CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Three-dimensional path planning with enhanced gravitational search algorithm for unmanned aerial vehicle

Keming Jiao[1,2] , Jie Chen[1,2], Bin Xin[3,4], Li Li[1,2], Yifan Zheng[1,2] and Zhixin Zhao[1,2]

[1]School of Electronics and Information Engineering, Tongji University, Shanghai, China
[2]Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai, China
[3]School of Automation, Beijing Institute of Technology, Beijing, China
[4]National Key Lab of Autonomous Intelligent Unmanned Systems, Beijing, China
Corresponding author: Bin Xin; Email: brucebin@bit.edu.cn

Abstract

Path planning for the unmanned aerial vehicle (UAV) is to assist in finding the proper path, serving as a critical role in the intelligence of a UAV. In this paper, a path planning for UAV in three-dimensional environment (3D) based on enhanced gravitational search algorithm (EGSA) is put forward, taking the path length, yaw angle, pitch angle, and flight altitude as considerations of the path. Considering EGSA is easy to fall into local optimum and convergence insufficiency, two factors that are the memory of current optimal and random disturbance with chaotic levy flight are adopted during the update of particle velocity, improving the balance between exploration and exploitation for EGSA through different time-varying characteristics. With the identical cost function, EGSA is compared with seven peer algorithms, such as moth flame optimization algorithm, gravitational search algorithm, and five variants of gravitational search algorithm. The experimental results demonstrate that EGSA is superior to the seven comparison algorithms on CEC 2020 benchmark functions and the path planning method based on EGSA is more valuable than the other seven methods in diverse environments.

1. Introduction

Unmanned aerial vehicle (UAV) is a kind of airplane without human pilots on board [1]. In recent decades, benefiting from the development of intelligence science and technology, the UAV has the advantage of flexibility, high safety, and low cost compared with the manned vehicle, showing great potential in military and civilian fields, such as reconnaissance, surveillance, rescue, and search in various complex environments [2–5]. Generally speaking, as shown in Fig. 1, path planning targets at offering an appropriate path from the starting point to the goal point, simultaneously avoiding the obstacles and meeting the requirements [6–8], e.g., time, energy [9], and distance, which is one of the critical technologies determining the autonomy level of UAV in performing a mission.

Over the past few years, many scholars have done a lot of research and put forward a variety of methods for UAV path planning [10]. Graph-based algorithms, such as probabilistic roadmaps algorithm [11], A star algorithm [12], rapidly exploring random trees based algorithms [13], are a kind of path planning method, converting the search area into a graph and regarding the grid cell as a node in the graph structure [14]. Regular grids are often applied in the environmental model, but they are inappropriate in the larger area environment due to high memory and computation and low precision. To eliminate the shortcomings of regular grids, Shah and Gupta employed quadtree to improve the computing efficiency of A star on visibility graph nodes [15]. Lee et al. applied the Dijkstra algorithm to obtain the path, using the quadtree-based graph that is generated by the quadtree structure [16]. Potential field-based

Figure 1. *An example for UAV path planning.*

method is a class of effective path planning method, and artificial potential field method regards the target and obstacle as the attractive field and repulsive field, respectively, pulling the UAV to move toward the destination and away from obstacles to obtain a proper path [17]. To solve the problem that this algorithm is easy to get stuck at the local optimum, Sun et al. integrated the distance factor and jump strategy into artificial potential field, avoiding the local minimum [18]. Considering that the UAV moves very quickly and cannot fly to a specific location and stop moving, Woods and La put forward an extended potential field controller, utilizing the relative velocity rather than position between the UAV and a target or obstacles [19].

Artificial neural network is inspired by the biological neural network and can be considered as an information processing system composed of interconnected neurons, having learning and reasoning ability [20–23]. One of the challenges for online path planning is that only limited information is detected in the real-time environment, Liu et al. designed a residual convolutional neural network-based method to predict the optimal path of similar scenarios, by learning the optimal paths in the offline scenarios [24]. For tracking errors generated by ignoring the UAV trajectory tracking controller, Liu et al. put forward a trajectory mapping network-based control oriented highly feasible path planning method [25]. Different from the artificial neural network with learning ability, fuzzy logic mainly takes expert experience and knowledge as the core of reasoning and decision making. Adjusting the parameters of member functions and rules in fuzzy logic needs lots of trial and error, Sathyan et al. designed a genetic fuzzy clustering for path planning, adopting the genetic algorithm to rectify the parameters in fuzzy logic [26]. To determine the parameters of differential evolution in path planning, Adhikari et al. put forward a fuzzy adaptive differential evolution, transforming the path planning problem into a multiobjective optimization and applying fuzzy logic to optimize the parameters of differential evolution [27]. Evolutionary algorithm, with the guidance of the principles of evolution and natural genetics, is a kind of random search and optimization technology and successfully solves various engineering problems [28–31]. For path planning in a disaster situation, a novel dynamic group-based cooperative optimization algorithm for UAV path planning is proposed by Qadir et al, considering the distance between the start point, control point, and end point [32]. To get over the disadvantages of existing algorithms, such as falling into local optimal solutions and slow convergence speed, Yu et al. invented a novel hybrid particle swarm optimization for UAV path planning, combining simulated annealing algorithm to improve algorithm performance [33].

There are many mature researches on two-dimensional (2D) path planning [34, 35], and some simple three-dimensional (3D) path planning problems can indeed be dealt with 2D path planning algorithm, overlooking some important information, e.g., flight altitude, pitch angle, and height of obstacle, However, in complex 3D space, more constraints and spatial information should be considered to ensure the optimality and feasibility of the path. For a complex optimization problem, these heuristic algorithms, e.g., moth flame optimization algorithm, are prone to fall into local minima and have the problem of slow convergence rates. Although gravitational search algorithm has some advantages over other heuristic algorithms, the standard gravitational search algorithm still has a few issues, e.g., no

effective acceleration mechanism and poor local search ability [36, 37]. Therefore, a 3D path planning method based on enhanced gravitational search algorithm (EGSA) is put forward in this paper. The path length, yaw angle, pitch angle, and flight height are taken into account in path planning, and the memory and random disturbance are adopted to enhance the balance ability of exploration and exploitation for EGSA. The main contributions and innovations are as follows:

1) EGSA is proposed. Each particle in the population has the memory ability and random disturbance ability to improve the performance of the algorithm, and the performance is illustrated by comparing EGSA with seven peer algorithms on CEC 2020 benchmark functions. Moreover, the effect of acceleration, memory ability of particle, and random disturbance are also carefully analyzed.

2) A path planning method is put forward, considering the path length, yaw angle, pitch angle, and flight altitude for the UAV. The performance of the method is tested and analyzed in simple and complex environments.

3) The effect of population size on path planning is explained, determining the population size by the path planning results and computing resource.

The structure of this paper is organized as follows: The preliminaries, including gravitational search algorithm (GSA) and chaotic tent map, are introduced in Section 2. Section 3 explains the proposed algorithm EGSA for 3D path planning in detail. Subsequently, the experimental settings and results are given in Section 4. Finally, the conclusions and future works are elaborated in the last section.

2. Preliminaries

In this section, the description of the gravitation search algorithm and chaotic tent map is introduced, respectively.

2.1. Gravitational search algorithm

Gravitational search algorithm, proposed by Rashedi et al. in 2009, is inspired by the law of gravity and motion [38, 39]. The solutions are viewed as particles, interacting with each other and moving in D dimensional search space, whose performance is estimated by the mass. The better the particle, the greater mass. Under the action of universal gravitation, the particles will move toward the particle with a large mass, approaching the optimal solution. The flowchart of GSA is presented in Fig. 2.

Assume that there are N particles in D dimensional space, whose positions and velocities are described as Eqs. (1) and (2).

$$X_i(t) = \left(x_i^1(t), x_i^2(t), \ldots, x_i^d(t), \ldots, x_i^D(t)\right), i = 1, 2, \ldots, N \tag{1}$$

$$V_i(t) = \left(v_i^1(t), v_i^2(t), \ldots, v_i^d(t), \ldots, v_i^D(t)\right), i = 1, 2, \ldots, N \tag{2}$$

The mass $M_i(t)$ of particle $X_i(t)$ is related to its fitness, for a minimization problem, mass can be calculated by Eqs. (3)-(6):

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{3}$$

$$m_i(t) = \frac{fit_i(t) - fit_{worst}(t)}{fit_{best}(t) - fit_{worst}(t)} \tag{4}$$

$$fit_{best}(t) = \min_{i \in \{1,\ldots,N\}} fit_i(t) \tag{5}$$

$$fit_{worst}(t) = \max_{i \in \{1,\ldots,N\}} fit_i(t) \tag{6}$$

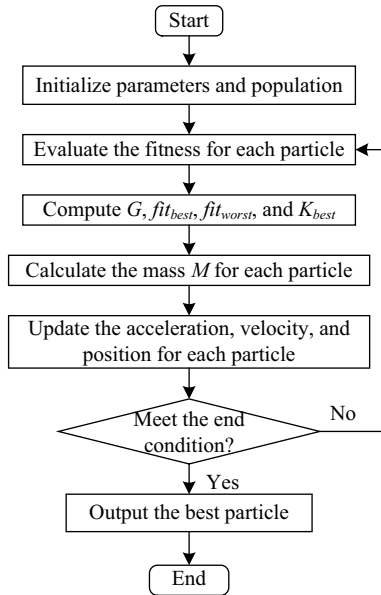where t stands for the number of iterations and $fit_i(t)$ represents the fitness of the particle $X_i(t)$.

