

Highlights

UAV 3-D path planning based on MOEA/D with adaptive areal weight adjustment

Yougang Xiao,Hao Yang,Huan Liu,Keyu Wu,Guohua Wu

- Introduction of multi-objective path planning with sharp peak and low tail PFs.
- AAWA enables individuals to evolve towards sparser areas on PFs.
- The AAWA improves the convergence and diversity of the population.
- The MOEA/D-AAWA achieves shorter paths and safer paths.

UAV 3-D path planning based on MOEA/D with adaptive areal weight adjustment

Yougang Xiao^a, Hao Yang^a, Huan Liu^a, Keyu Wu^b and Guohua Wu^a

^aSchool of Traffic and Transportation Engineering, Central South University, Changsha 410075, China

^bCollege of Systems Engineering and the College of Electronic Sciences, National University of Defense Technology, , Changsha 410073, China

ARTICLE INFO

Keywords:

MOEA/D
UAV
3-D path planning
Multi-objective optimization
Adaptive areal weight

ABSTRACT

Unmanned aerial vehicles (UAVs) are desirable platforms for time-efficient and cost-effective task execution. 3-D path planning is a key challenge for task decision-making. This paper proposes an improved multi-objective evolutionary algorithm based on decomposition (MOEA/D) with an adaptive areal weight adjustment (AAWA) strategy to make a tradeoff between the total flight path length and the terrain threat. AAWA is designed to improve the diversity of the solutions. More specifically, AAWA first removes a crowded individual and its weight vector from the current population and then adds a sparse individual from the external elite population to the current population. To enable the newly-added individual to evolve towards the sparser area of the population in the objective space, its weight vector is constructed by the objective function value of its neighbors. The effectiveness of MOEA/D-AAWA is validated in twenty synthetic scenarios with different number of obstacles and four realistic scenarios in comparison with other three classical methods.

1. Introduction

Due to the advantages of low cost, small size, high flexibility and strong concealment [1,2], UAVs have been widely exploited for various fields, such as meteorological detection [66], emergency rescue [60], traffic monitoring [64], and target reconnaissance [65]. As an important part of task planning, path planning aims to provide feasible flight paths for UAVs. Reasonable path planning schemes can effectively avoid potential threats and reduce flight cost [3]. However, it is still a significant challenge to provide decision-makers with a diverse and feasible set of paths, while considering the UAV maneuverability and obstacle avoidance.

Currently, path planning methods have achieved effective results in various fields. However, unlike traditional unmanned ground vehicles [54, 55], the motion space of UAVs has an additional dimension (i.e., height), i.e., UAVs can perform ascent and descent movements. Therefore, UAV path planning in 3-D space is more in line with the flight reality [56]. In addition, limiting UAVs in the 2-D space with a fixed altitude will lead to the loss of promising solutions. The consideration of the third dimension will inevitably bring challenges in additional maneuverability of UAVs, such as the minimum flight altitude, maximum horizontal angle, and maximum vertical angle. In order to simulate real UAV flight, the abovementioned three factors are considered in this article.

Although the difficulty of path planning increases with an additional space dimension [47], there are still many relevant studies [5, 49–51]. However, these studies only focus on a single objective or construct the objective function as a weighted combination. Determining a weight vector to cater to decision-maker preferences is a hard task, especially if there are many requirements and constraints [28]. Furthermore, a flight path with a specific weight vector cannot meet the diverse task needs. From our point of view, it is essential to consider a multi-objective optimization problem (MOP) to provide a diverse set of solutions for path planning tasks. Specifically, our path planning problem simultaneously optimizes two conflicting objectives (i.e., the path length and the degree of threats). The shorter path can diminish task execution time, and the safer path can reduce damage rates of UAVs.

Due to the conflicting objectives of MOPs, the optimization of one objective typically accompanies by the deterioration of at least one of the other objectives [63]. Thus, a sole solution with optimal values for all objectives almost does not exist in a MOP. The main goal of solving MOPs is to obtain a set of compromise solutions. Owing to the population-based nature, multi-objective evolutionary algorithms (MOEAs) can generate a set of solutions in one

*Corresponding author.

 guohuawu@csu.edu.cn (G. Wu)

run [61, 62]. Therefore, MOEAs, such as NSGA-II [27], MOEA/D [28], MOPSO [52], have been developed to deal with MOPs, and have been applied in path planning in recent years.

During the iteration of MOEAs, the population diversity plays a significant role in the convergence and uniformity of the population [61]. First, lacking diversity weakens the population ability to leave the current local optimal subspace, thereby preventing the population from further approaching to the Pareto front (PF). In addition, the population diversity determines whether individuals can be uniformly distributed on PF simultaneously. In particular, when the target MOP has a complex PF [31, 57], such as the PF with a sharp peak and a low tail in our paper, it is hard to search for the sharp peak and the low tail. The complexity of PFs may lead to the loss of the population diversity and uniformity, and even the inability of population to leave the current local optimal subspace.

The above observations motivate us to propose an improved MOEA/D with a new adaptive weight adjustment strategy. MOEA/D decomposes a MOP into a number of scalar optimization subproblems represented by a set of uniformly distributed weight vectors. Due to the decomposition-based nature of MOEA/D, it is well feasible to improve the population diversity by reasonably adjusting the weight vectors. Therefore, we design a new weight adjustment strategy. The weight vectors are periodically adjusted in the iteration to provide an accurate direction for the evolution of individuals to sparse areas. First, in the current population, the crowded individuals with high similarity to other individuals are removed. And then the same number of individuals who have high dissimilarity to the individuals of the current population are added to the current population. To ensure the population convergence, the newly-added individuals are selected from an external elite population, which stores nondominated individuals during the iteration. To enable the newly-added individuals to evolve towards a sparser area of the current population in the objective space, we improved the calculation method of its weight vector, which use the objective function value of its neighbor to calculate its weight vectors.

Based on the above analysis, the contributions of this paper can be summarized as follows.

1) We introduce a bi-objective UAV path planning problem under 3-D scenarios, which aims to minimize path length and the degree of terrain threats simultaneously.

2) To improve population diversity, an adaptive areal weight adjustment strategy is designed. To be specific, a crowded individual in the current population and its weight vector are deleted, while the individual with the maximum sparsity selected from the external elite population and its weight vector calculated by the objective function of neighbors are added. This strategy contributes to navigating the evolution of solutions towards sparse areas on PFs.

3) We conduct extensive experiments compared with other three state of the art methods in twenty synthetic scenarios with different number of obstacles and four realistic scenarios to verify the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 states the mathematical model of UAV 3-D multi-objective path planning. Section 4 presents the proposed algorithm in detail. Section 5 reports a series of experiments and comparative results. Finally, conclusions are drawn in Section 6.

2. Literature review

Path planning can be generalized as seeking a collision-free between two positions based on certain evaluation criteria [48]. So far, the path planning methods can be grouped into five categories: 1) methods based mathematical model: linear programming algorithm [6], dynamic programming algorithm [7], etc; 2) methods based on geometric search: Voronoi diagram [8, 9], probabilistic roadmaps [10, 11], rapidly-exploring random tree [12, 13], elliptic tangent graph method [26], etc; 3) methods based on heuristic algorithms: Dijkstra [14], A-Star [15, 16], etc; 4) methods based on the artificial potential field [25]; 5) methods based on evolutionary and swarm intelligence algorithms: particle swarm optimization [16, 17], ant colony optimization [18, 19], genetic algorithm [20–22], differential evolution [23, 24], etc.

In recent years, UAV 3-D path planning methods have been attracted much attention. Nevertheless, some of them achieve the purpose of reducing problem complexity by strategies that diminish a dimension of 3-D planning space, that is, converting computationally expensive 3-D path planning problems into more manageable 2-D path planning problems. The most commonly used is altitude fixing strategy. Chen and Gesbert [44] fixed the UAV at a predetermined height in a 3-D urban space to simplify the 3-D path planning problem. This strategy has extended to multiple UAVs. Zhao et al. [58] assumed that all UAVs share the same urban airspace at the same altitude. It might actually cause a high density of air traffic. Dai et al. [45] fixed each UAV at a unique flight height to avoid collisions among UAVs. Although this strategy can simplify the 3-D path planning problem, the solution space will inevitably shrink significantly,

resulting in the loss of promising solutions. Besides, the fixed altitude may be hard to maintain due to topography, weather, or task targets.

The complexity of path planning has significantly increased in a non-simplified 3-D space. First, the diverse range of UAV airborne movements in a 3-D space means an exponential growth in potential paths available. Second, the 3-D space places higher requirements on the constraints of UAV maneuverability and obstacle avoidance [47]. With considering multiple optimization objectives, the difficulty of solving the problem increases further. The traditional method of solving a MOP is to transform it into a single objective problem by the weighted sum method. For instance, Xu et al. [59] designed an objective function of the UAV path planning problem by weightedly summing the threat cost and the fuel cost. In order to improve UAV range and avoid detection by enemy radars, Roberge et al. [22] also minimized a weighted-sum objective function of fuel consumption and average flying altitude. Although this weighted sum method has been developed to find achievable paths for path planning, the determination of ideal weights is another consideration. Some researchers tries to find the ideal weights. For example, Masehian and Sedighizadeh [4] constructed an optimization objective function by means of assigning weights to the length and smoothness. The optimal weight of (1, 0.25) was obtained through extensive simulation experiments. Nazarahari et al. [46] proposed a method to balance the order of magnitude of three objective function terms to determine a set of optimal weights. The optimization objective is still the weighted sum of the terms.

However, real-world problems commonly involve several conflicting factors, which are difficult to optimize simultaneously. In general, a solution that is optimal for all objectives does not exist. Thus, it is necessary to generate a set of nondominated solutions for decision-makers, known as multi-objective manner. Ahmed and Deb [27] proposed a path planning method based on nondominated sorting genetic algorithm II (NSGA-II) for discrete grid environments, optimizing both path length and degree of threats. Guo et al. [52] developed a chaotic and sharing-learning particle swarm optimization (CSPSO) algorithm to optimize path length, smoothness, stranding risk, economic cost, and safety. These two methods only were used in 2-D space. Multi-objective path planning in 3-D space has been considered in several studies. Adhikari et al. [53] aimed to simultaneously minimize the fuel and the threat cost as well as finding the shortest path. Sun et al. [30] provided multiple feasible paths for the GEO-UAV bistatic SAR system with different tradeoffs between navigation for UAV and bistatic SAR imaging performance. Peng and Qiu [28] designed a local infeasibility utilization mechanism to optimize the flight path length and the degree of terrain threats. Wan et al. [29] also optimized these two terms according to an improved multi-objective ant colony algorithm combining the preferred search direction and random neighborhood search mechanism. Although these methods achieve competitive results in the experiment, it is still a significant challenge for multi-objective path planning in 3-D space to obtain a well-distributed and diverse set of solutions.

In brief, we propose an improved MOEA/D named MOEA/D-AAWA for short. The proposed MOEA/D-AAWA can obtain a well-distributed and diverse set of nondominated solutions on complex PFs.

