



Article

A Two-Stage Co-Evolution Multi-Objective Evolutionary Algorithm for UAV Trajectory Planning

Gang Huang , Min Hu ^{*}, Xueying Yang , Yijun Wang and Peng Lin

Department of Aerospace Science and Technology, Space Engineering University, Beijing 101416, China; hgnchk@163.com (G.H.); htgcyxy@163.com (X.Y.); wyjhgd@163.com (Y.W.); lpnefu@163.com (P.L.)

* Correspondence: jlhm09@163.com

Abstract: With the increasing complexity of unmanned aerial vehicle (UAV) missions, single-objective optimization for UAV trajectory planning proves inadequate in handling multiple conflicting objectives. There is a notable absence of research on multi-objective optimization for UAV trajectory planning. This study introduces a novel two-stage co-evolutionary multi-objective evolutionary algorithm for UAV trajectory planning (TSCEA). Firstly, two primary optimization objectives were defined: minimizing total UAV flight distance and obstacle threats. Five constraints were defined: safe distances between UAV trajectory and obstacles, maximum flight altitude, speed, flight slope, and flight corner limitations. In order to effectively cope with UAV constraints on object space limitations, the evolution of the TSCEA algorithm is divided into an exploration phase and an exploitation phase. The exploration phase employs a two-population strategy where the main population ignores UAV constraints while an auxiliary population treats them as an additional objective. This approach enhances the algorithm's ability to explore constrained solutions. In contrast, the exploitation phase aims to converge towards the Pareto frontier by leveraging effective population information, resulting in multiple sets of key UAV trajectory points. Three experimental scenarios were designed to validate the effectiveness of TSCEA. Results demonstrate that the proposed algorithm not only successfully navigates UAVs around obstacles but also generates multiple sets of Pareto-optimal solutions that are well-distributed across objectives. Therefore, compared to single-objective optimization, TSCEA integrates the UAV mathematical model comprehensively and delivers multiple high-quality, non-dominated trajectory planning solutions.



Citation: Huang, G.; Hu, M.; Yang, X.; Wang, Y.; Lin, P. A Two-Stage Co-Evolution Multi-Objective Evolutionary Algorithm for UAV Trajectory Planning. *Appl. Sci.* **2024**, *14*, 6516. <https://doi.org/10.3390/app14156516>

Academic Editor: Sylvain Bertrand

Received: 19 May 2024

Revised: 12 July 2024

Accepted: 23 July 2024

Published: 25 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: UAV trajectory planning; multi-objective optimization; exploration and exploitation stages; B-spline fitting curve

1. Introduction

With the rapid development of UAV technology, the high maneuverability, stability, and flexibility of UAVs have resulted in their widespread use in many fields, such as surveying and mapping, medical rescues, geological exploration, and hydrological resource monitoring [1,2]. Among them, UAV trajectory planning is crucial to ensure that UAVs complete their flight tasks, and it usually involves two key aspects: trajectory generation and trajectory optimization. Trajectory generation includes UAV obstacle avoidance and the solution of key trajectory points, while trajectory optimization determines the flight efficiency and safety of UAVs. UAV trajectory planning has been widely researched but is mainly regarded as a single-objective optimization problem [3], i.e., it only focuses on the optimization of the performance of one objective or the optimization of the performance of multiple objectives after weighting. With the increasing complexity of UAV missions, single-objective optimization methods are unable to handle multiple conflicting optimization objectives simultaneously. UAVs need to consider a variety of factors in actual flight, such as the shortest flight distance, flight speed, flight altitude, etc. Single-objective optimization

for UAV trajectory planning usually focuses on only one of these aspects, ignoring other important optimization requirements, resulting in poor overall planning results.

Multi-objective optimization for UAV trajectory planning (MOUPT) simultaneously considers the performance of multiple optimization objectives. Through multi-objective optimization algorithms and constraint processing techniques, it yields a set of planning solutions that satisfy different optimization objectives as much as possible. Not only can it satisfy the shortest flight distance, but it also can satisfy constraints such as the flight speed of the UAV, which makes the solution more flexible and the decision-making more comprehensive. At present, MOUPT algorithms are roughly divided into the following three categories.

The first category is MOUPT based on node algorithms, which are characterized by short response times and high robustness and universality. This type of MOUPT is suitable for a variety of complex scenarios where obstacle avoidance is the main optimization objective. Yuan et al. [4] extended the obstacle grid in a map in order to maintain a safe distance between the UAV and obstacles. Fan et al. [5] proposed MOUPT based on the PF-RRT* algorithm in complex environments, introduced a target weight bias strategy to simplify the search space near the obstacles by using a dichotomous method, and proposed an improved artificial potential field. Yan et al. [6] proposed a new algorithm for fixed-wing UAV trajectory planning based on a genetic algorithm and the theory of the Dobbins curve. This method uses an initial circle node, a target circle node, a threat circle node, and key trajectory points to jointly construct a coding scheme.

The second category is MOUPT based on intelligent optimization algorithms, which can extend multiple optimization objectives according to the decision maker's needs for meeting the various constraints and requirements of UAV trajectories. This type is suitable for balancing multiple optimization objectives. Xu et al. [7] used the multi-objective optimization theory to model a UAV trajectory planning problem, designed multiple objective functions and weight-vector-adaptive combinations for the model, and proposed MOUPT with dimensional exploration as the main optimization objective. Chen et al. [8] proposed an enhanced version of the chimpanzee optimization algorithm to solve MOUPT in 3D environments by combining differential variational operators and adaptive norms to enhance the algorithm's search capability and prevent key trajectory point solutions from falling into multiple local optima. Thoma, A. et al. [9] proposed an improved version of the 3D Vector Field Histogram * (3DVFH *) method for local trajectory planning of UAV in medical applications, in which a variant of weighted products, weighted Chebyshev distance, and factorial achievement scaling were investigated in addition to the classical weighted sum.

The third category is MOUPT based on deep learning algorithms, which provide more flexible, intelligent, and adaptive multi-objective UAV trajectory-planning schemes. Zhou et al. [10] proposed MOUPT based on a deep reinforcement learning (DRL) algorithm, which introduced multiple target optimization allocation networks in a double-delay deterministic policy gradient algorithm while establishing task assignment weight coefficients to prioritize important tasks. Sina et al. [11] used a deep deterministic policy gradient (DDPG)-based algorithm for set environments with 3D continuous actions, while a reward function based on inner product weights was proposed in order to balance features such as the UAV target tracking distance and UAV obstacle avoidance. Lv et al. [12] proposed an information theoretical exploration algorithm for UAV platforms, called Entropy Explorer (EE), for determining UAV obstacle avoidance in MOUPT.

Most of the above studies used weight vectors to balance multiple optimization objectives, but the parameters of weight vectors are easily affected by the decision maker's preferences. Meanwhile, the UAV constraint divides the target space into multiple infeasible domains, which makes solving for non-dominated critical waypoints quite difficult.

At present, a small number of studies have regarded MOUPT as a constrained multi-objective optimization problem. By simulating the natural evolutionary process, multiple performance objectives are optimized while satisfying the constraints of the planning

system. The basic working principle is as follows. First, a random initial population (a collection of candidate UAV solutions) is generated, and the optimal individuals are selected for crossover and mutation by evaluating their fitness and constraint satisfaction to generate a new generation of individuals (new candidate UAV solutions). Through repeated iterations, the fitness of individuals is continuously optimized to find multiple non-dominated solutions that satisfy the MOUPT constraints. However, they suffer from problems such as a long response time and high computational complexity during the planning process [13–15]. In order to further improve the performance of MOUPT, a novel multi-objective evolutionary algorithm with two-stage co-evolution for UAV trajectory planning (TSCEA) is proposed in this paper. The innovations and contributions of this research are summarized as follows.

- (1) A multi-objective optimization framework for UAV trajectory planning is proposed. Meanwhile, the distance of the UAV from an obstacle is regarded as a strong constraint, and the framework can be extended with multiple conflicting objectives and multiple constraints.
- (2) A two-phase co-evolutionary constraint-handling technique is proposed, where a two-population strategy is adopted in the exploration phase, the primary population is not restricted by the UAV constraints, and the secondary population treats the constraints as an additional objective. The exploitation stage then gradually converges to a well-distributed Pareto-optimal solution, thus contributing to the acquisition of key UAV trajectory points.
- (3) Several sets of test instances are constructed to simulate UAV trajectory planning by several benchmark algorithms, while four performance metrics are used to test the performance of the algorithms and verify the effectiveness of the proposed algorithms.

The rest of this paper is organized as follows. The relevant research background, including constrained multi-objective optimization problems and archive-based two-stage evolutionary algorithms, is presented in Section 2. The general framework of the TSCEA is presented in Section 3. Test examples and experimental analyses are described in Section 4. The full study is summarized in Section 5.

2. Related Research Background

In this study, MOUPT is regarded as a constrained multi-objective optimization problem. We first introduce the basic theory of constrained multi-objective optimization problems. Then, a constrained multi-objective evolutionary algorithm with a powerful search capability is introduced, which proposes a two-stage co-evolutionary constraint-processing technique to provide a reference for solving the set of key trajectory points.

2.1. Constrained Multi-Objective Optimization Problem

Unlike unconstrained single-objective optimization problems, constrained multi-objective optimization problems (CMOPs) need to satisfy different constraints and optimize multiple objectives, which are usually coupled and in competition with each other, i.e., an increase in the performance of one optimization objective may cause a decrease in the performance of other optimization objectives.

The final solution of CMOPs is not just an optimal solution but a set of Pareto-optimal solutions consisting of compromise solutions [16–18]. CMOPs originated for the determination of the optimal design, modeling, and planning of complex systems, and they are now widely used in a variety of practical works, such as the design of urban bus routes [19], the optimization of the gait of bipedal robots [20], the resource allocation and scheduling of unmanned aerial vehicles, etc., [21]. CMOPs can be defined as follows [22,23]:

$$\begin{aligned}
 & \text{Minimize } F(x) = (f_1(x), \dots, f_m(x))^T \\
 & \text{subject } x \in S, \\
 & \quad g_i(x) \leq 0, i = 1, \dots, p \\
 & \quad h_j(x) = 0, i = p+1, \dots, q
 \end{aligned} \tag{1}$$

where $F(x)$ denotes the objective function vector, $f_m(x)$ denotes m simultaneous optimization objectives to be optimized. $x = (x_1, \dots, x_n)^T$ is the n -dimensional decision vector, n is the number of decision variables, $x \in S$, S denotes the search space, and $g_i(x)$ and $h_j(x)$ are the i -th inequality and j -th equation constraints, respectively. p denotes the number of inequality constraints, q is the number of constraints.

