the search process of a D-variable problem. M is the number of objectives. First, the time complexity of the initialization phase is O(N). Then, the time complexity of the directional evolutionary strategy is O(T*(N*D)). The time complexity of adaptive stochastic ranking is O(M*(N*N)). The time complexity of updating the dung beetles' position is O(T*N). The overall time complexity is O(DENSDBO-ASR) = O(N + T*N*(1+D) + M*N*N). The time complexity of the proposed algorithm is within acceptable performance improvements.

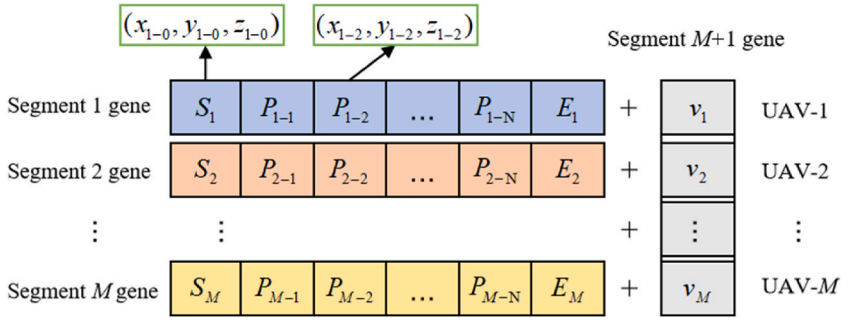


Fig. 7. Collaborative path chromosome encoding.

4.8. UAV path planning based on DENSDBO-ASR

In this section, the UAV path planning method based on the DENSDBO-ASR algorithm is introduced. The solutions need to be encoded into individuals for the population-based evolutionary algorithms. This paper uses a real-valued chromosome with variable-length coding to represent the cooperative trajectory of multiple UAVs. Assuming M UAVs on the mission, a chromosome consists of $(M+1)$ segments of genes. As presented in Fig. 7, an individual represents a planning path, including the starting point S, the trajectory points $P_{i-1}P_{i-2}...P_{i-N}$, and the target endpoint E. The first M genes indicate the flight paths of M UAVs, while the M+1st genes indicate the flight speed of M UAVs. Therefore, the dimension of a particle is $(3*M*N+M)$, where N is the number of track points for each UAV. The algorithm aims to find optimal candidates that satisfy the constraints during evolution, and then a decision-maker selects an optimal solution for UAVs. The specific path-planning steps are summarized as follows:

Step 1: Initialize the parameters of the multi-UAV collaborative path planning, including the number of UAVs, the starting and end points of each UAV, the number of way points, and the flight speeds of the UAVs.
Step 2: Initialize the algorithm's parameters, including the population size, the maximum number of iterations, and the dimension of each particle. Then, randomly generate N individuals within the defined search space range.
Step 3: Calculate the objective function values and constraint values for each individual. Then, implement the DENSDBO-ASR algorithm to obtain feasible solutions.
Step 4: Select the best candidate solution to generate UAV flight paths. A weighted linear approach is used, where an evaluation function determines the optimal solution. The smaller the function value, the better the path performance. Thus, the evaluation function f_{val} is calculated by $f_{val} = w_1 f_1 + w_2 f_2$, where w_1 and w_2 are the weights of the two objective functions, respectively. The final optimal solution for collaborative UAV path planning can be selected by $\min(f_{val})$. Based on the previous descriptions, the theoretical optimal value of f_2 is 0, while f_1 varies based on the start and end points of the flight. We prioritize the safety and effectiveness of performing tasks (hence, w_2 is higher at 0.9, while w_1 is 0.1).

5. Analysis of simulation experiments

5.1. Experimental design

To verify the effectiveness of DENSDBO-ASR, this paper conducted tests on the CF benchmark problems [46]. The comparison algorithms include three variations of the NSDBO algorithm: NSDBO with the adaptive stochastic ranking mechanism (NSDBO-ASR), NSDBO with the constrained dominance principle (NSDBO-CDP), NSDBO with the stochastic ranking (NSDBO-SR), and three variations of the NSGA-II algorithm: NSGAII with the constrained dominance principle (NSGAII-CDP) [47], NSGAII with the stochastic ranking (NSGAII-SR) [47], and NSGAII with a new fitness function with two rankings (NSGAII-ToR) [48].The population size is 100, and the maximum function evaluation is 300. For fairness, each benchmark function is run independently 30 times. Furthermore, the parameter settings of all algorithms are specified in Table 2. The inverse generation distance (IGD) is the statistical metric to assess the performance of algorithms.

5.2. Sensitivity analysis of parameters

This section analyzes the sensitivity of the control parameters (p_d, γ) used in the DENSDBO-ASR algorithm. For comparison, we separately evaluate the optimal values of γ and p_d. We choose nine different values of γ (0.1, 0.2, …, 0.9) and select CF1, CF2, CF4, and CF6 as our test cases. The IGD results are shown in Fig. 8. It can be seen that the best performance is achieved for CF2 and CF4 in Fig. 8 (b) and (c) when the value of γ is set to 0.3. Meanwhile, for CF1 and CF6 in Fig. 8 (a) and (d), $\gamma = 0.3$ obtains the second and third positions, respectively. Thus, $\gamma = 0.3$ is selected as the suggested option.

Subsequently, with a fixed value of $\gamma = 0.3$, six different parameters ($p_d = 0.5, 0.6, \ldots, 1$) are chosen for testing. The sensitivity analysis is shown in Fig. 9. Notably, a value of 0.5 for p_d yields the optimal results for CF1 in Fig. 9 (a). For CF4 and CF6 in Fig. 9 (c) and (d), $p_d = 0.5$ obtains second place after $p_d = 0.6$. For CF2 in Fig. 9 (b), $p_d = 0.6$ significantly deteriorates the performance, while $p_d = 0.5$ showcases robustness. Therefore, the optimal parameter for p_d in this paper is 0.5.

Table 1
The IGD results obtained by different improvement methods.

Prob.	Indexes	DENSDBO-ASR	NSDBO	DXNSDBO	DMNSDBO	DENSDBO	NSDBO-ASR
CF2	Avg (±Std)	**6.59E-02** (±1.54E-02)	1.13E-01 (±2.21E-02)	6.88E-02 (±1.49E-02)	9.51E-02 (±1.56E-02)	7.43E-02 (±**1.35E-02**)	1.04E-01 (±1.42E-02)
CF5	Avg (±Std)	**6.02E-01** (±2.72E-01)	2.45E+00 (±1.01E+00)	6.08E-01 (±2.29E-01)	2.29E+00 (±7.76E-01)	6.15E-01 (±**2.26E-01**)	2.31E+00 (±8.60E-01)
CF7	Avg (±Std)	**5.15E-01** (±**2.55E-01**)	2.99E+00 (±1.03E+00)	5.94E-01 (±3.24E-01)	2.42E+00 (±8.78E-01)	6.20E-01 (±2.92E-01)	2.90E+00 (±1.08E+00)

Table 2
Parameter settings of all algorithms.

Algorithm	Parameters
DENSDBO-ASR	$p_c = p_{cv} = 0.9$, $p_m = 0.1$, $p_d = 0.5$, $\alpha = 0.95$, $\gamma = 0.3$
NSDBO-ASR	$\gamma = 0.3$
NSDBO-CDP	/
NSDBO-SR	$Pf = 0.475$
NSGAII-CDP	$p_c = 0.9$, $p_m = 0.1$, $mu = 20$, $mum = 20$
NSGAII-SR	$p_c = 0.9$, $p_m = 0.1$, $mu = 20$, $mum = 20$, $Pf = 0.475$
NSGAII-ToR	$p_c = 0.9$, $p_m = 0.1$, $mu = 20$, $mum = 20$

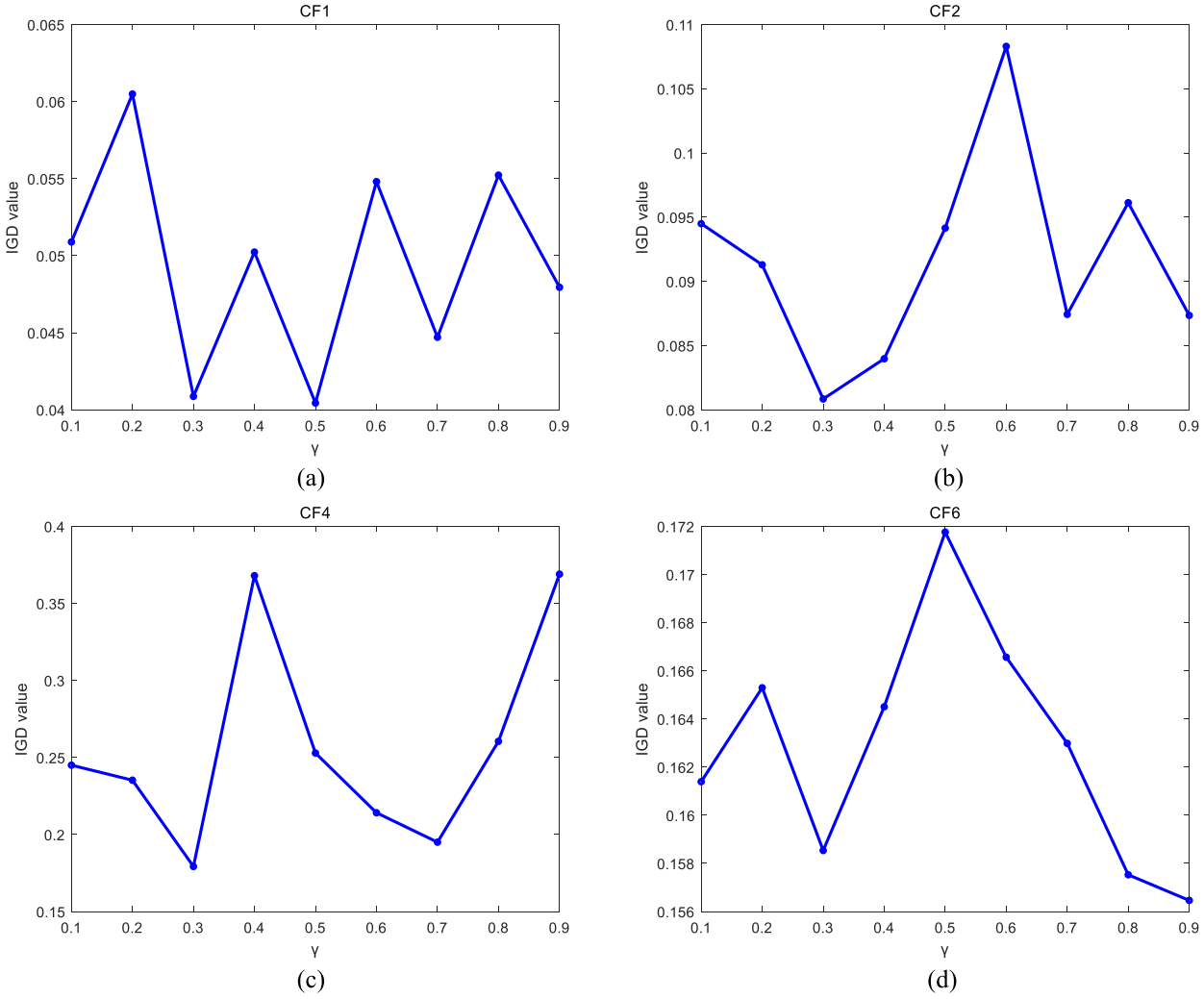


Fig. 8. Sensitivity analysis of the parameter γ on CF1, CF2, CF4, and CF6. (a) CF1; (b) CF2; (c) CF4; (d) CF6.

5.3. Ablation experiments analysis

To verify the effectiveness of the improved method, the ablation experiments were conducted. The comparison algorithms consist of the basic NSDBO and several variations of the NSDBO algorithm, including NSDBO with the directional crossover (DXNSDBO), NSDBO with the directional mutation (DMNSDBO), NSDBO with the directional crossover and mutation (DENSDBO), NSDBO with the adaptive stochastic ranking (NSDBO-ASR), and the proposed DENSDBO-ASR. The test functions are selected as CF2, CF5, and CF6. The average value (Avg) and standard deviation (Std) are the statistical metrics for comparison. To ensure fairness, all algorithms are run independently 30 times. The results obtained by different improvement methods are shown in Table 1.

From Table 1, it can be seen that DENSDBO-ASR achieves the first rank on all functions. Then, the methods using different

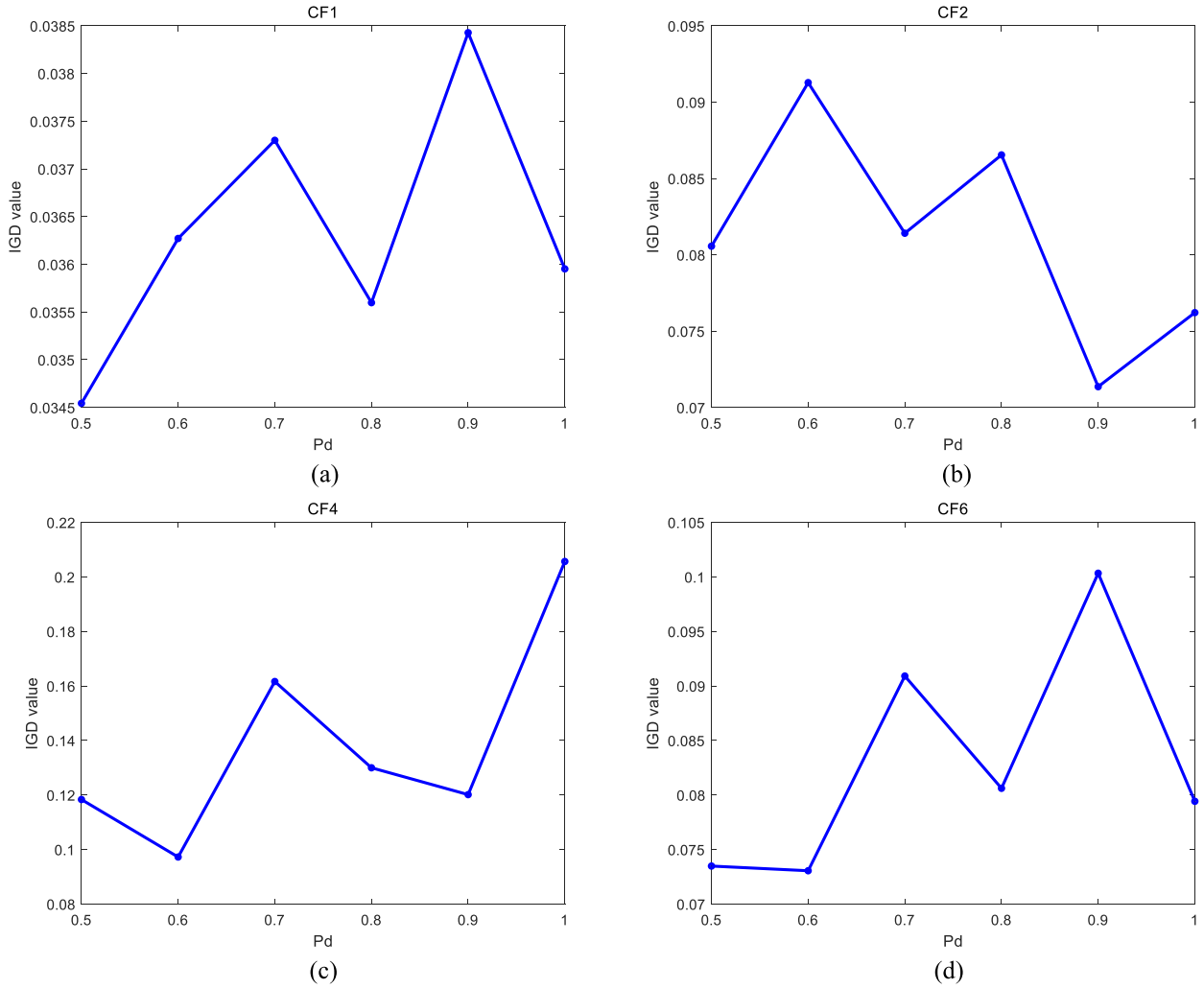


Fig. 9. Sensitivity analysis of the parameter p_d on CF1, CF2, CF4, and CF6. (a) CF1; (b) CF2; (c) CF4; (d) CF6.

enhancement strategies have improved performance compared to NSDBO. From the comparison of DXNSDBO, DMNSDBO, and DENSDBO, it is evident that the DXNSDBO obtains the lowest average value and demonstrates superior optimization. On the other hand, DENSDBO has the lowest standard deviation and demonstrates stability. The reason for this is that the DX tends to preserve the beneficial features of the parents during the crossover process, allowing the algorithm to perform a refined search in the area that yields the optimal solution. The DM randomly introduces new features to enhance the diversity of the population. DENSDBO combines the DX and DM strategies, leveraging the benefits of both exploration and exploitation to improve stability. The comparison between NSDBO and NSDBO-ASR reveals that the adaptive stochastic ranking mechanism enhances the accuracy of NSDBO, thereby proving its effectiveness. To summarize, all the improvement methods have proven to be effective. However, the DENSDBO-ASR algorithm, improved by hybrid strategies, has the most superior performance.