3. Model of UAV path planning

3.1. Representation of UAV path

B-spline curve is used to represent the UAV path by given control points [32]. Suppose that the UAV flies from the starting point $p_0 = (x_0, y_0, z_0)$ to the target point $p_{n+1} = (x_{n+1}, y_{n+1}, z_{n+1})$. B-spline curve, i.e., the UAV path, is generated from $(n + 2)$ control points $((x_0, y_0, z_0), \dots, (x_i, y_i, z_i), \dots, (x_{n+1}, y_{n+1}, z_{n+1}))$. The UAV path is composed of $(s + 1)$ path points $((x'_0, y'_0, z'_0), \dots, (x'_j, y'_j, z'_j), \dots, (x'_s, y'_s, z'_s))$, which can be calculated as Eq.(1):

$$\begin{cases} x'_j = \sum_{i=0}^{n+1} x_i B_{i,k}(t_j) \\ y'_j = \sum_{i=0}^{n+1} y_i B_{i,k}(t_j), j = 0, 1, \dots, s, \\ z'_j = \sum_{i=0}^{n+1} z_i B_{i,k}(t_j) \end{cases} \quad (1)$$

where k is the degree of B-spline curve. The larger value of k , the smoother of the curve. $B_{i,k}(t_j)$ is the blending function of the curve and defined recursively in terms of a non-decreasing sequence of real numbers by using Eq.(2)-(4). The

related parameters are set to $k = 3$, $s = 150$.

$$B_{i,1}(t_j) = \begin{cases} 1, & u_i \leq t_j < u_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

$$B_{i,k}(t_j) = \frac{t_j - u_i}{u_{i+k-1} - u_i} B_{i,k-1}(t_j) + \frac{u_{i+k} - t_j}{u_{i+k} - u_{i+1}} B_{i+1,k-1}(t_j), \quad (3)$$

$$u_i = \begin{cases} 0, & i \leq k, \\ i - k, & k < i \leq s - 1, \\ s - k, & i \geq s, \end{cases} \quad (4)$$

where t is a set of discrete points in the range between 0 to $(n + k + 2)$ with a constant step.

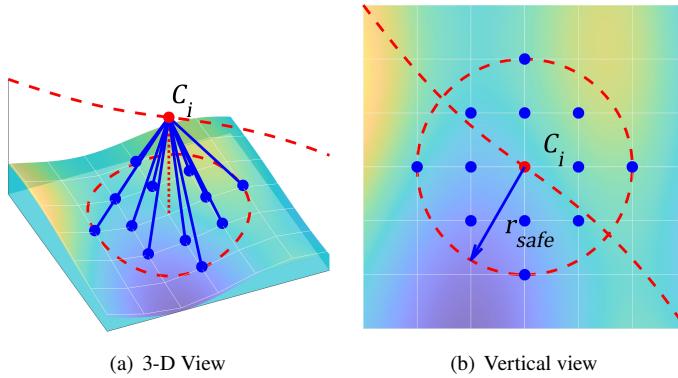


Fig. 1. Terrain grid points within the safe range

3.2. Objective function

In this paper, UAV path planning problem is considered as a bi-objective optimization problem that minimizes overall path length f_1 and degree of threats f_2 as follows.

$$\min F = (f_1, f_2). \quad (5)$$

The length f_1 can be calculated by the Euclidean operator:

$$f_1 = \sum_{i=0}^{s-1} \sqrt{\Delta x_i'^2 + \Delta y_i'^2 + \Delta z_i'^2}, \quad (6)$$

where $\Delta x_i' = x'_{i+1} - x'_i$, $\Delta y_i' = y'_{i+1} - y'_i$, $\Delta z_i' = z'_{i+1} - z'_i$.

The f_2 is used to measure the degree of threats [33]:

$$f_2 = \sum_{i=0}^s \sum_{j=1}^{n_g} \left(\frac{r_{safe}}{r_{i,j}} \right)^2, \quad (7)$$

where r_{safe} is the minimum safe distance between the drone and terrain obstacles. [30]. $r_{i,j}$ is the Euclidean distance between the path point C_i and the terrain grid point G_j . n_g is the number of terrain grid points. As shown in Fig. 1, all terrain grid points G_j within a circle with the path point C_i as its center and the radius of r_{safe} participate in calculating. The ratio of r_{safe} to $r_{i,j}$ is used as risk assessment of the terrain grid points to the path point.

3.3. Constraint

(1) Minimum flight altitude

The UAV path should maintain a safe flight altitude with the terrain. The minimum flight altitude constraint violation of a path point C_i , i.e., $g_1^i(p)$, is calculated according to Eq.(8). The minimum flight altitude constraint violation of a flight path $g_1(p)$ is obtained according to Eq.(9).

$$g_1^i(p) = \begin{cases} 0, & z'_i - z(x'_i, y'_i) > r_{safe}, \\ 1, & \text{otherwise,} \end{cases} \quad (8)$$

$$g_1(p) = \sum_{i=0}^s g_1^i(p), \quad (9)$$

where $z(x'_i, y'_i)$ is the topographic elevation of C_i .

(2) Maximum turning angle

Limited by the inherent maneuverability of the UAV, it can only turn within a certain angle range in the horizontal direction. The constraint violation of maximum turning angle is calculated as shown in Eq.(10) and (11).

$$g_2^i(p) = \begin{cases} 0, & 0 \leq \alpha_i \leq \alpha_{max}, \\ 1, & \text{otherwise,} \end{cases} \quad (10)$$

$$g_2(p) = \sum_{i=1}^{s-1} g_2^i(p), \quad (11)$$

where $g_2^i(p)$ is the constraint violation of the i -th turning angle on the path. $g_2(p)$ is the maximum turning angle constraint of the path; α_i is the i -th turning angle on the path, which calculated by Eq.(12). α_{max} is the maximum turning angle, which is 45° in this paper.

$$\alpha_i = \arccos \frac{\Delta x'_{i-1} \Delta x'_i + \Delta y'_{i-1} \Delta y'_i}{\sqrt{\Delta x'_{i-1}^2 + \Delta y'_{i-1}^2} \sqrt{\Delta x_i'^2 + \Delta y_i'^2}}. \quad (12)$$

(3) Maximum climbing slope

There should also be a limit on the UAV climbing slope in the vertical direction, i.e., maximum climbing slope. The constraint violation of maximum climbing slope is calculated by Eq.(13).

$$g_3^i(p) = \begin{cases} 0, & 0 \leq \beta_i \leq \beta_{max}, \\ 1, & \text{otherwise,} \end{cases} \quad (13)$$

$$g_3(p) = \sum_{i=1}^s g_3^i(p), \quad (14)$$

where $g_3^i(p)$ is the constraint violation of the i -th climbing slope on the path. $g_3(p)$ is the maximum climbing slope constraint of the path. β_i is the i -th climbing slope on the path, which calculated by Eq.(15). β_{max} is the maximum climbing slope, which is 45° in this paper.

$$\beta_i = \arctan \frac{|z'_i - z'_{i-1}|}{\sqrt{(x'_i - x'_{i-1})^2 + (y'_i - y'_{i-1})^2}}. \quad (15)$$

Finally, the constraint violation $g_k(p)$ is normalized to $g'_k(p)$, and then calculate the total constraint violation $g(p)$ according to Eq.(16).

$$g(p) = \sum_{k=1}^3 g'_k(p). \quad (16)$$

4. Design of MOEA/D-AAWA

4.1. Basic idea

Zhang and Li [34] proposed MOEA/D, which decomposes a MOP into a number of scalar optimization subproblems. All subproblems are optimized by collaboratively utilizing their neighborhood information and then collectively approach the PF. A subproblem is usually defined by a weight vector and the neighborhood relationship is usually determined by the Euclidean distance between weight vectors.

However, a set of uniformly distributed weight vectors employed in the classic MOEA/D cannot guarantee the uniformity and diversity of the optimal solutions on the PF, especially for complex PFs, such as discontinuous PFs and PFs with a sharp peak and a low tail. Thus, our main goal is to obtain a uniformly distributed and diverse optimal solution set on the PF of the multi-objective path planning problem.

Qi et al. [31] proposed an adaptive weight adjustment (AWA) strategy, which periodically removes and adds weight vectors. The newly-added weight vector is calculated by the objective function value of the newly-added individual. In our proposed AAWA, we adopt the same subproblem deletion strategy as AWA, and propose a new subproblem adding strategy (i.e., areal subproblem adding strategy). For the newly-added individual, its weight vector is calculated by the objective function value of its neighbors in the current population rather than itself, and thus the newly-added individual can evolve towards the sparser area on PFs.

4.2. WS-transformation

In this section, we introduce WS-transformation to construct new weight vectors.

Common aggregation methods include weighted sum method, Tchebycheff method, and boundary intersection approach, among which the aggregation function of Tchebycheff method for minimization problem is defined as follows:

$$\min_{\mathbf{x} \in \Omega} g^{tc}(\mathbf{x} | \lambda, \mathbf{z}^*) = \min_{\mathbf{x} \in \Omega} \max_{1 \leq i \leq m} \{ \lambda_i (f_i(\mathbf{x}) - z_i^*) \}, \quad (17)$$

where $\Omega \in \mathbb{R}^n$ is the decision space and $\mathbf{x} = (x_1, \dots, x_n)^T \in \Omega$ is a decision variable which represents a solution to the target MOP. $g^{tc}(\mathbf{x} | \lambda, \mathbf{z}^*)$ is Tchebycheff value. $\mathbf{z}^* = (z_1^*, \dots, z_m^*)^T$ is the reference point, i.e., $z_i^* = \min \{ f_i(\mathbf{x}) | \mathbf{x} \in \Omega \}$, $i = 1, \dots, m$. $f_i(\mathbf{x})$ is the objective function of \mathbf{x} in i -th dimension of objective space for all $i = 1, \dots, m$. $\lambda = (\lambda_1, \dots, \lambda_m)^T$ is the weight vector, $\lambda_i \geq 0$ and $\sum_{i=1}^m \lambda_i = 1$, $i = 1, 2, \dots, m$. m is the dimension of the objective space.

Fig. 2 shows the individual convergence in a bi-objective space under the Tchebycheff approach. λ is the weight vector. The individual approaches the PF along the solution mapping vector λ' . l_1 and l_2 are two different contour lines of λ' . In the objective space, the Tchebycheff contour line is right-angled serrated, implying that Tchebycheff values of all points on l_1 are equal and l_2 is the same. Besides, the Tchebycheff values of all points on l_1 are less than l_2 .

Based on the MOEA/D using the Tchebycheff method, WS-transformation is introduced to describe how the weight vector and solution mapping vector are transformed. The relevant formula is as follows:

$$\lambda = WS(\lambda') = WS(F - \mathbf{z}^*) = \left(\frac{\frac{1}{f_1 - z_1^*}}{\sum_{i=1}^m \frac{1}{f_i - z_i^*}}, \dots, \frac{\frac{1}{f_m - z_m^*}}{\sum_{i=1}^m \frac{1}{f_i - z_i^*}} \right), \quad (18)$$

where λ' is the solution mapping vector, which is the mapping of the weight vector in the objective space. F is the objective function value. For an individual with a known objective function value F , its solution mapping vector can be calculated by $\lambda' = F - \mathbf{z}^*$, and then the corresponding weight vector λ can be obtained by applying WS-transformation on λ' .

4.3. Adaptive areal weight adjustment

AAWA includes a subproblem deletion strategy and an areal subproblem adding strategy. In the objective space, the subproblem deletion strategy deletes overcrowded subproblems of the current population Pop . Meanwhile, the areal subproblem adding strategy selects sparse individuals from the external elite population EP , which store a specific number of nondominated individuals during the iteration process. The objective function value of selected sparse

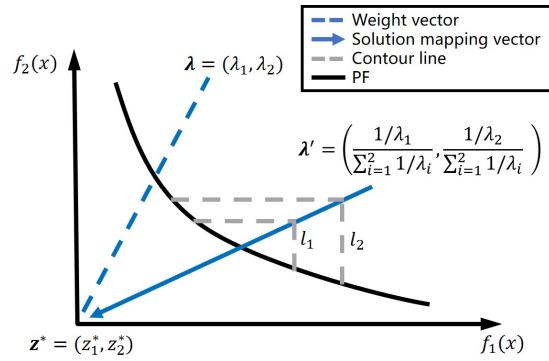


Fig. 2. Individual convergence in a bi-objective space under the Tchebycheff approach

individuals' neighbors is used to calculate their solution mapping vectors. Then the corresponding weight vector can be obtained by the WS-transformation.

