Knowledge-Based Systems 204 (2020) 106209

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios

Xiaobing Yu [a,b,*], Chenliang Li [b], JiaFang Zhou [b]

[a] Key Laboratory of Meteorological Disaster(KLME), Ministry of Education & Collaborative Innovation Center on Forecast and Evaluation of Meteorological Disasters(CIC-FEMD), Nanjing University of Information Science & Technology, Nanjing, China
[b] School of Management Science and Engineering, Nanjing University of Information Science & Technology, Nanjing, China

ARTICLE INFO

Article history:
Received 1 April 2020
Received in revised form 4 June 2020
Accepted 30 June 2020
Available online 3 July 2020

Keywords:
UAV path planning
Disaster emergency management
Differential evolution algorithm
Constrained optimization

ABSTRACT

Disasters have caused significant losses to humans in the past decades. It is essential to learn about the disaster situation so that rescue works can be conducted as soon as possible. Unmanned aerial vehicle (UAV) is a very useful and effective tool to improve the capacity of disaster situational awareness for responders. In the paper, UAV path planning is modelled as the optimization problem, in which fitness functions include travelling distance and risk of UAV, three constraints involve the height of UAV, angle of UAV, and limited UAV slope. An adaptive selection mutation constrained differential evolution algorithm is put forward to solve the problem. In the proposed algorithm, individuals are selected depending on their fitness values and constraint violations. The better the individual is, the higher the chosen probability it has. These selected individuals are used to make mutation, and the algorithm searches around the best individual among the selected individuals. The well-designed mechanism improves the exploitation and maintains the exploration. The experimental results have indicated that the proposed algorithm is competitive compared with the state-of-art algorithms, which makes it more suitable in the disaster scenario.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

In the past decades, various natural disasters have happened and caused considerable damages to humans. Earthquake, tsunami, hurricane, and extreme temperature have resulted in lots of losses, such as Indonesia was hit by Tsunami on December 26, 2004, the Hurricane Katrina happened in the U.S. Gulf Coast on August 29, 2005, Wenchuan earthquake occurred in 2008 [1,2]. It is vital to enhance the ability of disaster prediction, evaluation, and response. When a disaster happens, the most critical issue is to preserving lives. In the disaster context, it is known that the first 72 h are very significant. Research and rescue work should be conducted as quickly as possible. The main problem for disaster responders is to lack disaster situational awareness and communication during the disaster. Nowadays, one of the most effective approaches is the unmanned aerial vehicle (UAV). UAVs have various advantages, such as low cost, better robust and can carry out complicated tasks. Using UAVs, disaster responders can better understand which infrastructures are damaged and the number of people affected by the disaster [2–5]. UAVs can improve the ability of disaster prediction, assessment, and response. Among these applications, UAV path planning is very important, which has been investigated worldwide.

There are many challenges along a path, such as mountains and bad weather. Path planning is a rather tough optimization problem that seeks an optimal route while satisfying various constraints. The UAV path problem is defined as the sum of flight length, safety, altitude and three penalties, which are the climbing/gliding slope, the turning angle and the terrain constraint [6]. The length, height, and smoothness of the planning path are modelled, in which constraints mainly derive from obstacles and the ground [7]. To determine the optimal configuration of the longitudinal elements of each UAV at the lowest cost, D. Zhang & Duan considered the threat sources and coordination constraints of UAVs [8]. The UAV path planning was formulated as a weighted anisotropic shortest path problem to seek an optimal path in the presence of moving obstacles and constraints [9]. The four types of constraints considered in the UAV path planning problem are: the maximum bending angle, the minimum route segment length, the minimum flight altitude and the maximum ascent/diving angle [10]. The core of path planning is to find a path that can meet all constraints, including pitch angle, yaw angle, minimum track length, turning radius, height, and voyage [11]. Designers must consider the relationship between threat exposure and fuel consumption of the UAV [12]. The UAV path

* Corresponding author at: School of Management Science and Engineering, Nanjing University of Information Science & Technology, Nanjing, China.
E-mail address: yuxb111@163.com (X. Yu).

https://doi.org/10.1016/j.knosys.2020.106209
0950-7051/© 2020 Elsevier B.V. All rights reserved.

optimization problem minimizes the set of four terms: the non-feasible curves, length of the curve, flight paths, and curves with a prescribed minimum curvature radius [13]. The objective programme of UAV formation path planning can be defined as the sum of the terrain cost, collision cost, minimal flight path length cost and radar detection cost among UAVs, in which the constraints are incorporated into the objective function [14]. The four constraints of safe distance between multiple UAVs, maximum communication distance between UAVs, collision avoidance with obstacles, and sensor coverage on target are considered in UAV path planning [15]. Total distance, total delivery time, number of used drones, and maximum speed are considered to form a UAV planning model [16]. In order to guarantee the safety of flight, the hard constraints that must be met and the soft constraints to be optimized are raised separately as two criteria [17]. Path planning is a constrained optimization problem that represents two main aspects of path planning problems: UAV navigation and SAR imaging [18]. Mathematically, UAV path planning can be modelled as a constraint satisfaction problem. A fitness function is used to ensure that all constraints are fulfilled, thereby minimizing the optimization criteria for the problem [19]. When the UAV visits all targets and returns to the base in a continuous terrain monitored by radar, objectives including minimum total distance and radar detection threat are considered [20]. The UAV path planning cost function is expressed as the weighted sum of the fuel cost, threat cost and deviation cost [21]. In the performance evaluation of a UAV flight path, fuel cost and threat cost are commonly used indicators [22]. The grey wolf optimizer (GWO) and symbiotic organisms search (SOS) are combined to solve the problem, whose fitness functions are the fuel cost and the threat cost. The GWO improves the convergence rate and the SOS enhances the exploitation capacity [21]. A reinforcement learning method based on GWO is implemented to solve the path planning problem, in which four operations are introduced. The experimental results have revealed that the algorithm can find a good route in a complicated environment. These well-designed mechanisms allow algorithms to achieve good performance [22].

The UAV applications described above are mainly focused on the military field. The UAV path planning problem for disaster management is rarely discussed. However, disasters occur more frequently, resulting in widespread damage. Therefore, it is necessary to study the problem in the disaster scenario because UAVs play an influential role in disaster information collection, analysis, forecasting, and so on.

The most crucial issue for UAV path planning is achieving the optimal solution with minimal comprehensive costs. Previous studies use the A* algorithm, dynamic programming, and geometric algorithms to solve the problem [23]. However, these methods are time-consuming and suffer from relatively slow speed and execution [24]. In recent years, bio-inspired intelligence models have been increasingly used to solve the problem. Computational intelligence can effectively overcome these defects, which can be used in UAV path planning. Various research has focused on computational intelligence to solve the problem, in which the genetic algorithm (GA), particle swarm optimization (PSO), GWO algorithm and ant colony optimization (ACO) are widely used [21,22,24]. Every algorithm has its own features. Population extremal optimization (PEO) has attracted increasing attention. PEO changes bad individuals in the population, which distinguishes it from conventional evolutionary algorithms (EAs). It is also a very promising optimization algorithm as many researches have indicated that PEO can achieve superior performances in the optimization field [25,26]. It is also a good candidate for the optimization problem of UAV. Compared with PSO and GA, differential evolution (DE) algorithm is simple and easy to implement. It also provides various mutation operators, which distinguish it from other EAs.

UAV path planning is a constrained optimization problem. The penalty method and constraint domination are two main approaches to work with constraints [27]. Some adopt the penalty method [10,20], which introduces a penalty function into the original objective function to penalize infeasible solutions [28,29]. The constraints, such as the climbing/gliding slope, the turning angle and the terrain constraint, are considered as a part of objective functions [10]. Constraint domination is used to compare the feasible solution and infeasible solution, as proposed by Deb [30]. The main idea of constraint domination is that feasible solutions are better than infeasible solutions, and infeasible solutions with low constraint violations are greater than those with high violations.