5.4. Benchmark problem test analysis

5.4.1. Description of the benchmark suite

To comprehensively evaluate the performance of DENSDBO-ASR, the CF test set [46] was employed. The CF test set has two objective functions, CF1-5 with one constraint and CF6-7 with two constraints, each of which includes ten variables. The Pareto front of CF1 in the objective space comprises *N* point sets. These point sets represent all possible solutions, where each solution achieves a balance between the two objective functions. CF1 is mainly used to evaluate the algorithm's exploration and exploitation capabilities. Unlike CF1, the Pareto fronts of CF2 and CF3 have separate *N* parts in the objective space, which means that there are more than one set of separate Pareto optimal solutions. This presents a significant challenge for multi-objective optimization algorithms, which must locate optimal solutions within these discontinuous regions. CF2 and CF3 are typically used to evaluate the algorithms' robustness. The Pareto fronts of CF4 and CF5 exhibit continuously varying linear behavior and are primarily used to assess the algorithms' convergence

speed. The CF6 and CF7 test functions are more complex because they narrow down the decision space and increase the number of constraints. The CF6 and CF7 have nonlinear PFs and are mainly used to assess the algorithm's ability to solve complicated problems. Overall, the CF test function set is designed to evaluate the performance of multi-objective optimization methods over an extensive variety of problems. The CF test set enables a complete evaluation and comparison of the algorithms in terms of convergence speed, solution quality, and search capabilities, thereby improving the algorithms' applicability and efficiency in handling real-world issues.

5.4.2. Benchmark suite test result analysis

This subsection provides the statistical test results of the comparison algorithms in the CF test suite. Table 3 presents the average value (Avg) and standard deviation (Std) of the IGD results of seven algorithms on seven benchmark problems. The best results of each benchmark function are marked in bold. From the comparison of NSDBO-ASR, NSDBO-CDP, and NSDBO-SR, it can be seen that NSDBO-ASR performs the best on CF1, CF2, CF5, and CF6, and obtains second place on CF3 and CF4. This proves that the adaptive stochastic ranking strategy can effectively enhance global search capability and improve convergence accuracy. In addition, it can be observed that the DENSDBO-ASR algorithm achieves the best results on all optimized functions. It is worth noting that the accuracy of DENSDBO-ASR on CF3, CF5, CF6, and CF7 is an order of magnitude higher than other algorithms. This suggests that the DE strategy further enhances the ability to jump out of the local optimum and improves the quality of the solutions. Overall, DENSDBO-ASR demonstrates effectiveness and superiority on the test problems.

5.4.3. Convergence curve analysis

In this subsection, the convergence curves of all the algorithms are plotted to compare their efficiency in the search process. Fig. 10 shows the convergence curves of IGD values for all algorithms on CF2, CF3, CF5, and CF7. It is evident that the IGD values of all the algorithms decrease with the increase in iterations, indicating their effective search capabilities during the optimization process. Since the problems are minimization problems, smaller IGD values signify superior performance. From Fig. 10(a–d), it is evident that the DENSDBO-ASR algorithm exhibits the most rapid convergence speed and the best convergence accuracy compared to other optimization algorithms. This demonstrates that the improvement strategies can significantly enhance the convergence performance of the algorithm, particularly the DE strategy, which greatly boosts the algorithm's ability to escape local optima. In general, DENSDBO-ASR stands out as the strongest performer among all the methods, consistently achieving the global optimum at a faster rate than its counterparts.

5.4.4. Box plot analysis

In this subsection, the box line charts of 30 independent runs are used to compare the distribution characteristics of the solutions, as shown in Fig. 11. From Fig. 11(a–d), it is evident that DENSDBO-ASR ranks first in the median metric when compared to its competitors. This suggests that the DENSDBO-ASR algorithm displays the capability to identify a solution set that is in proximity to the optimal solution and exhibits robustness and stability in iterations. It can also be seen that the DENSDBO-ASR algorithm wins in both maximum and lowest metrics, demonstrating its impressive search capability. Furthermore, the height of the box shows the level of data discretization, which is used to evaluate the algorithm's exploration and exploitation capabilities. As shown in Fig. 11(a–d), it is evident that the box height of the DENSDBO-ASR algorithm is the smallest of all the functions. The data distribution of DENSDBO-ASR is highly concentrated, suggesting that the algorithm has strong exploration and development capability and superior optimization results. In summary, DENSDBO-ASR outperforms other algorithms in terms of optimization ability and stability when solving global optimization issues.

5.4.5. Wilcoxon rank sum test

To further test the effectiveness of the DENSDBO-ASR, the Wilcoxon rank sum test [49] is used to evaluate and discover statistical differences between the DENSDBO-ASR and other algorithms. Table 4 shows the p-values of the Wilcoxon rank sum test at the $\alpha = 5\ \%$

Table 3
The IGD results obtained by seven methods on CF test sets with two objectives.

Prob.	Indexes	DENSDBO-ASR	NSDBO-ASR	NSDBO-CDP	NSDBO-SR	NSGAII-CDP	NSGAII-SR	NSGAII-ToR
CF1	Avg	**3.35E-02**	5.09E-02	5.60E-02	5.19E-02	4.69E-02	4.18E-02	7.43E-02
	(±Std)	(±8.95E-03)	(±1.09E-02)	(±7.45E-03)	(±1.85E-02)	(±6.83E-03)	(±2.26E-02)	(±6.23E-03)
CF2	Avg	**6.42E-02**	9.52E-02	9.83E-02	1.01E-01	1.30E-01	1.34E-01	1.57E-01
	(±Std)	(±1.34E-02)	(±1.34E-02)	(±1.88E-02)	(±2.48E-02)	(±2.41E-02)	(±1.81E-02)	(±3.08E-02)
CF3	Avg	**4.58E-01**	1.22E+00	1.17E+00	1.28E+00	1.26E+00	1.38E+00	1.54E+00
	(±Std)	(±1.88E-01)	(±2.75E-01)	(±2.77E-01)	(±3.34E-01)	(±3.82E-01)	(±2.89E-01)	(±3.49E-01)
CF4	Avg	**1.19E-01**	2.45E-01	2.51E-01	2.42E-01	3.71E-01	3.06E-01	2.75E-01
	(±Std)	(±1.73E-02)	(±8.01E-02)	(±1.01E-01)	(±6.88E-02)	(±1.58E-01)	(±1.58E-01)	(±9.88E-02)
CF5	Avg	**5.01E-01**	2.01E+00	2.11E+00	2.07E+00	3.19E+00	3.12E+00	3.73E+00
	(±Std)	(±1.99E-01)	(±8.64E-01)	(±6.20E-01)	(±7.16E-01)	(±7.58E-01)	(±6.36E-01)	(±9.05E-01)
CF6	Avg	**8.97E-02**	1.51E-01	1.62E-01	1.70E-01	1.73E-01	1.64E-01	2.50E-01
	(±Std)	(±1.73E-02)	(±5.14E-02)	(±3.86E-02)	(±5.20E-02)	(±4.89E-02)	(±4.02E-02)	(±5.58E-02)
CF7	Avg	**6.02E-01**	3.02E+00	2.68E+00	2.85E+00	3.67E+00	3.91E+00	3.43E+00
	(±Std)	(±2.50E-01)	(±1.30E+00)	(±1.26E+00)	(±8.86E-01)	(±1.24E+00)	(±1.25E+00)	(±1.03E+00)

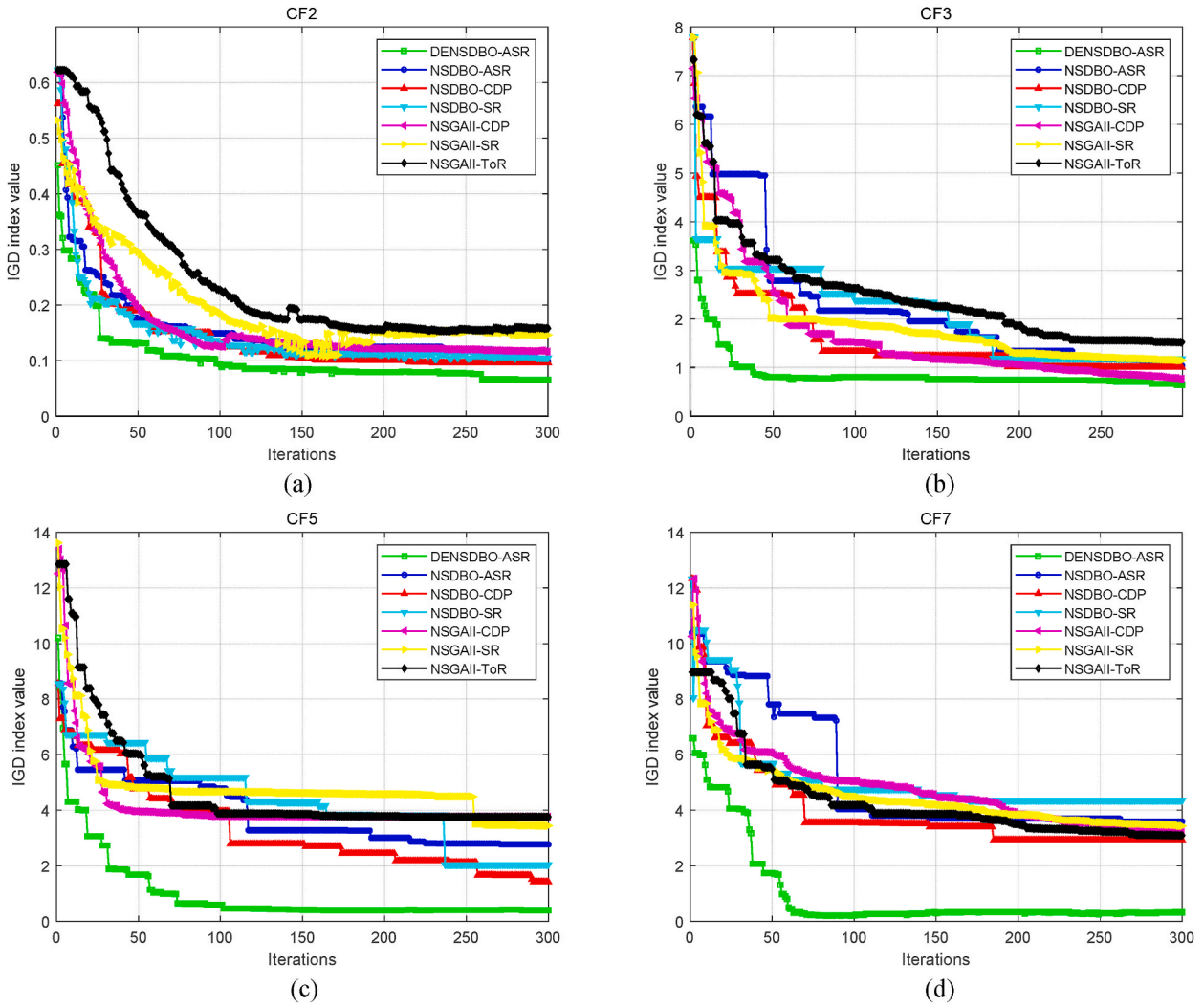


Fig. 10. The convergence curves of IGD values on CF2, 3, 5, and 7. (a) CF2; (b) CF3; (c) CF5; (d) CF7.

significance level. If $p < 0.05$, the original hypothesis is rejected and the alternative hypothesis is accepted. In Table 3, "+/−/ = " indicates the number of DENSDBO-ASR with better/worse/comparable performance compared with other algorithms, respectively. From Table 4, it can be seen that almost all the p-values are less than 0.05, which indicates that there is a significant difference between DENSDBO-ASR and other optimization algorithms. Therefore, DENSDBO-ASR has remarkable performance compared to other algorithms.

5.4.6. Friedman test

To see the overall rank of each algorithm, the Friedman mean rank test [50] was performed. Table 5 illustrates the results of Friedman's test. From Table 5, it is clear that DENSDBO-ASR ranks first in evaluating all algorithms. It can also be seen that the algorithms based on NSDBO are ranked at the front, while the algorithms based on NSGA-II are ranked at the back. This demonstrates the effectiveness and superiority of the developed NSDBO algorithm.

Fig. 12 presents a comparison of the statistical results of the selected algorithms on all test functions. From Fig. 12, it is evident that the DENSDBO-ASR algorithm is ranked first on all test functions. It can also be seen that NSDBO-ASR ranks second after DENSDBO-ASR. It demonstrates that the DE and ASR strategies can effectively enhance the quality of solutions. Overall, it can be confirmed that DENSDBO-ASR is statistically superior to all the comparison algorithms.