(1) Subproblem deletion strategy

The sparsity level of individuals in Pop among Pop is evaluated by vicinity distance [35] as follows:

$$SL(x^i, \text{Pop}) = \prod_{j \in N_k^i} L_{i,j}, x^i \in \text{Pop}, \quad (19)$$

where $L_{i,j}$ is Euclidean distance between the individual x^i and its neighbor x^j . N_k^i consists of indexes of the k -nearest individuals of x^i among Pop , and k is usually set to the objective function dimension. The smaller sparsity level implies the greater similarity between x^i and its neighbors.

The process of deleting subproblems is as follows. First, the sparsity level of each individual in Pop among Pop is calculated by Eq.(19). Then, the individual with the minimum sparsity level form Pop is removed and its weight vector is deleted.

As shown in Fig. 3(a), there is a population $\text{Pop} = \{x^1, x^2, x^3, x^4\}$. Since the distance between x^1 and x^3 is always greater than the distance between x^2 and x^3 in the objective space, the $SL(x^1, \text{Pop})$ is always greater than $SL(x^2, \text{Pop})$. Similarly, the $SL(x^4, \text{Pop})$ is always greater than $SL(x^3, \text{Pop})$. Thus, boundary points as valuable individuals in the population will not be deleted. We just need to compare the sparsity level of x^2 and x^3 . Apparently, $SL(x^2, \text{Pop}) = L_{2,1}L_{2,3} < SL(x^3, \text{Pop}) = L_{3,2}L_{3,4}$, thus delete x^2 and its weight vector.

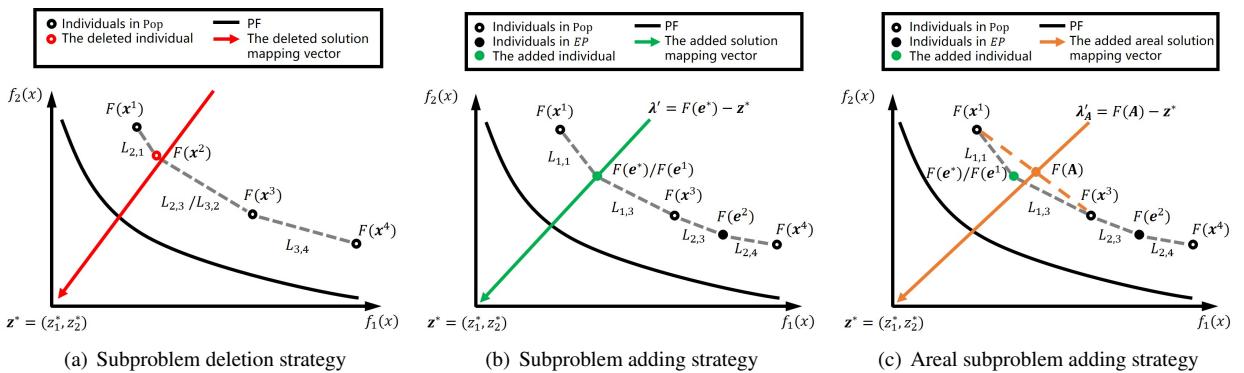


Fig. 3. Adaptive weight adjustment.

(2) Areal subproblem adding strategy

After deleting subproblems from Pop , the same number of new subproblems, which are related to new individuals and their weight vectors, are added to Pop . According to the sparsity, the newly-added individuals are selected from

EP , which is used to store elite individuals (nondominated solutions). The size of EP is set to $1.5 \times |Pop|$. More specifically, the sparsity level of each individual in EP among Pop is firstly calculate according to Eq.(20). The elite individual with higher sparsity level, has greater dissimilarity from its neighbors in Pop . Then the elite individual with the maximum sparsity level is added to Pop and its weight vector will be constructed.

$$SL(e^i, Pop) = \prod_{j \in N_k^i} L_{i,j}, e^i \in EP, \quad (20)$$

where $L_{i,j}$ is Euclidean distance between the individual e^i and its neighbor x^j . N_k^i consists of indexes of the k -nearest individuals of e^i among Pop , and k is usually set to the objective function dimension.

As shown in Fig. 3(b), after deleting x^2 from Pop , $Pop = \{x^1, x^3, x^4\}$. Supposing $EP = \{e^1, e^2\}$, x^1 and x^3 are neighbors of e^1 , x^3 and x^4 are neighbors of e^2 . Apparently, $SL(e^1, Pop) = L_{1,1}L_{1,3} > SL(e^2, Pop) = L_{2,3}L_{2,4}$, thus e^1 is selected as added individual, i.e., e^* .

AWA constructs a new weight vector based on the objective function value of the added individual. The following calculation method can be obtained by substituting the objective function value $F(e^*)$ into Eq.(18).

$$\lambda = WS(\lambda') = WS(F(e^*) - z^*) = \left(\frac{\frac{1}{f_1(e^*) - z_1^*}}{\sum_{i=1}^m \frac{1}{f_i(e^*) - z_i^*}}, \dots, \frac{\frac{1}{f_m(e^*) - z_m^*}}{\sum_{i=1}^m \frac{1}{f_i(e^*) - z_i^*}} \right). \quad (21)$$

However, when the added individual e^* has a high similarity with one of its neighbors, the added subproblem also has a high similarity with the neighbor subproblem. This cannot guarantee the population diversity. As shown in Fig. 4, suppose that n^1 and n^2 are neighbors of e^* , and $F(n^1)$ and $F(n^2)$ are objective function point of the neighbors respectively. n^1 and n^2 evolve towards P_1 and P_2 on the PF respectively. When e^* and n^1 are close in the objective space, that is, there is a high similarity between e^* and n^1 , and the newly constructed solution mapping vector λ' also has a high similarity with the solution mapping vector of n^1 . Then that e^* evolves towards P_3 which is close to P_1 is adverse to population diversity.

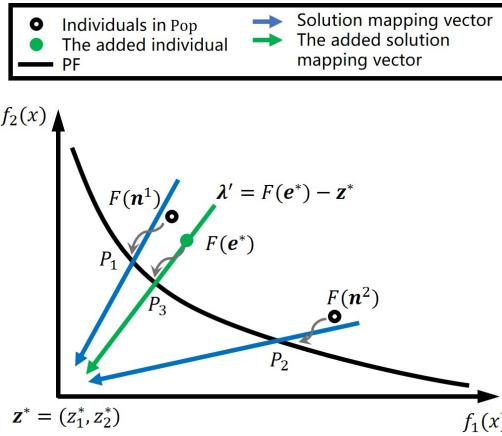


Fig. 4. The deficiency of subproblem adding strategy of AWA

To solve the above issue, we propose an areal subproblem adding strategy, which constructs a new weight vector based on the objective function values of the added individual's neighbors. In the m -dimensional objective space, k neighbors form an area, i.e., A . The centroid coordinate value of A can construct a new weight vector, which is called "the areal weight vector". To be specific, according to Eq.(22), we firstly calculate the centroid coordinate of area A composed of k neighbors. Then, we can obtain the corresponding solution mapping vector $\lambda'_A = F(A) - z^*$. Finally, we can construct the new weight vector based on WS-transformation, which can be defined as:

$$F(A) = \frac{1}{n} \sum_{i=1}^m F(n^i), \quad (22)$$

Algorithm 1 Adaptive areal weight adjustment

Input: current population Pop ; current weight vector set; external elite population EP ; the number of adjusting subproblems nus

Output: the adjusted population Pop ; adjusted weight vector set

- 1: **for** $i \leftarrow 1$ to $|Pop|$ **do**
- 2: **if** $g^{te}(\mathbf{x}^j | \lambda^i, \mathbf{z}^*) < g^{te}(\mathbf{x}^i | \lambda^i, \mathbf{z}^*)$, $j = 1, 2, \dots, |Pop| + |EP|$ **then**
- 3: $\mathbf{x}^i \leftarrow \mathbf{x}^j, F(\mathbf{x}^i) \leftarrow F(\mathbf{x}^j)$
- 4: **end if**
- 5: **end for**
- 6: **for** $k \leftarrow 1$ to nus **do**
- 7: Calculate sparsity $SL(\mathbf{x}, Pop), \mathbf{x} \in Pop$ by Eq.(19);
- 8: Remove the individual and its weight vector with the minimum sparsity $SL(\mathbf{x}, Pop)$;
- 9: **end for**
- 10: Remove the individuals in EP which are dominated by the individuals in Pop ;
- 11: **for** $k \leftarrow 1$ to nus **do**
- 12: Calculate sparsity $SL(e, Pop), e \in EP$ by Eq.(20);
- 13: Add the individual with the maximum sparsity $SL(e, Pop)$ to the Pop ;
- 14: Calculate the areal weight vector by Eq.(22)-(23) and add it to the weight vector set;
- 15: **end for**

$$\lambda_A = WS(\lambda'_A) = WS(F(A) - \mathbf{z}^*) = \left(\frac{1}{\sum_{i=1}^m \frac{1}{f_i(A) - z_i^*}}, \dots, \frac{1}{\sum_{i=1}^m \frac{1}{f_i(A) - z_i^*}} \right), \quad (23)$$

where n^i is i -th neighbor of the added individual among Pop . $F(n^i)$ is the objective function values of the i -th ($i = 1, \dots, m$) neighbor of the added individual e^* . $F(A)$ is the centroid coordinate of A , which is formed by k neighbors in m -dimensional objective space. In this paper, set $k = m$. λ_A is the areal weight vector.

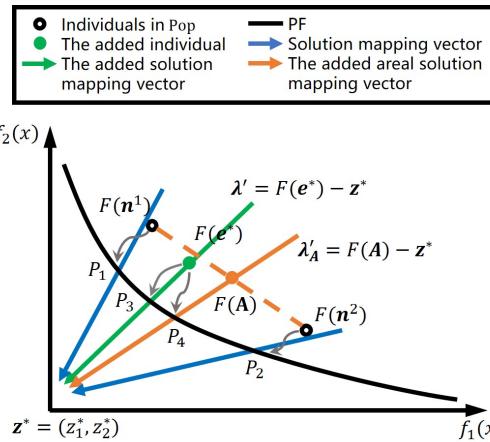


Fig. 5. Comparison of two subproblem adding strategies in terms of the added individuals' evolution

As shown in Fig. 3(c), after electing the added individual e^* , its two adjacent neighbors \mathbf{x}^1 and \mathbf{x}^3 in the Pop form an area A without any other individuals. The areal centroid $F(A)$ is represented as the midpoint of the line segment composed of the two neighbors. λ_A points from $F(A)$ to the reference point, providing evolutionary direction for the selected individual. Finally, the new weight vector can be calculated according to WS-transformation.