Both methods make infeasible solutions less likely to be selected in the EAs selection stage. In EAs, mutation, crossover, and selection operators are often involved, in which mutation is used to realize global search and crossover is used for local search. When the constrained problem becomes increasingly complicated, the feasible space becomes narrower. In biology, some individuals have high fitness values, while others have low fitness values. High-fitness individuals have a high probability of producing offspring. The better the individual, the greater chance it has to generate offspring. Feasible solutions have higher probabilities to be selected to produce offspring than infeasible solutions. The offspring produced are more likely to be feasible solutions. The simulated binary crossover (SBX) and arithmetic crossover are often applied to produce offspring near their parents in the decision space. Most objective values of these offspring will be in the vicinity of their parents in the objective space [31].

The above observation provides our motivation. An adaptive selection mutation operator is designed to give individuals with good fitness values higher probabilities to be selected. All individuals are ranked in order of the fitness values of feasible solutions and constraint violations of infeasible solutions. The algorithm searches for the best individual from the selected individuals. The novel mutation operator can enhance local search capabilities while maintaining the diversity of the population. The proposed algorithm is applied to solve the UAV path planning problem.

Based on the above analysis, the main contributions of the paper can be summarized as follows:

(1) The UAV path planning is formulated as a constrained optimization problem in the disaster scenario.

(2) An adaptive selection mutation constraint differential evolution (CDE) algorithm is proposed.

(3) The UAV path planning problem is solved by CDE. Numerical simulations have illustrated that CDE is competitive compared with the state-of-art algorithms.

The rest of the paper is organized as follows. Section 2 illustrates the UAV path planning model. CDE is proposed in Section 3. In Section 4, how to solve the path planning problem based on CDE is introduced. In Section 5, numerical simulations are provided, and conclusions are made in Section 6.

2. UAV path planning model

2.1. Constrained optimization problem (COP)

UAV path planning is the COP. In real applications, many problems involve constraints. Generally, the formulation of the problem can be described as follows:

$$\min f(x) \tag{1}$$

$$s.t. \begin{cases} g_j(x) \leq 0, j = 1, 2, \ldots, l \\ h_j(x) = 0, j = l+1, \ldots, p \\ L_i \leq x_i \leq U_i, i = 1, 2, \ldots, D \end{cases} \tag{2}$$

where $x = \{x_1, x_2, \ldots, x_D\}$ is the decision variable, f is the decision function, $g_j(x)$ is the inequality constraint, $h_j(x)$ is the equality constraint, U_i and L_i is the upper and lower boundary of x_i, p is the number of constraints, and D is the dimension of the problem.

If a solution x satisfies the $g_j(x)$ and $h_j(x)$ simultaneously, it is regarded as a feasible solution. Otherwise, it is the infeasible solution. For feasible solutions, the constraint violation (cv) is zero. On the opposite, the cv of the infeasible solution is more than zero. The cv of infeasible solution is calculated as follows:

$$cv_j = max\left(0, g_j(x)\right), j = 1, 2, \ldots, l \tag{3}$$

$$cv_j = max(0, \left|h_j(x)\right|), j = l + 1, \ldots, p \tag{4}$$

where cv_j is the jth constraint violation.

2.2. Path representation

Path parameterization technology is often adopted to describe the path because it can depict paths well through different parameters. There are two methods to represent the path [8]. The first is the rotated coordinated frame, which can reduce the dimensions of the problem. However, it cannot guide UAVs effectively to fly through obstacles as it is only along the rotated X-axis. Therefore, its path representation ability is limited. The second is the B-Spline. As it is flexible and can maps UAV paths with arbitrary and smooth shapes, it is widely adopted in path representation [13,18]. Moreover, the approach can be implemented by a few parameters, which can save computational time. It is essential in disaster scenarios. The B-Spline is introduced as follows:

Assume $n + 2$ control points with coordinates (xc_0, yc_0, zc_0), $\ldots, (xc_k, yc_k, zc_k), \ldots, (xc_{n+1}, yc_{n+1}, zc_{n+1})$. These control points can formulate the B-Spline. The discrete serial point with coordinates (xp_t, yp_t, zp_t) can be generated as:

$$\begin{cases} xp_t = \sum_{j=0}^{n+1} xc_j \cdot B_{j,k}(t) \\ yp_t = \sum_{j=0}^{n+1} yc_j \cdot B_{j,k}(t) \\ zp_t = \sum_{j=0}^{n+1} zc_j \cdot B_{j,k}(t) \end{cases} \tag{5}$$

Where $B_{j,k}(t)$ is the blending function of the curve, k is the sequence of the curve, which is related to the curve's smoothness. The blending functions are defined recursively in terms of a set of $Knot$ values and are expressed as:

$$B_{j,1}(t) = \begin{cases} 1 & if\ Knot(j) \le t \le Knot(j+1) \\ 1 & if\ Knot(j) \le t < Knot(j+1)\ and\ t = n - k + 3 \\ 0 & otherwise \end{cases} \tag{6}$$

$$B_{j,k}(t) = \frac{(t - Knot(j)) \times B_{j,k-1}(t)}{Knot(j+k-1) - Knot(j)} + \frac{(Knot(i+k) - t) \times B_{i+1,k-1}(t)}{Knot(j+k) - Knot(j+1)} \tag{7}$$

$$\begin{cases} Knot(j) = 0 & if\ j < k \\ Knot(j) = j - k + 1 & if\ k \le j \le n \\ Knot(j) = n - k + 2 & if\ n < j \end{cases} \tag{8}$$

Where t varies from 0 to $n-k+3$ in constant steps, which offers a series of discrete points [13].

2.3. Path planning model

There are many examples of UAV applications in various disasters scenarios, such as search, rescue and assessment operations. When disasters occur, UAVs should be sent to disaster sites as quickly as possible. The flight distance of UAVs should be as short as possible. However, there are many challenges along the path, such as mountains and bad weather. Therefore, the purpose of path planning is to minimize the flight distance and risk while meeting the constraints of UAVs and disasters.

Through the B-Spline curve, the path can be represented as a series of discrete points, such as $(p_0, p_1, p_2, \ldots, p_k, \ldots, p_{N+1})$. The coordinates of p_k are (xp_k, yp_k, zp_k). p_0 is the starting point and p_{N+1} is the ending point i.e., the location of the disaster site. The first objective is to minimize the flight distance between the starting point and the disaster site. Generally, the shorter the distance, the less time is required. The distance can be calculated by the Euclidean operator as follows:

$$f_1 = \sum_{k=0}^{N} d_k \tag{9}$$

$$d_k = \sqrt{(xp_{k+1} - xp_k)^2 + (yp_{k+1} - yp_k)^2 + (zp_{k+1} - zp_k)^2} \tag{10}$$

Where k is the index of discrete points from 0 to N.

Given that the size of a UAV, its horizontal direction cannot be omitted. By following Ref. [18], a safe distance r_{safe} is introduced to guarantee that the UAV avoids the terrain boundary.

Let p_k be the discrete point of the UAV path with the coordinates (xp_k, yp_k, zp_k) in Fig. 1(a). Point p_k can be projected to the horizon (x-axis and y-axis). Fig. 1(b) shows a local magnification, where D_{tp} is the projection point of p_k, and the dashed line is the projection of the terrain mesh. r_{safe} is introduced to ensure that the UAV can avoid the terrain boundary. The six filled dot points in Fig. 1(b) are in the range of r_{safe}. In other words, these points will threaten the safety of UAVs. The threat risk can be expressed as follows:

$$t_k = \sum_{j=1}^{N_{gk}} \left(\frac{r_{safe}}{r_{k,j}}\right)^2 \tag{11}$$

Where N_{gk} is the total number of terrain mesh point in the range of r_{safe}, and $r_{k,j}$ is the distance between the kth discrete projection point and the jth terrain mesh point. The overall threat can be obtained as follows:

$$f_2 = \sum_{k=0}^{N+1} t_k \tag{12}$$

The UAV path must meet at least three conditions. The first is that the UAV cannot collide with the terrain, which requires that these discrete points be above the terrain boundary. The value of the z-axis should be greater than the corresponding terrain.