For solution x , its overall constraint violation (CV) can be calculated as follows:

$$CV(x) = \sum_{i=1}^p \max\{0, g_i(x)\} + \sum_{j=p+1}^q \max\{0, |h_j(x)| - \eta\} \tag{2}$$

where η is a very small positive value, usually $\eta = 10^{-6}$, and it converts an equality constraint to an inequality constraint. If $CV(x) = 0$, then solution x is a feasible solution; otherwise, x is a non-feasible solution. Meanwhile, the dominance relationship is the focus of multi-objective evolutionary algorithms (any two decision variables $x, y \in \Omega$) if the following is true:

$$\begin{cases} f_i(x) \leq f_i(y), \forall i \in \{1, 2, \dots, m\} \\ f_j(x) \leq f_j(y), \forall j \in \{1, 2, \dots, m\} \end{cases} \tag{3}$$

Then, x is said to dominate y , which is denoted as $x \prec y$. If x and y do not dominate each other, then x and y are said to be Pareto non-dominated.

Figure 1 is a schematic diagram of a CMOP for generating feasible solutions. Figure 1a represents the constrained multi-objective decision space. Figure 1b represents the objective space; the blue points within the elliptical dashed lines represent the Pareto-optimal solutions, and the red curves represent the Pareto front. Note that the set of Pareto-optimal solutions refers to the constrained subset of the decision space, while the Pareto front is composed of the constrained subset in the objective function space.

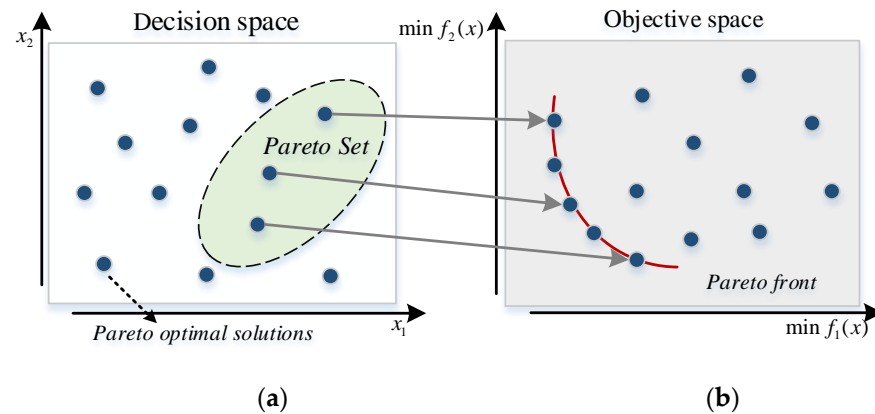


Figure 1. Multi-objective optimization problems with feasible solutions. (a) denotes the decision space; (b) denotes the objective space.

2.2. AT-CMOEA

The constraints of UAVs cause the search space to generate multiple infeasible domains that partition the objective space into multiple unevenly distributed subspaces. Most CMOEAs focus on the diversity, convergence, and feasibility of the solutions (planning solutions), ignoring the role of the infeasible solutions (the planning solutions of $CV \neq 0$), which leads to insufficient selection pressure on the nondominated solutions.

An archive-based two-stage evolutionary algorithm with constraints (AT-CMOEA) provides an effective global search strategy that fully considers the balanced relationship among objectives and between constraints and objectives, and this algorithm utilizes information from both feasible and infeasible solutions to improve the diversity and convergence of the solutions [12]. In this section, the AT-CMOEA is briefly introduced.

The AT-CMOEA divides the evolutionary process into two stages—exploration and exploitation. The exploration stage explores the search space extensively through two populations, one biased towards constraints and one biased towards the objective, so that the search resources are concentrated in the promising regions as much as possible. The exploitation stage obtains a set of well-distributed Pareto-optimal solutions based on the useful individual information found in the exploration stage, and the two populations with different characteristics cooperate to converge to the constrained Pareto front (CPF). This is organized as follows:

- (1) The exploration stage consists of two populations: the main population, which does not consider any constraints and aims to explore the search space quickly and efficiently, and the assistant population, which takes all the constraints as an additional objective and retains some infeasible solutions close to the feasible domain, with the aim of jumping out of the local optimum and finding a promising feasible domain. Finally, all the solution information is archived for further use in the exploitation phase.
- (2) The exploitation stage is more concerned with fast convergence to CPF, where the main population consists of the populations from the exploitation stage and the assistant population considers all constraints in order to obtain a set of well-distributed Pareto-optimal feasible solutions. In addition, by exchanging the offspring produced by the two populations, the assistant population continuously helps the main population to converge from the infeasible domain to the CPF by using the promising information it carries.

The AT-CMOEA uses different populations to determine the CPF from different stages and fully considers the effects of constraints and infeasible solutions on the objective space. The main and auxiliary populations co-evolve schematically, as shown in Figure 2.

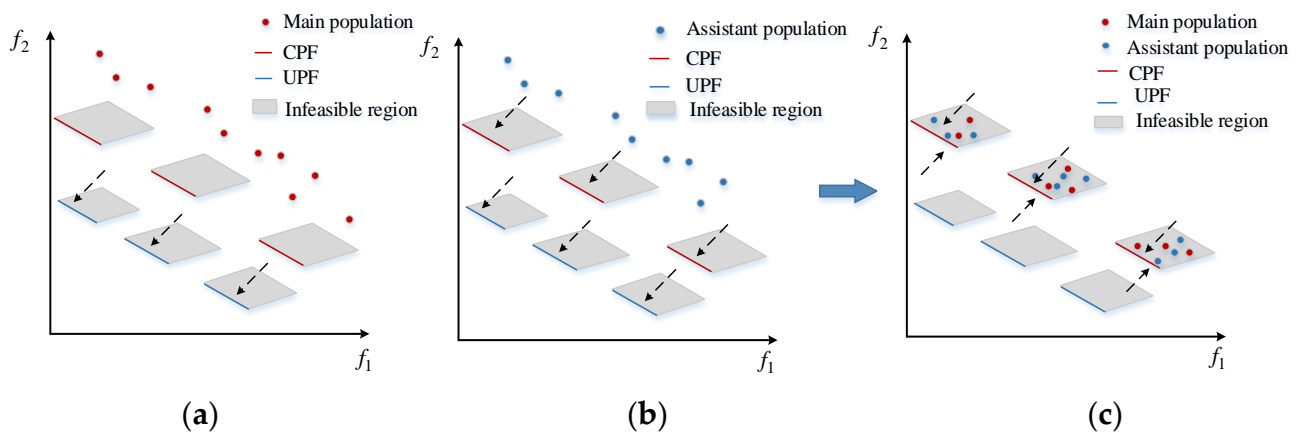


Figure 2. Schematic representation of the co-evolution of the main and assistant populations. (a) represents the process of moving the main population; (b) represents the process of moving the assistant population; (c) represents the process of moving both populations.

The red dots in Figure 2a,b denote the main population, the blue dots denote the assistant population, the grey area denotes the feasible domain, the red solid line denotes the Pareto front while considering the constraints, the blue solid line denotes the Pareto front without considering the constraints, and the dashed arrows denote the direction of the run of the populations. Figure 2c shows that the main and assistant populations were simultaneously clustered in the feasible domain constraining the Pareto front.

There are many constraints coupled in MOUTP; if the constraints of the UAV are ignored, then the generated key trajectory points will ignore the actual flight situation of the UAV, which will have an impact on the stability and safety of the UAV. The constraint-handling technique of the two-stage co-evolution of the AT-CMOEA is of high significance.

3. General Framework of MOUTP Based on TSCEA

The overall framework of MOUTP based on the TSCEA proposed in this study is shown in Figure 3 and consists of three main parts: (1) constructing the multi-objective mathematical model of MOUTP and establishing the UAV constraints and multiple objectives to be optimized; (2) using the mutation operator and selection operator of the differential evolutionary algorithm to obtain a set of offspring key trajectory points; and (3) performing the evolutionary process of the TSCEA. The main steps are described in detail below.

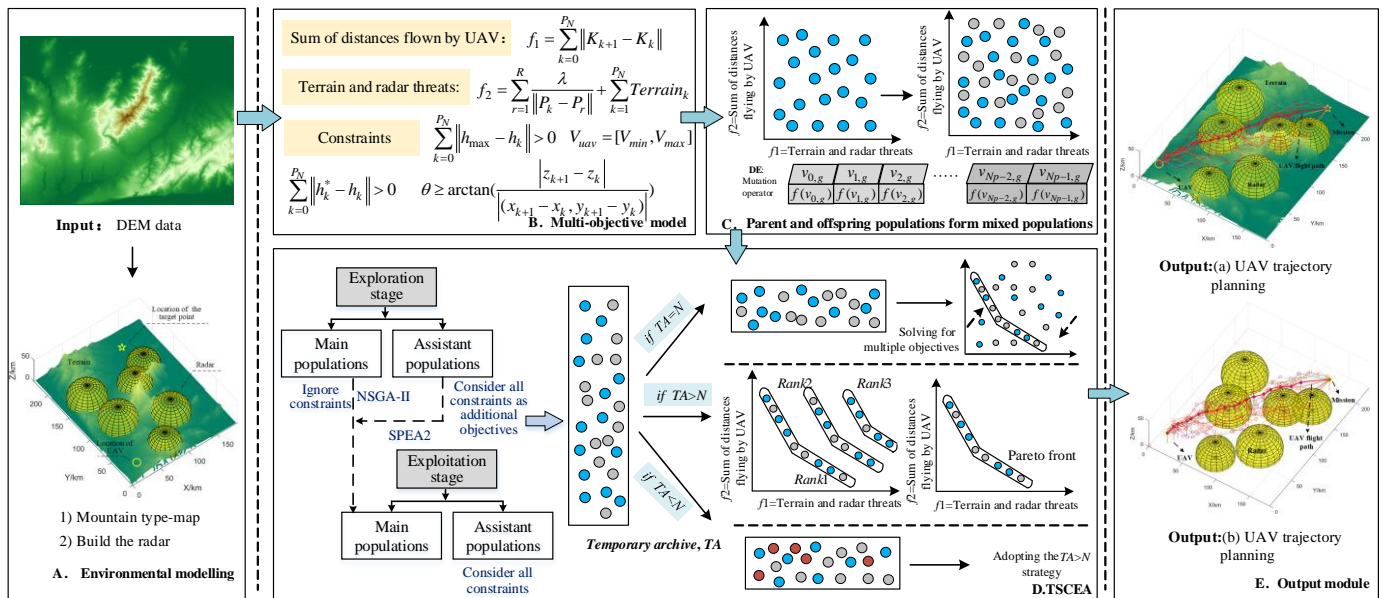


Figure 3. Execution framework of UAV trajectory planning based on TSCEA.