6. UAV path planning results and analysis

A series of experiments are conducted to evaluate the performance of the DENSDBO-ASR algorithm for collaborative multi-UAV path planning. Five representative state-of-art methods (the NSGAII-CDP [47], NSGAII-ToR [48], SPEA2-CDP [51], MOGWO [52],

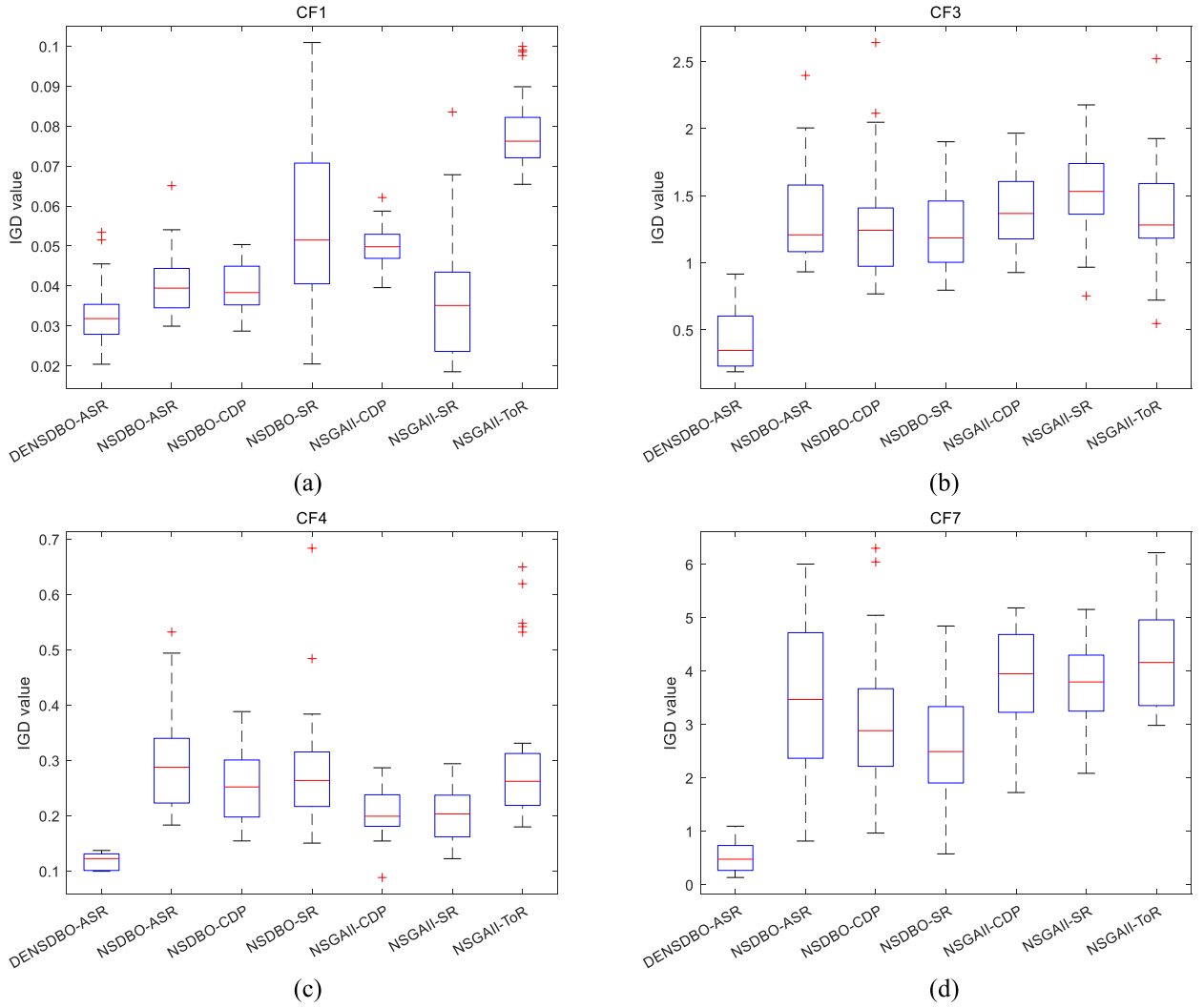


Fig. 11. Box plots of IGD values for CF1, 3, 4, and 7. (a) CF1; (b) CF3; (c) CF4; (d) CF7.

Table 4
Wilcoxon rank sum test results.

Fun No.	NSDBO-ASR	NSDBO-CDP	NSDBO-SR	NSGAII-CDP	NSGAII-SR	NSGAII-ToR
	p Value R	*p* Value R	*p* Value R	*p* Value R	*p* Value R	*p* Value R
CF1	1.89E-04	9.79E-05	4.42E-06	3.82E-09	4.29E-01	3.02E-11
CF2	5.53E-08	2.44E-09	9.26E-09	6.70E-11	2.87E-10	6.07E-11
CF3	3.02E-11	3.69E-11	4.50E-11	3.02E-11	3.69E-11	9.92E-11
CF4	3.02E-11	3.02E-11	3.02E-11	5.57E-10	1.29E-09	3.02E-11
CF5	2.37E-10	7.39E-11	1.09E-10	3.02E-11	3.02E-11	3.02E-11
CF6	7.39E-11	3.16E-10	4.50E-11	4.98E-11	3.02E-11	3.02E-11
CF7	3.69E-11	3.69E-11	1.21E-10	3.02E-11	3.02E-11	3.02E-11
+/−/ =	7/0/0	7/0/0	7/0/0	7/0/0	6/1/0	7/0/0

Table 5
Friedman test results.

	DENSDBO-ASR	NSDBO-ASR	NSDBO-CDP	NSDBO-SR	NSGAII-CDP	NSGAII-SR	NSGAII-ToR
Avg	1	2.8571	3.4286	3.8571	5.2857	5.1429	6.4286
Rank	1	2	3	4	6	5	7

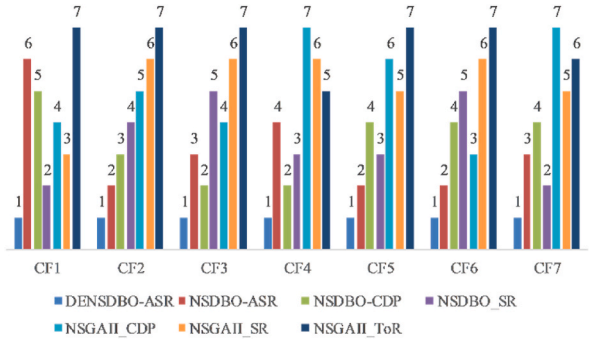


Fig. 12. The statistic results of the seven algorithms for all test functions.

and MAOA [53]) are used for the comparison. These algorithms cover both recently proposed algorithms, such as MAOA, MOGWO, and NSGAII-ToR, and the most utilized optimizers in the field, such as NSGAII-CDP and SPEA2-CDP. For fairness, all algorithms have the same population size and the maximum number of iterations. Each algorithm runs five times independently, and the results are compared using the mean value.

6.1. Scenario settings

This paper uses simulated digital terrain to model the UAV flight environment. The original digital terrain is modeled as shown in equation (50).

$$z_1(x,y) = \sin(y+a) + b\cdot\sin(x) + c\cdot\cos\left(d\cdot\sqrt{x^2+y^2}\right) + e\cdot\cos(y) + f\cdot\sin\left(g\cdot\sqrt{x^2+y^2}\right) \tag{50}$$

where, x and y are the coordinates of the points projected on the horizontal plane, and z is the corresponding terrain height; a, b, c, d, e, f, g are the constant coefficients, which control the undulation of the terrain in the digital map.

For higher natural mountains in the flight environment, the exponential function is used to describe the mathematical model, which can be expressed by equation (51).

$$z_2(x,y) = \sum_{i=1}^{n} h_i \exp\left[-\left(\frac{x-x_i}{x_{si}}\right)^2 - \left(\frac{y-y_i}{y_{si}}\right)^2\right] \tag{51}$$

where, $z_2(x,y)$ is the corresponding terrain height in the map, (x_i, y_i) is the center coordinate of the ith peak, h_i is the terrain parameter to control the height, x_{si} and y_{si} are the attenuation along the x-axis and y-axis direction of the ith peak to control the slope, respectively; n denotes the total number of peaks.

Then, the environment map is obtained by merging the original digital terrain with the peak terrain as shown in equation (52).

$$z(x,y) = \max(z_1(x,y), z_2(x,y)) \tag{52}$$

The terrain coefficients are set to $a = 10$, $b = 0.2$, $c = 0.1$, $d = 0.6$, $e = 1$, $f = 0.1$, and $g = 0.1$. The other main parameters in the environment model are given in Table 6.

In addition, threats such as radar and forbidden flying areas can affect the UAV's flight. In our work, the number and position of the threats are represented by red cylinders, with the specific parameters shown in Table 7.

In the simulation, the planned environment region is set to 200 km*200 km*6 km. Suppose there are three sets of mission scenarios, and the number of cooperative UAVs in different mission scenarios is 2, 3, or 4. The UAV parameter settings are presented in Table 8. The corresponding starting point and endpoint in the three cases for different numbers of UAVs are given in Table 9.

Table 6
Parameter settings for the peak dataset.

No. of peaks	(xi,yi) (km)	(xsi,ysi) (km)	H (km)
1	(40,70)	(9,7)	2.4
2	(70,100)	(18,15)	3.5
3	(160,90)	(11,15)	3.3
4	(110,160)	(15,13)	4.9
5	(150,40)	(14,7)	5.4
6	(110,180)	(15,7)	3.9
7	(160,150)	(12,10)	2.9

6.2. Evaluation metrics

When comparing the effectiveness of three-dimensional path planning methods for multiple UAVs, various metrics can be used to evaluate the resulting Pareto front. Two such metrics are the pure diversity (PD) [54] and the spacing (SP) [55], as represented by equations (53) and (54). The PD metric measures the diversity among solutions in the Pareto front, indicating whether the distribution of solutions is uniform and covers the entire front. In UAV path planning, diversity ensures that the algorithm's solutions cover different path choices, thereby increasing the flexibility and adaptability of path planning. The SP measures the distance among solutions, ensuring that the distance between solutions is sufficient to avoid excessive concentration. In UAV path planning, spacing ensures that generated paths are independent of each other, avoiding conflicts and collisions and enhancing the overall safety and reliability of path planning. It is expected that PD is larger while S is smaller.

$$\begin{aligned} PD(S) &= \max_{x^i \in S}\left(PD\left(S - x^i\right) + d\left(x^i, S - x^i\right)\right) \\ d(x, S) &= \min_{x^i \in S}\left(dissimilarity\left(x, x^i\right)\right) \end{aligned} \tag{53}$$

$$\begin{aligned} SP(S) &= \sqrt{\sum_{i=1}^{|S|} \frac{(d_i - \overline{d})}{|S| - 1}} \\ d_i &= \min_{x^j \in S, x^i \neq x^j}\left(\sum_{k=1}^{m} \left|PF_k\left(x^i\right) - PF_k\left(x^j\right)\right|\right) \end{aligned} \tag{54}$$

where, $d(x, S)$ denotes the dissimilarity from x to a community S. PF_k is the Pareto front obtained by the algorithm, and d_i is the Euclidean distance between the ith individual and the nearest individual to individual i in PF.

6.3. Simulated experimental results

Three experimental scenarios are considered with 2, 3, and 4 UAVs. The search space increases according to the number of UAVs. To fully explore the search space, it is necessary to set different population sizes and maximum iterations [20]. Considering the computational cost, for all the multi-objective methods, the population size is set to 80 for 2 UAVs, 120 for 3 UAVs, and 160 for 4 UAVs. The maximum number of iterations is 300, and the commanded arrival time $t_C = 1200$s. The 3-D and top-view results of the paths planned by all algorithms are shown in Figs. 13–15. It can be seen from Fig. 13(a–f) that the paths in regions C_{u2_1}, C_{u2_2}, C_{u2_3}, C_{u2_4}, and C_{u2_5} for SPEA2-CDP, NSGAII-ToR, and NSGAII-CDP collided with obstacles. The paths in regions R_{u2_2} and R_{u2_3} for MAOA and MOGWO are worse than the paths in R_{u2_1} for DENSDBO-ASR, and there is a risk of collision. Fig. 14(a–f) shows the paths of three UAVs for collaborative planning. It can be found that there are apparent collisions in regions C_{u3_1}, C_{u3_2}, C_{u3_3}, C_{u3_4}, C_{u3_5}, C_{u3_6}, C_{u3_7}, and C_{u3_8} with the SPEA2-CDP, NSGAII-ToR, NSGAII-CDP, MAOA, and MOGWO algorithms. In addition, from the comparison of the regions of R_{u3_1}, R_{u3_2}, and R_{u3_3}, the DENSDBO-ASR algorithm obtains better paths than MAOA and MOGWO, with the lowest risk.

Fig. 15(a–f) gives the results of four UAVs collaborating on the mission. It can be observed that the SPEA2-CDP, NSGAII-ToR, and MAOA algorithms have obvious collisions in regions C_{u4_1}, C_{u4_2}, C_{u4_3}, C_{u4_4}, C_{u4_5}, C_{u4_6}, and C_{u4_7}. In addition, from the comparison of the areas of R_{u4_1}, R_{u4_2}, R_{u4_3}, and R_{u4_4}, it is apparent that DENSDBO-ASR obtains preferable paths with the shortest path length and lowest risk. Therefore, the DENSDBO-ASR algorithm exhibits its superiority by generating better-performing paths in all scenarios. In addition, the Pareto fronts obtained by the DENSDBO-ASR and the SPEA2-CDP, NSGAII-ToR, NSGAII-CDP, MAOA, and MOGWO are presented in Fig. 16(a–c). It can be seen that the two methods of MOGWO and MAOA are comparable, while DENSDBO-ASR achieves a balance in optimizing the two objective functions and obtains a higher quality Pareto front.

Table 10 lists the statistical results, including HV, SP, the optimal value of objective function 1 (OFV1), and the optimal value of objective function 2 (OFV2). It can be found that the DENSDBO-ASR algorithm shows superior performance in all metrics in the scenario of two UAVs. In addition, in the collaborative scenarios of three and four UAVs, the DENSDBO-ASR algorithm achieves the first rank in PD, OFV1, and OFV2 metrics. For the SP index, the DENSDBO-ASR algorithm achieves second and third place in three and four UAV scenarios. The proposed method comprehensively considers various practical constraints, utilizes exploration and development capabilities, and effectively improves the optimization ability of DENSDBO-ASR in multimodal objective space. Therefore, better PD, S, OFV1, and OFV2 values can be obtained in three-dimensional multi-UAV collaborative path planning. In summary, all the results above prove the superiority of the DENSDBO-ASR algorithm.

Fig. 17(a–f) exhibits the convergence curves of the AOFV1 and AOFV2, where the horizontal axis represents the iterations and the vertical axis represents the optimal function value among all feasible solutions in the population. It can be seen that the DENSDBO-ASR

Table 7
Parameter settings for the obstacle datasets.

No. of obstacles	(x_m, y_m) (km)	R_m (km)	H (km)
1	(50,100)	10	2.5
2	(120,120)	10	2.5
3	(120,20)	15	2.5

Table 8
The parameters of UAV.

Indicators	Parameters
Maximum flight altitude	$h_{\max} = 3000$m
Minimum flight altitude	$h_{\min} = 2000$m
Minimum safe flight height	$h_{safe} = 200$m
Maximum turning angle	$\varphi_{\max} = 60^{\circ}$
Maximum climbing angle	$\theta_{\max} = 45^{\circ}$
Minimum straight flight distance	$l_{\min} = 2000$m
Minimum safe flight distance between UAVs	$d_{us} = 5000$m
Maximum flight speed	$v_{\max} = 238.00$ m/s
Minimum flight speed	$v_{\min} = 102.00$ m/s

Table 9
The starting and target point of UAVs for the three cases.