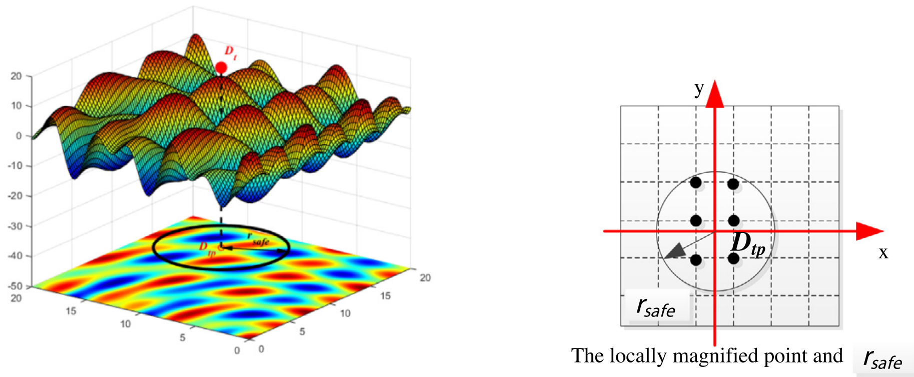
$$cv_1 = \sum_{k=1}^{N} c_{1k} \tag{13}$$

$$c_{1k} = \begin{cases} 0, & if\ z_{pk} - f\left(x_{\mathrm{pk}}, y_{\mathrm{pk}}\right) - z_{min} > 0 \\ z_{pk} - f\left(x_{\mathrm{pk}}, y_{\mathrm{pk}}\right) - z_{min}, & otherwise \end{cases} \tag{14}$$

Where (x_{pk}, y_{pk}, z_{pk}) is the coordinate value of discrete point p_k, z_{min} is the minimum safe height in the vertical direction, and $f\left(x_{pk}, y_{pk}\right)$ is the terrain function. The terrain value of the z-axis can be calculated.

The second condition is that the angle of the UAV is limited. It should not exceed a predetermined maximum angle θ_{max}. The turning angle can be calculated by these discrete points as follows:

$$cv_2 = \sum_{k=1}^{N-1} c_{2k} \tag{15}$$



(a) (b)

Fig. 1. The risk modelling..

$$c_{2k} = \begin{cases} 0, & if \ \cos^{-1}\left(\frac{\overrightarrow{p_k p_{k+1}} \cdot \overrightarrow{p_k p_{k-1}}}{\|\overrightarrow{p_k p_{k+1}}\| \|\overrightarrow{p_k p_{k-1}}\|}\right) - \theta_{max} < 0 \\ \cos^{-1}\left(\frac{\overrightarrow{p_k p_{k+1}} \cdot \overrightarrow{p_k p_{k-1}}}{\|\overrightarrow{p_k p_{k+1}}\| \|\overrightarrow{p_k p_{k-1}}\|}\right) - \theta_{max}, & otherwise \end{cases} \tag{16}$$

Where p_k, p_{k+1} are two discrete and continuous points along the path, and the range of k is between *0* and *N*.

The third condition is the limited UAV slope. The UAV manoeuvrability is limited by the minimal climbing slope β_i and the maximal climbing slope α_k [32]. Both slopes α_k and β_k are related to the altitude z_{pk}. They can be computed as follows:

$$\beta_k = 2.5063 \times 10^{-9} z_{pk}^2 - 6.3014 \times 10^{-6} z_{pk} - 0.3257 \tag{17}$$

$$\alpha_k = -1.5377 \times 10^{-10} z_{pk}^2 - 2.6997 \times 10^{-5} z_{pk} + 0.4211 \tag{18}$$

The UAV slope at the kth point (x_{pk}, y_{pk}, z_{pk}) can be obtained by Eq. (19):

$$S_k = \frac{z_{p(k+1)} - z_{pk}}{\sqrt{(x_{p(k+1)} - x_{pk})^2 + (y_{p(k+1)} - y_{pk})^2}} \tag{19}$$

All points along the path are subject to the constraint. If the slope of the point is out of the range $[\beta_k, \alpha_k]$, it will be penalized. Otherwise, there is no penalization. The penalty function can be expressed as follows:

$$cv_3 = \sum_{k=1}^{N-1} c_{3k} \tag{20}$$

$$c_{3k} = \begin{cases} 0, & if \, \beta_k < S_k < \alpha_k \\ S_k - \alpha_k, & S_k > \alpha_k \\ \beta_k - S_k, & S_k < \beta_k \end{cases} \tag{21}$$

Based on the above discussion, the UAV path planning model can be depicted as follows:

$$\begin{cases} f = w_1 f_1 + w_2 f_2 \\ \quad cv_1 = 0 \\ \quad cv_2 = 0 \\ \quad cv_3 = 0 \end{cases} \tag{22}$$

Where w_1 and w_2 are relative weights between path length and risk, $w_1 + w_2 = 1$. cv is used to measure the violation degree of infeasible routes against constraints. The smaller the value of cv, the better the path, and vice versa. The problem is a typical COP. The mining Eq. (22), while satisfying three constraints through EAs, can result in a set of B-Spline control points, which can generate the desired path.

3. The proposed algorithm

DE, a type of EA, is a straightforward algorithm that is widely used to optimize unconstrained optimization problems. The main procedures of DE are mutation, crossover, and selection. Compared with the selection and crossover operators, the mutation operator has attracted increasing attention. There are various mutation operators, DE/rand/1, DE/rand/2, DE/best/1, DE/current-to-best/1, DE/rand-to-best1/, and DE/rand-to-best/2/, which are utilized to generate new population [33,34].

Taken DE/rand/1 as an example:

$$v_i = x_{r_1} + F \cdot \left(x_{r_2} - x_{r_3}\right) \tag{23}$$

Where r_1, r_2, r_3 are three integers, $i \neq r_1 \neq r_2 \neq r_3$, whose range are between 1 and the population size, and F is a scale factor. Three individuals $x_{r_1}, x_{r_2}, x_{r_3}$ are randomly selected from the current population. There is no prejudice for which individuals are feasible or not. The operator is unfair when the population has both feasible and infeasible individuals at the same time. It is desirable that increasingly more feasible solutions should be selected to generate offspring. Just as in biology, some individuals will have high fitness, while others have low fitness. Low-fitness individuals have a high probability of dying in their generation. High-fitness individuals have a high chance of surviving and producing offspring.

However, the DE/rand/1 operator results in poor exploitation as it has no search direction. It is necessary to strengthen the exploitation ability. In the COP, feasible solutions contain information that satisfies constraints, and they should have more possibilities to generate offspring. Conversely, infeasible solutions violate constraints. These solutions have imperfect information. According to the constraint domination theory, the worse the constraint violation, the worse the individual [30]. These individuals should have a lower chance to be selected to generate offspring. Based on the motivation, the current population is divided into feasible and infeasible individuals. For feasible individuals, fitness values are used to determine the possibility that they are selected. For infeasible individuals, the worse the constraint violation, the lower the chance of selection.

Ranks depend on fitness values and the constraint violation. According to the constraint domination theory, feasible individuals with smaller fitness values are better than feasible solutions with larger fitness values; feasible individuals are better than infeasible individuals, and infeasible individuals with smaller constraint violations are better. The better the individual is, the smaller the ranking is. Suppose that there are N solutions, n_1 feasible solutions and n_2 infeasible solutions, in which $N = n_1 + n_2$. n_1 feasible solutions are ranked by descending order of their fitness values. n_2 infeasible solutions are sorted by descending order based on their constraint violations. As feasible solutions are better than infeasible solutions, the ranking number of n_1 feasible solutions is lower than that of n_2 infeasible solutions.

Algorithm 1 Sort individuals

Input: individuals x, the fitness values f and corresponding constraint violations cv, population size N
Output: the sorted $newx$ and their fitness values $newf$
Divide individuals into feasible and infeasible solutions according to their cv
Sort feasible solutions according to their fitness values by descending order and form individuals order x_1 and corresponding fitness values f_1
Sort infeasible solutions according to their constraint violations by descending order and form individuals order x_2 and corresponding fitness values f_2
$newx = [x_1; x_2]$;
$newf = [f_1; f_2]$;

The ranking method for the global unconstrained optimization problem is adopted [35]. The selection probability $P(j)$ can be calculated as follows:

$$P(j) = \frac{N-j}{N} \tag{24}$$

Where N is the population size, and $j(j = 1, 2, \ldots, N)$ is the ranking index of individuals. The better the individual is, the higher the selection probability is. Algorithm 2 is implemented to select three vectors from the current population. The operator is similar to the tournament selection in GA. It is used to accelerate the convergence of the algorithm and improve the exploitation capacity.