Figure 2. *The flowchart of GSA.*

As the role of gravity, at each generation t, the force $F_{ij}(t)$ between particle $X_i(t)$ and $X_j(t)$ can be defined as follows:

$$F_{ij}^d(t) = G(t) \frac{M_i(t) \cdot M_j(t)}{R_{ij}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t) \right) \tag{7}$$

where $R_{ij}(t)$ denotes the Euclidean distance between the particle $X_i(t)$ and $X_j(t)$; ε is a constant, preventing the dominator from being 0. $G(t)$ represents the gravitational constant, which can adjust the gravity and be described as Eq. (8):

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \tag{8}$$

where G_0 refers to the initial gravitational constant, α is the adaptive adjustment coefficient, and T represents the maximum iterations.

To improve the performance of the algorithm, the total force acting on the particle $X_i(t)$ from the top K best particles is defined as follows:

$$F_i^d(t) = \sum_{j \in K_{best}, j \neq i} rand_j F_{ij}^d(t) \tag{9}$$

where K_{best} is the set storing the top K best particles, decreasing linearly over iterations; $rand_j$ stands for a random value between 0 and 1.

According to the law of motion, the acceleration a of the particle $X_i(t)$ can be achieved by Eq. (10), using its mass and resultant force.

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{10}$$

Finally, the position $x_i^d(t)$ and velocity $v_i^d(t)$ of the ith particle can be updated by Eqs. (11) and (12), respectively.

$$v_i^d(t+1) = rand_i \cdot v_i^d(t) + a_i^d(t) \tag{11}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{12}$$

where $rand_i$ is a random value within the interval [0, 1].

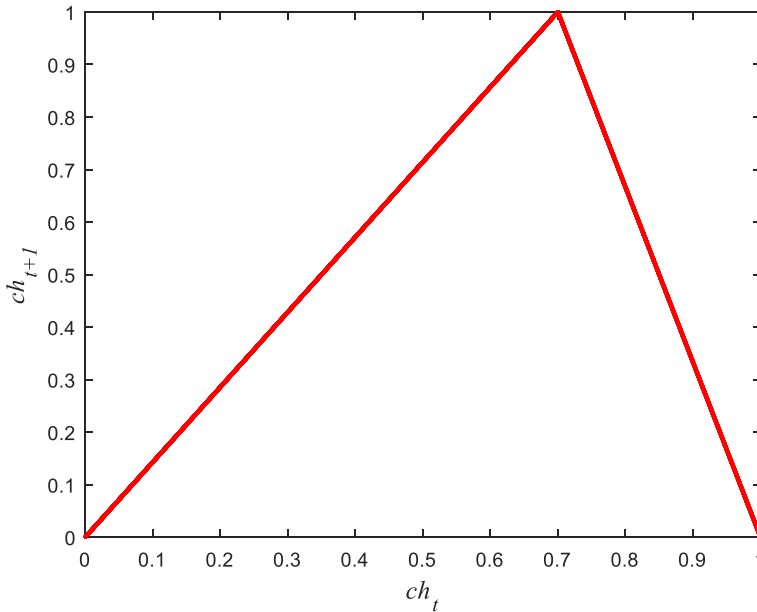**Figure 3.** *The tent map, in a case at $a = 0.7$.*

2.2. Tent map

The tent map is one of the noninvertible and piecewise linear discrete maps [40], which is given by Eq. (13).

$$ch_{t+1} = f(ch_t) = \begin{cases} \dfrac{ch_t}{a}, 0 \le ch_t \le a \\ \dfrac{1 - ch_t}{1 - a}, a < ch_t \le 1 \end{cases} \tag{13}$$

where a is referred to as the control parameter, and ch_t represents the output chaotic sequence. Figure 3 characters the orbit of the tent map during the iteration, and the maximal value is obtained at $ch_t = 0.7$.

3. Proposed method

This section first gives the environment model and defines the overall cost function. Then the proposed path planning algorithm is described. Finally, the computational complexity of EGSA is analyzed.

3.1. Modeling the environment

Suppose, S and T are the start and terminal points of the task for a UAV, whose corresponding coordinates are (x_1, y_1, z_1) and (x_D, y_D, z_D), respectively. The coordinates of S and T in 3D space are the main objects to be considered, just as shown in Fig. 4, ignoring the obstacles in the environment. $D - 1$ equal parts are got by evenly dividing $[x_1, x_D]$ along the X-axis, the vertical planes $(\beta_1, \beta_2, \ldots, \beta_d, \ldots, \beta_D)$ that are perpendicular to X-axis are attained according to the points $(x_1, x_2, \ldots, x_d, \ldots, x_D)$ in the X-axis. The point $P_d = (x_d, y_d, z_d)$ is taken from the plane β_d. A path in three-dimensional space can be achieved by connecting these points $(P_1, \ldots, P_d, \ldots, P_D)$ in turn, which can be described as $P = \{S, (x_2, y_2, z_2), \ldots, (x_d, y_d, z_d), \ldots, (x_{D-1}, y_{D-1}, z_{D-1}), T\}$. Generally, the division method in X-axis can also be revised to fulfill our needs.
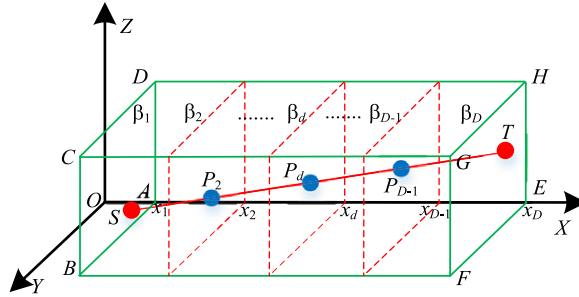
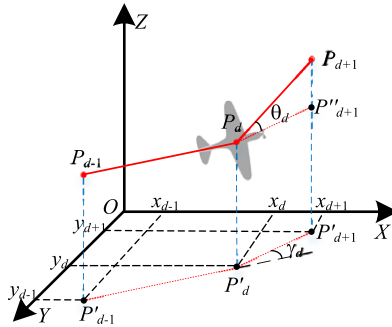**Figure 4.** *The three-dimensional environmental model.*

Figure 5. *Performance constraints of UAV.*

The optimal path problem can be transformed into optimizing a coordinate sequence to satisfy constraints and minimize the objective function in 3D space. Since some coordinates are set in advance and too large variable dimension would magnify the computational complexity, the path variable P can be expressed as $P = (y_2, z_2, y_3, z_3, \ldots, y_d, z_d, \ldots, y_{D-1}, z_{D-1})$, reducing $D - 2$ variables.

3.2. Cost function

When performing a task, time and energy consumption are two important factors to be considered, which are bound up with the path length. The greater the path length, the more time and energy required, and vice versa. So, the path length is a very critical factor in the optimal path, which can be described as follows:

$$F_1 = \sum_{d=1}^{D-1} F_1^d \tag{14}$$

$$F_1^d = \sqrt{(x_d - x_{d+1})^2 + (y_d - y_{d+1})^2 + (z_d - z_{d+1})^2} \tag{15}$$

where F_1^d denotes the Euclidean distance between the node d and its next node $d + 1$, (x_d, y_d, z_d) and $(x_{d+1}, y_{d+1}, z_{d+1})$ are the coordinates of node d and its next node $d + 1$, respectively.