	No.	Start point (km)	Target Point (km)
2 UAVs	1	(20, 25, 2)	(100, 170, 3)
	2	(25, 25, 2)	(170, 180, 3)
3 UAVs	1	(25, 25, 2)	(100, 175, 3)
	2	(25, 20, 2)	(170, 175, 3)
	3	(20, 20, 2)	(150, 20, 3)
4 UAVs	1	(25, 20, 2)	(170, 180, 3)
	2	(20, 25, 2)	(100, 175, 3)
	3	(25, 25, 2)	(150, 20, 3)
	4	(20, 20, 2)	(125, 90, 3)

algorithm consistently exhibits rapid convergence speed and ultimately obtains the lowest cost among all algorithms, showcasing its outstanding performance. This robust and reliable behavior makes DENSDBO-ASR the most prominent optimization method for dealing with cooperative path planning challenges.

To further demonstrate the robustness of the DENSDBO-ASR algorithm, the number of UAVs is increased to 6, 8, and 10. The SPEA2-CDP, NSGAII-ToR, and MOGWO are selected as comparison algorithms. The population size is set to 150. The maximum number of iterations is set to 200 for 6 UAVs, 300 for 8 UAVs, and 500 for 10 UAVs. All algorithms are run independently five times. In large-scale scenarios, finding a feasible path is the most important. Therefore, feasible rate, OFV1 and OFV2, are selected as the final evaluation metrics. Where, feasible rate is the percentage of feasible solutions in the Pareto set.

Table 11 presents the results of experiments with the number of UAVs of 6, 8, and 10. From Table 11, it is evident that DENSDBOASR has a feasible rate of 100 % in all scenarios, and both OFV1 and OFV2 are optimal. When the number of UAVs is 6, the feasible rates of SPEA2-CDP, NSGAII-ToR, and DENSDBO-ASR are all 100 %. However, MOGWO only achieves a feasible rate of 5.20 %. When comparing OFV1 and OFV2, DENSDBO-ASR ranks first, followed by MOGWO. When the UAV number reaches 8, the feasible rates of SPEA2-CDP, NSGAII-ToR, and DENSDBO-ASR remain at 100 %, while MOGWO decreases to 0. This suggests that MOGWO is prone to falling into local optimal solutions when dealing with large-scale issues. To address this, it is necessary to improve the updating strategy, modify the population size, or increase the parallel processing capacity. When the number of UAVs is 10, the feasible rates of SPEA2-CDP and NSGAII-ToR decline to 66.67 %, while DENSDBO-ASR maintains a feasible rate of 100 %. Furthermore, DENSDBO-ASR still demonstrates superior performance in terms of OFV1 and OFV2 metrics. To summarize, DENSDBO-ASR is robust and proficient in managing collaborative path planning with an increased number of UAVs.

7. Discussion

7.1. Implication

Metaheuristics are emerging as attractive vehicles for multi-UAV cooperative path planning due to their adaptability, flexibility, and global search capabilities. Although several improved metaheuristic algorithms have been successful in collaborative multi-UAV path planning [56–58], existing studies suggest that there are limitations in balancing different optimization objectives and considering practical constraints [20,59]. Namely, Ren et al. [20] designed a path-planning model that optimizes multiple objectives simultaneously by using the improved AEMO-EDA algorithm. However, optimizing multiple objectives simultaneously can significantly complicate the optimization problem, leading to a difficult trade-off between convergence speed and solution quality. Li et al. [59] used the reference point technique and a distance cost matrix for path planning. However, the model ignores the collaborative limitations among multiple UAVs. This increases the risk of collisions and reduces the overall path planning efficiency. Therefore, it is crucial to enhance safety while minimizing cost and risk of collision in order to improve efficiency in mission execution.

The benchmark function tests, the Wilcoxon rank sum test, and the Friedman test confirm that the proposed improvement strategies effectively enhance the performance of the algorithm and expand its range of applications. This finding was not unexpected since it has been shown that the improved strategies significantly enhance the performance of the original algorithms [14,15]. Abbaszadeh Shahri

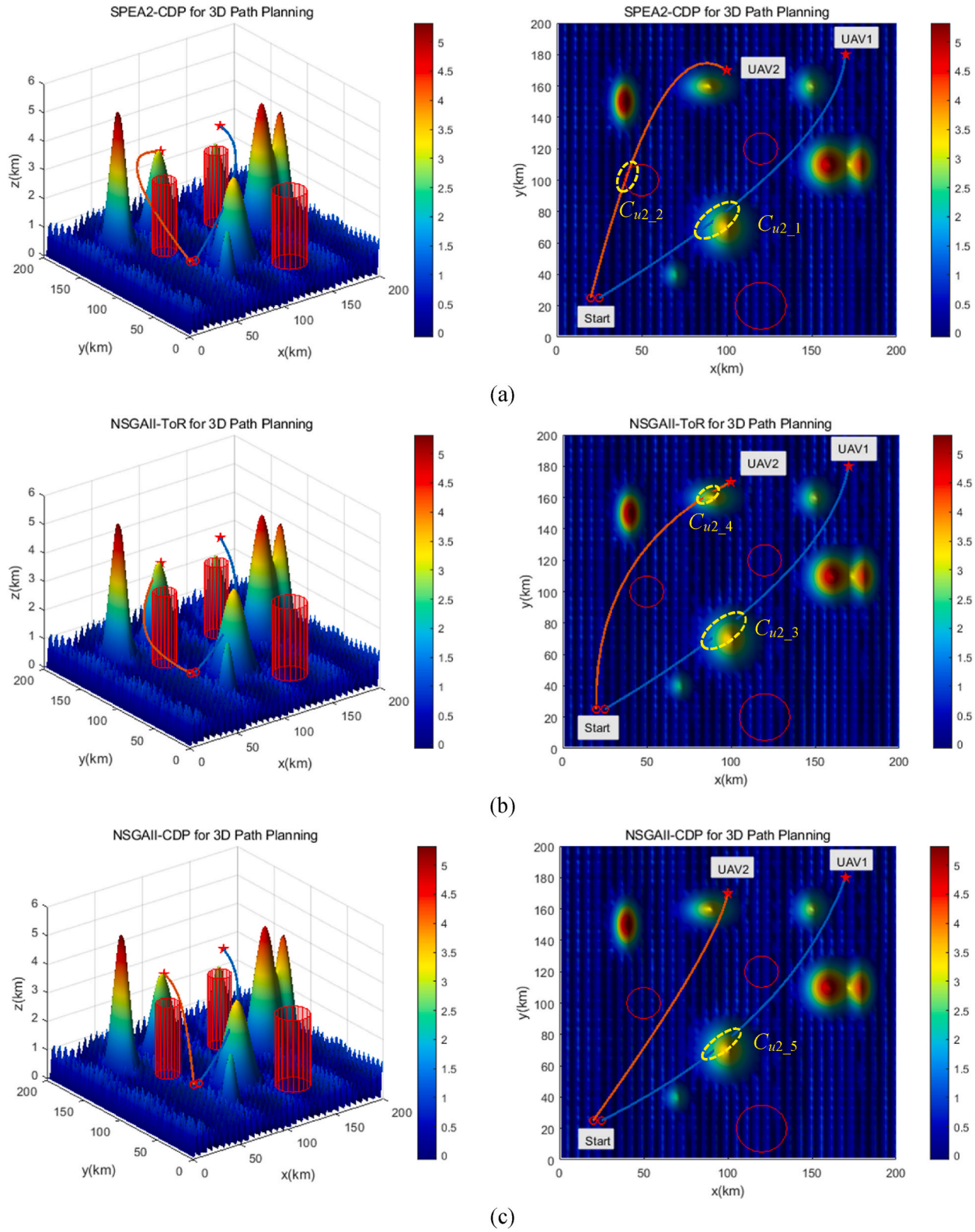
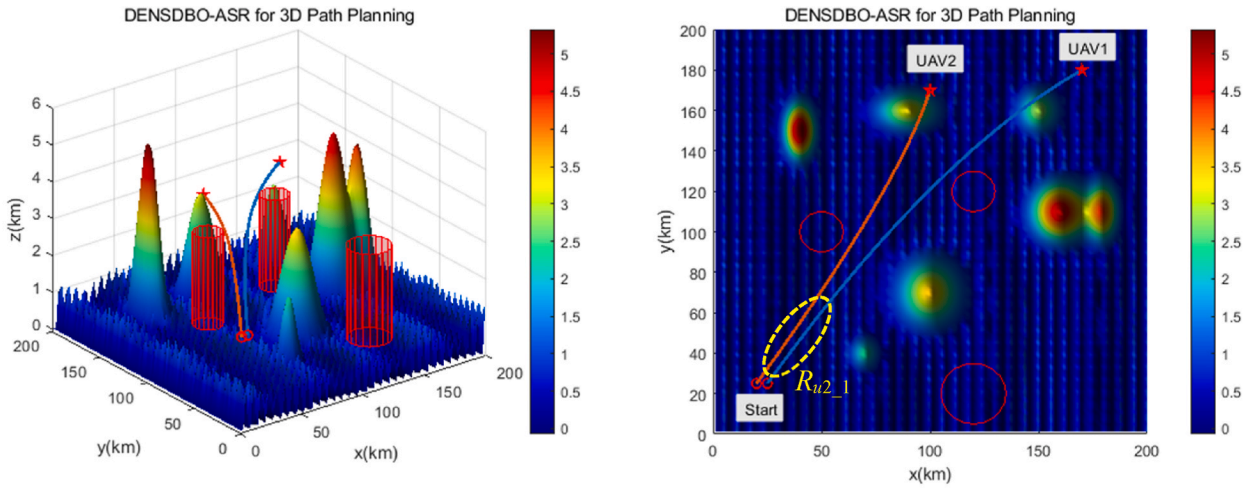
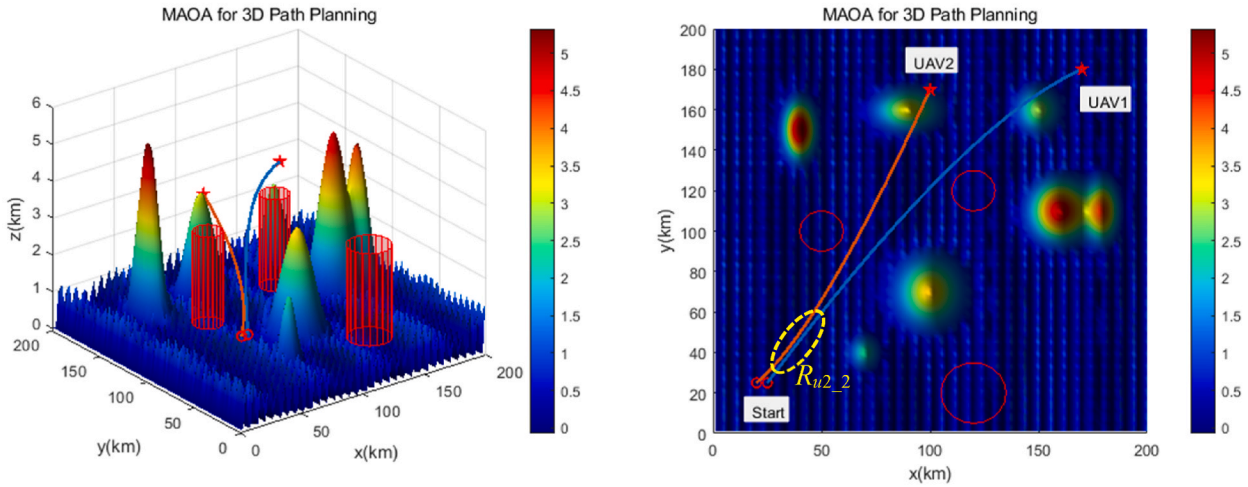


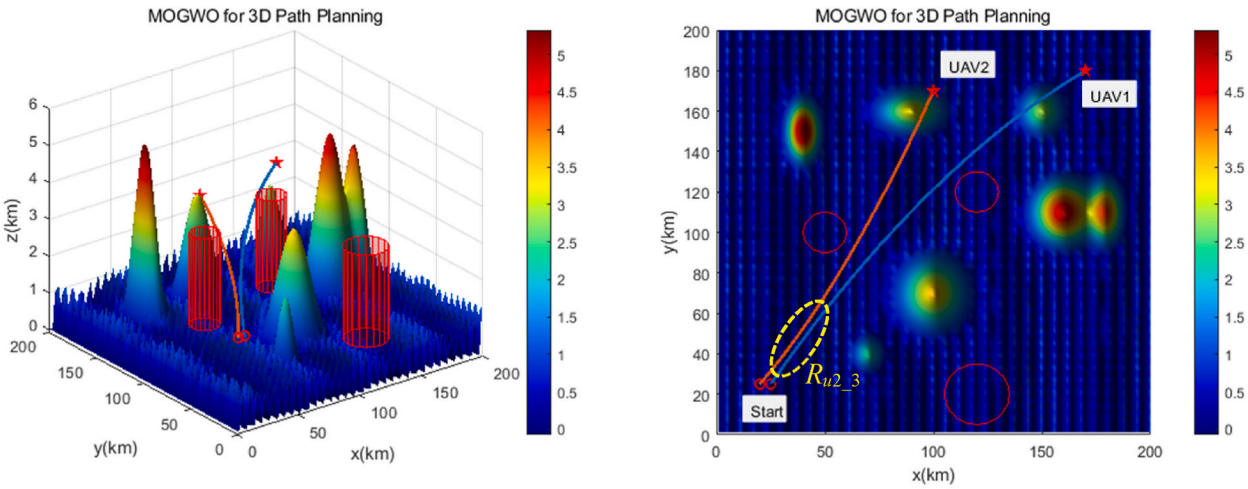
Fig. 13. Results of the collaborative path planning with two UAVs. (a) SPEA2-CDP; (b) NSGAII-ToR; (c) NSGAII-CDP; (d) DENSDBO-ASR; (e) MAOA; (f) MOGWO.