Algorithm 2 Ranking selection method

Input: The vector index i, N
Output: The selected index r_1, r_2, r_3
for $j = 1{:}N$
  $p(j) = \frac{N-j}{N}$;
end
$r_1 = floor(rand \times N) + 1$;
while($rand > p(r_1) | r_1 == i$)
  $r_1 = floor(rand \times N) + 1$;
end
$r_2 = floor(rand \times N) + 1$;
while($rand > p(r_2) | r_2 == r_1 | r_2 == i$)
  $r_2 = floor(rand \times N) + 1$;
end
$r_3 = floor(rand \times N) + 1$;
while($r_3 == r_1 | r_3 == r_2 | r_3 == i$)
  $r_3 = floor(rand \times N) + 1$;
end

When three integers r_1, r_2, r_3 are selected, three individuals $x_{r_1}, x_{r_2}, x_{r_3}$ will be utilized to generate vector v_i. When three individuals from algorithm 2 are obtained, they have higher probabilities of being feasible solutions, especially in the later phase of the evolution. It cannot be guaranteed that three individuals are feasible as randomness is one of the main features of EAs. According to the above ranking method, feasible solutions have higher probabilities, while infeasible solutions have lower probabilities.

For the mutation phase, DE/rand/1 and DE/best/1 are widely used. DE/rand/1 is to search around x_{r_1}, while DE/best/1 is used to search for the best individual found so far. The convergence of DE/best/1 is better when the diversity is weak. However, DE/rand/1 performs the random search, which results in weak convergence. The proposed algorithm can search for the best individual among the three individuals $x_{r_1}, x_{r_2}, x_{r_3}$, which can combine the mechanism of both DE/best/1 and DE/rand/1. Thus, the convergence of the algorithm can be improved, and the diversity of the population does not deteriorate. In Algorithm 3, we directly compare the fitness values of three individuals $x_{r_1}, x_{r_2}, x_{r_3}$. The algorithm searches around the individual with the best fitness value among the three individuals $x_{r_1}, x_{r_2}, x_{r_3}$. The algorithm pushes the population towards the best solution.

Algorithm 3 Mutation

Input : $x_{r_1}, x_{r_2}, x_{r_3}$ and their fitness value $f(x_{r_1}), f(x_{r_2}), f(x_{r_3})$
Output: The vector v
if($f(x_{r_1}) \geq f(x_{r_2})$ && $f(x_{r_1}) \geq f(x_{r_3})$)
  $v = x_{r_1} + F \cdot (x_{r_2} - x_{r_3})$
  return
end
if($f(x_{r_2}) \geq f(x_{r_1})$ && $f(x_{r_2}) \geq f(x_{r_3})$)
  $v = x_{r_2} + F \cdot (x_{r_1} - x_{r_3})$
  return
end
if($f(x_{r_3}) \geq f(x_{r_2})$ && $f(x_{r_3}) \geq f(x_{r_1})$)
  $v = x_{r_3} + F \cdot (x_{r_1} - x_{r_2})$
  return
end

The crossover operator is used to generate a trial vector u as follows:

$$u_i^j = \begin{cases} v_i^j & if\ rand_j(0,1) \leq CR\ or\ (j = j_{rand}) \\ x_i^j & others \end{cases} \tag{25}$$

Where $i(i = 1, 2, \ldots, N)$ is the index of the population size, N is the population size, $j(j = 1, 2, \ldots, D)$ is the index of the problem dimension and D is the dimension of the problem, CR is the crossover operator, and j_{rand} is a random number between 1 and D. j_{rand} is to ensure that u_i^j is different from x.

As the conventional DE is used to optimize the unconstrained optimization problem, we can extend it to the constrained optimization problem. Generally, the penalty method and constraint domination are often used to make comparisons between feasible and infeasible solutions [36]. As the latter is robust, it is adopted here.

$$x_i = \begin{cases} u_i & if\ f(u_i) \leq f(x_i) \text{ and } cv(u_i) = cv(x_i) = 0 \\ u_i & if\ cv(x_i) > cv(u_i) > 0 \\ u_i & if\ cv(u_i) = 0\ and\ cv(x_i) > 0 \\ x_i & otherwise \end{cases} \tag{26}$$

Where i is the index of the population size ($i = 1, 2, 3, \ldots, N$), N is the population size, and $f(u_i)$, $f(x_i)$, $cv(u_i)$, $cv(x_i)$ are fitness values and constraint violations of individual x_i and trial vector u_i. If both are feasible solutions, x_i will be replaced by u_i when the fitness value of u_i is lower than x_i. If individual x_i is an infeasible solution, u_i will take the place of x_i when u_i is a feasible solution or the constraint violation of u_i is smaller than x_i.

Based above analysis, the pseudocode of the proposed algorithm is depicted as follows:

The proposed algorithm CDE

```
Generate the initial population x
Evaluate the fitness value f and corresponding constraint violations cv
While the stop criterion is not met do
    [x, f] = Sort(x, f, cv);
    for i = 1: N
    [r1 r2 r3] = RankingSelection(i);
      v_i = Mutation(x_r1, x_r2, x_r3, f(x_r1), f(x_r2), f(x_r3));
     end
     for i = 1: N
        for j = 1: D
          if(rand_j [0, 1) ≤ CR or (j = j_rand))
             u_ij = v_ij;
          else
          u_ij = x_ij;
         end
      end
     for i = 1: N
        Evaluate u_i
        if(u_i is better than x_i)
          x_i = u_i;
        end
     end
   end while
```

First, the population is randomly generated in the search space as the initial population, and their fitness values and corresponding constraint violations are evaluated. By calling the sort function, the population is sorted from the best to the worst. The vector is generated through the well-designed mutation and crossover. The offspring will be obtained by the selection operator and enter into the next evolution.

The computational complexity of the proposed algorithm for one generation is presented as follows. Given the population size to be N, the dimension to be D. The complexity of dividing the population into feasible and infeasible individuals is $O(N)$. The sorting of these individuals is $O(\log(N))$. The probability calculation is $O(N)$. The complexity of the DE algorithm is $O(N \times D)$. Therefore, the total complexity of the proposed algorithm is $O(2N + \log(N) + N \times D)$, which does not significantly increase compared with the conventional DE.

4. UAV path planning based on CDE

We now use the proposed CDE to address the UAV path planning problem. In Section 2, the problem was formulated as the COP. The details of the procedure are presented as follows:

Step 1: Set disaster scenarios. In general, massive disasters often happen in remote mountainous areas. A meshed 3-D terrain can be generated to simulate the disaster scenario. The coordinates of starting point p_0 are given beforehand. The coordinates of the destination point p_{N+1} are also specified.

Step 2: Path plan modelling. Set the UAV path parameters, such as safe distance r_{safe} and maximal turning angle θ_{max}. Exploit the B-spline to generate UAV paths and establish the problem as an optimization problem with constraints.

Step 3: Initialization parameters of CDE. Initialize the parameters of CDE, including the maximal generation G_{max}, population size, scaling factor F, and crossover rate Cr. CDE randomly generates a number of individuals in the specified range to prevent the UAV from going out of control. The values of each individual are the physical coordinates of the freely moving B-spline control points. Each B-spline curve is constructed by each individual, and the starting point and the destination point. The fitness value and corresponding constraint can be evaluated by Eq. (22).

Step 4: Optimize the path planning by CDE. The problem is optimized in four stages: adaptive selection, adaptive mutation, crossover, and selection.

Step 5: Stopping criteria. The procedures of step 4 will end when the iteration reaches G_{max}. The output of the proposed algorithm will be the generated path, which will guide the UAV.