Due to the performance constraints of UAV, the UAV will lose control when the yaw angle or pitch angle is too large. Therefore, the smoothness of the path needs to be taken into account, just as shown in Fig. 5, the yaw angle and pitch angle are two primary factors to affect the smoothness and feasibility of the path. P_{d-1}, P_d, and P_{d+1} are three adjoining points, whose corresponding projection points are P'_{d-1}, P'_d, and P'_{d+1} in the plane OXY, respectively, and the projection of point P_d on the line $P_{d+1}P'_{d+1}$ is P''_{d+1}. The yaw angle γ_d, indicating the turning of the UAV in the horizontal plane, denotes the angle between

$\overrightarrow{P'_{d-1}P'_d}$ and $\overrightarrow{P'_dP'_{d+1}}$, which can be calculated by Eqs. (16) and (17). So the turning cost function could be computed by Eq. (19).

$$\gamma_d = \arccos\left(\frac{\overrightarrow{P'_{d-1}P'_d} \cdot \overrightarrow{P'_dP'_{d+1}}}{\left\|\overrightarrow{P'_{d-1}P'_d}\right\| \cdot \left\|\overrightarrow{P'_dP'_{d+1}}\right\|}\right) \tag{16}$$

$$\gamma_d = \arccos\left(\frac{(x_d - x_{d-1}, y_d - y_{d-1}) \cdot (x_{d+1} - x_d, y_{d+1} - y_d)^T}{\|x_d - x_{d-1}, y_d - y_{d-1}\| \cdot \|x_{d+1} - x_d, y_{d+1} - y_d\|}\right) \tag{17}$$

$$F_2^d = \begin{cases} 0, & \text{if } \gamma_d \leq \gamma_{max} \\ k_t \cdot \gamma_d, & \text{otherwise} \end{cases} \tag{18}$$

$$F_2 = \sum_{d=2}^{D-1} F_2^d \tag{19}$$

where $\|\cdot\|$ is the Euclidean distance, k_t and γ_{max} are the coefficient and threshold of yaw angle, respectively, and $(x_{d-1}, y_{d-1}, z_{d-1})$ denotes the coordinates of the point P_{d-1}.

The pitch angle, denoting the climbing angle in the vertical plane, represents the angle between the projection $\overrightarrow{P_dP''_{d+1}}$ of $\overrightarrow{P_dP_{d+1}}$ on the horizontal plane and $\overrightarrow{P_dP_{d+1}}$, which can be calculated using Eqs. (20) and (21). The corresponding climbing cost function could be described by Eq. (23).

$$\theta_d = \arctan\left(\frac{|z_{d+1} - z_d|}{\left\|\overrightarrow{P'_dP''_{d+1}}\right\|}\right) \tag{20}$$

$$\theta_d = \arctan\left(\frac{|z_{d+1} - z_d|}{\|(x_d - x_{d+1}, y_d - y_{d+1})\|}\right) \tag{21}$$

$$F_3^d = \begin{cases} 0, & \text{if } \theta_d \leq \theta_{max} \\ k_c \cdot \theta_d, & \text{otherwise} \end{cases} \tag{22}$$

$$F_3 = \sum_{d=1}^{D-1} F_3^d \tag{23}$$

where $|\cdot|$ is the absolute value operation, k_c and θ_{max} stand for the coefficient and threshold of pitch angle, respectively.

A UAV cannot touch buildings or break through the maximum altitude of flight, and the higher the flight altitude, the greater the possibility of being monitored by radar. Considering the constraints of geography, radar, and UAV performance, it is significant for a UAV to fly only within the feasible altitude range. Assume that h_{min} and h_{max} are the maximum and minimum feasible flight altitudes, respectively. Then the height cost function can be described by Eq. (25):

$$F_4^d = \begin{cases} 0, & \text{if } h'_{min} \leq z_d \leq h_{max} \\ k_h \cdot max(z_d - h_{max}, h'_{min} - z_d), & \text{otherwise} \end{cases} \tag{24}$$

$$h'_{min} = h_{min} + h(x_i, y_i)$$

$$F_4 = \sum_{d=1}^{D} F_4^d \tag{25}$$

where z_d is the current flight altitude or the coordinate of UAV in the Z-axis, and k_h denotes the flight height coefficients of the UAV.

In conclusion, the overall cost function can be represented by Eq. (26), considering the path length, UAV performance, and environmental factors for the path P.

$$F(P) = \sum_{t=1}^{4} F_t(P) \tag{26}$$

3.3. Enhanced gravitational search algorithm for UAV path planning

As we know, exploration denotes the process of accessing the entire search space, while exploitation is the process that visiting the neighborhood of the most promising points, and these two processes have a great influence on the population-based algorithm with evolutionary behavior [41–43]. The particles in particle swarm optimization, having memory function, can employ the social information to move toward the current optimal particle, which can promote optimization. Since the current optimal solution is uncertain to be the global optimal solution, perturbation is necessary, aiding the particle to jump out of the local optimal solution. Therefore, an EGSA is proposed, where each particle has the memory ability and random disturbance ability to improve the performance of the algorithm.

Levy flight is a non Gaussian random process, whose essence is a random walk derived from Levy stable distribution. The step length is vital for Levy flight, which determines whether a particle can jump out of the local solution. Since the proper step length is intractable to set, chaos is a good idea, considering randomness, ergodicity, and sensitivity to the initial value for chaos. Therefore, the combination of chaos and Levy flight is applied as a random disturbance, boosting the balance ability of exploration and exploitation. The velocity Eq. (11) can be rewritten as Eq. (27).

$$v_i^d(t+1) = rand_i \cdot v_i^d(t) + c_1(t) \cdot a_i^d(t) + c_2(t) \cdot \left(gbest^d - x_i^d(t)\right) + c_3(t) \cdot ch_t \cdot Levy(\beta) \tag{27}$$

$$c_1(t) = -5 \cdot \left(\frac{t}{T}\right)^3 + 5 \tag{28}$$

$$c_2(t) = 2.6 \cdot \left(\frac{t}{T}\right)^2 \tag{29}$$

$$c_3(t) = 2.58 \cdot \left(\frac{t}{T}\right)^{10} \tag{30}$$

$$Levy(\beta) = \frac{\mu}{|\nu|^{1/\beta}} \tag{31}$$

where $gbest^d$ denotes the dth dimension of the current optimal solution; c_1, c_2, and c_3 are named as the coefficients of acceleration, memory, and disturbance, respectively; μ and ν follow the normal distribution as follows:

$$\mu \sim N(0, 1)$$

$$\nu \sim N\left(0, \sigma_\varsigma^2\right)$$

$$\sigma_\varsigma = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\beta 2^{(\beta-1)/2}}\right)^{1/\beta} \tag{32}$$

where Γ is referred to as the standard gamma function.

In summary, the pseudocode of our proposed EGSA for UAV path planning in 3D space is shown in Algorithm 1.

Algorithm 1. EGSA for UAV path planning

Input: dimension D, maximum number of iterations T, population size N, gravitational constant G_0 and α, start points (x_1, y_1, z_1), terminal point (x_D, y_D, z_D)

Output: the best particle or path

1: % Initialization
2: Generate the population X and velocity V;
3: Calculate the overall cost or fitness of each particle using Eq. (26);
4: Obtain the current optimal particle *gbest*;
5: % Main loop;
6: **while** $t < T$ **do**
7:     Update the gravitation constant $G(t)$ by Eq. (8);
8:     Calculate the coefficients c_1, c_2, and c_3 of acceleration, memory, and disturbance, respectively, using Eqs. (28)-(30);
9:     Get the $fit_{best}(t)$ and $fit_{worse}(t)$ by Eqs. (5) and (6), respectively;
10:     **for all** $i = 1: N$ **do**
11:       Obtain the mass $M(t)$ by Eq. (3);
12:     **end for**
13:     **for all** $i = 1: N$ **do**
14:       **for all** $d = 1: D$ **do**
15:         Compute the acceleration $a_i^d(t)$ using Eq. (10);
16:         Update the particle x_i^d and velocity v_i^d by Eqs. (12) and (27), respectively;
17:       **end for**
18:     **end for**
19:     **for all** $i = 1: N$ **do**
20:       Evaluate the fitness $fit_i(t)$ for each particle using Eq. (26);
21:     **end for**
22:     Update the current optimal particle *gbest*;
23: **end while**

3.4. Computational complexity

Here, for the convenience of complexity analysis, let D and N be the dimension of decision variables and the population size. For EGSA, the mass calculation, acceleration computing, update of velocity and position, and evaluation consume the main calculation, whose maximum computational complexity is $O(ND)$ in one loop. Therefore, the overall computational complexity of EGSA is $O(ND)$ for one main cycle.

4. Experiments and discussion

This section characterizes the experimental setting and the performance of our proposed method. First, the parameter setting is introduced. The performance of EGSA is checked by comparing it with seven peer algorithms. Next, the effects of acceleration, memory ability of particle, and random disturbance are observed. Then our proposed path planning method is compared with seven methods under diverse environments. Finally, the influence of population size on our proposed algorithm is analyzed.