3.1. Multi-Objective Model

3.1.1. Establishment of the Objective Function

In this section, two optimization objective functions of MOUTP are considered: the UAV flight distance and the sum of the UAV's distance to the obstacle threat. From the perspective of actual UAV flight, the UAV and obstacle flight safety distance constraints and the UAV's maximum flight height, flight slope, and flight angle constraints were established. This study constructed the MOUTP model using two optimization objective functions and five constraints.

The first objective function was the sum of the UAV's flight distances, f_1 :

$$f_1 = \sum_{k=0}^{P_N} \|K_{k+1} - K_k\| \quad (4)$$

where P_N denotes the total number of key trajectory points in the UAV's trajectory, K_0 denotes the position of the UAV, K_{P_N+1} denotes the position of the mission point, $[K_1, K_2, \dots, K_{P_N}]$ denotes the position of each key trajectory point, and $\|\cdot\|$ denotes the calculation of the Euclidean distance between two points.

The second objective function is the sum of the UAV's distance from the terrain and the simulated radar obstacle threat, f_2 . In order to avoid collisions between the UAV and obstacles, the generated key trajectory points should be far away from the terrain and obstacle points:

$$f_2 = \sum_{r=1}^R \frac{\lambda}{\|P_k - P_r\|} + \sum_{k=1}^{P_N} Terrain_k \quad (5)$$

where R represents the number of radars, P_k represents the coordinates of the UAV at key trajectory point k , P_r represents the position of the center of radar r , λ represents the threat

coefficient of the radar, and $Terrain_k$ represents the threat coefficient of the terrain and the key trajectory point k .

3.1.2. Establishment of Constraints

The first constraint is the safe distance of the UAV trajectory from obstacles:

$$\sum_{k=0}^{P_N} \|h_k^* - h_k\| > 0 \tag{6}$$

where h_k^* denotes the elevation value below the k -th key trajectory point and h_k denotes the actual distance below the k -th key trajectory point.

The second constraint is the maximum flight altitude of the UAV:

$$\sum_{k=0}^{P_N} \|h_{\max} - h_k\| > 0 \tag{7}$$

where h_{\max} denotes the maximum flight altitude of the UAV.

The third constraint is the speed constraint of the UAV:

$$V_{uav} = [V_{\min}, V_{\max}] \tag{8}$$

where V_{\min} denotes the minimum speed of the UAV and V_{\max} denotes the maximum speed of the UAV.

The fourth constraint is the UAV flight slope θ , which is the angle between the UAV and the horizontal:

$$\theta \geq \arctan\left(\frac{|z_{k+1} - z_k|}{|(x_{k+1} - x_k, y_{k+1} - y_k)|}\right) \tag{9}$$

where z_k^i denotes the altitude of the k -th key trajectory point of the i -th UAV in three-dimensional space, x_k^i denotes the horizontal coordinate of the k -th key trajectory point of the i -th UAV in three-dimensional space, and y_k^i denotes the vertical coordinate of the k -th key trajectory point of the i -th UAV in three-dimensional space.

The fifth constraint is the UAV flight corner η :

$$\eta \geq \arctan\left(\frac{a_i^k(y)}{\sqrt{(a_i^k(x))^2 + (a_i^k(z))^2}}\right) \tag{10}$$

where $a_i^k(x)$, $a_i^k(y)$, and $a_i^k(z)$ denote the acceleration of the k -th path point of the i -th UAV in the direction of the x , y , and z coordinate axes, respectively.

3.2. TSCEA

The AT-CMOEA cannot directly solve the MOUTP problem because it adopts a two-stage constraint-handling technique with two populations, which promotes the diversity and convergence of the solution but also increases the algorithm's response time and computational complexity. In this research, a multi-objective evolutionary algorithm with two-stage co-evolution for UAV trajectory planning (TSCEA) is proposed.

In the TSCEA algorithm, constraints are effectively handled, and multi-objectives are optimized through a combination of exploration and exploitation phases. The selection and optimization of critical trajectory points is the core of the algorithm, which ensures that the UAV's trajectory planning in complex environments can both avoid obstacles and achieve optimized objectives. Obtaining better key trajectory points through the exploration and exploitation phases of the TSCEA algorithm not only improves the efficiency and accuracy of trajectory planning but also enhances the adaptability of UAVs in dynamic and complex environments. The main operational diagram of the algorithm is given in Figure 4.

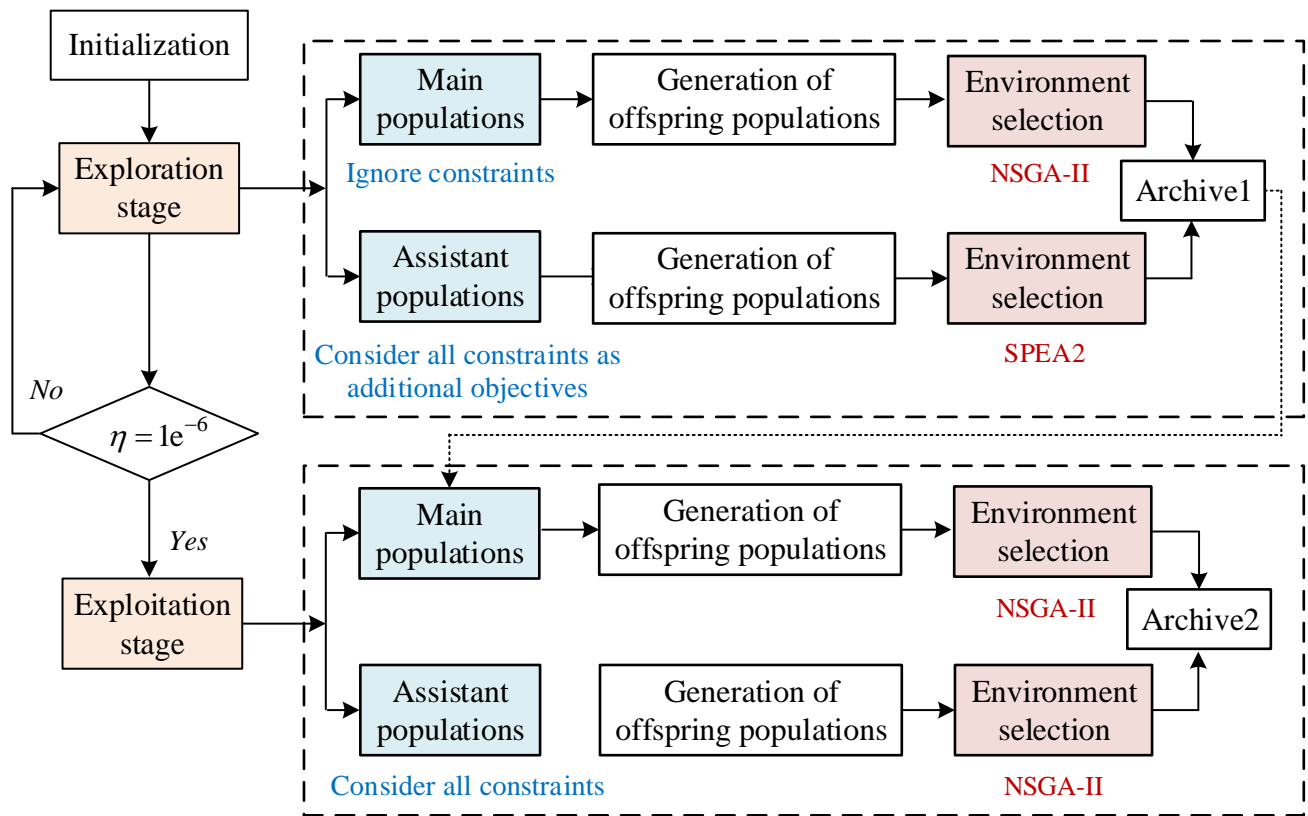


Figure 4. Runtime diagram of the TSCEA.

(1) Exploration stage of the TSCEA

The exploration stage aims to increase the searchability of the population across the search space. The main population ignores the UAV constraints in order to find as many feasible domains as possible. The assistant population uses the constraints as additional objective functions to find more promising solutions. At this point, the constrained bi-objective MOUPT optimization problem is converted to an unconstrained multi-objective MOUPT optimization problem. The main and assistant populations generate the offspring populations through the crossover operator and the variation operator of the differential evolutionary algorithm, respectively. The main population was updated by NSGA-II [24], and the assistant population was updated using SPEA2 [25]. The exploratory stage of the TSCEA is shown in Algorithm 1.

Algorithm 1: Exploration Stage of the TSCEA.

Input: Terrain (Terrain dataset), Radar (Radar dataset), UAV (The location of the UAV), Mission (The location of the mission), f_1 (Objective function 1), f_2 (Objective function 2), C_{uav} (UAV constraint set), Pop (The size of population), Gen (The number of iterations). P_m (Mutation probability);

Output: $update_Mx_1$ (Updated parent population1), $update_Mx_2$ (Updated parent population2);

1 Define terrain datasets and radar datasets and unify the simulation environment;

$Terrain = Terrain3D; Radar = CRadar; E = [Terrain, Radar];$

2 Initialize the population to obtain two random parent populations Mx_1, Mx_2 ;

$Mx_1 = initial (); Mx_2 = initial ();$

```

3 Obtaining new offspring populations using the classical DE algorithm mutation operator
  offMx1;
  for i = 1: Pop
    [r1,r2,r3] = Gainthreenumber(i);/* Get three random numbers */
    if rand() < Pm
      offMx1 = Mx1{r3} + Pm*(Mx1{r2}-Mx1{r1});/* Mutation operator for differential
      evolutionary algorithm */
    end
  end
4 update_Mx1 ← Update_NSII(Mx1,offMx1);/* Update Mx1 through NSGA-II */
5 Converting Mx2 constrained MOUPT (M-objects) to unconstrained MOUPT (M + 1-objects);
  for i = 1: Pop
    offMx2(i) = DEFitness(UAV, Mission, f1, f2, Cuav, Mx2(i));
  end
6 update_Mx2 ← Update_SPEA2(Mx2,offMx2);/* Update Mx2 through SPEA2 */
7 Archive1 = {update_Mx1, update_Mx2};

```

(2) Exploitation stage of the TSCEA

The purpose of the exploitation stage is to converge to the CPF by utilizing the high-quality populations from the archive in the exploration stage. The main population consists of the populations from Archive 1. At this point, both the main and assistant populations need to take into account all the constraints in order to obtain a set of well-distributed feasible solutions. Both the main and assistant populations were used in NSGA-II. The TSCEA utilized the stages shown in Algorithm 2.