(d)

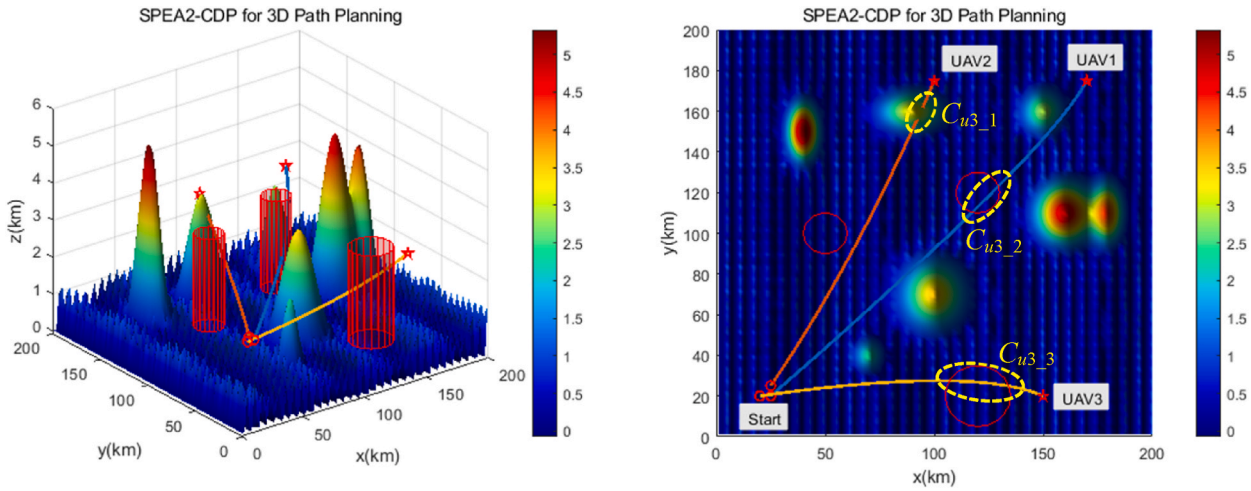


(e)

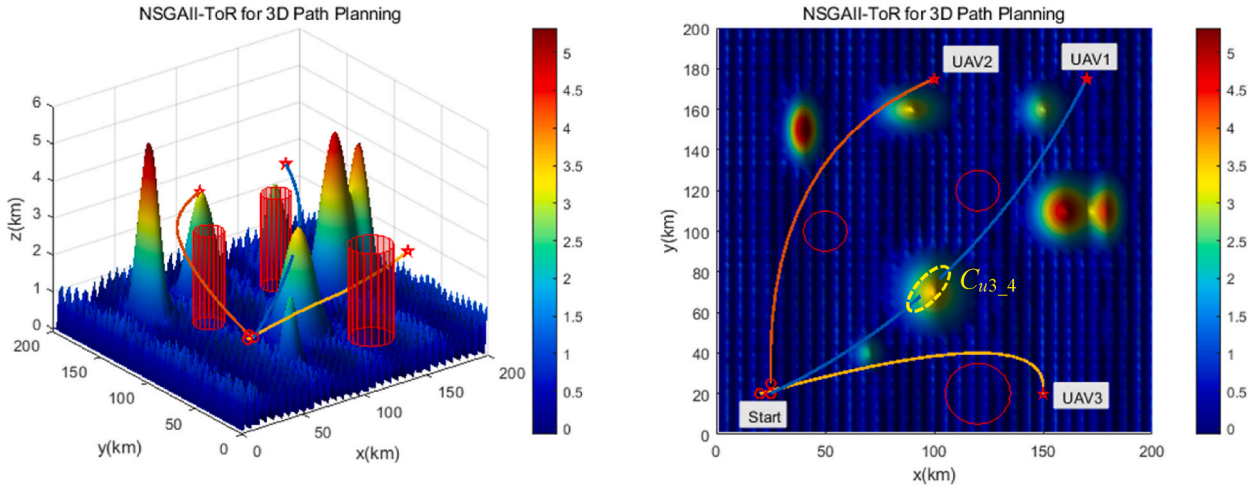


(f)

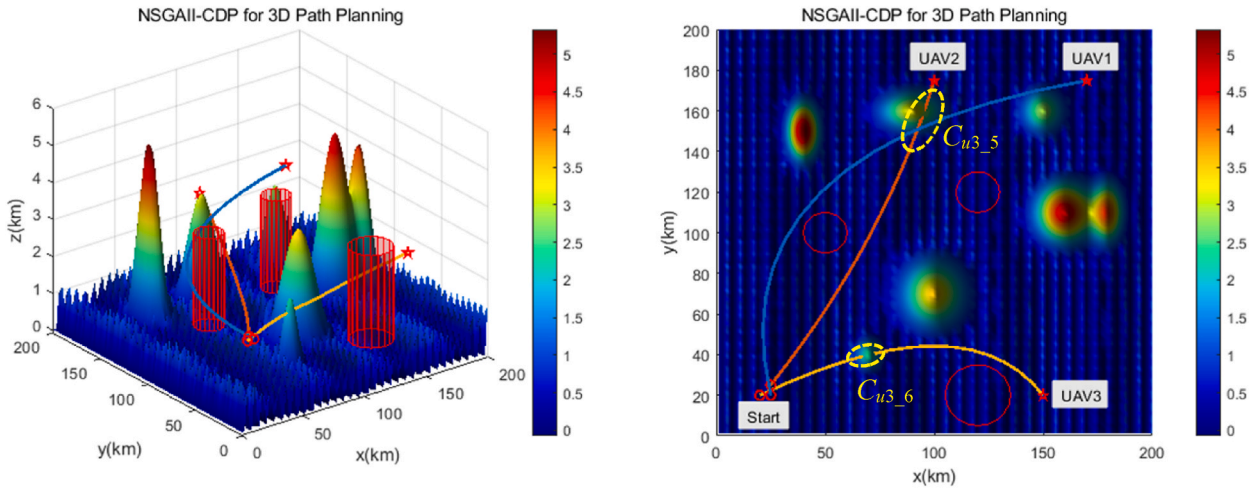
Fig. 13. (*continued*).



(a)

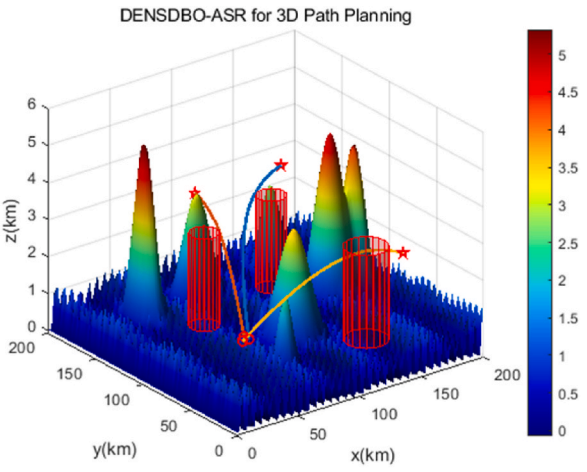


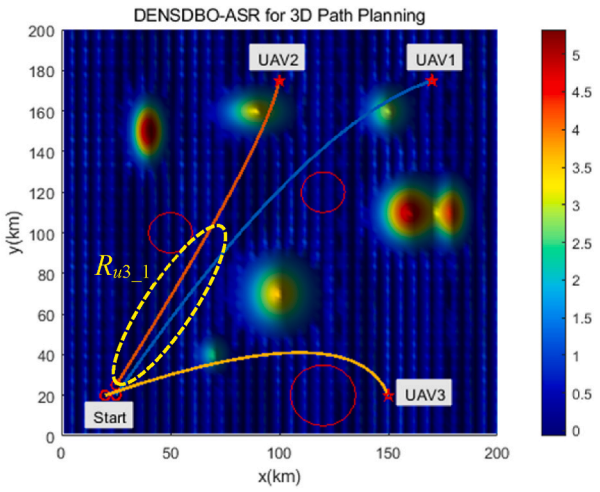
(b)



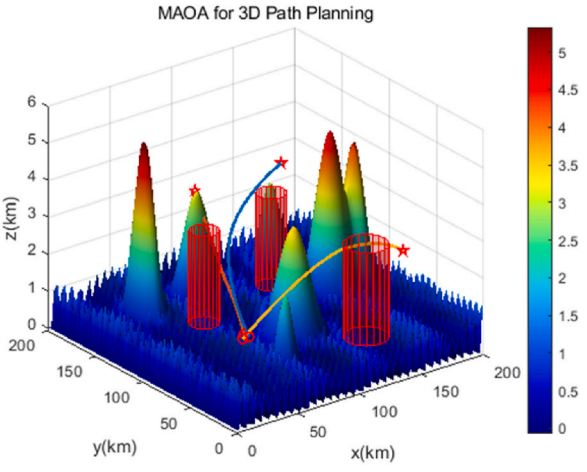
(c)

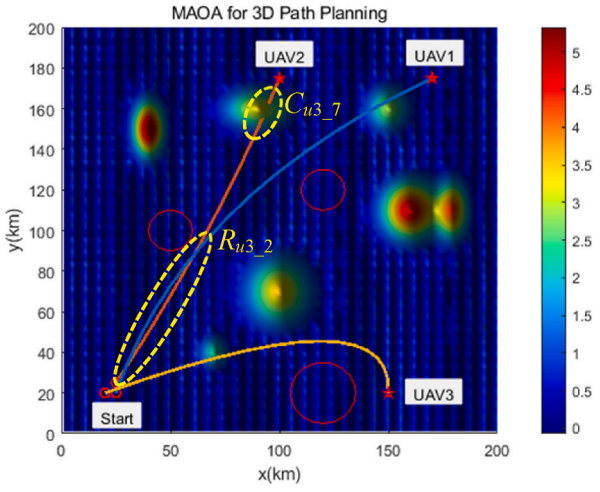
Fig. 14. Results of the collaborative path planning with three UAVs. (a) SPEA2-CDP; (b) NSGAII-ToR; (c) NSGAII-CDP; (d) DENSDBO-ASR; (e) MAOA; (f) MOGWO.



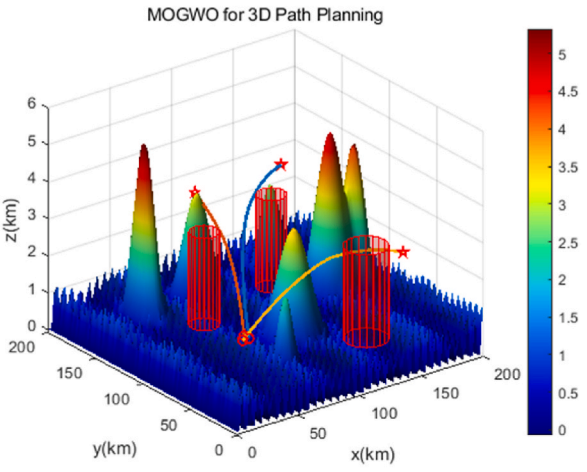


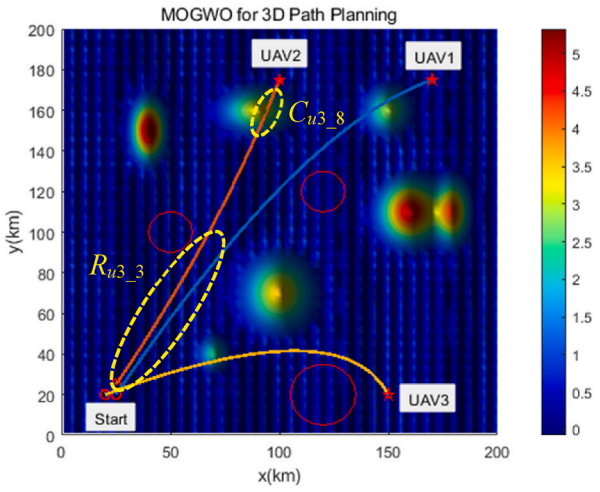
(d)





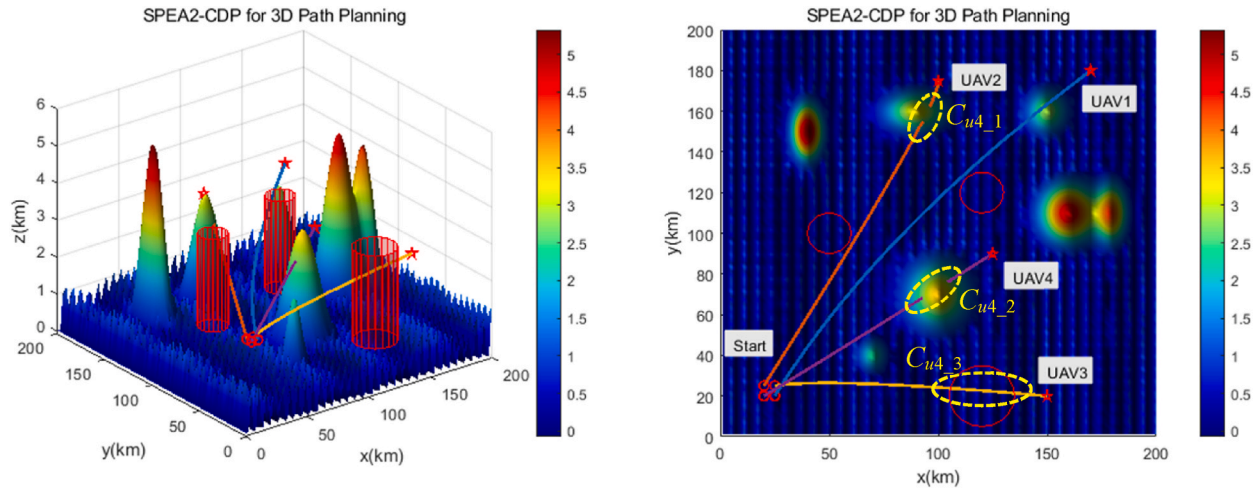
(e)



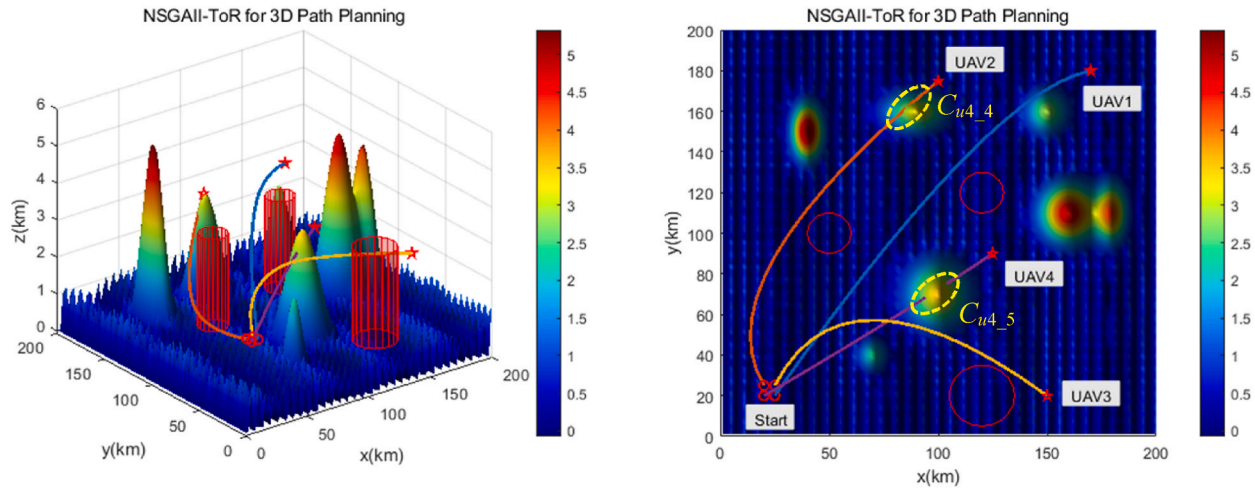


(f)

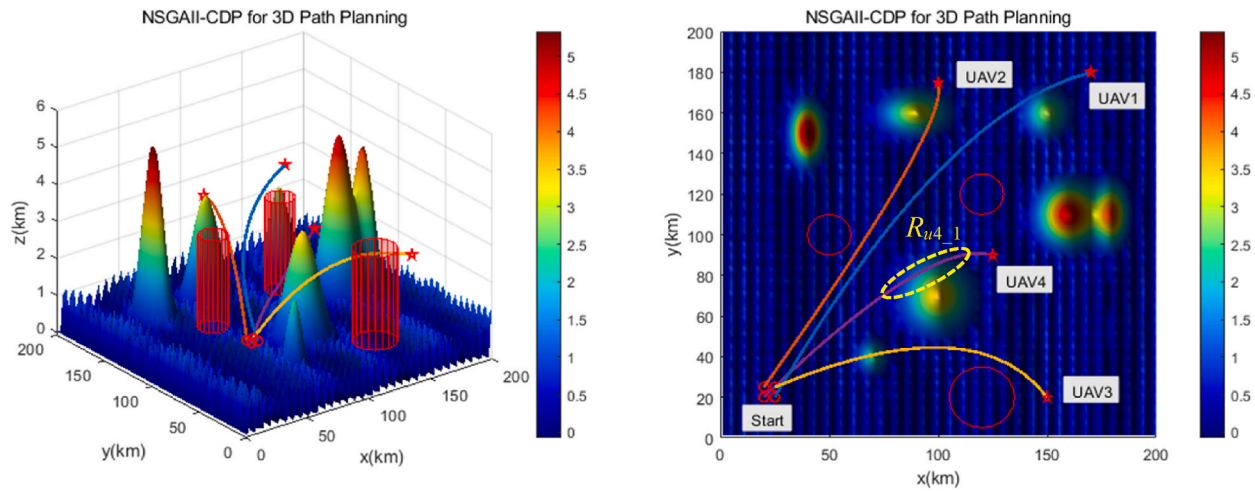
Fig. 14. (*continued*).