5. Experiments and analysis

5.1. Experimental setup

Generally, the testing terrain must be given first. It is often represented by a 3-D model [13] as follows:

$$z(x, y) = \sin(y + a) + b \cdot \sin(x) + c \cdot \cos(y) + d \cdot \cos(y) + e \cdot \cos\left(f \cdot \sqrt{y^2 + x^2}\right) + g \cdot \sin\left(g \cdot \sqrt{y^2 + x^2}\right) \quad (27)$$

Where a, b, c, d, e, f, g are constants. They are defined to generate a smooth surface to simulate the terrain. Here, $a = 1, b = 1, c = 1.8, d = 1.8, e = 1, f = 1.8, g = 1$. The start point of UAV and the destination point are [4 4 1], [18 17 10], respectively.

For the parameters of the UAV, the maximal turning angle θ_{max} is 30°, and the minimal safe distance r_{safe} is 200 m. Eight points are used to generate the B-spline; in addition to the starting and ending points, there are six free points. Therefore, the dimensions of each algorithm are set to $6 \times 3 = 18$. There are three experiments corresponding to three disaster scenarios. We have set three weights ($[w_1 = 0.8, w_2 = 0.2], [w_1 = 0.5, w_2 = 0.5], [w_1 = 0.2, w_2 = 0.8]$) among two objective functions. Three weights denote different disaster scenarios and balance the travelling distance and risk of the UAV.

Two indicators are used to measure the performance of the algorithms:

$$Feasiblerate = (feasibleruns)/(totalruns) \quad (28)$$

$$f\left(x^*\right) = min(f(x)) \quad (29)$$

Where feasible runs denote that at least one feasible solution is found in the given function evaluations, and $f(x^*)$means the best results obtained in this run.

5.2. Comparison algorithms

Four representative state-of-the-art DE variants are chosen to draw comparisons. Feasible and infeasible DE (FIDE) adopts different mutation strategies for feasible solutions and infeasible solutions [27]. Multi-objective optimization techniques are combined with DE to design CMODE to cope with the COP [37]. $(\mu + \gamma)$-constraint DE and an adaptive trade-off model are integrated to form the $(\mu+\gamma)$-CDE algorithm [38]. DE with ranking (RankDE) mutation operators is proposed to guide the search direction [35].

The parameter settings of the proposed algorithm, FIDE, and RankDE are as follows: the population size is 50, crossover constant CR increases from 0.4 to 0.8 linearly, and scale factor F is reduced from 1 to 0.7. The parameter configurations of $(\mu + \gamma)$-CDE and CMODE are based on their original references [37,38].

To draw fair and thorough comparisons, the maximal function evaluations are set according to the constraints posed to the UAV path planning. If the first constraint (c1) is considered in Eqs. (13)–(14), the maximal function evaluations are set to 10 000. If both c1 and the second constraint (c2) in Eqs. (15)–(16) are considered, they are set to 20 000. If three constraints c1, c2 and the third constraint in Eqs. (15)–(16) are considered during optimization, they are set to 30 000. Each algorithm runs ten times independently for each disaster scenario.

Table 1
The two performance indicators of five algorithms.

W	Constraint		CDE	RankDE	$(\mu+\gamma)$-DE	FIDE	CMODE
[0.8, 0.2]	c1	Mean	**20.484**	20.939+	21.948+	21.652+	21.636+
		Std	**0.0312**	0.168	0.199	0.203	0.0345
		Success rate	100%	100%	100%	100%	100%
	c1+c2	Mean	**20.00**	20.04=	20.77+	20.90+	20.47+
		Std	**0.013**	0.016	0.0457	0.056	0.029
		Success rate	100%	100%	100%	100%	100%
	c1+c2+c3	Mean	**26.45**	27.9=	28.50=	33.60=	26.5=
		Std	**0.128**	11.16	18.23	38.14	0.335
		Success rate	100%	100%	100%	60%	100%
[0.5, 0.5]	c1	Mean	**14.139**	14.244=	14.973+	14.726+	14.659+
		Std	**0.0452**	0.0928	0.0927	0.0538	0.0346
		Success rate	100%	100%	100%	100%	100%
	c1+c2	Mean	**13.568**	13.677=	14.11+	14.785+	14.058+
		Std	**0.0228**	0.0168	0.0298	0.1123	0.017
		Success rate	100%	100%	100%	100%	100%
	c1+c2+c3	Mean	19.56	20.937+	20.149=	25.795+	**19.40**=
		Std	3.578	**1.264**	9.02	107.262	5.08
		Success rate	100%	100%	90%	30%	100%
[0.2, 0.8]	c1	Mean	**6.702**	6.717=	7.029+	6.938+	6.863+
		Std	0.0163	0.0126	0.0112	0.0045	**0.0011**
		Success rate	100%	100%	100%	100%	100%
	c1+c2	Mean	**6.393**	6.377=	6.774+	7.020+	6.624+
		Std	0.0105	0.0105	0.0322	0.0114	**0.0084**
		Success rate	100%	100%	100%	100%	100%
	c1+c2+c3	Mean	**9.218**	10.313+	11.442=	12.427=	10.090=
		Std	1.831	1.971	1.368	5.843	**0.459**
		Success rate	100%	100%	100%	90%	100%
+/= /−				3/6/0	6/3/0	7/2/0	6/3/0

5.3. Experimental results and analysis

The statistical results of the two performance indicators are obtained and listed in Table 1, where the mean values of $f(x^*)$ and feasible rate are recorded, and the best values are highlighted. It can be seen that when the constraints c1, c1+c2 are considered, the success rate is 100%, indicating that all five algorithms can find safe and smooth UAV paths in the run. However, when three constraints are imposed on the optimization problem, CDE can obtain the highest success rate up to 100%. CMODE ranks second. However, the success rate of FIDE is only 30%, which is the worst among the five algorithms when $w_1 = 0.5$.

The mean and standard deviation of $f(x^*)$ can indicate the search capacity and robustness of a bio-inspired algorithm. The mean results of $f(x^*)$ are the lowest in the first and third scenarios, and the standard deviation is the lowest in the first scenario. The proposed algorithm has obtained the best performance in all three situations. Therefore, CDE has achieved superior performances in the statistical sense.

When three constraints are considered in the optimization process, it is tough to find a feasible solution for the algorithm. The feasible space becomes very small, and various infeasible solutions are generated in the evolution. As FIDE uses only two mutations DE/rand/1 and Gauss mutation for feasible and infeasible solutions, respectively, the search ability is limited when the problem becomes complicated. Although the selection operator RankDE is implemented and the individuals with better information can be selected with high probabilities, there is no search direction. The convergence capacity is relatively weak. $(\mu+\gamma)$-DE adopts three mutation strategies and binomial crossover to generate the offspring. There is no selection operator during the mutation phase. CMODE combines the multi-objective optimization theory into the algorithm. The whole process is somewhat complicated. In CDE, the better the individual, the greater its chance of being selected to enter the mutation stage. The mutation operator can make the algorithm search around the best individual among the three selected individuals. The novel mutation mechanism can improve the local search ability and maintain the exploration capacity. The algorithm gives the population more chances to enter feasible space. The opportunity for generating feasible solutions increases relative to the other algorithms. Therefore, the performance is superior to the four state-of-the-art DE variants.

The best UAV paths generated by CDE, RankDE, $(\mu+\gamma)$-DE, FIDE, and CMODE, are plotted in Figs. 2–4 when $w_1 = 0.8$ and $w_2 = 0.2$. As is shown, the five algorithms can successfully find a safe and smooth path when the constraints are c1 and c1+c2. The routes do not change significantly when the constraint is changed from c1 to c1+c2. This indicates that solutions generated with constraint c1 also satisfy constraint c2. However, when three constraints are considered together, all paths are changed significantly, which means that solutions generated using constraints c1 and c2 do not satisfy constraint c3. It is complicated to meet constraint c3. The paths generated have at least a corner. FIDE is the worst among them, as there are many corners along the route. These corners will make the UAV flight very difficult. Compared with RankDE, FIDE, and $(\mu+\gamma)$-DE, the path from CMODE is smoother.