Algorithm 2: Exploitation Stage of the TSCEA.

Input: *update_Mx₁* (Updated parent population1), *update_Mx₂* (Updated parent population2).
 Output: *Archive₂*.

1 Assigning populations from archive 1 to Mx₁;

```
Mx1 ← Archive1;
```

2 Mx₁ obtains a subpopulation *offMx₁* by the mutation operator;

```
for i = 1: Pop
```

```
  if rand() < Pm
```

```
    offMx1 = Mx1{r3} + Pm*(Mx1{r2}-Mx1{r1});/
```

```
  End
```

```
End
```

3 Ditto for obtaining *offMx₂*;

4 *update_Mx₁* ← Update_NSII(Mx₁,*offMx₁*);/* Update Mx₁ through NSGA-II */

5 *update_Mx₂* ← Update_NSII(Mx₂,*offMx₂*);/* Update Mx₂ through NSGA-II */

6 *Archive₂* = {*update_Mx₁*, *update_Mx₂*};

3.3. B-Spline Curve

MOUPT yields a set of key trajectory points through the TSCEA, but some of the key trajectory points are widely spaced, and the fitted trajectory segments have a large folding angle with the trajectory segments, which is not in line with the maneuvering performance of the UAV. The UAV trajectory can be smoothed using the B-spline fitting method, which

can fine-tune the trajectory in a specific region without affecting the overall trajectory. The order of the B-spline can be adjusted according to the actual needs of the UAV so as to balance the accuracy of the fitting and the computational efficiency. This is suitable for fitting the uneven distribution of the key trajectory points [26]. The mathematical expression of the UAV 3D m -order B-spline curve equation is as follows:

$$\begin{cases} X(s) = \sum_{i=0}^m X_i B_{i,m}(s) = X_0 B_{0,m}(s) + X_1 B_{1,m}(s) + \dots + X_m B_{m,m}(s) \\ Y(s) = \sum_{i=0}^m Y_i B_{i,m}(s) = Y_0 B_{0,m}(s) + Y_1 B_{1,m}(s) + \dots + Y_m B_{m,m}(s) \\ Z(s) = \sum_{i=0}^m Z_i B_{i,m}(s) = Z_0 B_{0,m}(s) + Z_1 B_{1,m}(s) + \dots + Z_m B_{m,m}(s) \end{cases} \quad (11)$$

where $0 \leq s \leq 1, i = 1, \dots, m$; X_i, Y_i and Z_i denote the position vectors of each key trajectory point; and $B_{i,m}(s)$ is the m -times Bernstein basis function, also known as the B-spline segmentation mixing function, whose expression is as follows:

$$\begin{aligned} B_{i,m}(s) &= \frac{1}{m!} \sum_{j=0}^{m-i} (-1)^j \cdot C_{m+1}^j (s+m-i-j)^m \\ &= \frac{1}{m!} [(s+m-i)m - (m+1)(s+m-i-1)m + \dots \\ &\quad \dots + (-1)^{m-i} \cdot C_{m+1}^{m-i} s^m] \end{aligned} \quad (12)$$

where $C_m^n = \frac{m!}{n!(m-n)!}, C_m^0 = 1$.

When $m = 3$, the basis function of the B-spline can be expressed as follows:

$$\begin{cases} B_{(0,3)}(s) = \frac{1}{6}(-s^3 + 3s^2 - 3s + 1) \\ B_{(1,3)}(s) = \frac{1}{6}(3s^3 - 6s^2 + 4) \\ B_{(2,3)}(s) = \frac{1}{6}(-3s^3 + 3s^2 + 3s + 1) \\ B_{(3,3)}(s) = \frac{1}{6}s^3 \end{cases} \quad (13)$$

When the UAV key trajectory points are unevenly distributed, i.e., the key trajectory points are not equidistant from the key trajectory points, they can be further iteratively updated based on the following basis function:

$$B_{i,1}(s) = \begin{cases} 1, & \tau_i \leq s \leq \tau_{i+1} \\ 0, & \text{elsewise} \end{cases} \quad (14)$$

$$B_{i,m}(s) = \frac{s - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1}(s) + \frac{\tau_{i+m} - s}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}(s) \quad (15)$$

where the nodal equation is as follows:

$$\tau_i = \begin{cases} 0, & i < m \\ i - m + 1, & m \leq i \leq b \\ b - m + 2, & b < i \end{cases} \quad (16)$$

The corresponding expansion of Equations (8)–(12) can be obtained as follows:

$$\begin{bmatrix} X(s) \\ Y(s) \\ Z(s) \end{bmatrix} = \sum_{i=0}^3 B_{i,3}(s) \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \frac{1}{6} [S^3, S^2, S] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_0 & Y_0 & Z_0 \\ X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ X_3 & Y_3 & Z_3 \end{bmatrix} \quad (17)$$

3.4. Algorithmic Time and Space Complexity Analysis of TSCEA

In this section, we provide an analysis of the time and space complexity of the TSCEA algorithm. The main components of the TSCEA algorithm are the exploration phase, the exploitation phase, and the optimization process of the key trajectory points.

(1) Time complexity analysis

Exploration phase: In this phase, the algorithm performs an extensive search of the search space, assuming that the population size is N and each individual evolves within T generations; the time complexity of the exploration phase is $O(N \times T)$.

Exploitation phase: In this phase, the algorithm uses the population information stored in the exploration phase to perform fine-grained optimization. The population size is N and each individual evolves within T' generations; the time complexity of the exploitation phase is $O(N \times T')$.

The optimization of key trajectory points includes B-spline fitting curves and other optimization steps, assuming that the optimization of key trajectory points requires K iterations and the complexity of each iteration is $O(M)$, where M is the number of trajectory points, the time complexity of this step is $O(K \times M)$.

Combining the above parts, the overall time complexity of the TSCEA algorithm is $O(N \times T) + O(N \times T') + O(K \times M)$.

(2) Space complexity analysis

The space complexity of the TSCEA algorithm depends mainly on the population size and the stored population information.

Population storage: In the exploration and exploitation phase, N individuals need to be stored, each containing d -dimensional decision variables; then, the space complexity of population storage is $O(N \times d)$.

Auxiliary storage: Including key trajectory point information and other intermediate variables, assuming that an additional S units of space are required, the space complexity of auxiliary storage is $O(S)$.

Combining the above parts, the overall space complexity of the TSCEA algorithm is $O(N \times d) + O(S)$.

4. Experiments and Analysis

In order to validate the effectiveness of the proposed TSCEA, this section first presents three test instances. Then, four comparative algorithms and algorithm parameters of the classical DE algorithm [27], the DD-CMOEA [28], the cDADSEA [29], and the DPCPRA [30] are introduced. Finally, we conducted several experiments and evaluated the performance of the MOUPT planning results using four performance metrics: the hypervolume (HV) [31], the inverted generational distance (IGD) [32], the feasible solution proportion (FP) [33], and the average planning time.

4.1. Experimental Settings

(1) Test instances and model parameters

In this section, three different test instances are designed with the aim of examining the UAV's traversal of mountainous terrain and radar scanning areas, as shown in Figure 5. Specifically, only the digital elevation model of real terrain was considered in Instance 1, as shown in Figure 5a. Instance 2 contained the real terrain digital elevation model and three radar models, as shown in Figure 5b. Instance 3 added three radar models to Instance 2, and its main purpose was to analyze the effect of increasing radar density on the performance of MOUPT. Instance 3 is shown in Figure 5c.

In Figure 5, the yellow circle indicates the location of the UAV, the yellow pentagram indicates the location of the mission point, and the yellow hemisphere indicates the location of each radar model. Table 1 shows the real terrain digital elevation model dimensions and the coordinates and scanning radius of each radar model for different instances.

sanfranciscos.dem.gz denotes the real terrain digital elevation model dataset [180,230,50], denotes the terrain dimensions, R_c denotes the radar’s coordinates, and R_r denotes the radar’s scanning radius.

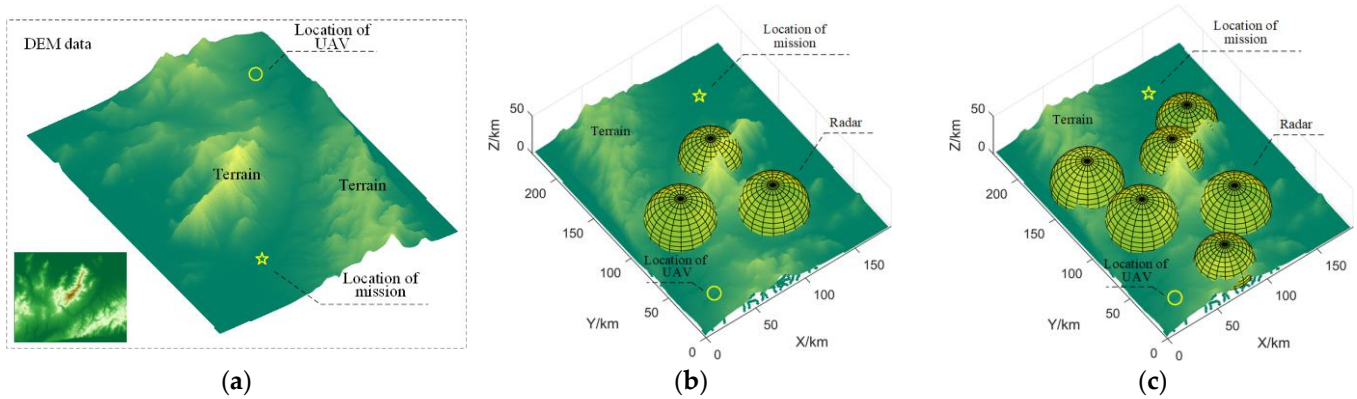


Figure 5. Schematic representation of different test instances. (a) denotes terrain-type obstacles; (b) denotes terrain-type obstacles_3 radars; (c) denotes terrain-type obstacles_6 radars.

Table 1. Scope of DEM of real terrain and parameters for modeling radar obstacles.