(a)

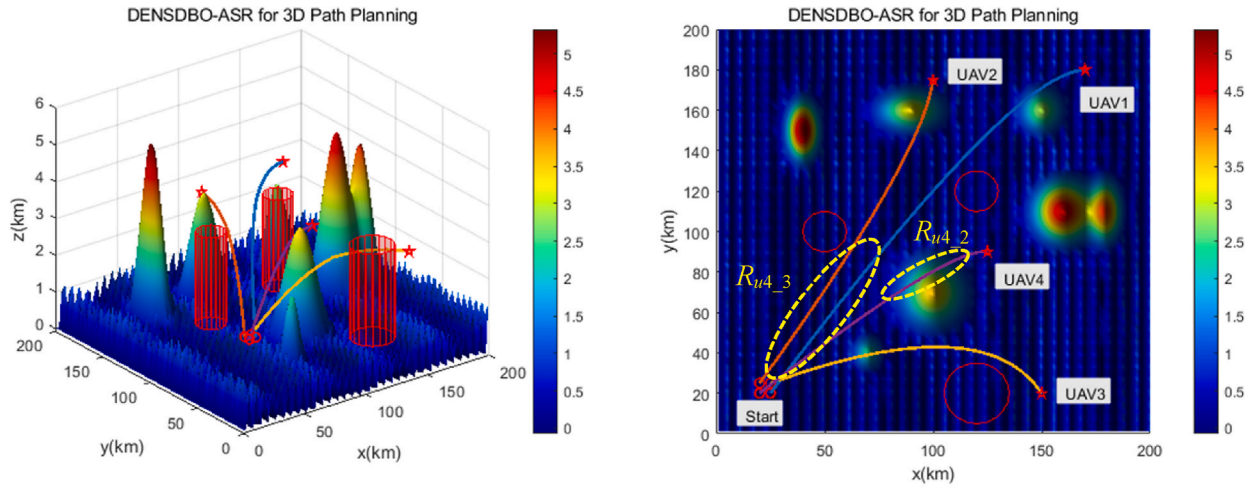


(b)

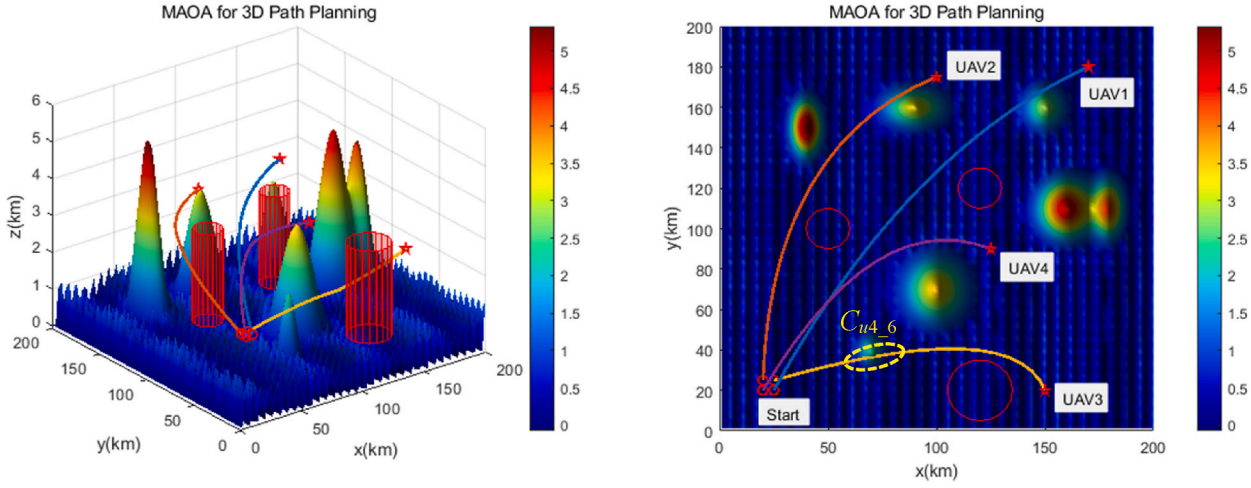


(c)

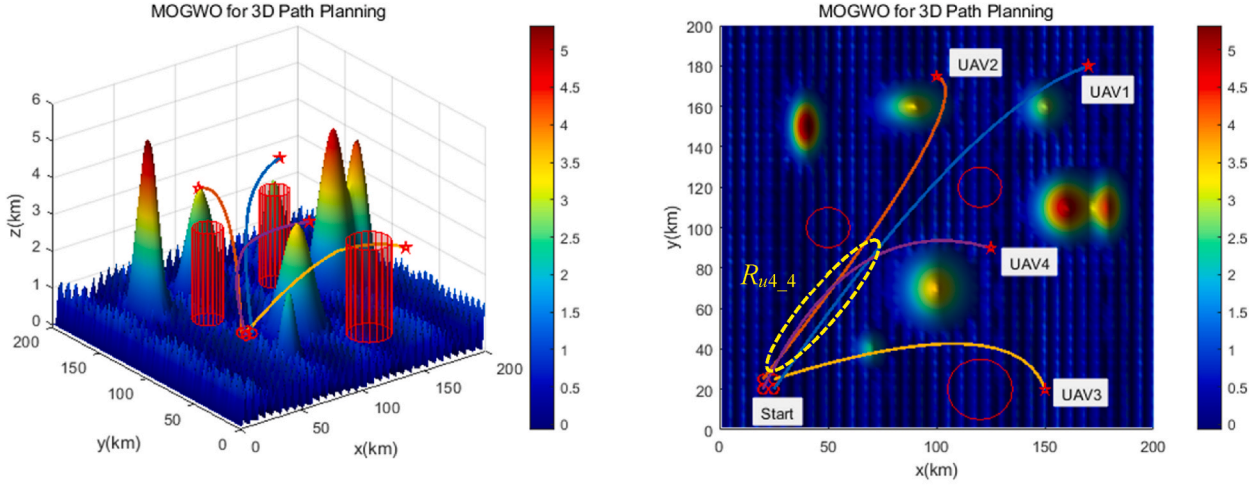
Fig. 15. Results of the collaborative path planning with four UAVs. (a) SPEA2-CDP; (b) NSGAII-ToR; (c) NSGAII-CDP; (d) DENSDBO-ASR; (e) MAOA; (f) MOGWO.



(d)

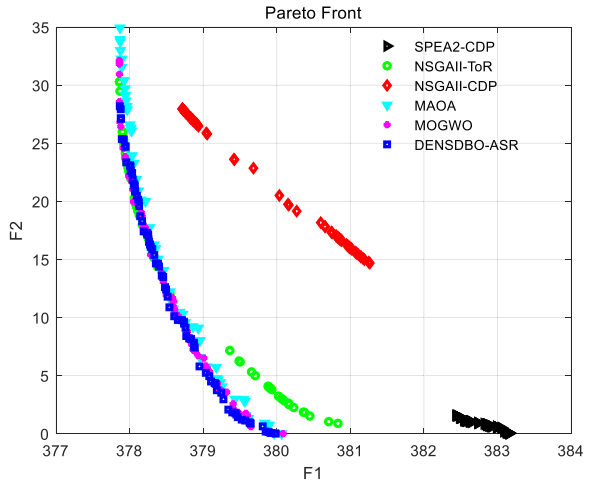


(e)

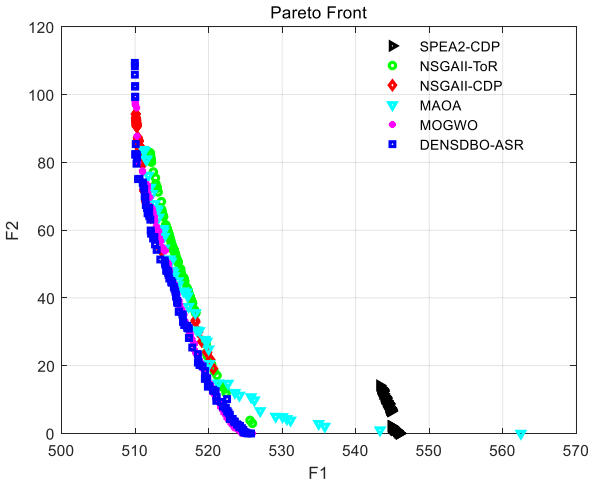


(f)

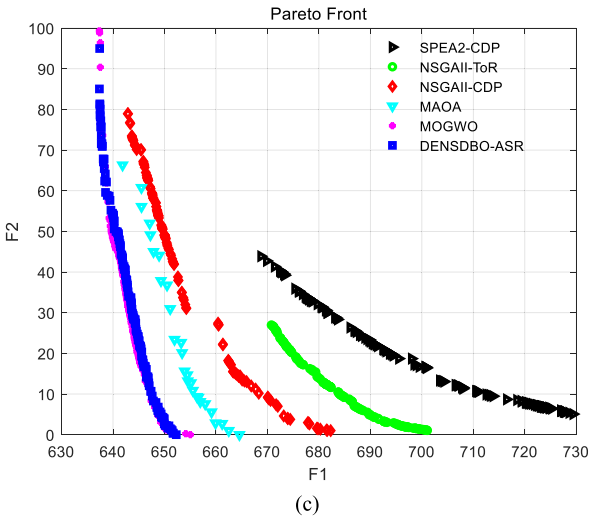
Fig. 15. (*continued*).



(a)



(b)



(c)

Fig. 16. Comparisons of the Pareto fronts. (a) Two UAVs; (b) Three UAVs; (c) Four UAVs.

Table 10
Comparisons of multi-objective indicators for the simulated experiments.

	Method	Indexes	Metric			
			PD↑	SP↓	OFV1↓	OFV2↓
Two UAVs	SPEA2-CDP [51]	Avg (±Std)	3.71E+03 (±4.97E+03)	6.40E-03 (±3.59E-03)	3.84E+02 (±6.28E+00)	7.20E+00 (±6.52E+00)
	NSGAII-ToR [48]	Avg (±Std)	7.54E+03 (±2.79E+03)	7.39E-03 (±4.06E-03)	3.83E+02 (±5.27E+00)	5.80E+00 (±7.78E+00)
	NSGAII-CDP [47]	Avg (±Std)	8.64E+03 (±5.79E+03)	8.27E-03 (±4.97E-03)	3.80E+02 (±2.59E+00)	4.00E+00 (±4.98E+00)
	DENSDBO-ASR	Avg (±Std)	**1.41E+04** (±2.19E+03)	**6.06E-03** (±8.25E-04)	**3.78E+02** (±1.06E-03)	**0.00E+00** (±0.00E+00)
	MAOA [53]	Avg (±Std)	1.25E+04 (±4.44E+02)	1.76E-02 (±1.73E-02)	**3.78E+02** (±6.08E-02)	**0.00E+00** (±0.00E+00)
	MOGWO [52]	Avg (±Std)	1.18E+04 (±9.77E+02)	9.40E-03 (±2.90E-03)	**3.78E+02** (±9.99E-02)	6.51E-02 (±1.45E-01)
Three UAVs	SPEA2-CDP	Avg (±Std)	1.19E+04 (±1.13E+04)	**2.37E-03** (±2.96E-03)	5.14E+02 (±2.87E+00)	4.08E+01 (±2.80E+01)
	NSGAII-ToR	Avg (±Std)	2.99E+04 (±1.36E+04)	6.15E-03 (±3.19E-03)	5.24E+02 (±9.38E+00)	9.40E-01 (±5.77E-01)
	NSGAII-CDP	Avg (±Std)	3.54E+04 (±1.14E+04)	9.95E-03 (±8.14E-03)	5.23E+02 (±8.98E+00)	7.74E+00 (±1.23E+01)
	DENSDBO-ASR	Avg (±Std)	**5.87E+04** (±1.02E+04)	4.50E-03 (±6.22E-04)	**5.10E+02** (±3.83E-03)	**0.00E+00** (±0.00E+00)
	MAOA	Avg (±Std)	5.81E+04 (±1.27E+04)	1.69E-02 (±2.73E-03)	5.11E+02 (±4.98E-01)	**0.00E+00** (±0.00E+00)
	MOGWO	Avg (±Std)	5.78E+04 (±1.21E+04)	6.26E-03 (±2.29E-03)	**5.10E+02** (±1.36E-01)	7.51E-02 (±1.05E-01)
Four UAVs	SPEA2-CDP	Avg (±Std)	3.03E+04 (±1.57E+04)	4.18E-03 (±2.58E-03)	6.64E+02 (±3.43E+01)	7.35E+00 (±7.35E+00)
	NSGAII-ToR	Avg (±Std)	4.06E+04 (±1.61E+04)	7.16E-03 (±3.47E-03)	6.56E+02 (±1.95E+01)	2.21E+00 (±3.16E+00)
	NSGAII-CDP	Avg (±Std)	3.88E+04 (±1.47E+04)	**3.97E-03** (±7.76E-04)	6.48E+02 (±4.35E+00)	2.67E+00 (±4.65E+00)
	DENSDBO-ASR	Avg (±Std)	**5.52E+04** (±1.89E+04)	4.55E-03 (±9.95E-04)	**6.37E+02** (±1.77E-02)	**0.00E+00** (±0.00E+00)
	MAOA	Avg (±Std)	5.23E+04 (±1.70E+04)	3.31E-02 (±1.88E-02)	6.39E+02 (±2.84E+00)	**0.00E+00** (±0.00E+00)
	MOGWO	Avg (±Std)	5.29E+04 (±7.26E+03)	5.15E-03 (±1.77E-03)	**6.37E+02** (±9.29E-02)	1.43E-01 (±3.20E-01)

et al. [15] employed an adaptive parameter adjustment technique to improve the firefly algorithm that strikes a balance between exploration and exploitation. Chaharmahali et al. [14] proposed a restart mechanism to enhance the algorithm's capability to escape local optima. The results of path planning experiments indicate that DENSDBO-ASR outperforms other comparative algorithms, highlighting the effectiveness of metaheuristic algorithms in handling multi-objective path planning issues. Surprisingly, DENSDBO-ASR exhibits robustness in managing multi-objective functions. Previous research has shown that the theoretical optima values of the multi-objective functions have similar magnitudes [24,60,61]. While there is no conclusive data indicating that this resemblance has an impact on the algorithm's measured performance, we cannot entirely dismiss this possibility. Researchers have demonstrated that the algorithms exhibit strong performance when dealing with two objective functions that have close theoretical optima [24]. Our examination reveals a disparity in the magnitude of the theoretical optimal values of the two objective functions. The data results indicate that the values of OFV1 and OFV2 for DENSDBO-ASR are optimal in all scenarios, with OFV2 being able to converge to the ideal value of 0. This highlights the robustness of DENSDBO-ASR in addressing objective functions that have unbalanced magnitudes.