To draw a comprehensive conclusion, nonparametric statistical tests at a significant level of 5% have been implemented to

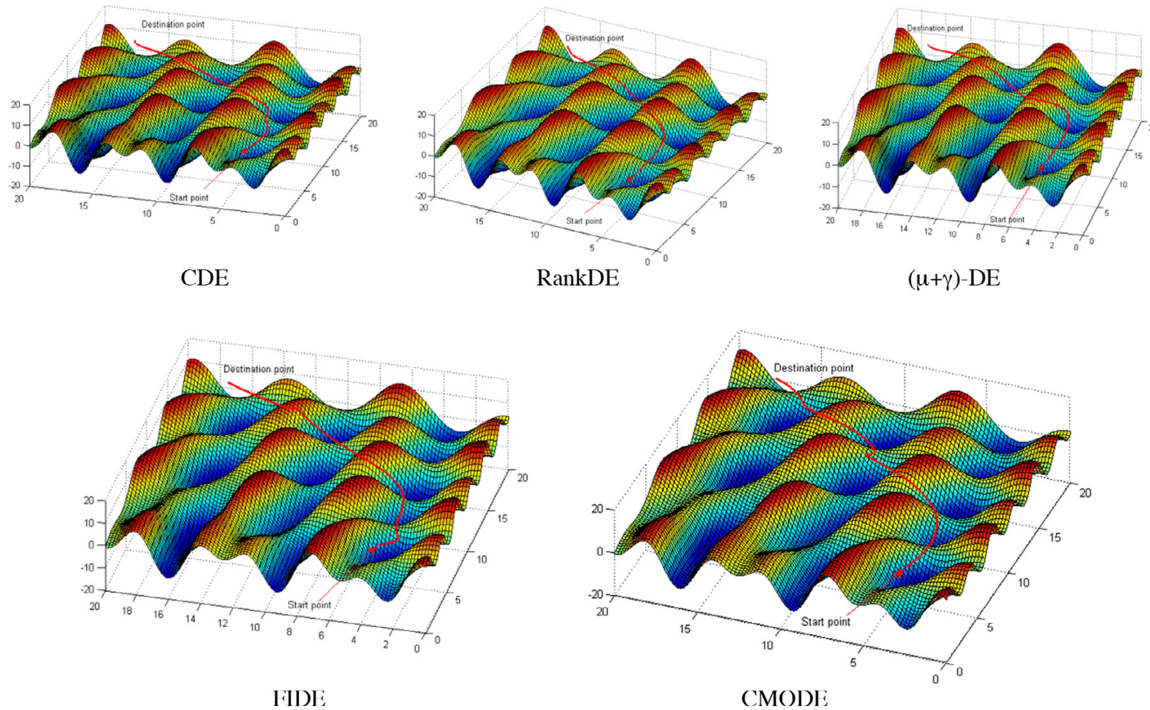


Fig. 2. The UAV paths from five algorithms when c1 is considered.

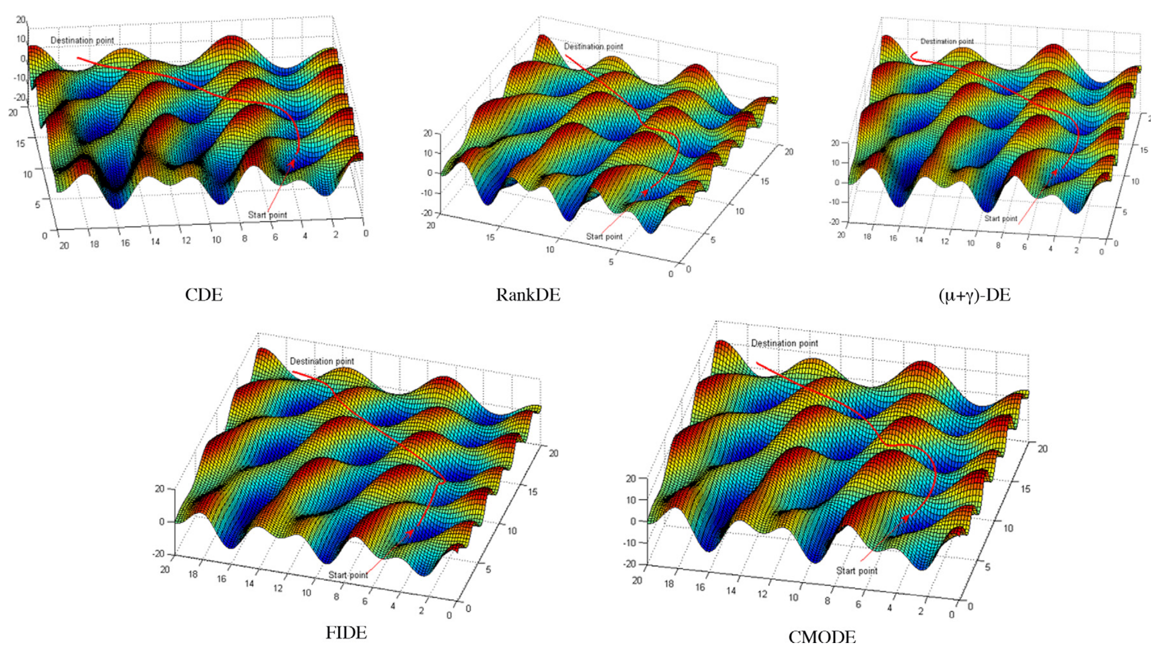


Fig. 3. The UAV paths from five algorithms when both c1 and c2 are considered.

validate the significance of the better performance achieved by the proposed algorithm with respect to the other algorithms [39]. The results are listed in Table 1, in which "+ (Better)", "= (Same)", and "- (Worse)" reveal that the proposed algorithm performs significantly better than, almost the same as, and worse than the compared algorithm, respectively. It can be seen that the proposed algorithm is better than the compared algorithms in most scenarios.

5.4. Discussion

To further validate the effectiveness of CDE, an additional experiment has been performed. The experiment studies the function evaluations that each algorithm needs to find a feasible solution. All parameters of the five algorithms are the same as in the above section. Three constraints are considered, as they are the most difficult to optimize. The results are plotted in Fig. 5. It can be observed that CDE outperforms the other algorithms. It needs the fewest function evaluations to find a feasible solution. It is worth noting that time is very limited during disasters, which requires that the path plan be generated as soon as possible. With

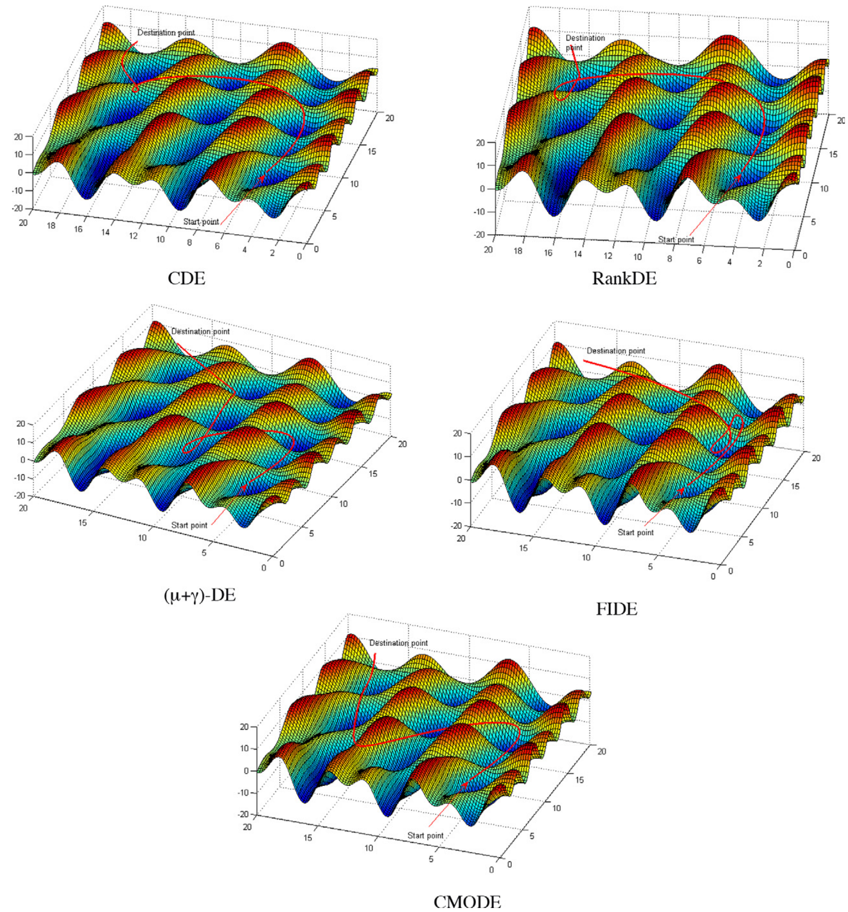


Fig. 4. The generated paths from five algorithms when three constraints are considered.