Terrain-Type Obstacles	Terrain-Type Obstacles_3 Radars		Terrain-Type Obstacles_6 Radars		
	#	R_c (km)	R_r (km)	R_c (km)	R_r (km)
sanfranciscos.dem.gz [180,230,50]	1	[120,70,4]	30	[120,70,4]	30
	2	[110,140,4]	26	[110,140,4]	26
	3	[52,103,5]	30	[52,103,5]	30
	4	-	-	[80,30,5]	25
	5	-	-	[140,160,5]	25
	6	-	-	[40,160,5]	30

(2) Parameters for comparison algorithms

The classical DE algorithm reduces the limitations of UAV constraints through the mutation operator, crossover operator, and other perturbation strategies, which makes it easier for individuals to jump out of the local optima. The DE algorithm is more suitable for solving optimization problems with complex environments and more constraints. The DD-CMOEA adopts a dual-population co-evolutionary approach, which uses two populations, mainPop and auxPop, to explore the feasible and infeasible domains, respectively, and the algorithm uses infeasible solutions that violate the constraints to participate in the evolution, which enhances the diversity of the populations. The cDADSEA employs a competing-constraint dual-archive two-stage evolutionary algorithm, which evolved in different evolutionary stages by means of a convergence-driven archive (CA) and diversity-driven archive (DA). The DDPCPRA designs a dynamic constraint-handling mechanism and a resource allocation mechanism whose main and assistant populations directly optimize all constraints. The basic parameters of each comparison algorithm in our experiment were set as follows:

- Population size: $N = 100$;
- Number of iterations of the algorithm: $Gen = 500$;
- Probability of mutation: $Pm = 0.9$;
- Number of independent runs of the algorithm: 30.

The remaining specific parameters corresponded to those described in the respective original literature.

(3) Evaluation indicators for CMOEAs

- Hypervolume (HV)

$$HV(P, q) = \int_{R^n} 1_H(P, q)(z) dz \tag{18}$$

In the above equation, P represents the solution set, q represents the predefined reference point in the target space, $1_H(P, q)$ is the eigenfunction of $H(P, q)$, and the HV indicator calculates the area covered by the population to the reference point. The larger the HV value, the better the diversity of the algorithm.

- Inverted generational distance (IGD)

$$IGD(P, P^*) = \frac{\sum_{x \in P^*} \min_{y \in P} dis(x, y)}{|P^*|} \tag{19}$$

In the above equation, P is the non-dominant solution, P^* represents the reference point uniformly sampled from the PF, and the IGD index is the average value of the minimum Euclidean distance between the calculated reference point and the non-dominant solution. The smaller the IGD value, the better the convergence of the algorithm.

- Feasible solution proportion (FP)

$$FP = \frac{N_{fs}}{N_s} * 100\% \tag{20}$$

In the above equation, N_{fs} denotes the number of feasible solutions and N_s denotes the number of solutions, including both feasible and infeasible solutions. This metric can be used to obtain the percentage of all runs of a feasible solution, which helps to evaluate the effectiveness and adaptability of the algorithm.

- Average planning time

$$T_{average}(T_{total}, n) = \frac{\sum_{n \in total} (T_1 + T_2 + \dots + T_n)}{n} \tag{21}$$

In the above equation, T_{total} represents the superposition of the planned system running independently n times, and n represents the number of times the planning system is running. The lower the $T_{average}$ value, the more stable the response time of the planning system.

4.2. MOUTP Simulation Experiments Based on Classical DE Algorithm

Complex terrain structures (e.g., mountain heights, slopes, occlusions, etc.) and radar obstacles can cause algorithms to be affected by a large number of local extreme points during operation, making it easy for individuals to fall into local optima. In this study, the classical DE algorithm was used to simulate MOUTP in different instances, and the purpose of this section is to analyze the performance of MOUTP based on the global search algorithm. Moreover, the classical DE algorithm has fewer parameter settings and stronger adaptability.

Figure 6 shows the MOUTP simulation graphs for three different test instances, where the red circles indicate the trajectory points; the thin solid lines indicate the connectivity relationships between the UAV, the trajectory points, and the mission points; the red thick solid lines indicate the optimal UAV trajectory, fitted according to the key trajectory points; and the black circular dashed lines indicate that the trajectory segment crosses the obstacle region. In order to show the trajectory point evolution generated by the algorithm more intuitively during the iteration process, this study randomly shows the connectivity of 13 trajectory points in Gen iterations. On the premise that the fitted trajectories based on the

key trajectory points do not traverse the real terrain digital elevation model, the real terrain digital elevation model was hidden in Figure 6d–f in order to better observe whether the key trajectory points traversed the radar obstacles or not.

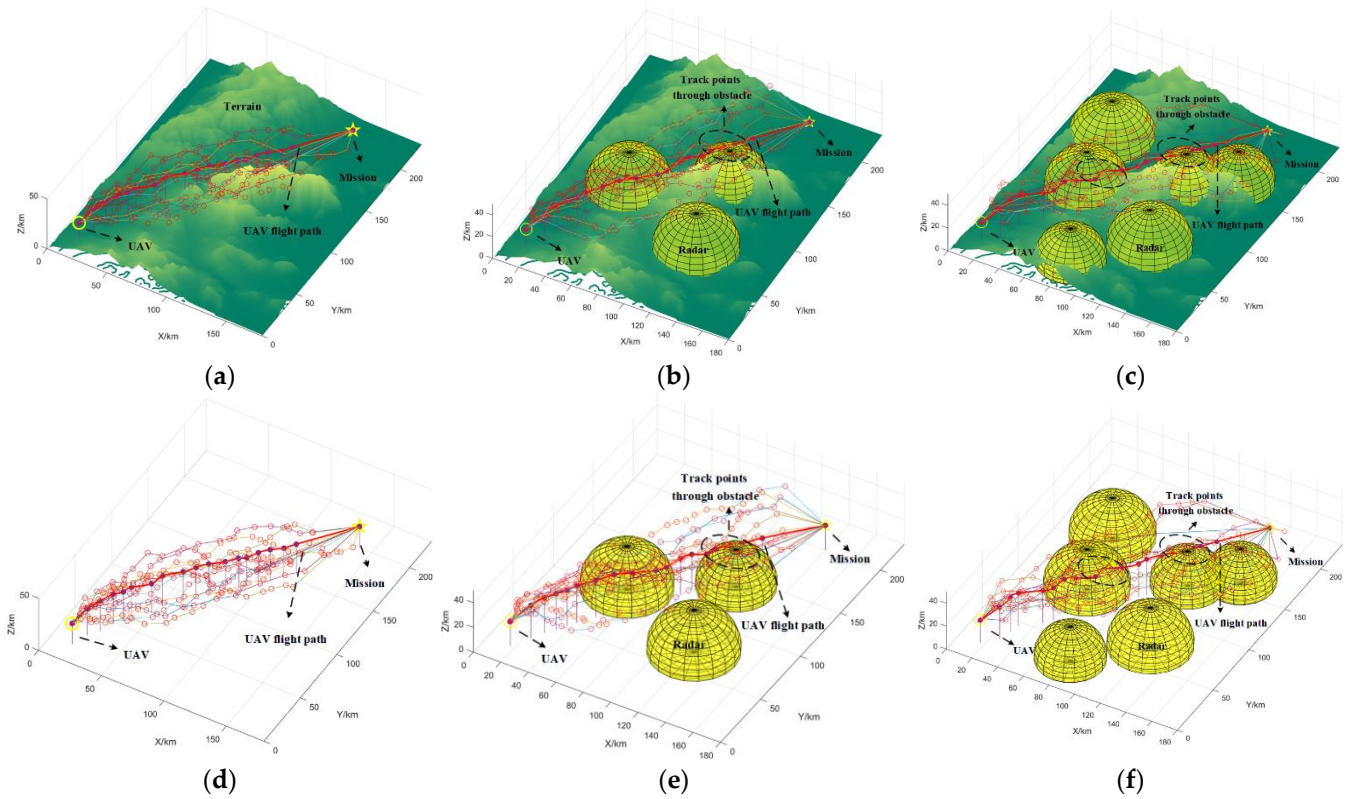


Figure 6. Simulation of MOUTP based on classical differential evolutionary algorithm. (a–c) represent the simulation figures in the three test instances of the traditional DE algorithm respectively. (d–f) conceal the terrain data in the three test instances respectively.

Table 2 represents the coordinates of the key trajectory points generated via MOUTP based on the classical DE algorithm and the average time of its operation for the three instances. The grey numbered areas indicate key trajectory points that crossed the obstacles. Comprehensively, it can be seen in Figure 6 and Table 2 that the classical DE algorithm jumped out of the local optimum and generated the final UAV flight trajectory in different instances. The average response time was shorter, but the trajectory traversed the radar obstacle in both Instance 2 and Instance 3, and the fitted trajectory segments resulted in a certain collision risk to UAV flights.

Table 2. MOUTP based on classical DE algorithm to generate coordinates of key trajectory points.

	UAV (km)		Coordinates of Key Trajectory Points											Target (km)	$T_{average}$ (s)
Test Instance 1	17.0	23.3	27.5	38.9	48.4	57.6	75.3	89.3	105.7	109.4	114.3	122.6	140.0	24 s	
	17.0	29.7	34.2	52.4	69.6	79.8	105.5	125.0	142.0	149.1	157.2	169.4	200.0		
	20.0	23.4	24.8	27.9	29.9	28.3	28.5	28.1	29.0	28.6	30.1	28.5	20.0		
Test Instance 2	17.0	33.6	37.1	58.9	66.8	64.2	71.5	82.6	97.4	101.4	113.0	123.5	140.0	26 s	
	17.0	29.2	44.9	66.5	79.4	80.4	110.5	110.4	130.4	150.4	168.4	189.0	200.0		
	20.0	19.3	20.3	20.3	18.5	18.5	21.0	20.7	20.3	21.4	20.2	21.0	20.0		
Test Instance 3	17.0	20.8	32.9	39.4	48.8	71.3	74.7	99.1	107.0	116.2	124.2	131.4	140.0	30 s	
	17.0	25.4	52.3	59.9	64.7	74.3	106.0	114.1	137.7	151.9	174.3	185.8	200.0		
	20.0	23.2	25.3	25.5	24.8	25.6	24.7	25.1	24.9	24.8	25.0	25.2	20.0		

4.3. MOUTP Simulation Experiments Based on CMOEAs

(1) Validation of MOUTP based on TSCEA

As described in this section, the TSCEA was used to simulate MOUTP under different instances. The TSCEA employed the classical DE algorithm mutation operator to enhance the diversity of the population, which, in turn, enhanced the global search capability, and the algorithm employed a two-phase strategy to take into account the limitations of the UAV constraints on the objective space. The TSCEA has the advantage of performing a global search while possessing the ability of constraint processing. Figure 7 and Table 3 represent the MOUTP simulation maps and key trajectory point locations, respectively, based on the TSCEA for three different instances.