4.1. Parameter setting

To verify the performance of the proposed algorithm, let EGSA compare with moth flame optimization algorithm (MFO) [44], GSA [38], chaotic kbest gravitational search algorithm (CKGSA) [45], gbest guided gravitational search algorithm (GGSA) [46], gaussian particle swarm optimization and

Table I. *Parameter settings for eight algorithms.*

Algorithms	Parameters
MFO	Constant $b = 1$.
GSA	$G_0 = 100, \alpha = 20$.
CKGSA	$G_0 = 100, \alpha = 20$, constant in logistic mapping $\mu = 4$.
GGSA	$G_0 = 100, \alpha = 20, c_1 = (-2t^3/T^3) + 2, c_2 = 2t^3/T^3$.
GPSOGSA	$G_0 = 1, \alpha = 20$.
HGSPSO	$G_0 = 100, \alpha = 20, c_1 = 2(1 - t/T) + 0.5, c_2 = -2(1 - t/T) + 2.5$.
LKGSA	$G_0 = 100, \alpha = 20$.
EGSA	$G_0 = 100, \alpha = 20, c_1 = (-5t^3/T^3) + 5, c_2 = 2.6t^2/T^2, c_3 = 2.58t^{10}/T^{10}, \beta = 1.5$, control parameter in tent mapping $a = 0.7$.

gravitational search algorithm (GPSOGSA) [47], hybrid gravitational search particle swarm optimization algorithm (HGSPSO) [48], and logarithmic kbest gravitational search algorithm (LKGSA) [49], and the parameters of these eight algorithms are listed in Table I, and the population size N and the maximum number of iterations T are 50 and 3000, respectively. The coefficients k_t, k_c, and k_h of yaw angle, pitch angle, and flight height are defined to 20, 20, and 30, respectively. The thresholds γ_{max}, θ_{max}, h_{min}, and h_{max} of yaw angle, pitch angle, minimum flight height, and maximum flight height are set to 60, 45, 5, and 200, respectively. To be fair, all the experiments are run on the computer with an Intel Core i7 2.6 GHz and Windows 10 operating system.

4.2. Experimental results on CEC 2020 benchmark functions

Here, the performance of EGSA is verified on CEC 2020 with dimension $D = 20$, compared with MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA [50]. These eight algorithms were run 30 times independently, and several indicators, including the best results, mean results, and standard deviation (Std) of the optimal values, are applied to evaluate the performance of these algorithms. Furthermore, the Wilcoxon rank sum test at a 0.05 significance level is employed between EGSA and the other seven algorithms, and the symbols '+', '−', and '∼' denote that the performance of the corresponding algorithm is significantly better, worse, and similar than that obtained by EGSA on the benchmark function, respectively [51, 52]. The experimental results on CEC 2020 obtained by these eight algorithms are recorded in Table II and Fig. 6.

From the view of the mean value in Table II, EGSA reveals a better performance than the other seven algorithms on most of these benchmark functions, winning the best place except CEC1 and CEC8. For function CEC1, the best performance is obtained by GSA, in fact, EGSA behaves similarly to GSA from the perspective of the Wilcoxon rank sum test. For function CEC8, GGSA performs best, but the performance of EGSA is not bad and ranks second among that displayed by eight algorithms, maybe the random disturbance cannot improve the performance of EGSA on CEC8. At the same time, the convergence curves in Fig. 6 verified the excellent performance of EGSA in CEC 2020 again.

As denoted by the symbols of "+/−/∼" at the bottom of Table II, EGSA performs better than MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA on 10, 8, 8, 7, 10, 10, and 7 benchmark functions and gets similar results with GSA, CKGSA, GGSA, and LKGSA on 2, 2, 2, and 3 out of the 10 test instances. As shown by the average ranking, the average ranking of EGSA is better than that of comparison algorithms, demonstrating that EGSA owns a better balance between exploitation and exploration and has superiority over the peer algorithms.

4.3. The effect of acceleration, memory ability of particle, and random disturbance

To investigate the influence of acceleration, memory ability of particle, and random disturbance, EGSA is compared with six variants on CEC 2020, meanwhile, explaining the rationality of parameter setting.

Table II. *Best values, mean values, standard deviations, and p-values of the minimum values obtained by EGSA and compared algorithms on CEC 2020 benchmark functions.*

Function	Quality	MFO	GSA	CKGSA	GGSA	GPSOGSA	HGSPSO	LKGSA	EGSA
CEC1	Best	1.453E + 02	1.001E + 02	1.087E + 02	1.012E + 02	1.274E + 10	1.463E + 10	1.001E + 02	1.094E + 02
	Mean	1.268E + 09	3.105E + 02	6.345E + 02	5.534E + 02	1.992E + 10	2.005E + 10	6.272E + 02	5.754E + 02
	Std	1.425E + 09	2.265E + 02	6.417E + 02	6.138E + 02	2.283E + 09	2.838E + 09	9.575E + 02	5.377E + 02
	p-value	1.279E-09 −	5.188E-02 ∼	6.414E-01 ∼	6.309E-01 ∼	3.020E-11 −	3.020E-11 −	5.793E-01 ∼	
CEC2	Best	1.634E + 03	2.830E + 03	3.302E + 03	2.377E + 03	4.791E + 03	4.526E + 03	3.659E + 03	1.346E + 03
	Mean	3.123E + 03	3.694E + 03	4.127E + 03	3.065E + 03	5.130E + 03	5.120E + 03	4.481E + 03	1.764E + 03
	Std	6.667E + 02	3.782E + 02	4.085E + 02	4.222E + 02	1.832E + 02	2.257E + 02	4.910E + 02	2.224E + 02
	p-value	1.957E-10 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	
CEC3	Best	7.463E + 02	7.258E + 02	7.301E + 02	7.233E + 02	1.245E + 03	1.304E + 03	7.661E + 02	7.233E + 02
	Mean	8.116E + 02	7.303E + 02	7.430E + 02	7.284E + 02	1.397E + 03	1.416E + 03	8.077E + 02	7.283E + 02
	Std	5.825E + 01	3.245E + 00	7.175E + 00	2.507E + 00	6.346E + 01	5.952E + 01	2.798E + 01	3.174E + 00
	p-value	3.020E-11 −	2.510E-02 −	9.919E-11 −	5.997E-01 ∼	3.020E-11 −	3.020E-11 −	3.020E-11 −	

Table II.   *Continued.*

Function	Quality	MFO	GSA	CKGSA	GGSA	GPSOGSA	HGSPSO	LKGSA	EGSA
CEC4	Best	1.904E + 03	1.901E + 03	1.901E + 03	1.901E + 03	2.431E + 04	1.662E + 04	1.902E + 03	1.901E + 03
	Mean	1.003E + 04	1.902E + 03	1.902E + 03	1.902E + 03	9.233E + 04	9.790E + 04	1.904E + 03	1.902E + 03
	Std	1.555E + 04	4.951E-01	6.071E-01	8.225E-01	4.376E + 04	4.422E + 04	1.365E + 00	4.485E-01
	p-value	3.020E-11 −	4.733E-01 ∼	6.567E-02 ∼	3.265E-02 −	3.020E-11 −	3.020E-11 −	4.573E-09 −	
CEC5	Best	1.946E + 04	1.250E + 05	5.835E + 04	1.088E + 05	1.086E + 06	2.831E + 05	2.202E + 04	2.145E + 04
	Mean	4.144E + 05	3.330E + 05	1.207E + 05	2.545E + 05	3.159E + 06	2.593E + 06	5.784E + 04	5.106E + 04
	Std	3.930E + 05	1.279E + 05	3.363E + 04	6.802E + 04	1.155E + 06	1.198E + 06	2.703E + 04	1.664E + 04
	p-value	6.010E-08 −	3.020E-11 −	1.464E-10 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	5.692E-01 ∼	
CEC6	Best	1.690E + 03	2.456E + 03	2.195E + 03	2.057E + 03	2.491E + 03	2.534E + 03	2.135E + 03	1.604E + 03
	Mean	2.026E + 03	2.743E + 03	2.717E + 03	2.396E + 03	2.864E + 03	2.873E + 03	2.701E + 03	1.782E + 03
	Std	1.736E + 02	1.583E + 02	2.525E + 02	2.040E + 02	1.490E + 02	1.475E + 02	2.605E + 02	9.923E + 01
	p-value	1.254E-07 −	3.020E-11 −	3.020E-11 −	3.690E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	
CEC7	Best	9.564E + 03	6.198E + 04	1.801E + 04	5.804E + 04	4.058E + 04	1.980E + 05	7.381E + 03	4.730E + 03
	Mean	2.313E + 05	1.750E + 05	4.785E + 04	1.265E + 05	6.673E + 05	6.459E + 05	2.162E + 04	9.951E + 03
	Std	2.695E + 05	7.499E + 04	2.013E + 04	4.917E + 04	2.485E + 05	2.711E + 05	1.047E + 04	4.584E + 03
	p-value	7.380E-10 −	3.020E-11 −	4.077E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.010E-07 −	