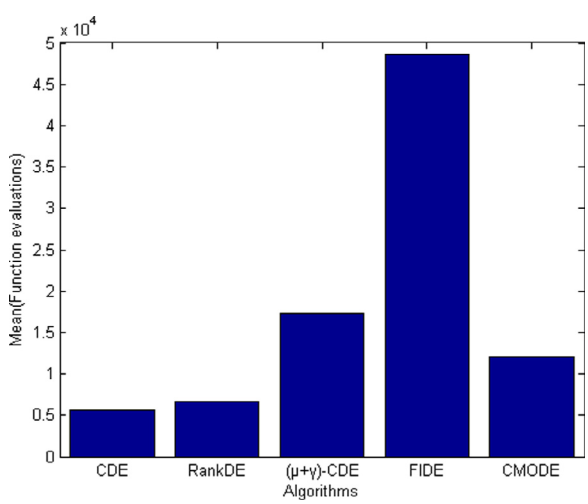


Fig. 5. The minimal function evaluations to find a feasible solution.

the help of the adaptive selection mutation operator, CDE can search towards the feasible space more quickly. RankDE is ranked the second.

To measure the advantage of adaptive selection mutation, the proposed algorithm is compared with the conventional DE. In the three constraint scenarios ($w_1 = 0.8$ and $w_2 = 0.2$), the same experiments are implemented on DE. All parameters of DE are same to those of CDE. The comparison results are listed in Table 2. The results of the conventional DE are worse than those of the proposed algorithm in all three scenarios. Especially when three limitations are considered together, the conventional DE cannot find feasible solutions in four runs. This is because the conventional DE algorithm has poor convergence ability. As the feasible space narrows, it is tough for DE to escape infeasible areas. Therefore, it cannot solve complicated constraint problems. However, the proposed algorithm with adaptive selection mutation can enhance the convergence rate and push the population towards feasible regions. Therefore, it can find better results than the conventional DE.

The convergence curves of the two algorithms are plotted in Fig. 6 when c1, c1+c2, c1+c2+c3 are considered. It can be seen that DE shows inferior convergence as it has no guide direction for the mutation operator and falls into the local optimum. CDE has achieved faster convergence speed and smaller fitness values. The results have verified that the proposed mutation operator can contribute to improving the search ability of CDE.

6. Conclusions

UAVs are among of the most useful tools for situational awareness in disaster scenarios. In this paper, UAV path pathing is discussed and constructed as a COP. The fitness functions include the travelling distance and risk, and three constraints are considered: height of the UAV, angle of the UAV, and limited UAV slope. The objective of UAV path planning is to minimize the fitness functions and satisfy three constraints. The UAV path is generated by free points and represented as a series of discrete points. In this paper, a novel CDE algorithm based on the DE algorithm is implemented to solve the problem. In CDE, an adaptive selection mutation operator is designed to improve the exploitation ability.

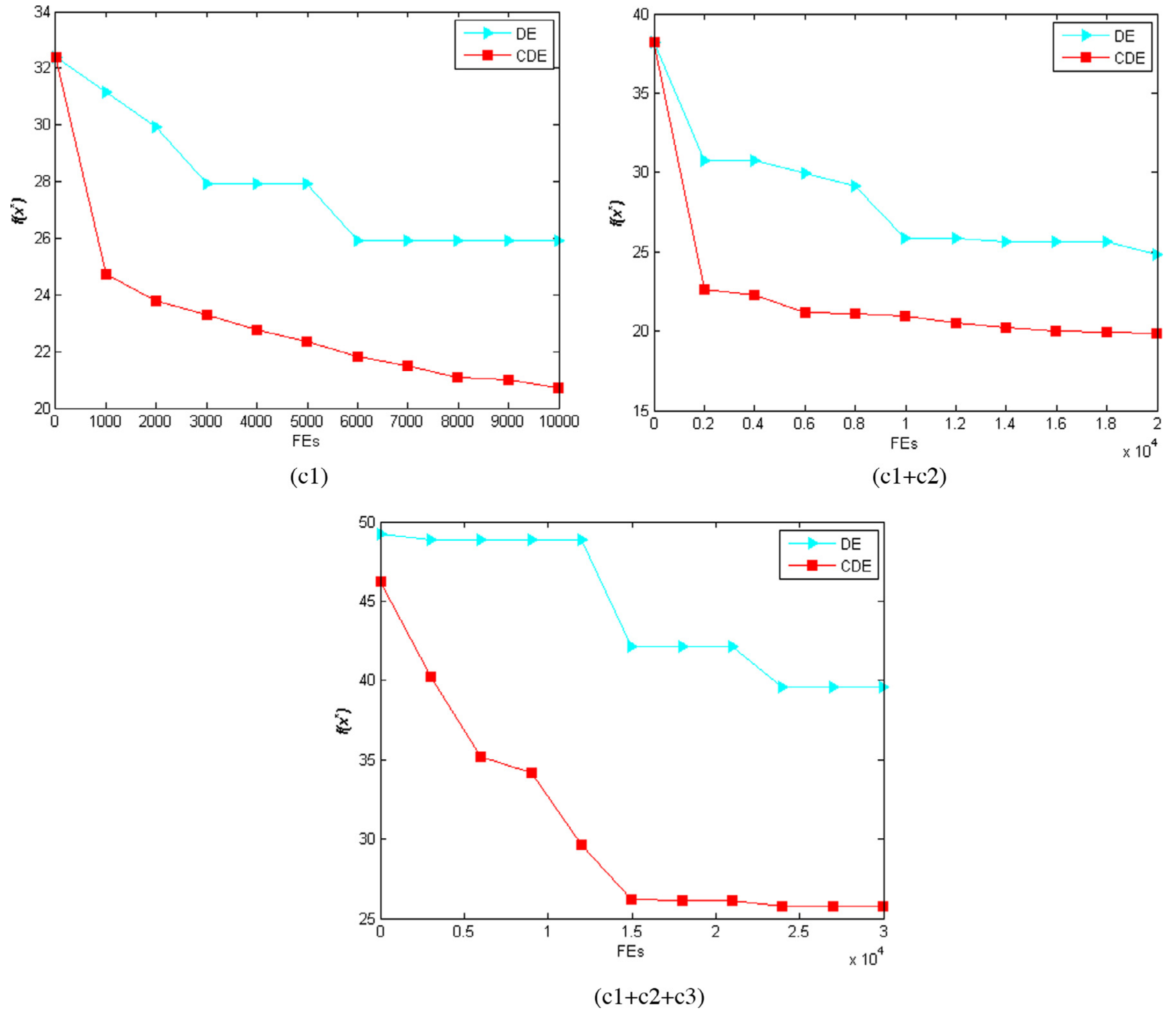


Fig. 6. The convergence curves of CDE and the conventional DE.

Table 2
The comparison results between CDE and the conventional DE.

Constraint	Criteria	CDE	DE
c1	Mean	**20.4839**	25.97
	Std	0.0312	0.19
	Success rate	**100%**	**100%**
c1+c2	Mean	**20.00**	24.86
	Std	0.013	0.160
	Success rate	**100%**	**100%**
c1+c2+c3	Mean	**26.45**	39.55
	Std	0.128	14.10
	Success rate	**100%**	60%

Different individuals have different probabilities to be selected to enter the mutation phase according to their fitness values and constraint violations.