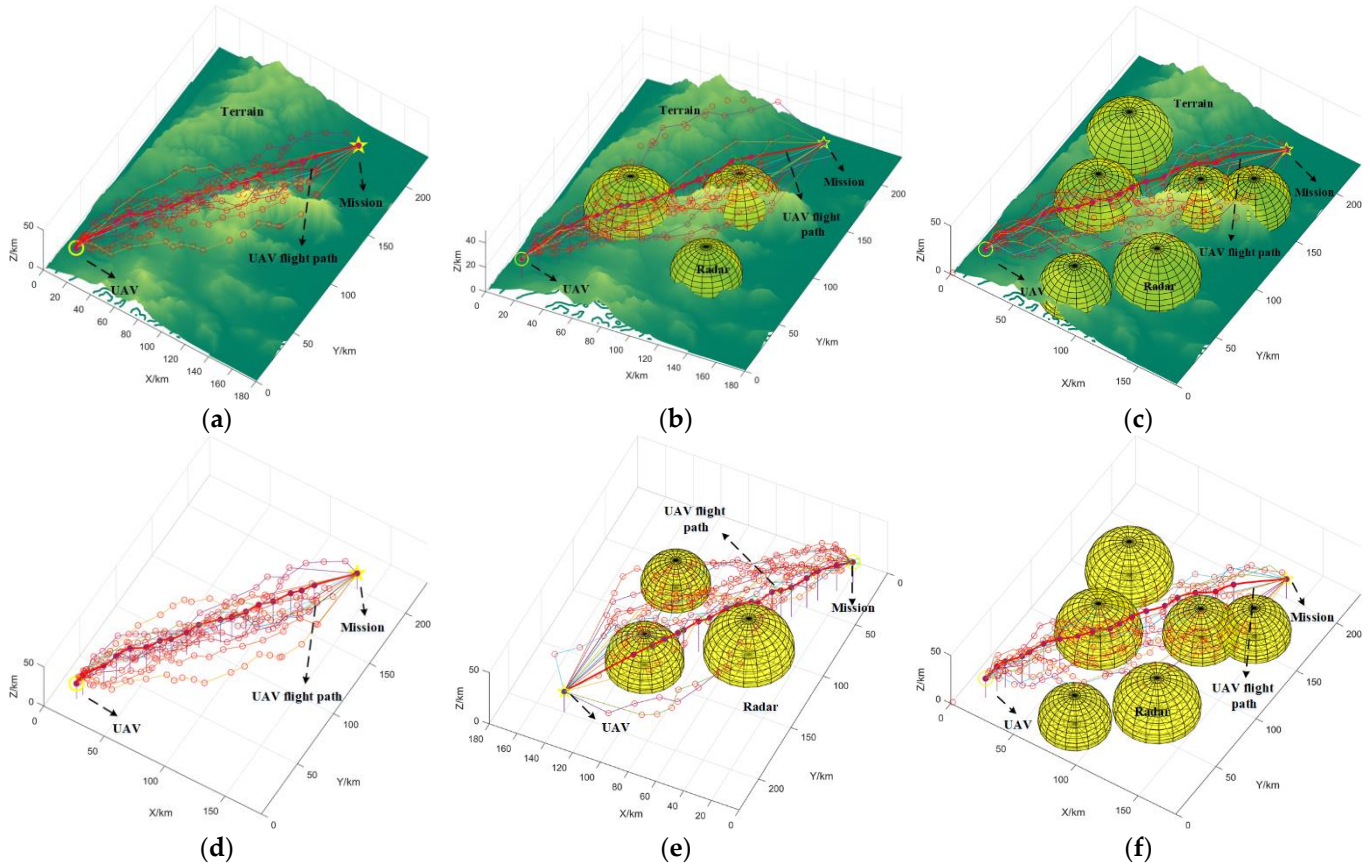


Figure 7. Simulation of MOUTP based on TSCEA. (a–c) represent the simulation figures in the three test instances of the traditional TSCEA algorithm respectively. (d–f) conceal the terrain data in the three test instances respectively.

Table 3. MOUTP based on TSCEA to generate coordinates of key trajectory points.

	UAV (km)		Coordinates of Key Trajectory Points										Target (km)	$T_{average}$ (s)
Test Instance 1	17.0	24.7	32.7	46.5	59.1	70.0	85.2	90.2	101.3	111.0	113.4	119.8	140.0	36 s
	17.0	30.3	37.5	56.1	74.4	94.1	108.7	114.8	129.1	152.3	161.3	171.9	200.0	
	20.0	25.8	26.2	28.2	28.9	27.8	28.1	28.1	28.7	28.1	27.7	28.6	20.0	
Test Instance 2	17.0	21.6	34.0	43.5	48.9	52.0	62.3	76.7	82.1	93.7	103.0	113.7	140.0	51 s
	17.0	23.0	40.4	65.3	71.6	75.8	94.8	105.6	117.3	131.6	151.0	160.8	200.0	
	20.0	24.9	30.1	29.8	27.5	28.1	29.8	28.9	28.0	27.4	28.3	28.6	20.0	
Test Instance 3	17.0	24.3	35.5	49.5	59.9	69.0	81.3	89.1	92.9	96.9	103.9	107.7	140.0	40 s
	17.0	27.7	43.5	60.2	78.3	95.7	115.5	134.3	141.7	147.3	156.6	164.5	200.0	
	20.0	28.4	29.5	28.4	28.4	28.8	28.7	28.5	29.3	29.1	28.3	30.3	20.0	

By combining Figure 7 and Table 3, it was found that the MOUPT fitted trajectories based on the TSCEA in three different instances did not cross the obstacles. The third of these instances, i.e., the location of the key trajectory point near the radar [110,140,4] in Figure 7c,f and the fitted trajectory segments fit the shape of the radar, indicating that the TSCEA had a better performance in balancing the constraints and objectives. Meanwhile, the average planning time of MOUPT based on the TSCEA was higher than that of the DE algorithm but still within a limited time.

(2) MOUPT based on different CMOEAs

As described in this section, three different CMOEAs were used to simulate MOUPT under different instances, as shown in Figure 8. Table 4 displays the HV, IGD, FP, and average planning time performance metrics for different CMOEAs.

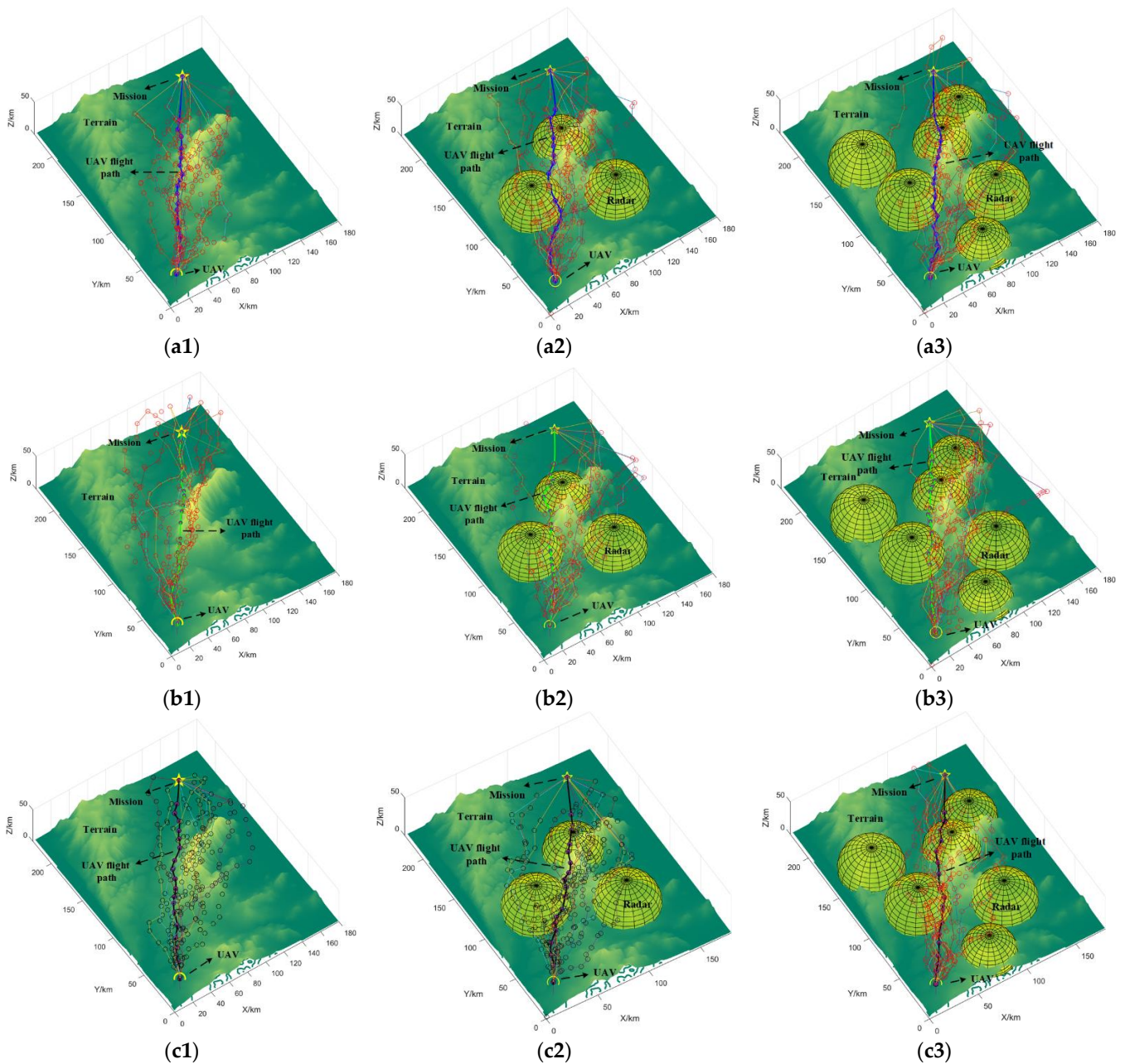


Figure 8. MOUPT simulation graph based on different CMOEAs. (a1–a3) denote the MOUPT simulations based on the DD-CMOEA; (b1–b3) denote the MOUPT simulations based on the cDADSEA algorithm; (c1–c3) denote the MOUPT simulations based on the DPCPRA.

Table 4. Metrics of CMOEAs for different instances.