7.2. Limitation

Despite taking into account several constraints, including flying distance, threat, time, and spatial collisions, our study found that the algorithm still has certain limitations. First, the algorithm may struggle to adapt to changes in a dynamic environment. The real world is always changing, with unexpected barriers, fluctuating wind speeds, and so on, which requires algorithms to be able to recognize and adjust in real time. However, the algorithm was not developed with this dynamism in mind, resulting in real-world applications that may not perform optimally. Second, the model's assumptions may not appropriately account for all real-world complications. It happens to be essential to make assumptions about the real world to simplify the problem. However, these assumptions may not fully capture all complexities, limiting the algorithm's utility. Additionally, the constraints of UAV test datasets and the issue of time complexity must be considered. Existing UAV test datasets are limited, thus affecting the algorithm's ability to generalize across different scenarios. Furthermore, when dealing with large-scale problems, the algorithm's time complexity can become an obstacle.

To address these limitations, further research efforts can be conducted using the following methods: Firstly, develop a dynamic path updating approach that can enable the algorithm to adjust to real-time changes in the environment. This can be achieved by enhancing the algorithm's feedback mechanism, enabling it to automatically adjust the trajectory in response to environmental changes. Furthermore, add certain restriction assumptions, such as communication restrictions, to make the model more realistic. In addition, it is necessary to improve the UAV test dataset by incorporating a wider range of conditions and mission scenarios. Moreover, the implementation of parallel computing can reduce the time complexity.

In real-world scenarios, the hardware limits and communication constraints between UAVs should also be considered. Both distance and interference can have an impact on UAV communication. Efficient communication protocols can be employed to improve the anti-interference capability of UAVs [62]. In addition, it is essential to take into account hardware constraints such as compute power, energy consumption, and sensor performance. The algorithm needs to reduce computational complexity so that it can be executed faster on limited computer resources.

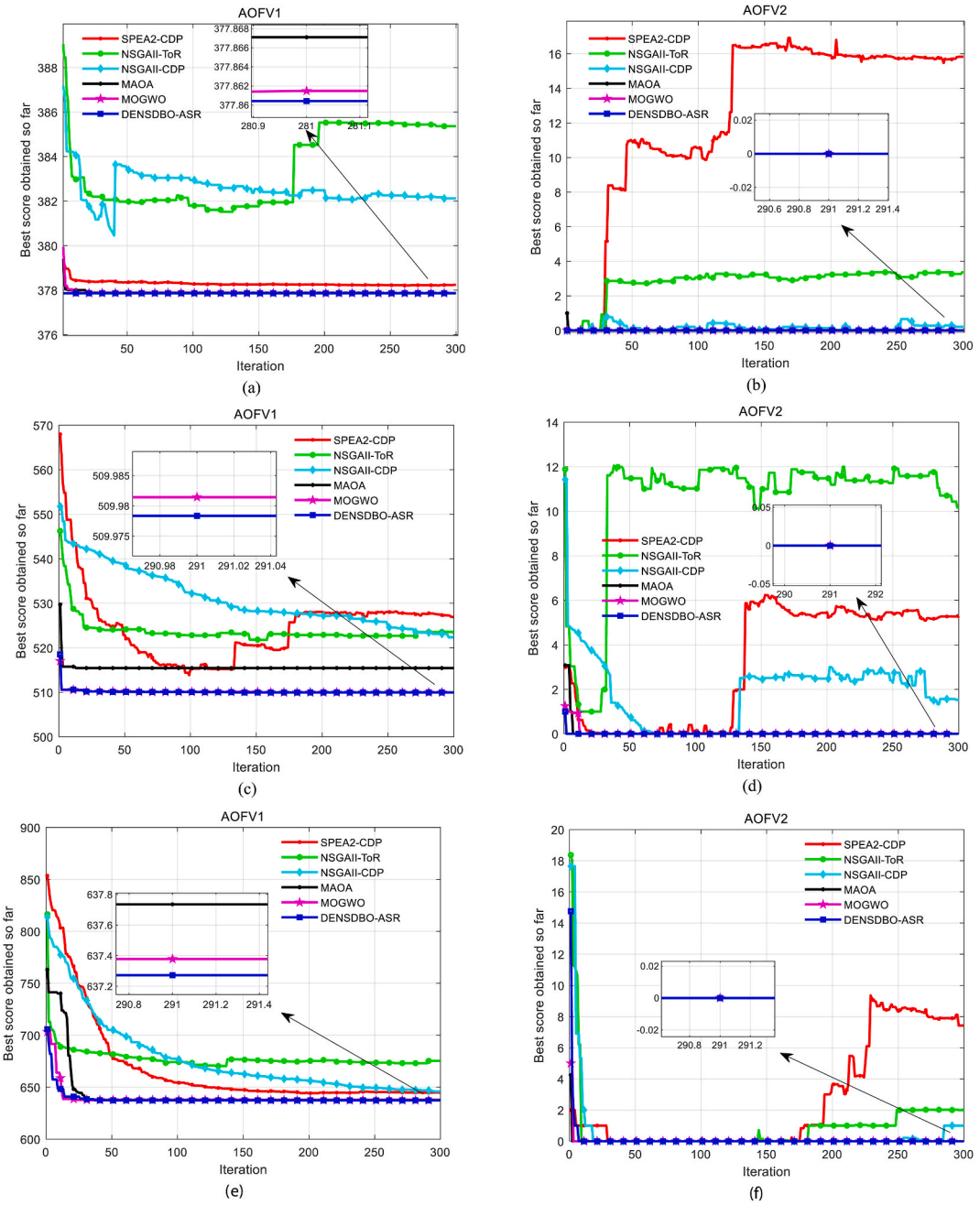


Fig. 17. Convergence curves of the AOFV. (a) AOFV1 with two UAVs; (b) AOFV2 with two UAVs; (c) AOFV1 with three UAVs; (d) AOFV2 with three UAVs; (e) AOFV1 with four UAVs; (f) AOFV2 with four UAVs.

7.3. Ethical issues of deploying UAVs

The deployment of UAVs on collaborative missions raises significant ethical issues that must be considered [63,64]. One of the primary concerns is the infringement on personal privacy and the abuse of surveillance capabilities [65]. It is crucial to establish clear guidelines to safeguard individual rights and privacy. Van Wynsberghe [63] investigated the physiological and behavioral impacts of UAVs on humans to ensure that drones are used responsibly. Ntizikira [66] proposed the SP-IoUAV model, which focuses on enhancing the security and privacy of UAVs. The main objective of this model is to secure the integrity and confidentiality of data with the Internet of UAV ecosystems. Privacy and civil liberties can be effectively protected by utilizing privacy-preserving technologies, such as anonymized data processing and encrypted communications.

Ensuring safety is of great importance in UAV operations, particularly in multi-UAV-coordinated missions. To reduce the risk of

Table 11
The results of experiments with the number of UAVs of 6, 8, and 10.

	Metrics	Indexes	Methods			
			SPEA2-CDP	NSGAII-ToR	DENSDBO-ASR	MOGWO
Six UAVs	Feasible rate		**100 %**	**100 %**	**100 %**	5.20 %
	OFV1	Avg (±Std)	1.06E+03 (±7.69E+01)	9.98E+02 (±5.95E+01)	**9.01E+02 (±1.29E-01)**	9.08E+02 (±1.01E+01)
	OFV2	Avg (±Std)	6.12E+01 (±1.02E+02)	5.16E+00 (±8.93E+00)	**0.00E+00 (±0.00E+00)**	2.85E+01 (±2.29E+01)
Eight UAVs	Feasible rate		**100 %**	**100 %**	**100 %**	0 %
	OFV1	Avg (±Std)	1.61E+03 (±1.41E+02)	1.68E+03 (±9.21E+01)	**1.30E+03 (±6.64E-01)**	–
	OFV2	Avg (±Std)	2.85E+02 (±3.69E+02)	3.03E+02 (±2.50E+02)	**0.00E+00 (±0.00E+00)**	–
Ten UAVs	Feasible rate		66.67 %	66.67 %	**100 %**	0 %
	OFV1	Avg (±Std)	2.06E+03 (±1.37E+02)	2.15E+03 (±1.74E+02)	**1.53E+03 (±2.02E+01)**	–
	OFV2	Avg (±Std)	4.20E+02 (±4.18E+02)	4.54E+02 (±2.59E+02)	**1.01E+01 (±2.32E-01)**	–

accidents and ensure the safety of the public, it is imperative to implement powerful safety measures such as precise flight planning, secure operations, and advanced collision avoidance systems [67]. Prioritizing these safety procedures allows us to minimize potential risks and enhance the general reliability of UAV operations.

Furthermore, UAV technology has had both advantageous and adverse effects on society. It offers substantial advantages in diverse domains, such as disaster management, agricultural monitoring, and infrastructure inspection. However, it is important to acknowledge the potential environmental impacts associated with their use, such as noise pollution and habitat destruction [68]. Effective regulations are critical for ensuring that UAVs are employed appropriately and minimizing potential risks. This includes promptly updating airspace regulations, setting clear guidelines for the use of UAVs in various areas, and implementing appropriate penalties for violations of safety and privacy standards. The development of UAV technology can be further facilitated by maintaining a robust regulatory framework.

In summary, our studies demonstrate the effectiveness of DENSDBO-ASR in cooperative path planning, and these findings provide theoretical support for practical navigation. Furthermore, the algorithm is expected to be applied to more complex, constrained multi-objective optimization situations.

8. Conclusion

This paper focuses on multi-UAV cooperative path planning and regards it as a constrained, multi-objective problem. A multi-objective optimization model is presented, which considers the limitations regulating UAV maneuverability and cooperative constraints among different UAVs. Four main cost functions are considered: the UAV path length cost, altitude cost, threat cost, and time cost. The sum of the length cost and altitude cost is one objective function, while the sum of the threat cost and time cost is another objective function. Subsequently, a novel DENSDBO-ASR algorithm is proposed to solve the multi-UAV cooperative path planning problem.

First, a multi-objective dung beetle optimizer is developed to solve the multi-objective optimization problem. Moreover, a directional evolutionary strategy is adopted to improve the diversity of the algorithm and enhance its search capability. Then, we propose an adaptive stochastic ranking mechanism to handle multiple constraints effectively. The experimental results on benchmark functions indicate that the DENSDBO-ASR algorithm has higher efficiency and superior performance than other methods. In addition, a detailed implementation of a cooperative multi-UAV path planning method using the DENSDBO-ASR algorithm is provided. Finally, the simulation experiments show that DENSDBO-ASR can obtain optimal paths for multiple UAVs in various scenarios. The comparison results also confirm that the DENSDBO-ASR algorithm has a significant advantage in solving the multi-UAV path planning problem.

In future work, we hope to develop more effective strategies to optimize the DENSDBO-ASR algorithm to reduce time complexity and broaden the algorithm's application scope. The use of realistic and diverse experimental datasets will be essential for improving the path-planning model. Furthermore, we will concentrate on the real-time adaptation of dynamic path planning, establishing moving obstacles for path planning, and increasing UAV planning efficiency. These efforts are crucial to enhancing our models and progressing UAVs to address the ever-changing safety challenges in dynamic scenarios.

Funding statement

This study was funded by National Natural Science Foundation of China, grant number No.62166006; Natural Science Foundation of Guizhou Province, grant number [2020]1Y254.

Data availability statement

The authors have no competing interests to declare that are relevant to the content of this article.

CRediT authorship contribution statement

Qianwen Shen: Writing – review & editing, Software, Methodology, Conceptualization. **Damin Zhang:** Writing – review &

editing, Supervision, Investigation, Funding acquisition. **Qing He:** Writing – review & editing, Visualization, Supervision, Funding acquisition. **Yunfei Ban:** Writing – review & editing, Supervision, Software. **Fengqin Zuo:** Visualization, Supervision, Investigation.

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.

References

[1] D. Hong, S. Lee, Y.H. Cho, D. Baek, J. Kim, N. Chang, Energy-efficient online path planning of multiple drones using reinforcement learning, IEEE Trans. Veh. Technol. 70 (10) (2021) 9725–9740, https://doi.org/10.1109/tvt.2021.3102589.

[2] K. Li, Heuristic task scheduling on heterogeneous UAVs: a combinatorial optimization approach, J. Syst. Architect. 140 (2023), https://doi.org/10.1016/j.sysarc.2023.102895.

[3] A.A. Shahri, F. Pashamohammadi, R. Asheghi, H.A. Shahri, Automated intelligent hybrid computing schemes to predict blasting induced ground vibration, Eng. Comput. 38 (SUPPL 4) (2022) 3335–3349, https://doi.org/10.1007/s00366-021-01444-1.

[4] J.J. Rao, C.Y. Xiang, J.Y. Xi, J.B. Chen, J.T. Lei, W. Giernacki, M. Liu, Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm, Knowl. Base Syst. 277 (2023), https://doi.org/10.1016/j.knosys.2023.110797.

[5] Y.C. Guo, X.X. Liu, W. Jiang, W.G. Zhang, HDP-TSRRT*: a time-space cooperative path planning algorithm for multiple UAVs, Drones 7 (3) (2023), https://doi.org/10.3390/drones7030170.

[6] F. Aljalaud, H. Kurdi, K. Youcef-Toumi, Bio-inspired multi-UAV path planning heuristics: a review, Mathematics 11 (10) (2023), https://doi.org/10.3390/math11102356.

[7] Z. Yu, Z. Si, X. Li, D. Wang, H. Song, A novel hybrid particle swarm optimization algorithm for path planning of UAVs, IEEE Internet Things J. 9 (22) (2022) 22547–22558, https://doi.org/10.1109/jiot.2022.3182798.

[8] Y.V. Pehlivanoglu, P. Pehlivanoglu, An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems, Appl. Soft Comput. 112 (2021), https://doi.org/10.1016/j.asoc.2021.107796.

[9] S.K. Shao, C.L. He, Y.J. Zhao, X.J. Wu, Efficient trajectory planning for UAVs using hierarchical optimization, IEEE Access 9 (2021) 60668–60681, https://doi.org/10.1109/ACCESS.2021.3073420.