To facilitate a great understanding of the method, we take a bi-objective minimization problem as an example as shown in Fig. 5. \mathbf{n}^1 and \mathbf{n}^2 are the two-nearest neighbors of the added individual e^* . The midpoint of the line segment composed of points $F(\mathbf{n}^1)$ and $F(\mathbf{n}^2)$ is the centroid $F(A)$. λ'_A is the solution mapping vector by our areal subproblem

adding strategy, guiding the added individual e^* to evolve towards P_4 on the PF. And the solution mapping vector λ' constructed by the objective function value of the added individual e^* guides e^* to evolve towards P_3 . The uniformity and diversity of solutions $\{P_1, P_2, P_4\}$ on the PF is better than that of solutions $\{P_1, P_2, P_3\}$. The reason is that the centroid of the area formed by neighbors is closer to the sparsest area of the current population, so that the constructed areal weight vector or its solution mapping vector can more accurately guide the evolution of the added individual to the sparse area on the PF. In addition, the areal weight vector is constructed by the centroids of sparse area. Regardless of where the newly-added individual e^* is located in the area, the evolutionary direction provided by the areal weight vector navigates to the sparsest part of the area. The pseudocode of the adaptive areal weight adjustment strategy is shown in Algorithm 1.

4.4. MOEA/D-AAWA

Based on the above analysis, the pseudocode of the MOEA/D-AAWA is shown in Algorithm 2 and the main steps are as follows.

Step 1 (Initialization): initialize Pop , EP , neighborhood list B and related parameters (line 1, lines 2-5). Generate a set of uniformly distributed solution mapping vectors. Obtain initial weight vectors according to WS-transformation (line 2).

Step 2 (Allocate computing resources): the pseudocode of the computing resources allocation strategy is shown in Algorithm 3 [36]. A utility function U is defined and computed for each subproblem. The indexes of evolutionary individuals I are selected from Pop according its utility function (line 8).

Step 3 (Update): for each evolutionary individual in I , its offspring y is generated by the SBX operator and the polynomial mutation operator [38] shown as Eq.(24)-(27) (lines 11-13). Update the reference z^* (line 14). Update the Pop by constraint-domination principle (CDP) shown in Algorithm 4 [39] (line 15). The definition of constraint-domination is given as follows. A solution x is said to constraint-dominate another solution y , if any of the following criterion is satisfied: 1) x is feasible and y is infeasible; 2) Both x and y are infeasible and x has a smaller degree of constraint violation than y ; 3) Both x and y are feasible and x dominates y .

$$\mathbf{y}'_k = 0.5 \times [(1 + \beta) \mathbf{x}_k^{r_1} + (1 - \beta) \mathbf{x}_k^{r_2}], \quad (24)$$

$$\beta = \begin{cases} (\text{rand} \times 2)^{1/(1+\eta_c)}, \text{rand} \leq 0.5, \\ [1/(2 - \text{rand} \times 2)]^{1/(1+\eta_c)}, \text{otherwise}, \end{cases} \quad (25)$$

$$\mathbf{y}_k = \mathbf{y}'_k + \boldsymbol{\delta}_k, \quad (26)$$

$$\boldsymbol{\delta}_k = \begin{cases} (2 \times \text{rand})^{1/(\eta_m+1)} - 1, \text{if rand} \leq 0.5, \\ 1 - [2(1 - \text{rand})]^{1/(\eta_m+1)}, \text{otherwise}, \end{cases} \quad (27)$$

where rand is a uniformly random number from (0, 1); η_c and η_m are the distribution index of the SBX operator and the polynomial mutation operator.

Step 4 (Adaptive areal weight adjustment): update EP by the new generation of offspring according to the vicinity distance based nondominated sorting (line 18). Periodically delete crowded subproblems and add new subproblems according to Algorithm 1 (line 20). Update B after adjusting Weight vectors (line 21).

Step 5 (Stop criteria): go to Step 2 and repeat the above steps until termination criteria is satisfied (lines 6-25).

5. Computational experiments

In this section, we evaluate the performance of MOEA/D-AAWA in synthetic (mountain and urban environments) and realistic scenarios. The proposed algorithm and the three compared algorithms are coded in MATLAB, and run on a PC computer with Core i5-10400F 2.90GHz CPU, 16G memory, and Windows 10 operating system.

Algorithm 2 MOEA/D-AAWA

Input: the size of current population N ; objective function dimension m ; the neighborhood size T ; the probability of selecting mate subproblem from its neighborhood δ ; the ratio of iteration times to evolve with only MOEA/D r_{evol} ; individual evolution ratio r_{pop} ; update number n_r ; the maximal number of subproblems needed to be adjusted nus ; the iteration interval of utilizing computing resource allocation strategy g_r ; the iteration interval of utilizing the adaptive areal weight vector adjustment strategy g_w

Output: population Pop and its objective function values

- 1: Initialize population randomly $Pop = \{\mathbf{x}^i, i = 1, \dots, N\}$; for each \mathbf{x}^i , evaluate $F(\mathbf{x}^i) = (f_1(\mathbf{x}^i), \dots, f_m(\mathbf{x}^i))^T$;
- 2: Initialize the weight vectors $\{\lambda^i, i = 1, \dots, N\}$ by applying the WS-transformation on uniformly distributed solution mapping vectors;
- 3: Calculate the Euclidean distances between any two weight vectors. For each $i = 1, \dots, N$, set the neighborhood list of the i -th subproblem as $B(i) = (i_1, i_2, \dots, i_T)^T$, where i_1, i_2, \dots, i_T are the indexes of T closest weight vectors to λ^i ;
- 4: Initialize the reference $\mathbf{z}^* = (z_1^*, \dots, z_m^*)^T$ by setting $z_i^* = \min \{f_i(\mathbf{x}^1), \dots, f_i(\mathbf{x}^N)\} - 10^{-7}$;
- 5: Set $g = 1$, $EP = \emptyset$ and $U^i = 1, i = 1, \dots, N$;
- 6: **while** termination criteria is not satisfied **do**
- 7: **if** $mod(g, g_r) == 0$ **then**
- 8: Allocate computing resources by Algorithm 3 and obtain indexes list of evolutionary individuals I ;
- 9: **end if**
- 10: **for** $i \in I$ **do**
- 11: Select mating scope: $P = \begin{cases} B(i), rand < \delta \\ \{1, 2, \dots, N\}, otherwise \end{cases}$, where $rand$ is a uniformly random number from $(0, 1)$;
- 12: Reproduce: randomly select two indexes r_1 and r_2 from P , and then generate a new individual \mathbf{y} from \mathbf{x}^{r_1} and \mathbf{x}^{r_2} by the SBX operator and the polynomial mutation operator;
- 13: Repair \mathbf{y} : if an element of \mathbf{y} is out of the boundary, randomly select its value inside the boundary;
- 14: Update the reference \mathbf{z}^* : for each $j = 1, \dots, m$, if $z_j^* > f_j(\mathbf{y})$, then set $z_j^* = f_j(\mathbf{y}) - 10^{-7}$;
- 15: Update the population Pop by Algorithm 4;
- 16: **end for**
- 17: **if** $g \geq r_{evol} \times g_{max}$ **then**
- 18: Update the external population EP by the new generation of offspring according to the vicinity distance based nondominated sorting;
- 19: **if** $mod(g, g_r) == 0$ **then**
- 20: Adjust weight vectors by Algorithm 1;
- 21: Update neighborhood: for each $i = 1, \dots, N$, find the T closest weights to λ^i and build the new $B(i)$.
- 22: **end if**
- 23: **end if**
- 24: Set $g = g + 1$.
- 25: **end while**

5.1. Experimental setting

Twenty-four experimental cases are designed in this paper. The first environment is mountain environments. Ten cases (C1-C10) are generated under the mountain environment in the range of $200 \text{ km} \times 200 \text{ km}$, whose numbers of mountain obstacles are 2-20 with a step of 2.

The mountain environment consists of original terrain and mountain obstacles. The elevation matrix of original terrain is constructed by Eq.(28) [41].

$$z_1(x, y) = \sin(y + a) + b\sin(x) + c\cos(y) + d\cos\left(e\sqrt{x^2 + y^2}\right) + f\sin\left(f\sqrt{x^2 + y^2}\right), \quad (28)$$

Algorithm 3 Allocation of computing resources

Input: The objective function values in the last allocation of computing resources f_{old} ; Current objective function values f ; Individual evolution ratio r_{pop}

Output: I

- 1: Calculate the increment of the utility function: $\Delta^i = \frac{f_{old}(\mathbf{x}^i) - f(\mathbf{x}^i)}{f_{old}(\mathbf{x}^i)}$;
- 2: Update the utility function: $U^i = \begin{cases} 1, & \Delta^i > 0.001 \\ \left(0.95 + 0.05 * \frac{\Delta^i}{0.001}\right) U^i, & \text{otherwise} \end{cases}$
- 3: Select the subproblems: set $I = \emptyset$ and select all indexes of the m boundary weight vectors. Choose other $[N \times r_{pop}] - m$ indexes using 2-tournament selection [37] according to U^i , and add them to I .

Algorithm 4 Update scheme of population

- 1: Set $c = 0$;
- 2: **while** $c < n_r$ or $P \neq \emptyset$ **do**
- 3: Randomly pick an index j from P ;
- 4: **if** $V(\mathbf{y}) < V(\mathbf{x}^j)$ **then**
- 5: $\mathbf{x}^j = \mathbf{y}, f(\mathbf{x}^j) = f(\mathbf{y}), V(\mathbf{x}^j) = V(\mathbf{y}), c = c + 1$
- 6: **else if** $V(\mathbf{y}) = V(\mathbf{x}^j) = 0$ **then**
- 7: **if** $g^{te}(\mathbf{y}|\lambda^j, z^*) < g^{te}(\mathbf{x}^j|\lambda^j, z^*)$ **then**
- 8: $\mathbf{x}^j = \mathbf{y}, f(\mathbf{x}^j) = f(\mathbf{y}), V(\mathbf{x}^j) = V(\mathbf{y}), c = c + 1$
- 9: **end if**
- 10: **end if**
- 11: **end while**

where (x, y) is the coordinate of a point on the horizontal plane. $z_1(x, y)$ is the corresponding terrain height. a, b, c, d, e, f are the terrain coefficients, and a combination of them can simulate different topographic features. In this paper, the terrain coefficients are set to $a = 3\pi, b = 0.1, c = 0.3, d = 0.9, e = 0.5$ and $f = 0.5$.

The elevation matrix of 3-D mountain obstacles is constructed by Eq.(29).

$$z_2(x, y) = \sum_{i=1}^k h(i) * \exp\left(-\frac{x - x_0(i)}{a(i)} - \frac{y - y_0(i)}{b(i)}\right), \quad (29)$$

where $z_2(x, y)$ is the corresponding mountain height. k is the number of mountain obstacles. $h(i)$ is the height of i -th mountain obstacle. $(x_0(i), y_0(i))$ is the coordinate of the center of the i -th mountain obstacle on the horizontal plane. $a(i)$ and $b(i)$ are the corresponding slope parameters along the x axis and y axis directions. By adjusting $h(i)$, $a(i)$ and $b(i)$, mountain obstacles with various contours can be constructed.

Then, the elevation matrix of mountain environments can be formed by merging the elevation matrix of original terrain and mountain obstacles according to Eq.(30).

$$z_m(x, y) = \max \{z_1(x, y), z_2(x, y)\}. \quad (30)$$

The second environment is urban environments. Ten cases (C11-C20) are generated under the urban environment in the range of 2000 m × 2000 m, whose numbers of urban obstacles are 2-20 with a step of 2.

We model 3-D urban obstacles as prisms. To further keep the safety of the UAV during the flight, we set the protected zone for each obstacle, whose width is defined as a safe distance r_{safe} . The relevant elevation matrix of urban environments can be calculated as follows. For each urban obstacle, if (x, y) is within k -th urban obstacle, set $z_u^k(x, y) = h_k$, otherwise 0. h_k is the height of i -th urban obstacle and $z_u^k(x, y)$ is elevation matrix of k -th urban obstacle. And then the elevation matrix of urban environments can be calculated as $z_u(x, y) = \max_k \{z_u^k(x, y)\}$.