Table II. *Continued.*

Function	Quality	MFO	GSA	CKGSA	GGSA	GPSOGSA	HGSPSO	LKGSA	EGSA
CEC8	Best	$2.300E+03$	$2.300E+03$	$2.300E+03$	$2.300E+03$	$3.626E+03$	$3.535E+03$	$2.300E+03$	$2.300E+03$
	Mean	$3.098E+03$	$2.506E+03$	$3.038E+03$	$2.300E+03$	$4.637E+03$	$4.610E+03$	$3.410E+03$	$2.300E+03$
	Std	$1.113E+03$	$7.829E+02$	$1.366E+03$	$7.753E\text{-}11$	$3.950E+02$	$4.113E+02$	$1.504E+03$	$3.832E\text{-}04$
	p-value	$8.477E\text{-}09\ -$	$8.485E\text{-}09\ -$	$3.987E\text{-}04\ -$	$3.020E\text{-}11\ +$	$3.020E\text{-}11\ -$	$3.020E\text{-}11\ -$	$6.627E\text{-}01\ \sim$	
CEC9	Best	$2.838E+03$	$2.500E+03$	$2.500E+03$	$2.700E+03$	$3.062E+03$	$3.057E+03$	$3.067E+03$	$2.700E+03$
	Mean	$2.888E+03$	$2.997E+03$	$3.075E+03$	$2.804E+03$	$3.149E+03$	$3.151E+03$	$3.188E+03$	$2.759E+03$
	Std	$2.575E+01$	$1.045E+02$	$1.147E+02$	$3.923E+01$	$4.290E+01$	$3.669E+01$	$6.538E+01$	$2.052E+01$
	p-value	$3.020E\text{-}11\ -$	$5.573E\text{-}10\ -$	$5.573E\text{-}10\ -$	$9.533E\text{-}07\ -$	$3.020E\text{-}11\ -$	$3.020E\text{-}11\ -$	$3.020E\text{-}11\ -$	
CEC10	Best	$2.911E+03$	$2.906E+03$	$2.918E+03$	$2.936E+03$	$3.737E+03$	$3.794E+03$	$2.902E+03$	$2.917E+03$
	Mean	$3.029E+03$	$2.982E+03$	$2.972E+03$	$2.976E+03$	$4.527E+03$	$4.415E+03$	$2.978E+03$	$2.946E+03$
	Std	$1.662E+02$	$1.819E+01$	$1.905E+01$	$1.400E+01$	$3.077E+02$	$3.323E+02$	$2.542E+01$	$2.513E+01$
	p-value	$3.265E\text{-}02\ -$	$1.873E\text{-}07\ -$	$1.248E\text{-}04\ -$	$1.194E\text{-}06\ -$	$3.020E\text{-}11\ -$	$3.020E\text{-}11\ -$	$7.221E\text{-}06\ -$	
Average ranking		4.900	3.800	3.900	2.700	7.400	7.400	4.600	1.300
$+/-/\sim$		0/10/0	0/8/2	0/8/2	1/7/2	0/10/0	0/10/0	0/7/3	
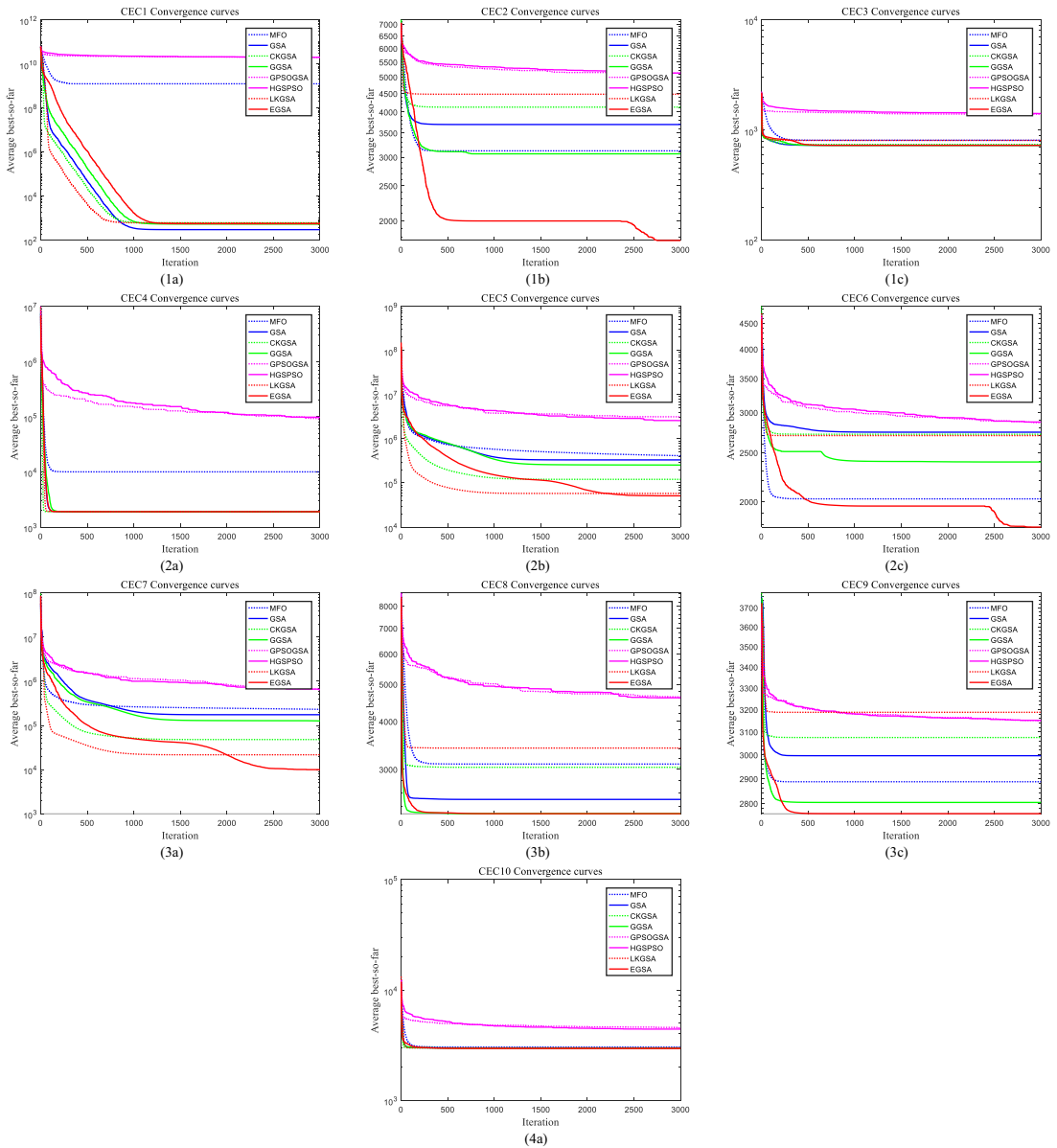
Figure 6. *The convergence curves of mean values obtained by EGSA and compared algorithms on CEC 2020 benchmark functions.*

The coefficients c_1, c_2, and c_3 are set to the minimum and maximum in the range of variation, respectively. The changed parameters and functions of six variants about EGSA are listed in Table III, and all the other parameters remain unchanged. Table IV and Fig. 7 reveal the experimental results attained by six variants and EGSA.

Our proposed algorithm EGSA achieves the top two on eight benchmark functions and wins the first place on six benchmark functions. For the functions CEC1 and CEC6, the EGSA-Variant5 and EGSA-Variant4 win the championship, respectively, however, these two algorithms do not perform well in the other five functions, which implies that the performance of EGSA cannot be promoted by over-strengthening the memory ability of particle or abandoning the random disturbance. For the algorithms EGSA-Variant2, EGSA-Variant3, and EGSA-Variant6, the performance of these three algorithms seem

Table III. *The changed parameter and function of six variants about EGSA.*

Algorithm	Changed parameter	Function
EGSA-Variant1	$c_1 = 5;$	Enlarge the impact of acceleration;
EGSA-Variant2	$c_1 = 0;$	Weaken or cancel the impact of acceleration;
EGSA-Variant3	$c_2 = 0;$	Remove the memory ability of particle;
EGSA-Variant4	$c_2 = 2.6;$	Strengthen the memory ability of particle;
EGSA-Variant5	$c_3 = 0;$	Abolish the role of random disturbance;
EGSA-Variant6	$c_3 = 2.58;$	Expand the role of random disturbance;

to be deteriorating, which implies that acceleration and memory ability of particle can evoke the performance, but large random disturbance is not desirable. For EGAS-Variant1, boosting the effect of acceleration, the performance is a little worse, which illustrates that overemphasized acceleration cannot facilitate performance.