[10] F. Yan, J. Chu, J.W. Hu, X.P. Zhu, Cooperative task allocation with simultaneous arrival and resource constraint for multi-UAV using a genetic algorithm, Expert Syst. Appl. 245 (2024), https://doi.org/10.1016/j.eswa.2023.123023.

[11] K. Karthik, C. Balasubramanian, Improved green anaconda optimization algorithm-based coverage path planning mechanism for heterogeneous unmanned aerial vehicles, Sustain. Comput.-Inform. Syst. 42 (2024), https://doi.org/10.1016/j.suscom.2024.100961.

[12] A. Abbaszadeh Shahri, R. Asheghi, M. Khorsand Zak, A hybridized intelligence model to improve the predictability level of strength index parameters of rocks, Neural Comput. Appl. 33 (8) (2021) 3841–3854, https://doi.org/10.1007/s00521-020-05223-9.

[13] S.P. Mirfallah Lialestani, D. Parcerisa, M. Himi, A. Abbaszadeh Shahri, A novel modified bat algorithm to improve the spatial geothermal mapping using discrete geodata in Catalonia-Spain, Model. Earth Syst. Environ. (2024) 1–14, https://doi.org/10.1007/s40808-024-01992-7.

[14] G. Chaharmahali, D. Ghandalipour, M. Jasemi, S. Molla-Alizadeh-Zavardehi, Modified metaheuristic algorithms to design a closed-loop supply chain network considering quantity discount and fixed-charge transportation, Expert Syst. Appl. 202 (2022), https://doi.org/10.1016/j.eswa.2022.117364.

[15] A. Abbaszadeh Shahri, M. Khorsand Zak, H. Abbaszadeh Shahri, A modified firefly algorithm applying on multi-objective radial-based function for blasting, Neural Comput. Appl. 34 (3) (2022) 2455–2471, https://doi.org/10.1007/s00521-021-06544-z.

[16] J. Xue, B. Shen, Dung beetle optimizer: a new meta-heuristic algorithm for global optimization, J. Supercomput. 79 (7) (2023) 7305–7336, https://doi.org/10.1007/s11227-022-04959-6.

[17] F. Zhu, G. Li, H. Tang, Y. Li, X. Lv, X. Wang, Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems, Expert Syst. Appl. 236 (2024), https://doi.org/10.1016/j.eswa.2023.121219.

[18] Y. Li, K. Sun, Q. Yao, L. Wang, A dual-optimization wind speed forecasting model based on deep learning and improved dung beetle optimization algorithm, Energy 286 (2024), https://doi.org/10.1016/j.energy.2023.129604.

[19] M. Alamgeer, N. Alruwais, H.M. Alshahrani, A. Mohamed, M. Assiri, Dung beetle optimization with deep feature fusion model for lung cancer detection and classification, Cancers 15 (15) (2023), https://doi.org/10.3390/cancers15153982.

[20] Y. Ren, L. Zhang, An adaptive evolutionary multi-objective estimation of distribution algorithm and its application to multi-UAV path planning, IEEE Access 11 (2023) 50038–50051, https://doi.org/10.1109/access.2023.3270297.

[21] F. Ge, K. Li, Y. Han, Solving interval many-objective optimization problems by combination of NSGA-III and a local fruit fly optimization algorithm, Appl. Soft Comput. 114 (2022), https://doi.org/10.1016/j.asoc.2021.108096.

[22] L. Xu, X. Cao, W. Du, Y. Li, Cooperative path planning optimization for multiple UAVs with communication constraints, Knowl. Base Syst. 260 (2023), https://doi.org/10.1016/j.knosys.2022.110164.

[23] S. Ghambari, M. Golabi, J. Lepagnot, M. Brevilliers, L. Jourdan, L. Idoumghar, An enhanced NSGA-II for multiobjective UAV path planning in urban environments. 32nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Electr Network, 2020, pp. 106–111.

[24] Y. Wan, Y. Zhong, A. Ma, L. Zhang, An accurate UAV 3-D path planning method for disaster emergency response based on an improved multiobjective swarm intelligence algorithm, IEEE Trans. Cybern. 53 (4) (2023) 2658–2671, https://doi.org/10.1109/tcyb.2022.3170580.

[25] H.M. Maldonado, S. Zapotecas-Martinez, A dynamic penalty function within MOEA/D for constrained multi-objective optimization problems. IEEE Congress on Evolutionary Computation (IEEE CEC), Electr Network, 2021, pp. 1470–1477.

[26] Q. Gu, J. Bai, X. Li, N. Xiong, C. Lu, A constrained multi-objective evolutionary algorithm based on decomposition with improved constrained dominance principle, Swarm Evol. Comput. 75 (2022), https://doi.org/10.1016/j.swevo.2022.101162.

[27] Y. Yang, J. Liu, S. Tan, A constrained multi-objective evolutionary algorithm based on decomposition and dynamic constraint-handling mechanism, Appl. Soft Comput. 89 (2020), https://doi.org/10.1016/j.asoc.2020.106104.

[28] D. Xia, X. Wu, M. Yan, C. Xiong, An adaptive stochastic ranking-based tournament selection method for differential evolution, J. Supercomput. 80 (1) (2024) 20–49, https://doi.org/10.1007/s11227-023-05390-1.

[29] U. Balande, D. Shrimankar, SRIFA: stochastic ranking with improved-firefly-algorithm for constrained optimization engineering design problems, Mathematics 7 (3) (2019), https://doi.org/10.3390/math7030250.

[30] Q. Gu, Q. Wang, N.N. Xiong, S. Jiang, L. Chen, Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems, Complex & Intellig. Syst. 8 (4) (2022) 2699–2718, https://doi.org/10.1007/s40747-020-00249-x.

[31] Z. Zhang, J. Jiang, J. Wu, X.Z. Zhu, Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm, ISA (Instrum. Soc. Am.) Trans. 134 (2023) 42–57, https://doi.org/10.1016/j.isatra.2022.07.032.

[32] L. Wu, X.D. Huang, J.G. Cui, C. Liu, W.S. Xiao, Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot, Expert Syst. Appl. 215 (2023), https://doi.org/10.1016/j.eswa.2022.119410.

[33] L.Y. Zhou, S.P. Leng, Q. Liu, Q. Wang, Intelligent UAV swarm cooperation for multiple targets tracking, IEEE Internet Things J. 9 (1) (2022) 743–754, https://doi.org/10.1109/JIOT.2021.3085673.

[34] D. Mandloi, R. Arya, A.K. Verma, Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3d environment, Int. J. Syst. Assuran. Eng. Manag. 12 (5) (2021) 990–1000, https://doi.org/10.1007/s13198-021-01186-9.
[35] G. Sanna, S. Godio, G. Guglieri, Neural network based algorithm for multi-UAV coverage path planning. 2021 International Conference on Unmanned Aircraft Systems (ICUAS), 2021, pp. 1210–1217.
[36] Y. Chen, Q. Dong, X. Shang, Z. Wu, J. Wang, Multi-UAV autonomous path planning in reconnaissance missions considering incomplete information: a reinforcement learning method, Drones 7 (1) (2023), https://doi.org/10.3390/drones7010010.
[37] K. Rajwar, K. Deep, S. Das, An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges, Artif. Intell. Rev. 56 (11) (2023) 13187–13257, https://doi.org/10.1007/s10462-023-10470-y.
[38] S. Jiaqi, T. Li, Z. Hongtao, L. Xiaofeng, X. Tianying, Adaptive multi-UAV path planning method based on improved gray wolf algorithm, Comput. Electr. Eng. 104 (2022), https://doi.org/10.1016/j.compeleceng.2022.108377.
[39] R. Kumar, L. Singh, R. Tiwari, Novel reinforcement learning guided enhanced variable weight grey wolf optimization (RLV-GWO) algorithm for multi-UAV path planning, Wireless Pers. Commun. 131 (3) (2023) 2093–2123, https://doi.org/10.1007/s11277-023-10534-w.
[40] H. Duan, J. Zhao, Y. Deng, Y. Shi, X. Ding, Dynamic discrete pigeon-inspired optimization for multi-UAV cooperative search-attack mission planning, IEEE Trans. Aero. Electron. Syst. 57 (1) (2021) 706–720, https://doi.org/10.1109/TAES.2020.3029624.
[41] B. Yu, S. Fan, W. Cui, K. Xia, L. Wang, A Multi-UAV cooperative mission planning method based on SA-WOA algorithm for three-dimensional space atmospheric environment detection, Robotica (2024), https://doi.org/10.1017/S0263574724000596.
[42] F. Peres, M. Castelli, Combinatorial optimization problems and metaheuristics: review, challenges, design, and development, Appl. Sci.-Basel 11 (14) (2021), https://doi.org/10.3390/app11146449.
[43] A.K. Das, D.K. Pratihar, Solving engineering optimization problems using an improved real-coded genetic algorithm (IRGA) with directional mutation and crossover, Soft Comput. 25 (7) (2021) 5455–5481, https://doi.org/10.1007/s00500-020-05545-9.
[44] A.K. Das, D.K. Pratihar, A directional crossover (DX) operator for real parameter optimization using genetic algorithm, Appl. Intell. 49 (5) (2019) 1841–1865, https://doi.org/10.1007/s10489-018-1364-2.
[45] A. Qi, D. Zhao, F. Yu, A.A. Heidari, Z. Wu, Z. Cai, F. Alenezi, R.F. Mansour, H. Chen, M. Chen, Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation, Comput. Biol. Med. 148 (2022), https://doi.org/10.1016/j.compbiomed.2022.105810.
[46] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, K.C. Tan, A survey on evolutionary constrained multiobjective optimization, IEEE Trans. Evol. Comput. 27 (2) (2023) 201–221, https://doi.org/10.1109/TEVC.2022.3155533.
[47] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, https://doi.org/10.1109/4235.996017.
[48] Z. Ma, Y. Wang, W. Song, A new fitness function with two rankings for evolutionary constrained multiobjective optimization, IEEE Transact. Syst. Man Cybernetics-Syst. 51 (8) (2021) 5005–5016, https://doi.org/10.1109/tsmc.2019.2943973.
[49] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, Int. J. Intell. Syst. 36 (10) (2021) 5887–5958, https://doi.org/10.1002/int.22535.
[50] J. Ma, D. Xia, Y. Wang, X. Niu, S. Jiang, Z. Liu, H. Guo, A comprehensive comparison among metaheuristics (MHs) for geohazard modeling using machine learning: insights from a case study of landslide displacement prediction, Eng. Appl. Artif. Intell. 114 (2022), https://doi.org/10.1016/j.engappai.2022.105150.
[51] S. Larrain, L. Pradenas, I. Pulkkinen, F. Santander, Multiobjective optimization of a continuous kraft pulp digester using SPEA2, Comput. Chem. Eng. 143 (2020), https://doi.org/10.1016/j.compchemeng.2020.107086.
[52] S. Mirjalili, S. Saremi, S.M. Mirjalili, LdS. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, Expert Syst. Appl. 47 (2016) 106–119, https://doi.org/10.1016/j.eswa.2015.10.039.
[53] N. Khodadadi, L. Abualigah, E.-S.M. El-Kenawy, V. Snasel, S. Mirjalili, An archive-based multi-objective arithmetic optimization algorithm for solving industrial engineering problems, IEEE Access 10 (2022) 106673–106698, https://doi.org/10.1109/ACCESS.2022.3212081.
[54] H. Wang, Y. Jin, X. Yao, Diversity assessment in many-objective optimization, IEEE Trans. Cybern. 47 (6) (2017) 1510–1522, https://doi.org/10.1109/tcyb.2016.2550502.
[55] W. Zhao, Z. Zhang, S. Mirjalili, L. Wang, N. Khodadadi, S.M. Mirjalili, An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems, Comput. Methods Appl. Mech. Eng. 398 (2022), https://doi.org/10.1016/j.cma.2022.115223.
[56] Z. Qadir, M.H. Zafar, S.K.R. Moosavi, K.N. Le, M.A.P. Mahmud, Autonomous UAV path-planning optimization using metaheuristic approach for predisaster assessment, IEEE Internet Things J. 9 (14) (2021) 12505–12514, https://doi.org/10.1109/JIOT.2021.3137331.
[57] R. Jarray, M. Al-Dhaifallah, H. Rezk, S. Bouallegue, Parallel cooperative coevolutionary grey wolf optimizer for path planning problem of unmanned aerial vehicles, Sensors 22 (5) (2022), https://doi.org/10.3390/s22051826.
[58] Z.A. Ali, Z. Han, Z. Di, Path planning of multiple UAVs using MMACO and DE algorithm in dynamic environment, Measur. Control 56 (3–4) (2023) 459–469, https://doi.org/10.1177/0020294020915727.
[59] K. Li, F. Ge, Y. Han, Ya Wang, W. Xu, Path planning of multiple UAVs with online changing tasks by an ORPFOA algorithm, Eng. Appl. Artif. Intell. 94 (2020), https://doi.org/10.1016/j.engappai.2020.103807.
[60] X.Y. Zhang, S. Xia, X.Z. Li, T. Zhang, Multi-objective particle swarm optimization with multi-mode collaboration based on reinforcement learning for path planning of unmanned air vehicles, Knowlege-Based Systems 250 (2022), https://doi.org/10.1016/j.knosys.2022.109075.
[61] R. Jarray, S. Bouallegue, H. Rezk, M. Al-Dhaifallah, Parallel multiobjective multiverse optimizer for path planning of unmanned aerial vehicles in a dynamic environment with moving obstacles, Drones 6 (12) (2022), https://doi.org/10.3390/drones6120385.
[62] H. Liu, Z. Yang, N. Zhao, Y. Gu, C. Yuen, Interference-aware multi-hop routing in UAV networks: a harmonic function-based potential field approach, IEEE Internet Things J. 1 (2024), https://doi.org/10.1109/JIOT.2024.3366580.
[63] A. van Wynsberghe, T. Comes, Drones in humanitarian contexts, robot ethics, and the human-robot interaction, Ethics Inf. Technol. 22 (1) (2020) 43–53, https://doi.org/10.1007/s10676-019-09514-1.
[64] D.B. Resnik, K.C. Elliott, Using drones to study human beings: ethical and regulatory issues, Sci. Eng. Ethics 25 (3) (2019) 707–718, https://doi.org/10.1007/s11948-018-0032-6.
[65] S.A.H. Mohsan, N.Q.H. Othman, Y. Li, M.H. Alsharif, M.A. Khan, Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends, Intellig. Serv. Robotics 16 (1) (2023) 109–137.
[66] E. Ntizikira, W. Lei, F. Alblehai, K. Saleem, M.A. Lodhi, Secure and privacy-preserving intrusion detection and prevention in the internet of unmanned aerial vehicles, Sensors 23 (19) (2023) 8077.
[67] H. Shakhatreh, A.H. Sawalmeh, A. Al-Fuqaha, Z. 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, https://doi.org/10.1109/ACCESS.2019.2909530.
[68] A. Jackman, N. Millner, A.M. Cunliffe, Y. Laumonier, E. Lunstrum, J. Paneque-Gálvez, S.A. Wich, Protecting people and wildlife from the potential harms of drone use in biodiversity conservation: interdisciplinary dialogues, Glob. Soci. Challenges J. 2 (1) (2023) 68–83.