To make the experiment more convincing, four cases (C21-C24) are generated under the realistic scenarios in the range of $10 \text{ km} \times 10 \text{ km}$, and the spatial resolution is 5 m. The latitude and longitude ranges of the four cases are shown in Table 1.

Table 1
The latitude and longitude ranges of C21-C24

case	the range of longitude and latitude
C21	(78.680000°E-78.608046°E, 34.300000°N-34.390212°N)
C22	(78.500000°E-78.608046°E, 33.670000°N-33.760212°N)
C23	(78.410000°E-78.518135°E, 33.760000°N-33.850212°N)
C24	(78.190000°E-78.297817°E, 33.500000°N-33.590212°N)

NSGA-III [40], C-MOEA/D [40] and MOEA/D-AWA [31] are used for the comparison. The parameters of this paper are set as follows:

- 1) Each algorithm runs over 30 independent times, and stops after k function evaluations. The times of function evaluation k and number of path control points (excluding starting and target points) DOP in all cases are given in Table 2.
- 2) The safe distance r_{safe} is set to 200 m, 10 m and 20 m in mountain environments (C1-C10), urban environments (C11-C20) and realistic scenarios (C21-C24) respectively;
- 3) The population size N is set to 20, the neighborhood size T is set to $0.1 \times N$;
- 4) The proposed MOEA/D-AWA has the same parameters as MOEA/D-AWA: $\eta_c = 20$, $\eta_m = 1$, $n_r = 2$, $r_{evol} = 0.8$, $r_{pop} = 0.2$, $nus = 0.1 \times N$, $g_r = 50$, $g_w = 100$;
- 5) The rest of the parameters in NSGA-III and C-MOEA/D remain the same as in the original papers.

Table 2
The values of times of function evaluation k and numbers of path control points

case	k	DOP
C1-C3,C11-C13	10000	2
C4-C6,C14-C16	12000	3
C7-C9,C17-C19	14000	4
C10,C20	16000	5
C21-C24	10000	4

5.2. Performance metrics

Due to the unknown PFs of path planning problems, hypervolume (HV) [42] and pure diversity (PD) [43] are used to evaluate the performance of the four algorithms in this paper, as shown in Eq.(31)-(33). In addition, the larger HV and PD implies the better convergence and diversity of PFs.

$$HV = \text{hypervolume} \left(\bigcup_{i=1}^{|PF|} s_i \right), \quad (31)$$

$$PD(X) = \max \{ PD(X - s_i) + d(s_i, X - s_i) \}, \quad (32)$$

$$d(s, X) = \min_{s_i \in X} \{ \text{dissimilarity}(s, s_i) \}, \quad (33)$$

where s_i is the solution in the Pareto optimal solution set. $d(s, X)$ represents the dissimilarity between the solution s and a community X . When calculating the HV, first normalize the values of multiple objective functions, and then calculate the area of the polygon formed by the values and the reference point (1,1). Moreover, when there are no feasible solutions in the population, set HV and PD to 0.

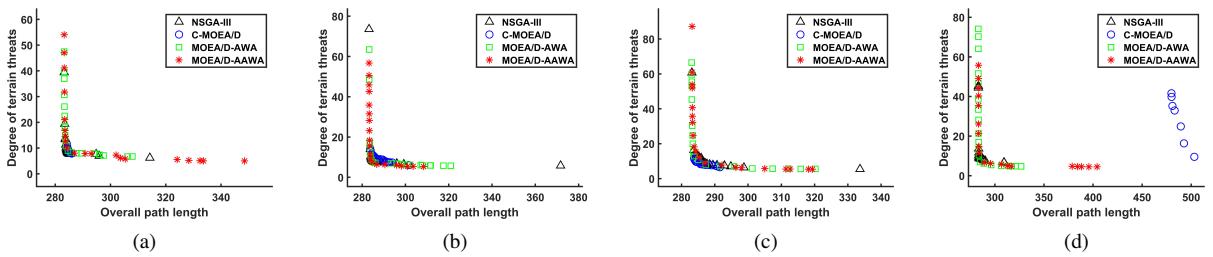
5.3. Results and analyses

5.3.1. Experiments under mountain environments

Table 3

Mean and Std values of HV and PD in mountain environments

Case	Number of threats	Metric	Algorithm			
			NSGA-III	C-MOEA/D	MOEA/D-AWA	MOEA/D-AAWA
C1	2	HV	0.9799±0.0039 (3)	0.7968±0.0226 (4)	0.9799±0.0091 (2)	0.9828±0.004 (1)
		PD($\times 10^4$)	1.5202±0.1658 (3)	0.5881±0.144 (4)	2.2678±0.7973 (2)	2.4318±0.7354 (1)
C2	4	HV	0.9663±0.0154 (3)	0.7895±0.0297 (4)	0.9723±0.0137 (2)	0.9763±0.0092 (1)
		PD($\times 10^4$)	1.4563±0.336 (3)	0.5424±0.1416 (4)	2.1858±1.024 (2)	2.1922±0.6543 (1)
C3	6	HV	0.9663±0.0071 (2)	0.7919±0.0512 (4)	0.9694±0.0139 (1)	0.9656±0.0165 (3)
		PD($\times 10^4$)	1.8452±0.3764 (3)	0.4038±0.1045 (4)	2.6521±1.3831 (1)	2.4123±1.0745 (2)
C4	8	HV	0.9729±0.0161 (3)	0.7696±0.042 (4)	0.977±0.0069 (2)	0.9789±0.0074 (1)
		PD($\times 10^4$)	1.5276±0.2884 (3)	0.5397±0.2363 (4)	2.2935±0.7193 (2)	2.5778±1.223 (1)
C5	10	HV	0.94±0.1782 (3)	0.7779±0.0578 (4)	0.9749±0.0119 (2)	0.9793±0.0068 (1)
		PD($\times 10^4$)	1.6526±0.4933 (3)	0.5266±0.2524 (4)	2.8152±0.8421 (1)	2.6343±0.6803 (2)
C6	12	HV	0.9736±0.0087 (3)	0.7773±0.06 (4)	0.9774±0.0086 (2)	0.9791±0.0063 (1)
		PD($\times 10^4$)	1.7189±0.4173 (3)	0.5836±0.2936 (4)	2.5545±0.7485 (1)	2.4437±0.7408 (2)
C7	14	HV	0.7768±0.3957 (3)	0.7384±0.0438 (4)	0.9799±0.0073 (1)	0.976±0.0259 (2)
		PD($\times 10^4$)	1.3276±0.7345 (3)	0.5986±0.2677 (4)	2.7111±0.8724 (2)	2.7573±1.3857 (1)
C8	16	HV	0.7737±0.3937 (3)	0.7045±0.0978 (4)	0.9715±0.012 (1)	0.9664±0.0227 (2)
		PD($\times 10^4$)	1.6206±0.9349 (3)	0.8282±0.3471 (4)	2.9873±1.0119 (2)	3.3547±1.4623 (1)
C9	18	HV	0.7918±0.3625 (3)	0.6986±0.1362 (4)	0.9593±0.0308 (2)	0.9629±0.0299 (1)
		PD($\times 10^4$)	1.6448±0.8734 (3)	1.0799±0.5171 (4)	3.1508±1.2755 (2)	3.3412±0.9261 (1)
C10	20	HV	0.4291±0.4698 (4)	0.6873±0.1092 (3)	0.8468±0.2903 (2)	0.9423±0.0598 (1)
		PD($\times 10^4$)	1.0252±1.4117 (3)	1.0026±0.5005 (4)	2.6693±1.4752 (2)	3.2183±1.4812 (1)
Overall ranking			(60)	(79)	(34)	(27)

**Fig. 6.** Final nondominated solutions in mountain environments. (a): C3; (b): C6; (c): C9; (d): C10.

Under the mountain environment, we generate ten cases to verify the effectiveness of our proposed method compared with NSGA-III, C-MOEA/D and MOEA/D-AWA as shown in Table 3. Four algorithms are conducted on the above-mentioned ten cases over 30 independent runs. The mean and standard deviation (Std) of HV and PD are represented in Table 3. The number within a bracket denotes the ranking of each algorithm for each case. For each case, the best result is in boldface. For each algorithm, its overall ranking is equal to the sum of rankings for all cases. A smaller overall ranking indicates a better result. In the ten cases (i.e., C1-C10), our proposed algorithm achieves the highest overall ranking, slightly higher than MOEA/D-AWA, followed by NSGA-III and C-MOEA/D, indicating that weight adjustment strategies can effectively improve algorithm performance. The proposed algorithm obtains the highest mean HV values on seven out of ten cases. Meanwhile, the proposed algorithm also achieves the highest mean PD values on seven out of ten cases. It manifests that the proposed algorithm can obtain the optimal convergence and diversity in most mountain environments.