#	Method	Metrics			
		HV	IGD	FP (%)	$T_{average}$ (s)
Test Instance 1	DD-CMOEA	9.8750×10^{-3} (7.45×10^{-3})	7.1922×10^{-1} (4.36×10^{-2})	100%	20
	cDADSEA	3.5523×10^{-3} (3.65×10^{-3})	7.6204×10^{-1} (2.69×10^{-2})	94.2%	25
	TSCEA	1.7388×10^{-2} (4.57×10^{-2})	6.9665×10^{-1} (1.68×10^{-1})	100%	27
	DPCPRA	2.9169×10^{-3} (4.53×10^{-3})	7.7373×10^{-1} (3.46×10^{-2})	100%	31
Test Instance 2	DD-CMOEA	2.2770×10^{-1} (3.87×10^{-2})	2.5567×10^{-1} (2.63×10^{-2})	100%	35
	cDADSEA	2.3768×10^{-1} (7.65×10^{-3})	2.9148×10^{-1} (3.01×10^{-2})	86%	30
	TSCEA	2.4992×10^{-1} (5.30×10^{-3})	2.4768×10^{-1} (1.86×10^{-2})	100%	32
	DPCPRA	2.5206×10^{-1} (5.45×10^{-3})	3.1852×10^{-1} (8.95×10^{-2})	100%	40
Test Instance 3	DD-CMOEA	2.0949×10^{-1} (7.92×10^{-3})	3.4199×10^{-1} (1.48×10^{-2})	100%	32
	cDADSEA	1.5448×10^{-1} (8.48×10^{-2})	4.5464×10^{-1} (1.75×10^{-1})	75%	40
	TSCEA	2.1166×10^{-1} (1.99×10^{-2})	3.2324×10^{-1} (7.55×10^{-2})	100%	37
	DPCPRA	1.1034×10^{-1} (9.26×10^{-2})	5.3951×10^{-1} (1.91×10^{-1})	95%	58

From the MOUTP simulation graph based on different CMOEAs in Figure 8 and the metrics of different CMOEAs in Table 4, the following was shown:

- In Test Instance 1, the HV values of the TSCEA were all larger than those of the other CMOEAs, and the values of the IGD were all smaller than those of the other algorithms, indicating that the TSCEA had better diversity and convergence than the other CMOEAs. The values of the FP of the DD-CMOEA, the TSCEA, and the DPCPRA algorithm were all 100%, and part of the key trajectory points of the cDADSEA crossed the obstacles, so there were a small number of infeasible solutions. Although the average running time of the TSCEA was longer than that of the DD-CMOEA and the TSCEA, the difference was small.
- In Test Instance 2, the HV values of the TSCEA were lower than those of the DPCPRA, suggesting that the TSCEA was less diverse than the DPCPRA; however, the values of the IGDs were still superior to those of the other CMOEAs. The lower FP values of the DD-CMOEA and the cDADSEA suggest that some of the key trajectory points traversed the obstacles. Meanwhile, the average time of the algorithms in Test Instance 2 was longer than that in Test Instance 1 because the simulation environment was more complex in Test Instance 2; however, the planning time was still shorter.
- The TSCEA's HV and IGD values were superior to those of the other algorithms in Test Instance 3. The FP of each CMOEA in Test Instance 1 remained at 100%, but as the difficulty of the simulation environment in other instances increased further, the FP value of the other two CMOEAs gradually decreased. The FP value of the TSCEA did not change. This indicates that the performance of the TSCEA metrics was more stable. Meanwhile, the average planning time of the TSCEA in Example 3 was also more stable.

Figure 9 is a graph of the Pareto fronts for MOUTP based on different CMOEAs. Figure 9a–c represent the CMOEA Pareto fronts for Test Instance 1, Test Instance 2, and Test Instance 3, respectively. Since the simulation environment in Instance 1 was relatively simple, the distribution of each CMOEA Pareto front was relatively uniform. However, the TSCEA had a more uniform distribution, and the two optimization objective values were smaller than those of the other algorithms, which indicates that the TSCEA was able to effectively balance the UAV range distance and the threat value of the distance to the obstacle. From Figure 9b,c, it can be seen that the UAV range distance and the UAV distance to the obstacle threat values of MOUTP based on the TSCEA were smaller than those of the other CMOEAs, which was mainly due to the TSCEA's global search and constraint-processing abilities.

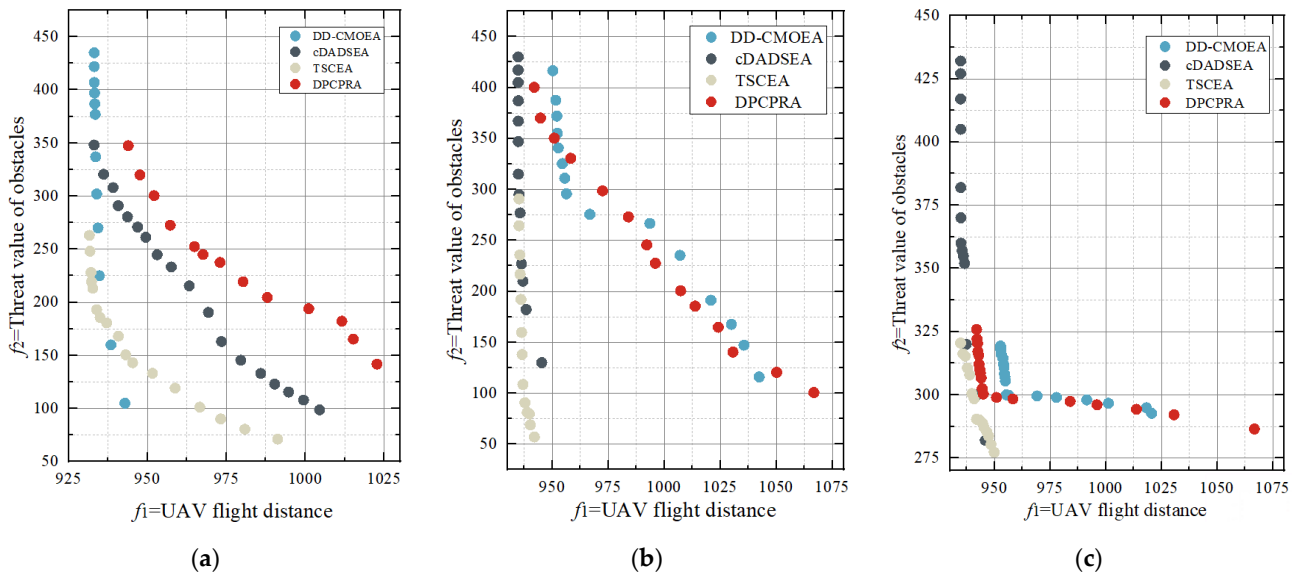


Figure 9. Distribution of Pareto fronts for different CMOEAs. (a) denotes the Pareto frontier for each algorithm in test instance 1; (b) denotes the Pareto frontier for each algorithm in test instance 2; (c) denotes the Pareto frontier for each algorithm in test instance 3.

Figure 10 shows the average planning time and the shortest planning time for each CMOEA in different test instances. From Figure 10, it can be seen that although the TSCEA-based MOUPT was not the shortest in terms of the average planning time and shortest planning time in different test instances, the average planning time and the shortest planning time of MOUPT based on the TSCEA were more stable in each test instance. By combining the HV, IGD, and PF values of the TSCEA above, it can be concluded that the MOUPT planning system based on the TSCEA is more stable in its operation and able to provide an effective method for solving the key trajectory points.

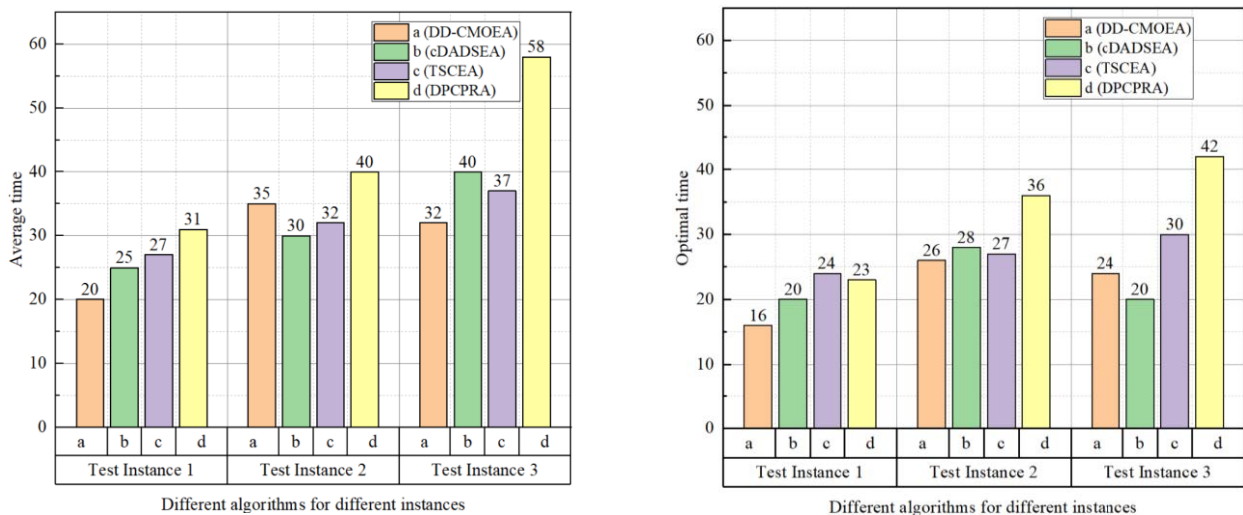


Figure 10. Average and minimum running time of CMOEAs under test instances.

5. Conclusions

In this study, we first construct a multi-objective UAV trajectory planning framework containing two optimization objectives and five constraints. Then, a TSCEA algorithm is proposed in conjunction with the MOUPT mathematical model. The TSCEA algorithm divides the evolutionary process into an exploration phase and an exploitation phase, where the exploration phase aims to perform an extensive search of the search space. The exploitation phase, on the other hand, obtains a set of well-distributed solution sets by using the population

information stored in the exploration phase. Finally, the flight trajectory is smoothed to ensure that the trajectory matches the maneuverability of the UAV by fitting a curve through a B-spline. Comparisons between performance metrics are performed through three different test instances and four benchmark algorithms. The simulation results show that the MOUPT planning system based on the TSCEA algorithm is more stable, the optimized trajectory can effectively avoid obstacles, can consider conflicting optimization objectives, and obtain multiple sets of better Pareto non-dominated solutions, which significantly improves the performance of the MOUTP planning system. The TSCEA algorithm has a significant advantage in UAV trajectory planning, which provides new ideas for the UAV planning field. In future research, we will continue to explore and optimize the algorithm, extend its application in other agent mission planning, and further promote the development of agent mission planning in the direction of large-scale and ultra-large-scale.