The results of the Wilcoxon test, shown in the last row of Table IV, indicate that EGSA is superior to six variants in 1, 10, 7, 6, 6, and 8 out of the 10 test functions and performs similarly to six variants in 9, 0, 3, 3, and 2 out of the 10 test functions. The average ranking, put on the penultimate row of Table IV, explains that the performance of EGSA surpasses that of six variants, and the convergence curves in Fig. 7 declare the outstanding performance of EGSA again. Therefore, the acceleration, memory ability of particle, and random disturbance can affect the performance, which should not be too weakened or strengthened.

4.4. Experimental results in the simple environment

Assume that there is no more than one obstacle in an environment, which is named as a simple environment. The coordinates of the starting point and end point for a UAV are defined as $(1, 5, 10)$ and $(201, 190, 70)$, respectively. The paths achieved by eight algorithms are shown in Fig. 8, and the relevant convergence curves are represented in Fig. 9. Intuitively, in Fig. 8, the path got by EGSA is much better than that obtained by MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA in terms of smoothness and length. From the perspective of convergence curves, just as presented in Fig. 9, EGSA converges faster than the other seven algorithms and wins the lowest overall cost value in the end, which means that EGSA has a good balance between exploration and exploitation for this question. So EGSA can perform better than MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA in this simple environment. As a matter of fact, we discover that F_2, F_3, and F_4 in the overall cost function are all equal to 0 for the last obtained paths, which implies that the proposed method can obtain the path satisfying the three constraints and the path length becomes the main factor affecting the optimality at last.

4.5. Experimental results in the complex environment

Suppose that plenty of obstacles exist in the environment, like hills and buildings, which is considered as a complex environment. In complex environment 1, the obstacles are scattered, while in complex environments 2, 3, and 4, the obstacles are relatively concentrated. The starting point and terminal point settings are identical to those in the simple environment. For the complex environment 1, the paths and convergence curves obtained by eight algorithms are drawn in Fig. 10 and Fig. 11. The path obtained by EGSA is shorter and smoother than those obtained by the other seven algorithms. The convergence rate of EGSA is faster than those of comparison algorithms and its corresponding cost value is the smallest among the eight algorithms. For the complex environments 2, 3, and 4, Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, and Fig. 17 represent the paths and convergence curves obtained by eight algorithms, respectively. Intuitively, the paths obtained by MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA are longer than those obtained by EGSA. At the same time, from the views of

Table IV. *Best values, mean values, standard deviations, and p-values of the minimum values obtained by six variants and EGSA on CEC 2020 benchmark functions.*

Function	Quality	EGSA-Variant1	EGSA-Variant2	EGSA-Variant3	EGSA-Variant4	EGSA-Variant5	EGSA-Variant6	EGSA
CEC1	Best	$1.073E+02$	$1.832E+03$	$1.098E+02$	$3.867E+02$	$1.011E+02$	$2.043E+06$	$1.094E+02$
	Mean	$4.290E+02$	$3.911E+03$	$5.187E+02$	$2.601E+03$	$2.748E+02$	$3.119E+06$	$5.754E+02$
	Std	$5.065E+02$	$1.552E+03$	$7.533E+02$	$2.211E+03$	$2.464E+02$	$7.880E+05$	$5.377E+02$
	p-value	$1.958E\text{-}01 \sim$	$3.690E\text{-}11 -$	$1.958E\text{-}01 \sim$	$2.028E\text{-}07 -$	$8.315E\text{-}03 +$	$3.020E\text{-}11 -$	
CEC2	Best	$1.445E+03$	$1.776E+03$	$1.467E+03$	$1.278E+03$	$1.347E+03$	$1.287E+03$	$1.346E+03$
	Mean	$1.807E+03$	$2.414E+03$	$2.076E+03$	$1.698E+03$	$1.966E+03$	$1.838E+03$	$1.764E+03$
	Std	$2.198E+02$	$4.053E+02$	$3.112E+02$	$2.153E+02$	$2.705E+02$	$2.792E+02$	$2.224E+02$
	p-value	$5.298E\text{-}01 \sim$	$4.573E\text{-}09 -$	$1.248E\text{-}04 -$	$2.905E\text{-}01 \sim$	$2.891E\text{-}03 -$	$3.255E\text{-}01 \sim$	
CEC3	Best	$7.242E+02$	$7.619E+02$	$7.248E+02$	$7.334E+02$	$7.233E+02$	$7.414E+02$	$7.233E+02$
	Mean	$7.291E+02$	$8.019E+02$	$7.286E+02$	$7.540E+02$	$7.296E+02$	$7.549E+02$	$7.283E+02$
	Std	$3.475E+00$	$2.860E+01$	$2.463E+00$	$9.704E+00$	$4.750E+00$	$9.432E+00$	$3.174E+00$
	p-value	$3.042E\text{-}01 \sim$	$3.020E\text{-}11 -$	$5.201E\text{-}01 \sim$	$4.077E\text{-}11 -$	$3.112E\text{-}01 \sim$	$3.020E\text{-}11 -$	
CEC4	Best	$1.902E+03$	$1.901E+03$	$1.902E+03$	$1.901E+03$	$1.901E+03$	$1.905E+03$	$1.901E+03$

Table IV. *Continued.*

Function	Quality	EGSA-Variant1	EGSA-Variant2	EGSA-Variant3	EGSA-Variant4	EGSA-Variant5	EGSA-Variant6	EGSA
	Mean	1.903E + 03	1.907E + 03	1.903E + 03	1.902E + 03	1.903E + 03	1.906E + 03	1.902E + 03
	Std	3.846E-01	4.412E + 00	8.290E-01	8.391E-01	6.575E-01	1.202E + 00	4.485E-01
	p-value	6.528E-08 −	8.485E-09 −	1.411E-09 −	9.049E-02 ∼	4.311E-08 −	3.020E-11 −	
CEC5	Best	2.246E + 04	9.560E + 03	1.213E + 05	3.992E + 03	3.787E + 04	2.924E + 04	2.145E + 04
	Mean	5.128E + 04	7.768E + 04	2.803E + 05	1.066E + 05	1.183E + 05	8.685E + 04	5.106E + 04
	Std	1.620E + 04	3.793E + 04	1.354E + 05	1.518E + 05	6.006E + 04	3.012E + 04	1.664E + 04
	p-value	9.234E-01 ∼	2.380E-03 −	3.020E-11 −	1.260E-01 ∼	7.043E-07 −	8.292E-06 −	
CEC6	Best	1.637E + 03	1.722E + 03	2.051E + 03	1.602E + 03	1.721E + 03	1.613E + 03	1.604E + 03
	Mean	1.841E + 03	1.949E + 03	2.327E + 03	1.714E + 03	2.002E + 03	1.727E + 03	1.782E + 03
	Std	1.431E + 02	1.773E + 02	1.662E + 02	9.366E + 01	1.705E + 02	1.085E + 02	9.923E + 01
	p-value	1.224E-01 ∼	7.739E-06 −	3.338E-11 −	6.669E-03 +	6.283E-06 −	1.087E-01 ∼	
CEC7	Best	5.739E + 03	4.100E + 03	2.945E + 04	4.542E + 03	5.155E + 03	1.305E + 04	4.730E + 03
	Mean	1.458E + 04	6.106E + 04	1.287E + 05	2.707E + 04	4.100E + 04	3.075E + 04	9.951E + 03
	Std	1.648E + 04	1.221E + 05	8.546E + 04	3.069E + 04	2.664E + 04	1.480E + 04	4.584E + 03
	p-value	1.154E-01 ∼	1.041E-04 −	3.020E-11 −	1.748E-05 −	4.998E-09 −	1.547E-09 −	