UAV 3-D path planning based on MOEA/D with adaptive areal weight adjustment

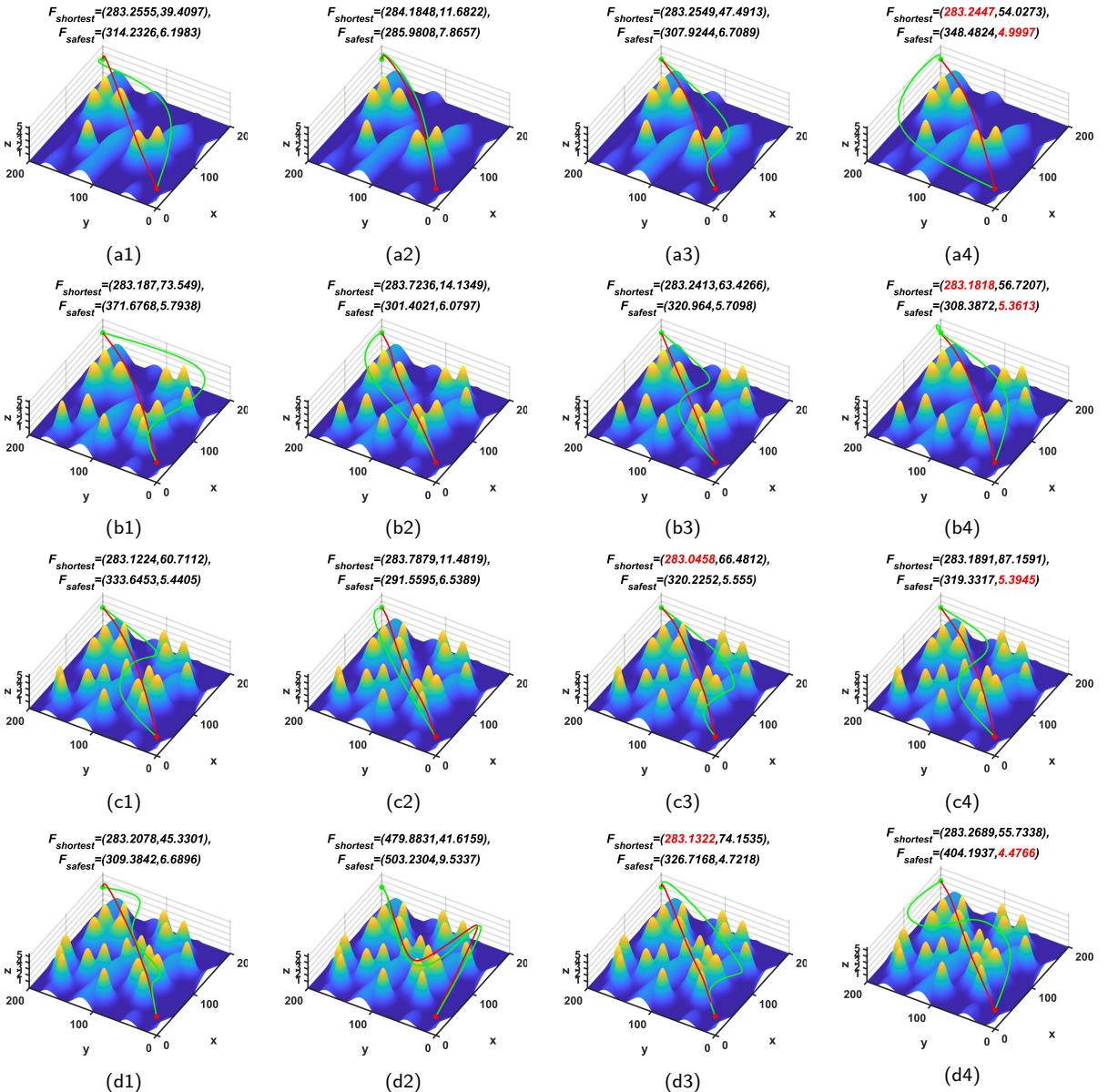


Fig. 7. The shortest path (red path) and the safest path (green path) in mountain environments. (a1)-(a4): C3; (b1)-(b4): C6; (c1)-(c4): C9; (d1)-(d4): C10. (a1)-(d1): NSGA-III; (a2)-(d2): C-MOEA/D; (a3)-(d3): MOEA/D-AWA; (a4)-(d4): MOEA/D-AAWA.

For each run, the score of each algorithm is equal to the sum of the rankings of HV and PD. Fig. 6 presents the final nondominated solutions of the four algorithms at the run with the median score. Four sets of final nondominated solutions in cases C3, C6, C9 and C10 are selected to visually compare the performance of the four algorithms. It can be observed that the final nondominated solutions in the objective space have a sharp peak and a low tail. Thus, it is difficult to achieve the shortest and safest path under the mountain environments. From Fig. 6, NSGA-III and C-MOEA/D can hardly reach these sharp peak and low tail areas. Conversely, the proposed algorithm can find more solutions in these areas. It further indicates that the proposed algorithm has a good performance in solution diversity. The reasons are given as follows. On the one hand, the proposed AAWA can guide individuals to approach such sparse areas according to the current population distribution in the objective space. On the other hand, the core idea of

MOEA/D is decomposition and cooperation. Individuals evolve cooperatively based on neighborhood relationships. And diverse neighborhood with our strategy provides more valuable information for evolution.

Fig. 7 presents the shortest path (red path) and the safest path (green path) at the run with the median score in cases C3, C6, C9 and C10. For each algorithm, its objective function values of the two paths are displayed at the top of the figure, where F_{shortest} is the objective function value of the shortest path and F_{safest} is that of the safest path. For each case, the shortest length and the lowest threat degree among the four algorithms are marked in red. From Fig. 7, in most path planning cases, the proposed algorithm can provide decision-makers with the shorter or safer paths, even both at the same time. It implies the superiority of the proposed algorithm again.

5.3.2. Experiments under urban environments

Table 4

Mean and Std values of HV and PD in urban environments

Case	Number of threats	Metric	Algorithm			
			NSGA-III	C-MOEA/D	MOEA/D-AWA	MOEA/D-AAWA
C11	2	HV	0.884±0.0332 (3)	0.8464±0.028 (4)	0.8962±0.0161 (2)	0.8966±0.0134 (1)
		PD($\times 10^4$)	5.0693±0.8773 (3)	4.9734±0.7184 (4)	7.405±1.4496 (2)	7.6929±1.7622 (1)
C12	4	HV	0.8915±0.027 (3)	0.8394±0.0254 (4)	0.8969±0.0151 (2)	0.9004±0.0169 (1)
		PD($\times 10^4$)	5.0002±0.5642 (4)	5.4638±1.307 (3)	7.4205±1.3327 (1)	7.3192±1.485 (2)
C13	6	HV	0.9013±0.0214 (2)	0.835±0.0384 (4)	0.8994±0.0169 (3)	0.904±0.0131 (1)
		PD($\times 10^4$)	5.3669±0.6157 (3)	4.7638±0.6612 (4)	7.7798±1.4584 (2)	8.0438±2.0033 (1)
C14	8	HV	0.8615±0.0681 (3)	0.8087±0.0607 (4)	0.894±0.0385 (2)	0.9046±0.0226 (1)
		PD($\times 10^4$)	4.9181±1.3454 (3)	4.7567±0.9142 (4)	8.2094±1.766 (2)	8.5427±1.7508 (1)
C15	10	HV	0.8322±0.1627 (3)	0.7964±0.1025 (4)	0.9032±0.0162 (1)	0.8994±0.03 (2)
		PD($\times 10^4$)	4.7939±1.5326 (3)	4.7188±1.1545 (4)	8.8987±1.8765 (2)	9.4026±3.8884 (1)
C16	12	HV	0.8139±0.2264 (3)	0.7893±0.0566 (4)	0.8914±0.0317 (2)	0.9063±0.0221 (1)
		PD($\times 10^4$)	4.578±1.6819 (3)	4.0444±0.7341 (4)	8.5236±2.043 (1)	8.3838±2.1779 (2)
C17	14	HV	0.4013±0.4098 (4)	0.6122±0.257 (3)	0.6203±0.4141 (2)	0.6874±0.387 (1)
		PD($\times 10^4$)	1.6159±1.8464 (4)	3.7112±4.1819 (3)	6.0677±4.4531 (2)	6.4354±4.0017 (1)
C18	16	HV	0.2856±0.4119 (4)	0.5329±0.3396 (3)	0.5373±0.4467 (2)	0.7378±0.3379 (1)
		PD($\times 10^4$)	1.3799±2.2004 (4)	2.6108±1.7486 (3)	6.0971±5.5594 (2)	7.8408±4.775 (1)
C19	18	HV	0.3079±0.4123 (4)	0.5004±0.3232 (1)	0.4594±0.4404 (3)	0.4674±0.4471 (2)
		PD($\times 10^4$)	1.5621±2.301 (4)	2.3766±1.6115 (3)	4.8437±4.9738 (2)	4.9161±5.0176 (1)
C20	20	HV	0.1331±0.3036 (4)	0.3784±0.3426 (1)	0.3739±0.4364 (2)	0.2566±0.4 (3)
		PD($\times 10^4$)	0.6663±1.6027 (4)	2.4107±3.2936 (2)	4.151±5.057 (1)	2.2967±3.6426 (3)
Overall ranking			(68)	(66)	(38)	(28)

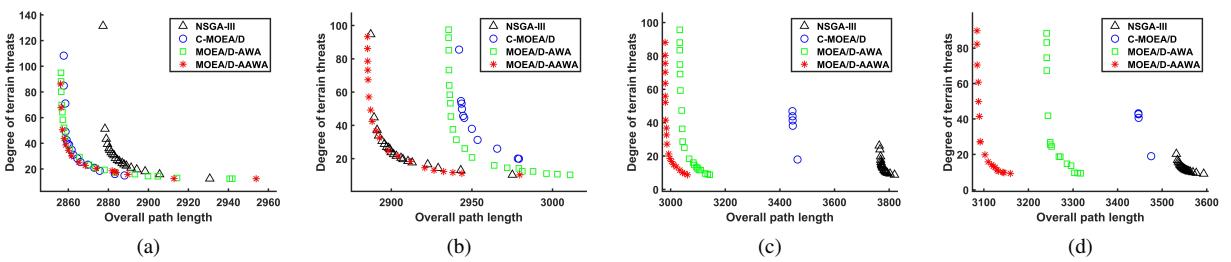


Fig. 8. Final nondominated solutions in urban environments. (a): C11; (b): C14; (c): C18; (d): C19.

Table 4 displays the experimental results of the four MOEAs over 30 independent runs regarding to the mean and Std values of HV and PD in urban environments. Similarly, rank the mean values, and mark the optimal mean value and the highest ranking in boldface. On the one hand, there is little difference of the mean HV values between

UAV 3-D path planning based on MOEA/D with adaptive areal weight adjustment

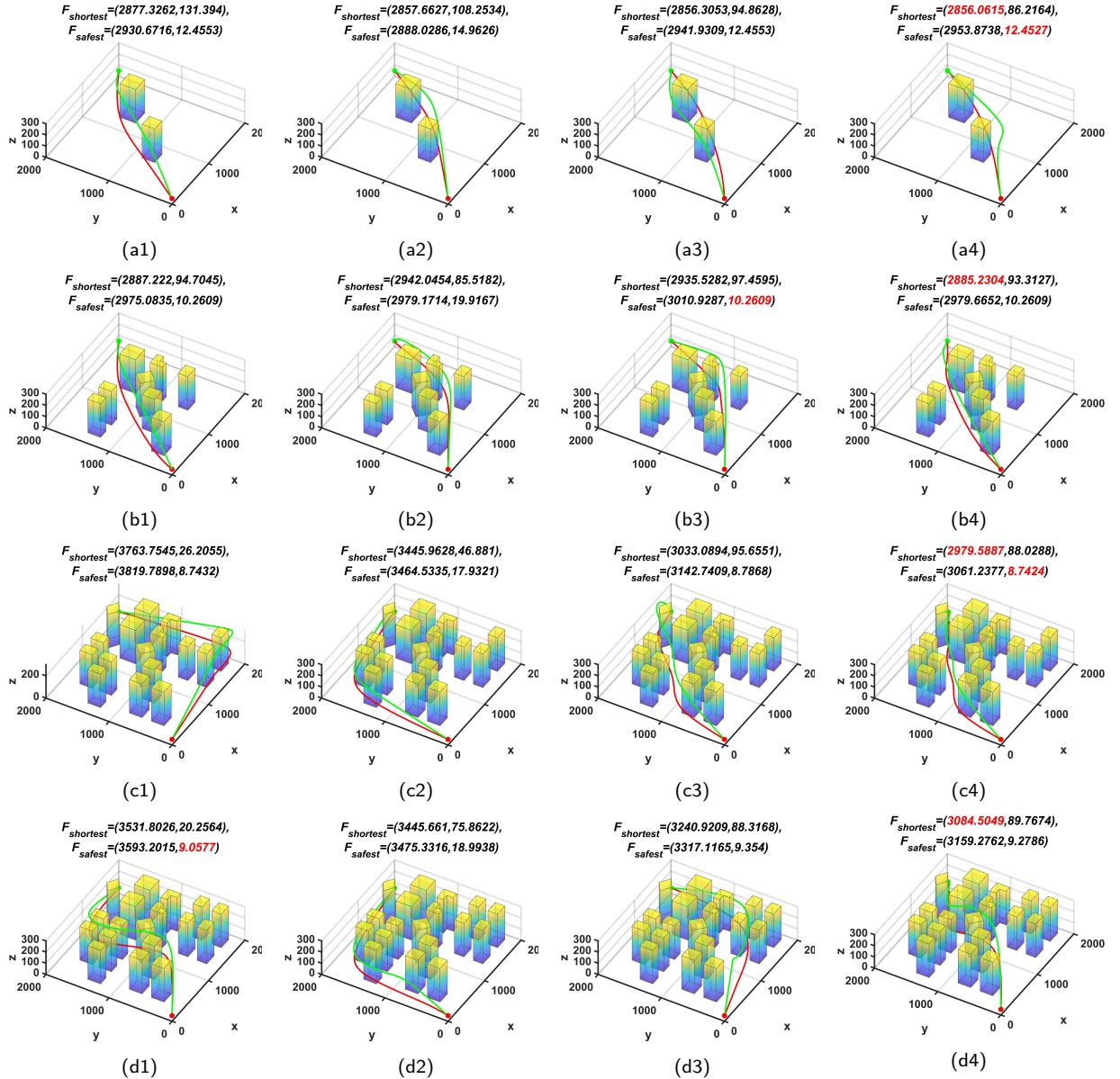


Fig. 9. The shortest path (red path) and the safest path (green path) in urban environments. (a1)-(a4): C11; (b1)-(b4): C14; (c1)-(c4): C18; (d1)-(d4): C19. (a1)-(d1): NSGA-III; (a2)-(d2): C-MOEA/D; (a3)-(d3): MOEA/D-AWA; (a4)-(d4): MOEA/D-AAWA.