Author Contributions: Conceptualization, G.H. and M.H.; methodology, G.H. and M.H.; project administration, M.H.; software, G.H. and M.H.; supervision, M.H. and Y.W.; validation, X.Y. and P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61403416.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: Thanks to Min Hu for his important technical help and for providing experimental equipment and related materials.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Acronym	Full Name
UAV	Unmanned aerial vehicle
TSCEA	Two-stage co-evolution multi-objective evolutionary algorithm
MOUTP	Multi-objective optimization for UAV trajectory planning
DRL	Deep reinforcement learning
DDPG	Deep deterministic policy gradient
EE	Entropy explorer
CMOEA	Constrained multi-objective evolutionary algorithm
CMOP	Constrained multi-objective optimization problem
CV	Constraint violation
AT-CMOEA	An archive-based two-stage evolutionary algorithm with constraints
CPF	Constrained Pareto front
UPF	Unconstrained Pareto front
NSGA-II	Non-dominated Sorting Genetic Algorithm II
SPEA2	Strength Pareto evolutionary Algorithm 2
DE	Differential evolution
DD-CMOEA	Dual-stage dual-population constrained multi-objective evolutionary algorithm
cDADSEA	Constrained dual-archive dual-stage evolutionary algorithm
DPCPRA	Dual-population evolutionary algorithm based on dynamic constraint processing and resources allocation
HV	Hypervolume
IGD	Inverted generational distance
FP	Feasible solution proportion
DEM	Digital elevation model
CA	Convergence-driven archive
DA	Diversity-driven archive

References

1. Zhang, W.; Peng, C.; Yuan, Y.; Cui, J.; Qi, L. A Novel Multi-Objective Evolutionary Algorithm with a Two-Fold Constraint-Handling Mechanism for Multiple UAV Path Planning. *Expert Syst. Appl.* **2024**, *238*, 121862. [\[CrossRef\]](#)
2. Bai, H.; Fan, T.; Niu, Y.; Cui, Z. Multi-UAV Cooperative Trajectory Planning Based on Many-Objective Evolutionary Algorithm. *Complex Syst. Model. Simul.* **2022**, *2*, 130–141. [\[CrossRef\]](#)
3. Wan, Y.; Zhong, Y.; Ma, A.; Zhang, L. An Accurate UAV 3-D Path Planning Method for Disaster Emergency Response Based on an Improved Multiobjective Swarm Intelligence Algorithm. *IEEE Trans. Cybern.* **2023**, *53*, 2658–2671. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Yuan, M.; Zhou, T.; Chen, M. Improved Lazy Theta* Algorithm Based on Octree Map for Path Planning of UAV. *Def. Technol.* **2023**, *23*, 8–18. [\[CrossRef\]](#)
5. Fan, J.; Chen, X.; Wang, Y.; Chen, X. UAV Trajectory Planning in Cluttered Environments Based on PF-RRT* Algorithm with Goal-Biased Strategy. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105182. [\[CrossRef\]](#)
6. Yan, P.; Yan, Z.; Zheng, H.; Guo, J. A Fixed Wing UAV Path Planning Algorithm Based On Genetic Algorithm and Dubins Curve Theory. *MATEC Web Conf.* **2018**, *179*, 03003. [\[CrossRef\]](#)
7. Xu, X.; Xie, C.; Luo, Z.; Zhang, C.; Zhang, T. A Multi-Objective Evolutionary Algorithm Based on Dimension Exploration and Discrepancy Evolution for UAV Path Planning Problem. *Inf. Sci.* **2024**, *657*, 119977. [\[CrossRef\]](#)
8. Chen, Q.; He, Q.; Zhang, D. UAV Path Planning Based on an Improved Chimp Optimization Algorithm. *Axioms* **2023**, *12*, 702. [\[CrossRef\]](#)
9. Thoma, A.; Thomessen, K.; Gardi, A.; Fisher, A.; Braun, C. Prioritising Paths: An Improved Cost Function for Local Path Planning for UAV in Medical Applications. *Aeronaut. J.* **2023**, *127*, 2125–2142. [\[CrossRef\]](#)
10. Kong, X.; Zhou, Y.; Li, Z.; Wang, S. Multi-UAV Simultaneous Target Assignment and Path Planning Based on Deep Reinforcement Learning in Dynamic Multiple Obstacles Environments. *Front. Neurobot.* **2024**, *17*, 11727–11740. [\[CrossRef\]](#)
11. Sabzekar, S.; Samadzad, M.; Mehditabrizi, A.; Tak, A.N. A Deep Reinforcement Learning Approach for UAV Path Planning Incorporating Vehicle Dynamics with Acceleration Control. *Unmanned Syst.* **2023**, *17*, 1–22. [\[CrossRef\]](#)
12. Lv, H.; Chen, Y.; Li, S.; Zhu, B.; Li, M. Improve Exploration in Deep Reinforcement Learning for UAV Path Planning Using State and Action Entropy. *Meas. Sci. Technol.* **2024**, *35*, 056206. [\[CrossRef\]](#)
13. Zhao, W.; Li, L.; Wang, Y.; Zhan, H.; Fu, Y.; Song, Y. Research on A Global Path-Planning Algorithm for Unmanned Aerial Vehicle Swarm in Three-Dimensional Space Based on Theta*-Artificial Potential Field Method. *Drones* **2024**, *8*, 125. [\[CrossRef\]](#)
14. Yu, X.; Li, C.; Zhou, J. A Constrained Differential Evolution Algorithm to Solve UAV Path Planning in Disaster Scenarios. *Knowl.-Based Syst.* **2020**, *204*, 106209. [\[CrossRef\]](#)
15. Zhang, J.; Wang, Z.; Han, G.; Qian, Y.; Li, Z. A Collaborative Path Planning Method for Heterogeneous Autonomous Marine Vehicles. *IEEE Internet Things J.* **2024**, *11*, 1465–1480. [\[CrossRef\]](#)
16. Ming, F.; Gong, W.; Wang, L. A Two-Stage Evolutionary Algorithm with Balanced Convergence and Diversity for Many-Objective Optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 6222–6234. [\[CrossRef\]](#)
17. Yang, Z.; Qiu, H.; Gao, L.; Chen, L.; Liu, J. Surrogate-Assisted MOEA/D for Expensive Constrained Multi-Objective Optimization. *Inf. Sci.* **2023**, *639*, 119016. [\[CrossRef\]](#)
18. Kong, X.; Yang, Y.; Lv, Z.; Zhao, J.; Fu, R. A Dynamic Dual-Population Co-Evolution Multi-Objective Evolutionary Algorithm for Constrained Multi-Objective Optimization Problems. *Appl. Soft Comput.* **2023**, *141*, 110311. [\[CrossRef\]](#)
19. Ming, F.; Gong, W.; Wang, L.; Gao, L. A Constraint-Handling Technique for Decomposition-Based Constrained Many-Objective Evolutionary Algorithms. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 7783–7793. [\[CrossRef\]](#)
20. Bao, Q.; Wang, M.; Dai, G.; Chen, X.; Song, Z.; Li, S. An Archive-Based Two-Stage Evolutionary Algorithm for Constrained Multi-Objective Optimization Problems. *Swarm Evol. Comput.* **2022**, *75*, 101161. [\[CrossRef\]](#)
21. Liang, J.; Zhang, L.; Yu, K.; Qu, B.; Shang, F.; Qiao, K. Interactive Niching-Based Two-Stage Evolutionary Algorithm for Constrained Multiobjective Optimization. *Swarm Evol. Comput.* **2023**, *83*, 101402. [\[CrossRef\]](#)
22. Zuo, M.; Gong, D.; Wang, Y.; Ye, X.; Zeng, B.; Meng, F. Process Knowledge-Guided Autonomous Evolutionary Optimization for Constrained Multiobjective Problems. *IEEE Trans. Evol. Comput.* **2024**, *28*, 193–207. [\[CrossRef\]](#)
23. Qiao, K.; Liang, J.; Liu, Z.; Yu, K.; Yue, C.; Qu, B. Evolutionary Multitasking with Global and Local Auxiliary Tasks for Constrained Multi-Objective Optimization. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 1951–1964. [\[CrossRef\]](#)
24. Ahmadi, M.H.; Ahmadi, M.A. Multi Objective Optimization of Performance of Three-Heat-Source Irreversible Refrigerators Based Algorithm NSGAI. *Renew. Sustain. Energy Rev.* **2016**, *60*, 784–794. [\[CrossRef\]](#)
25. Li, K.; Chen, R.; Fu, G.; Yao, X. Two-Archive Evolutionary Algorithm for Constrained Multi-objective Optimization. *IEEE Trans. Evol. Comput.* **2019**, *23*, 303–315. [\[CrossRef\]](#)
26. Dong, L.; Gan, X. An Energy-Saving Path Planning Method Using Adaptive Ant Colony System for Marine Surface Vehicles. *Soft Comput.* **2023**, *23*, 303–315. [\[CrossRef\]](#)
27. Yu, X.; Jiang, N.; Wang, X.; Li, M. A Hybrid Algorithm Based on Grey Wolf Optimizer and Differential Evolution for UAV Path Planning. *Expert Syst. Appl.* **2023**, *215*, 119327. [\[CrossRef\]](#)
28. Ming, M.; Wang, R.; Ishibuchi, H.; Zhang, T. A Novel Dual-Stage Dual-Population Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2022**, *26*, 1129–1143. [\[CrossRef\]](#)
29. Zhou, T.; He, P.; Niu, B.; Yue, G.; Wang, H. A Novel Competitive Constrained Dual-Archive Dual-Stage Evolutionary Algorithm for Constrained Multiobjective Optimization. *Swarm Evol. Comput.* **2023**, *83*, 101417. [\[CrossRef\]](#)

30. Qiao, K.; Chen, Z.; Qu, B.; Yu, K.; Yue, C.; Chen, K.; Liang, J. A Dual-Population Evolutionary Algorithm Based on Dynamic Constraint Processing and Resources Allocation for Constrained Multi-Objective Optimization Problems. *Expert Syst. Appl.* **2024**, *238*, 121707. [[CrossRef](#)]
31. Xiang, Y.; Zheng, J.; Hu, Y.; Liu, Y.; Zou, J.; Deng, Q.; Yang, S. Weak Relationship Indicator-Based Evolutionary Algorithm for Multimodal Multi-Objective Optimization. *Inf. Sci.* **2024**, *652*, 119755. [[CrossRef](#)]
32. Ye, Y.; Lin, Q.; Wong, K.-C.; Li, J.; Ming, Z.; Coello Coello, C.A. A Localized Decomposition Evolutionary Algorithm for Imbalanced Multi-Objective Optimization. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107564. [[CrossRef](#)]
33. Cao, J.; Qi, Z.; Chen, Z.; Zhang, J. A Multi-Modal Multi-Objective Evolutionary Algorithm Based on Scaled Niche Distance. *Appl. Soft Comput.* **2024**, *152*, 111226. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.