Table IV. *Continued.*

Function	Quality	EGSA-Variant1	EGSA-Variant2	EGSA-Variant3	EGSA-Variant4	EGSA-Variant5	EGSA-Variant6	EGSA
CEC8	Best	2.300E + 03	2.300E + 03	2.300E + 03	2.300E + 03	2.300E + 03	2.312E + 03	2.300E + 03
	Mean	2.300E + 03	3.414E + 03	2.300E + 03	2.301E + 03	2.300E + 03	2.312E + 03	2.300E + 03
	Std	5.146E-01	1.504E + 03	4.899E-04	7.420E-01	2.431E-01	3.978E-01	3.832E-04
	p-value	7.483E-02 $\sim$	3.020E-11 $-$	8.841E-07 $-$	3.020E-11 $-$	5.573E-10 $-$	3.020E-11 $-$	
CEC9	Best	2.700E + 03	3.004E + 03	2.700E + 03	2.847E + 03	2.700E + 03	2.747E + 03	2.700E + 03
	Mean	2.764E + 03	3.143E + 03	2.761E + 03	2.916E + 03	2.754E + 03	2.775E + 03	2.759E + 03
	Std	2.150E + 01	9.821E + 01	2.144E + 01	5.669E + 01	2.675E + 01	2.016E + 01	2.052E + 01
	p-value	1.858E-01 $\sim$	3.020E-11 $-$	2.226E-01 $\sim$	3.020E-11 $-$	1.844E-01 $\sim$	1.302E-03 $-$	
CEC10	Best	2.919E + 03	2.952E + 03	2.924E + 03	2.903E + 03	2.920E + 03	2.921E + 03	2.917E + 03
	Mean	2.953E + 03	2.989E + 03	2.959E + 03	2.960E + 03	2.946E + 03	2.958E + 03	2.946E + 03
	Std	2.575E + 01	1.795E + 01	1.866E + 01	2.824E + 01	2.423E + 01	2.396E + 01	2.513E + 01
	p-value	3.790E-01 $\sim$	8.352E-08 $-$	4.515E-02 $-$	4.060E-02 $-$	9.117E-01 $\sim$	4.207E-02 $-$	
Average ranking		3.000	6.200	4.700	3.900	3.700	4.800	1.700
+/$-$/$\sim$		0/1/9	0/10/0	0/7/3	1/6/3	1/6/3	0/8/2	
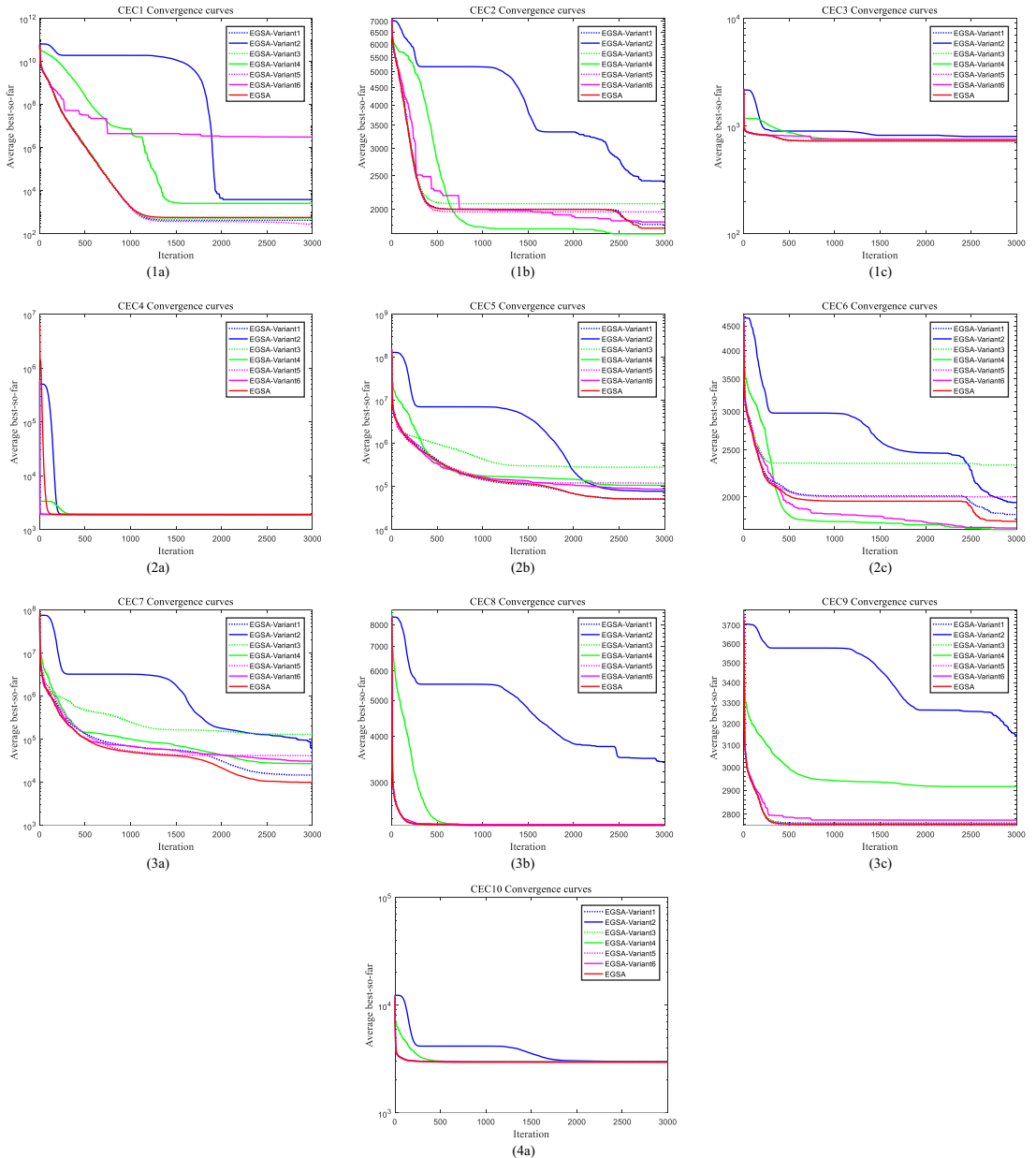
Figure 7. *The convergence curves of mean values obtained by six variants and EGSA on CEC 2020 benchmark functions.*

convergence speed and cost value, the performance of EGSA is much better than that of the other seven algorithms, and EGSA stands out as the top performer. Therefore, the performance of EGSA is the best among those eight algorithms in these four complex environments. The path length, like that in the simple environment, also becomes the primary contributor affecting the optimality for the last obtained path.

To objectively and accurately verify the performance of the proposed algorithm, the eight algorithms have been run 30 times in four kinds of environments, and Table V presents the mean, Std, and p-vlaue of the overall cost values. Obviously, the mean and Std of EGSA are much smaller than the results of
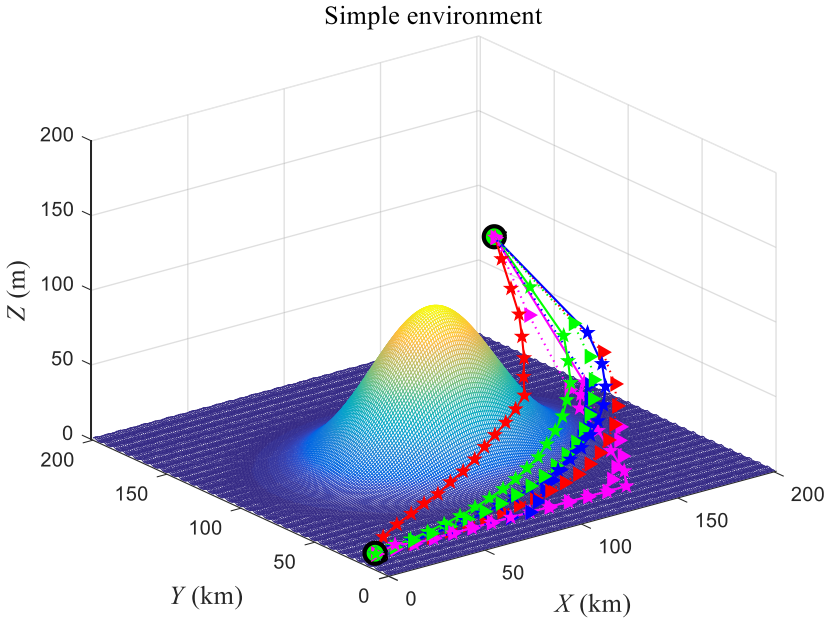
Figure 8. *Path planning results obtained by eight algorithms in the simple environment; blue triangle dotted line, blue pentagram solid line, green triangle dotted line, green pentagram solid line, purple triangle dotted line, purple pentagram solid line, red triangle dotted line, and red pentagram solid line are acquired by MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, LKGSA, and EGSA, respectively.*
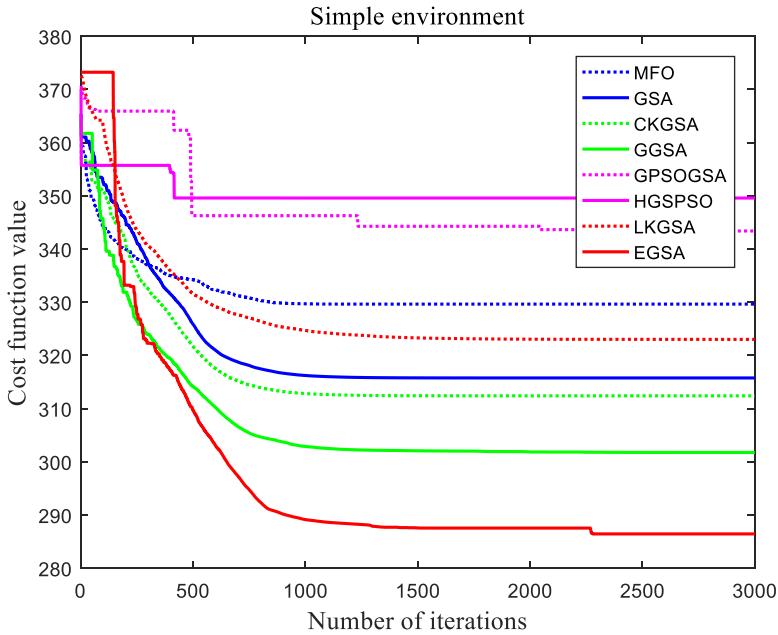
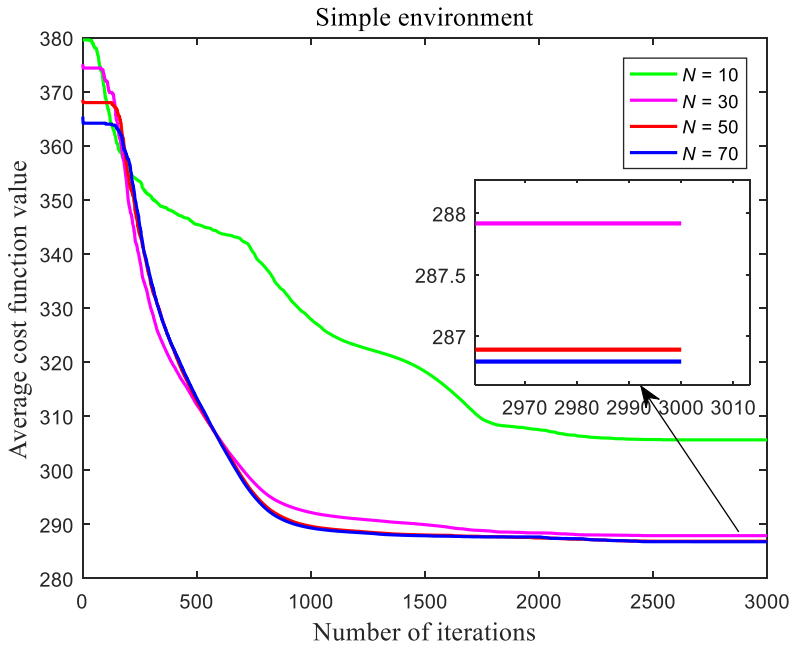**Figure 9.** *Convergence curves of eight algorithms in the simple environment.*