C-MOEA/D, MOEA/D-AWA and MOEA/D-AAWA in all cases. From the cases C17-C20, the mean HV values of NSGA-III are significantly smaller than that of the other three algorithms. It indicates that NSGA-III is not suitable for handling complex environments with dense urban obstacles. On the other hand, MOEA/D-AWA and MOEA/D-AAWA obtain better mean values of PD in all cases, which demonstrates the advantages of weight adjustment strategies in diversity. The reason is that weight adjustment strategies guide individuals to evolve towards sparse areas, where there are valuable solutions. In the all cases C11-C20, the proposed algorithm achieves seven optimal mean values of HV (except C15, C19 and C20), seven optimal mean values of PD (except C12, C16 and C20) and the best overall ranking. It indicates that our weight adjustment strategy (i.e., AAWA) is superior to AWA.

Fig. 8 presents the final nondominated solutions at the run with the median score in cases C11, C14, C18 and C19. And Fig. 9 presents the corresponding shortest path (red path) and safest path (green path). From Fig. 8, the PF also has a sharp peak and a low tail in the urban environments, but it is more gentle than that in the mountain environments. The final nondominated solutions obtained by the proposed algorithm achieves superior diversity. In addition, its convergence is better than other algorithms, especially in cases C18 and C19. By analyzing paths of cases C18 and C19 in Fig. 9, the reason is that the paths of proposed algorithm tend to pass through the middle of the map.

5.3.3. Experiments under realistic scenarios

Table 5

Mean and Std values of HV and PD in realistic scenarios

Case	Metric	Algorithm			
		NSGA-III	C-MOEA/D	MOEA/D-AWA	MOEA/D-AAWA
C21	HV	0.3184±0.4581 (4)	0.4612±0.3573 (3)	0.5164±0.4922 (2)	0.62±0.4799 (1)
	PD($\times 10^4$)	0.231±0.3388 (3)	0.2066±0.1697 (4)	0.785±0.8889 (2)	0.8378±0.7473 (1)
C22	HV	0.7328±0.4113 (4)	0.7566±0.0249 (3)	0.9771±0.0113 (1)	0.972±0.0136 (2)
	PD($\times 10^4$)	0.4866±0.2941 (3)	0.4145±0.1057 (4)	1.2474±0.2983 (1)	1.1769±0.4781 (2)
C23	HV	0.7557±0.3846 (4)	0.7628±0.0718 (3) (3)	0.9053±0.2466 (2)	0.968±0.0209 (1)
	PD($\times 10^4$)	0.4976±0.277 (3)	0.4591±0.113 (4)	0.9053±0.2466 (2)	1.4611±0.6646 (1)
C24	HV	0.5749±0.4463 (4)	0.7091±0.0219 (3)	0.9775±0.0084 (2)	0.9798±0.0064 (1)
	PD($\times 10^4$)	0.2828±0.2281 (3)	0.2422±0.059 (4)	1.027±0.3195 (2)	1.0444±0.3038 (1)
Overall ranking		(28)	(28)	(14)	(10)

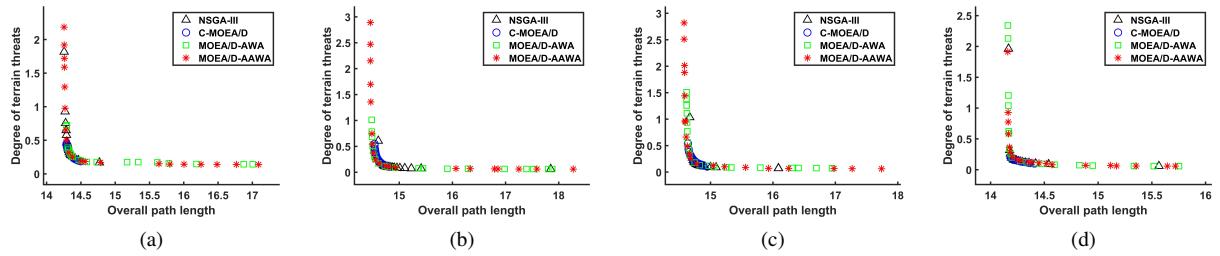


Fig. 10. Final nondominated solutions in realistic scenarios. (a): C21; (b): C22; (c): C23; (d): C24.

The mean and Std values of HV and PD over 30 independent runs in all four real-data cases are shown in Table 5. Sort and bold the data of the table in the same way. The proposed algorithm obtains the optimal result of both HV and PD mean values in cases C21, C23 and C24, and ranks first overall. Fig. 10 presents the final nondominated solutions at the run with the median score. NSGA-III and C-MOEA/D without a weight adjustment strategy cannot reach the sharp peak and low tail. It seems that MOEA/D-AWA reaches these two areas, however, sometimes it can only reach one of them. Taking Fig. 10(a)-(c) as examples, the coverage of sharp peak by MOEA/D-AWA is insufficient. The proposed algorithm cannot only obtain feasible solutions in the middle area of PFs, but also simultaneously search for valuable individuals in sharp peak and low tail.

Fig. 11 presents the shortest path (red path) and the safest path (green path) in all four real-data cases. As shown in Fig. 11(a2)-(d2), the shortest and safest path of C-MOEA/D are very similar. It is because that, its safest path is almost a straight line from the starting point to the target point, rather than avoiding dangerous terrain. The path of NSGA-III have the same defect shown in Fig. 11 (a1) and (c1). Conversely, the safest paths of MOEA/D-AWA and the proposed algorithm reduce the collision possibility by bypassing from the sides of map or raising the height of the UAV, thereby improving the flight safety. However, the proposed algorithm obtains both the shortest path and the safest path in cases C21, C22 and C23, and the shortest flight paths in case C24. It indicates that the proposed algorithm has more superior performance.

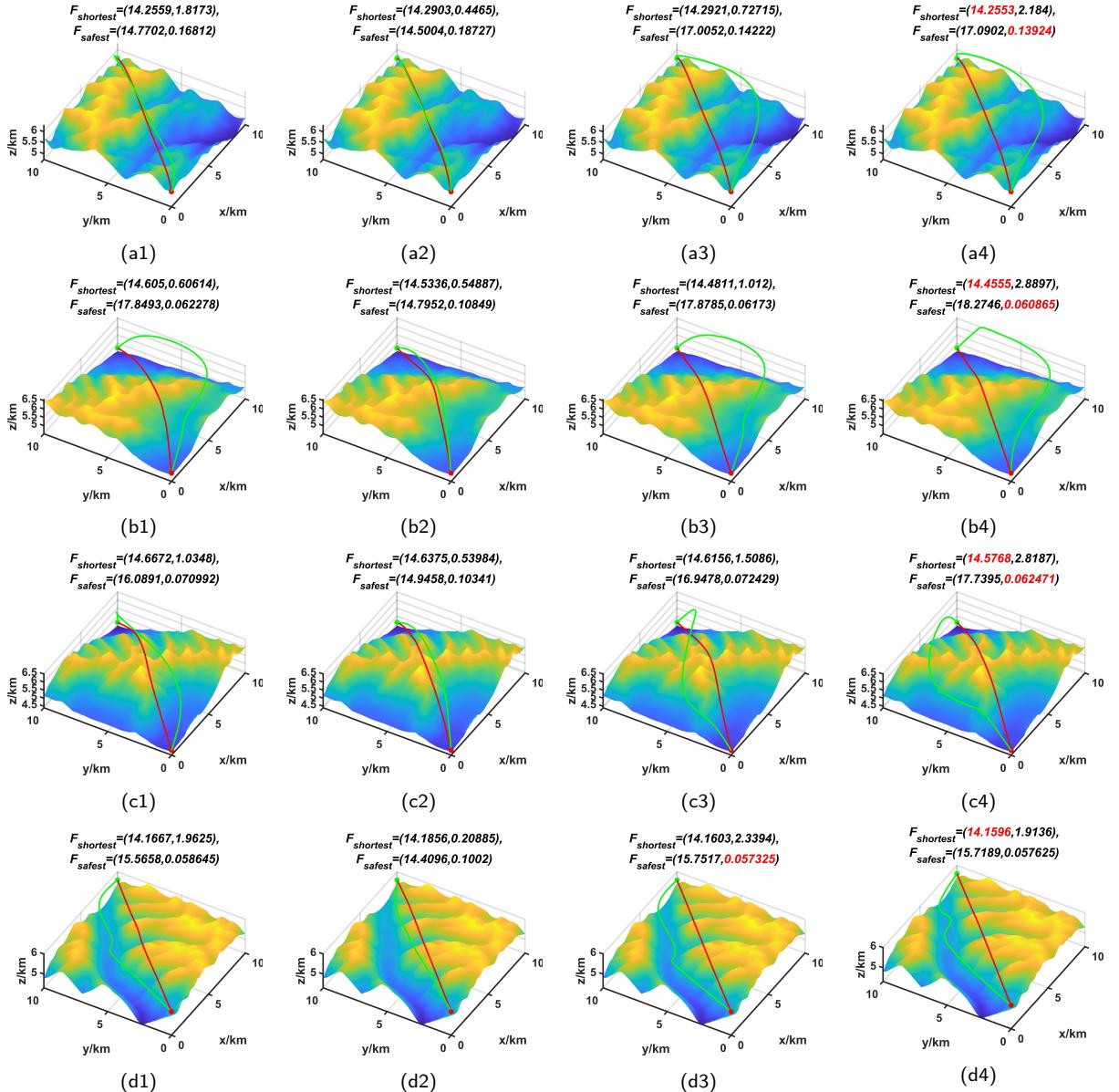


Fig. 11. The shortest path (red path) and the safest path (green path) in realistic scenarios. (a1)-(a4): C21; (b1)-(b4): C22; (c1)-(c4): C23; (d1)-(d4): C24. (a1)-(d1): NSGA-III; (a2)-(d2): C-MOEA/D; (a3)-(d3): MOEA/D-AWA; (a4)-(d4): MOEA/D-AAWA.

6. Conclusion

To solve UAV 3-D path planning problems with sharp peak and low tail PFs, we propose a multi-objective path planning method based on improved MOEA/D. In this paper, we introduce a multi-objective path planning problem minimizing overall path length and degree of threats for a UAV in 3-D task scenarios. In order to improve population diversity, we propose an adaptive areal weight adjustment strategy, i.e., deleting crowded subproblems in the population and adding new subproblems in the sparse area of the population. Furthermore, the newly-added weight vector is calculated by the objective function value of its neighbors to navigate the newly-added individual to the sparser area on PFs.