Four state-of-the-art DE variants, FIDE, CMODE, $(\mu+\gamma)$-CDE, and RankDE, are selected to draw comparisons. CDE is applied to solve the UAV path planning model. The five algorithms are compared according to their statistical results and generated paths. The comparisons results have indicated that the CDE algorithm is superior to the other four algorithms. The proposed algorithm can successfully find the optimal solution and generate smooth paths, which make it more appropriate in the application of disaster emergency management.

CRediT authorship contribution statement

Xiaobing Yu: Conceptualization, Methodology, Writing - original draft. **Chenliang Li:** Writing - review & editing. **JiaFang Zhou:** Data curation.

Declaration of competing interest

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

Acknowledgements

This research was funded by the China Natural Science Foundation (No.71974100, No.71503134), Natural Science Foundation in Jiangsu Province (No. BK20191402), Major Project of Philosophy and Social Science Research in Colleges and Universities in Jiangsu Province (2019SJZDA039), Key Project of National Social and Scientific Fund Program (16ZDA047), Qing Lan Project (R2019Q05) and the Joint Open Project of KLME & CIC-FEMD, NUIST (KLME202004).

References

[1] E.J. Raker, S.R. Lowe, M.C. Arcaya, S.T. Johnson, J. Rhodes, M.C. Waters, Twelve years later: The long-term mental health consequences of Hurricane Katrina, Soc. Sci. Med. 242 (2019) 112610.

[2] J. Sánchez-García, D.G. Reina, S.L. Toral, A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario, Future Gener. Comput. Syst. 90 (2019) 129–148.
[3] M. Erdelj, E. Natalizio, K. R.Chowdhury, I. F.Akyildiz, Help from the sky: Leveraging UAVs for disaster management, IEEE Pervasive Comput. 1 (2017) 24–32.
[4] W. Pereira Coutinho, J. Fliege, M. Battarra, Glider routing and trajectory optimisation in disaster assessment, European J. Oper. Res. 274 (2019) 1138–1154.
[5] X. Zhou, Z. Yi, Y. Liu, K. Huang, H. Huang, Survey on path and view planning for UAVs, Virtual Real. Intell. Hardw. 2 (2020) 56–69.
[6] X. Zhang, X. Lu, S. Jia, X. Li, A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning, Appl. Soft Comput. 70 (2018) 371–388.
[7] Y. Chen, J. Yu, Y. Mei, Y. Wang, X. Su, Modified central force optimization (MCFO) algorithm for 3D UAV path planning, Neurocomputing 171 (2016) 878–888.
[8] D. Zhang, H. Duan, Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning, Neurocomputing 313 (2018) 229–246.
[9] M. Radmanesh, M. Kumar, M. Sarim, Grey wolf optimization based sense and avoid algorithm in a Bayesian framework for multiple UAV path planning in an uncertain environment, Aerosp. Sci. Technol. 77 (2018) 168–179.
[10] J.-P. Wu, Z.-H. Peng, J. Chen, 3D multi-constraint route planning for UAV low-altitude penetration based on multi-agent genetic algorithm, in: Proceedings of the 18th World Congress The International Federation of Automatic Control Milano, Italy, 2011, pp. 11821–11826.
[11] X. Wu, W. Bai, Y. Xie, X. Sun, C. Deng, H. Cui, A hybrid algorithm of particle swarm optimization, metropolis criterion and RTS smoother for path planning of UAVs, Appl. Soft Comput. 73 (2018) 735–747.
[12] C. Xu, H. Duan, F. Liu, Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning, Aerosp. Sci. Technol. 14 (2010) 535–541.
[13] I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis, A.N. Kostaras, Evolutionary algorithm based offline/online path planner for UAV navigation, IEEE Trans. Syst. Man Cybern. B 33 (2003) 898–912.
[14] S. Shao, Y. Peng, C. He, Y. Du, Efficient path planning for UAV formation via comprehensively improved particle swarm optimization, ISA Trans. (2019).
[15] X. Wang, H. Zhao, T. Han, H. Zhou, C. Li, A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem, Appl. Soft Comput. 78 (2019) 240–260.
[16] B.N. Coelho, V.N. Coelho, I.M. Coelho, L.S. Ochi, K.R. Haghnazar, D. Zuidema, M.S.F. Lima, A.R. da Costa, A multi-objective green UAV routing problem, Comput. Oper. Res. 88 (2017) 306–315.
[17] H. Qiu, H. Duan, A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles, Inform. Sci. 509 (2020) 515–529.
[18] Z. Sun, J. Wu, J. Yang, Y. Huang, C. Li, D. Li, Path planning for GEO-UAV bistatic SAR using constrained adaptive multiobjective differential evolution, IEEE Trans. Geosci. Remote Sens. 54 (2016) 6444–6457.
[19] C. Ramirez Atencia, J. Del Ser, D. Camacho, Weighted strategies to guide a multi-objective evolutionary algorithm for multi-UAV mission planning, Swarm Evol. Comput. 44 (2019) 480–495.
[20] E. Dasdemir, M. Köksalan, D. Tezcaner Öztürk, A flexible reference point-based multi-objective evolutionary algorithm: An application to the UAV route planning problem, Comput. Oper. Res. 114 (2020) 104811.
[21] C. Qu, W. Gai, J. Zhang, M. Zhong, A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning, Knowl.-Based Syst. (2020a) 105530.
[22] C. Qu, W. Gai, M. Zhong, J. Zhang, A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning, Appl. Soft Comput. 89 (2020b) 106099.
[23] S. Bayili, F. Polat, Limited-Damage A*: A path search algorithm that considers damage as a feasibility criterion, Knowl.-Based Syst. 24 (2011) 501–512.
[24] Y. Zhao, Z. Zheng, Y. Liu, Survey on computational-intelligence-based UAV path planning, Knowl.-Based Syst. 158 (2018) 54–64.
[25] K. Lu, W. Zhou, G. Zeng, Y. Zheng, Constrained population extremal optimization-based robust load frequency control of multi-area interconnected power system, Int. J. Electr. Power Energy Syst. 105 (2019) 249–271.
[26] F. Zhao, G.-Q. Zeng, K.-D. Lu, EnLSTM-WPEO: Short-term traffic flow prediction by ensemble LSTM, NNCT weight integration, and population extremal optimization, IEEE Trans. Veh. Technol. 69 (2020) 101–113.
[27] X. Yu, X. Yu, Y. Lu, G.G. Yen, M. Cai, Differential evolution mutation operators for constrained multi-objective optimization, Appl. Soft Comput. 67 (2018) 452–466.
[28] Y. Wang, Z. Cai, Y. Zhou, W. Zeng, An adaptive tradeoff model for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 12 (2008) 80–92.
[29] Y.G. Woldesenbet, G.G. Yen, B.G. Tessema, Constraint handling in multi-objective evolutionary optimization, IEEE Trans. Evol. Comput. 13 (2009) 514–525.
[30] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.
[31] K. Zhang, G.G. Yen, Z. He, Evolutionary algorithm for knee-based multiple criteria decision making, IEEE Trans. Cybern. (2019) 1–14.
[32] E. Besada-Portas, L.d.l. Torre, J.M.d.l. Cruz, B.d. Andrés-Toro, Evolutionary trajectory planner for multiple UAVs in realistic scenarios, IEEE Trans. Robot. 26 (2010) 619–634.
[33] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011) 1679–1696.
[34] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66.
[35] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (2013) 2066–2081.
[36] D. Datta, J.R. Figueira, A real–integer–discrete-coded differential evolution, Appl. Soft Comput. 13 (2013) 3884–3893.
[37] Y. Wang, Z. Cai, Combining multiobjective optimization with differential evolution to solve constrained optimization problems, IEEE Trans. Evol. Comput. 16 (2012) 117–134.
[38] Y. Wang, Z. Cai, Constrained evolutionary optimization by means of ($\mu\mu$ + $\lambda\lambda$)-differential evolution and improved adaptive trade-off model, Evol. Comput. 19 (2011) 249–285.
[39] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.