To verify the effectiveness of our proposed algorithm, we conduct extensive experiments under synthetic (mountain and urban environments) and realistic scenarios. The experiment results indicate that the proposed algorithm is superior to the other three MOEAs in terms of the convergence and diversity. Besides, the proposed algorithm has the capability of searching shorter paths and safer paths in both synthetic and realistic scenarios.

The improvement of the proposed method in future work mainly includes the following two aspects. First, it is necessary to verify the performance of MOEA/D-AAWA in many-objective optimization problems. Second, more realistic flight constraints and parameters should be simulated to make the simulation closer to reality, such as acceleration of UAVs, flight speed. Furthermore, the effectiveness of the method needs to be verified in other types of scenarios, such as mazes, combat environments with radars and artillery.

CRediT authorship contribution statement

Yougang Xiao: Conceptualization, Resources, Supervision. **Hao Yang:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Huan Liu:** Validation, Writing - Review & Editing, Project administration. **Keyu Wu:** Supervision, Project administration. **Guohua Wu:** Writing - Review & Editing, Supervision, Funding acquisition.

References

- [1] H. Huang, A. V. Savkin, and C. Huang, “Reliable path planning for drone delivery using a stochastic time-dependent public transportation network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4941–4950, 2020.
- [2] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, “Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing,” *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3984–3997, 2019.
- [3] C. Ramirez-Atencia, G. Bello-Orgaz, M. D. R-Moreno, and D. Camacho, “Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms,” *Soft Computing*, vol. 21, pp. 4883–4900, 2017.
- [4] E. Masehian and D. Sedighizadeh, “Multi-objective PSO-and NPSO-based algorithms for robot path planning,” *Advances in Electrical and Computer Engineering*, vol. 10, no. 4, pp. 69–76, 2010.
- [5] M. D. Phung and Q. P. Ha, “Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization,” *Applied Soft Computing*, vol. 107, p. 107376, 2021.
- [6] M. Radmanesh and M. Kumar, “Flight formation of UAVs in presence of moving obstacles using fast-dynamic mixed integer linear programming,” *Aerospace Science and Technology*, vol. 50, pp. 149–160, 2016.
- [7] P. Fabiani, V. Fuertes, A. Piquereau, R. Mampey, and F. Teichteil-Königsbuch, “Autonomous flight and navigation of VTOL UAVs: from autonomy demonstrations to out-of-sight flights,” *Aerospace Science and Technology*, vol. 11, no. 2-3, pp. 183–193, 2007.
- [8] X. Chen and X. Chen, “The UAV dynamic path planning algorithm research based on Voronoi diagram,” in *The 26th Chinese Control and Decision Conference (2014 CCDC)*. IEEE, 2014, pp. 1069–1071.
- [9] Y. V. Pehlivanoglu, “A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV,” *Aerospace Science and Technology*, vol. 16, no. 1, pp. 47–55, 2012.
- [10] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [11] Z. Xu, D. Deng, and K. Shimada, “Autonomous UAV exploration of dynamic environments via incremental sampling and probabilistic roadmap,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2729–2736, 2021.
- [12] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, “Anytime motion planning using the RRT,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1478–1483.

- [13] Z. Zhang, J. Wu, J. Dai, and C. He, “A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment,” *IEEE Access*, vol. 8, pp. 122 757–122 771, 2020.
- [14] J. Li, H. Qin, J. Wang, and J. Li, “Openstreetmap-based autonomous navigation for the four wheel-legged robot via 3D-lidar and ccd camera,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2708–2717, 2021.
- [15] P. P.-Y. Wu, D. Campbell, and T. Merz, “Multi-objective four-dimensional vehicle motion planning in large dynamic environments,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 621–634, 2010.
- [16] Q. Zhang and S. Li, “Path planning based on improved particle swarm optimization for a mobile robot,” *WSEAS Transactions on Systems and Control*, vol. 2, no. 5, p. 347, 2007.
- [17] Z. Cheng, Y. X. Tang, and Y. L. Liu, “3-D path planning for UAV based on chaos particle swarm optimization,” in *Applied Mechanics and Materials*, vol. 232. Trans Tech Publ, 2012, pp. 625–630.
- [18] X. You, K. Liu, and S. Liu, “A chaotic ant colony system for path planning of mobile robot,” *International Journal of Hybrid Information Technology*, vol. 9, no. 1, pp. 329–338, 2016.
- [19] J. Zhao and X. Fu, “Improved ant colony optimization algorithm and its application on path planning of mobile robot,” *Journal of Computers*, vol. 7, no. 8, pp. 2055–2062, 2012.
- [20] J. A. Joines and C. R. Houck, “On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA’s,” in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE, 1994, pp. 579–584.
- [21] J. Xiao-Ting, X. Hai-Bin, Z. Li, and J. Sheng-De, “Flight path planning based on an improved genetic algorithm,” in *2013 Third International Conference on Intelligent System Design and Engineering Applications*. IEEE, 2013, pp. 775–778.
- [22] V. Roberge, M. Tarbouchi, and G. Labonté, “Fast genetic algorithm path planner for fixed-wing military UAV using GPU,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2105–2117, 2018.
- [23] X. Yu, C. Li, and J. Zhou, “A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios,” *Knowledge-Based Systems*, vol. 204, p. 106209, 2020.
- [24] J. Liu and J. Lampinen, “A fuzzy adaptive differential evolution algorithm,” *Soft Computing*, vol. 9, pp. 448–462, 2005.
- [25] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [26] H. Liu, X. Li, M. Fan, G. Wu, W. Pedrycz, and P. N. Suganthan, “An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3061–3073, 2020.
- [27] F. Ahmed and K. Deb, “Multi-objective path planning using spline representation,” in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 1047–1052.
- [28] C. Peng and S. Qiu, “A decomposition-based constrained multi-objective evolutionary algorithm with a local infeasibility utilization mechanism for UAV path planning,” *Applied Soft Computing*, vol. 118, p. 108495, 2022.
- [29] Y. Wan, Y. Zhong, A. Ma, and L. Zhang, “An accurate UAV 3-D path planning method for disaster emergency response based on an improved multiobjective swarm intelligence algorithm,” *IEEE Transactions on Cybernetics*, 2022.
- [30] Z. Sun, J. Wu, J. Yang, Y. Huang, C. Li, and D. Li, “Path planning for GEO-UAV bistatic SAR using constrained adaptive multiobjective differential evolution,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6444–6457, 2016.

- [31] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.
- [32] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 6, pp. 898–912, 2003.
- [33] X. Yu, C. Li, and G. G. Yen, "A knee-guided differential evolution algorithm for unmanned aerial vehicle path planning in disaster management," *Applied Soft Computing*, vol. 98, p. 106857, 2021.
- [34] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [35] S. Kukkonen, K. Deb *et al.*, "A fast and effective method for pruning of non-dominated solutions in many-objective problems," in *Parallel Problem Solving from Nature - PPSN IX*, vol. 4193. Springer, 2006, pp. 553–562.
- [36] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 203–208.
- [37] B. L. Miller, D. E. Goldberg *et al.*, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, no. 3, pp. 193–212, 1995.
- [38] K. Deb and H.-G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evolutionary Computation*, vol. 9, no. 2, pp. 197–221, 2001.
- [39] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [40] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2013.
- [41] S. Jiang, L. Feng, D. Yang, C. K. Heng, Y.-S. Ong, A. N. Zhang, P. S. Tan, and Z. Cai, "Towards adaptive weight vectors for multiobjective evolutionary algorithm based on decomposition," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 500–507.
- [42] K. Shang and H. Ishibuchi, "A new hypervolume-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 839–852, 2020.
- [43] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1510–1522, 2016.
- [44] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A LOS map approach," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.
- [45] H. Dai, H. Zhang, C. Li, and B. Wang, "Efficient deployment of multiple UAVs for IoT communication in dynamic environment," *China Communications*, vol. 17, no. 1, pp. 89–103, 2020.
- [46] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Systems with Applications*, vol. 115, pp. 106–120, 2019.
- [47] M. Jones, S. Djahel, and K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–39, 2023.
- [48] S. A. Bortoff, "Path planning for UAVs," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, vol. 1, no. 6. IEEE, 2000, pp. 364–368.

- [49] X. Wu, W. Bai, Y. Xie, X. Sun, C. Deng, and H. Cui, “A hybrid algorithm of particle swarm optimization, metropolis criterion and RTS smoother for path planning of UAVs,” *Applied Soft Computing*, vol. 73, pp. 735–747, 2018.
- [50] D. Zhang and H. Duan, “Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning,” *Neurocomputing*, vol. 313, pp. 229–246, 2018.
- [51] M. Radmanesh, M. Kumar, and M. Sarim, “Grey wolf optimization based sense and avoid algorithm in a Bayesian framework for multiple UAV path planning in an uncertain environment,” *Aerospace Science and Technology*, vol. 77, pp. 168–179, 2018.
- [52] X. Guo, M. Ji, Z. Zhao, D. Wen, and W. Zhang, “Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm,” *Ocean Engineering*, vol. 216, p. 107693, 2020.
- [53] D. Adhikari, E. Kim, and H. Reza, “A fuzzy adaptive differential evolution for multi-objective 3D UAV path optimization,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 2258–2265.
- [54] J. Liu, S. Anavatti, M. Garratt, and H. A. Abbass, “Modified continuous ant colony optimisation for multiple unmanned ground vehicle path planning,” *Expert Systems with Applications*, vol. 196, p. 116605, 2022.
- [55] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, “Sensor planning for a symbiotic UAV and UGV system for precision agriculture,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.
- [56] Y. Zhao, Z. Zheng, and Y. Liu, “Survey on computational-intelligence-based UAV path planning,” *Knowledge-Based Systems*, vol. 158, pp. 54–64, 2018.
- [57] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, “A survey of multiobjective evolutionary algorithms based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2016.
- [58] Y. Zhao, L. Jiao, R. Zhou, and J. Zhang, “UAV formation control with obstacle avoidance using improved artificial potential fields,” in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 6219–6224.
- [59] C. Xu, H. Duan, and F. Liu, “Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning,” *Aerospace Science and Technology*, vol. 14, no. 8, pp. 535–541, 2010.
- [60] T. Xiong, F. Liu, H. Liu, J. Ge, H. Li, K. Ding, and Q. Li, “Multi-drone optimal mission assignment and 3D path planning for disaster rescue,” *Drones*, vol. 7, no. 6, p. 394, 2023.
- [61] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing, and X. Zhu, “An adaptive resource allocation strategy for objective space partition-based multiobjective optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1507–1522, 2019.
- [62] W. Qiu, J. Zhu, G. Wu, H. Chen, W. Pedrycz, and P. N. Suganthan, “Ensemble many-objective optimization algorithm based on voting mechanism,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, pp. 1716–1730, 2020.
- [63] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, “Hyperplane assisted evolutionary algorithm for many-objective optimization problems,” *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3367–3380, 2019.
- [64] M. Fan, Y. Wu, T. Liao, Z. Cao, H. Guo, G. Sartoretti, and G. Wu, “Deep reinforcement learning for UAV routing in the presence of multiple charging stations,” *IEEE Transactions on Vehicular Technology*, 2022.
- [65] S. R. Pokhrel, J. Jin, and H. Le Vu, “Mobility-aware multipath communication for unmanned aerial surveillance systems,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6088–6098, 2019.
- [66] D. Sziroczak, D. Rohacs, and J. Rohacs, “Review of using small UAV based meteorological measurements for road weather management,” *Progress in Aerospace Sciences*, vol. 134, p. 100859, 2022.