Complex environment 1
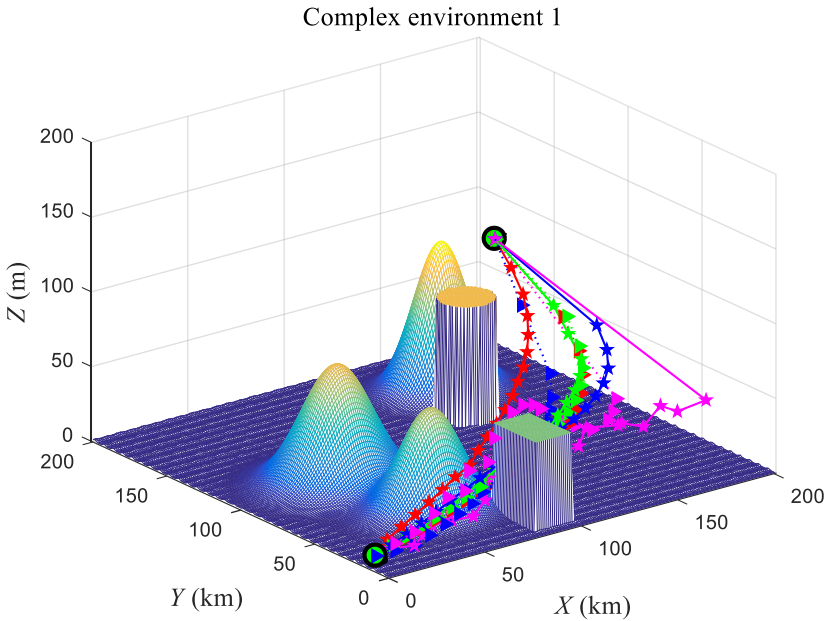
Figure 10. *Path planning results obtained by eight algorithms in the complex environment 1; blue triangle dotted line, blue pentagram solid line, green triangle dotted line, green pentagram solid line, purple triangle dotted line, purple pentagram solid line, red triangle dotted line, and red pentagram solid line are acquired by MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, LKGSA, and EGSA, respectively.*

Complex environment 1
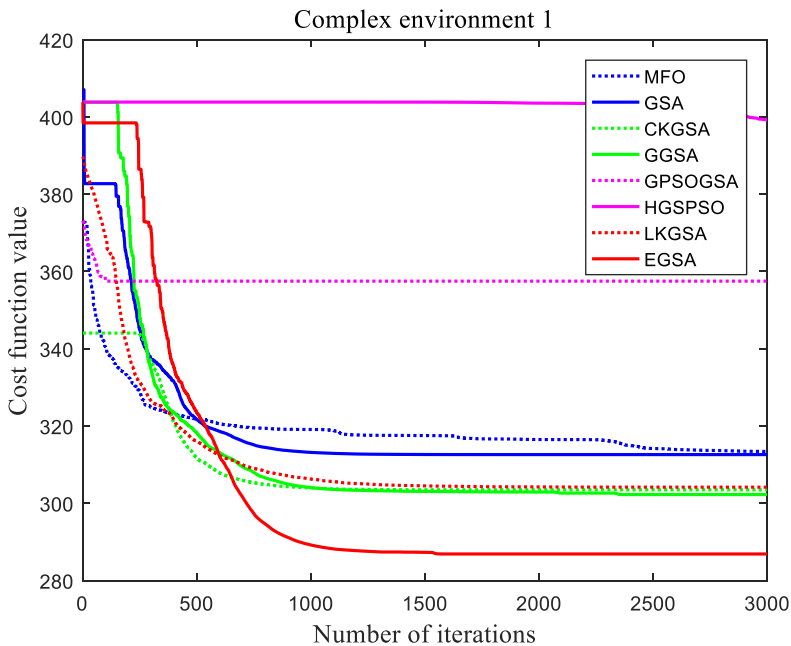
Figure 11. *Convergence curves of eight algorithms in the complex environment 1.*
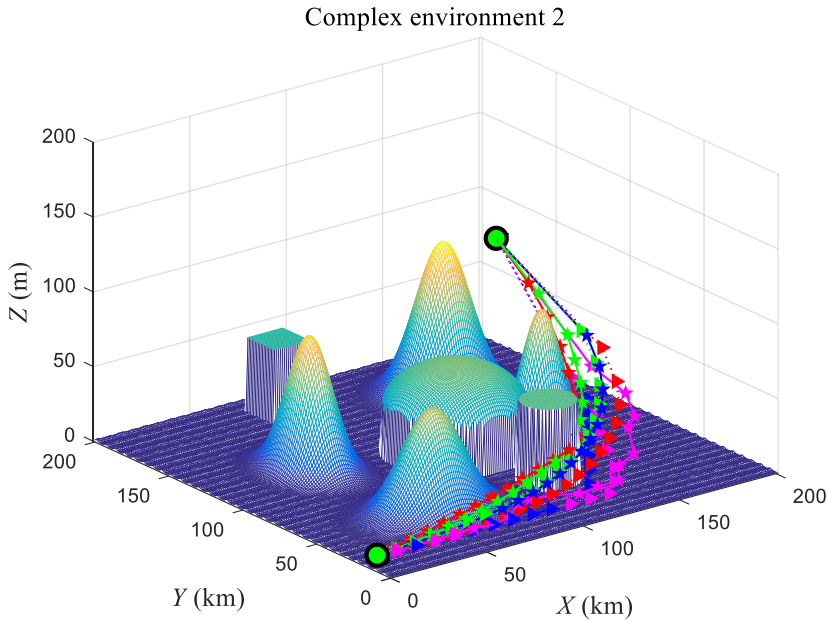
Figure 12. *Path planning results obtained by eight algorithms in the complex environment 2; blue triangle dotted line, blue pentagram solid line, green triangle dotted line, green pentagram solid line, purple triangle dotted line, purple pentagram solid line, red triangle dotted line, and red pentagram solid line are acquired by MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, LKGSA, and EGSA, respectively.*
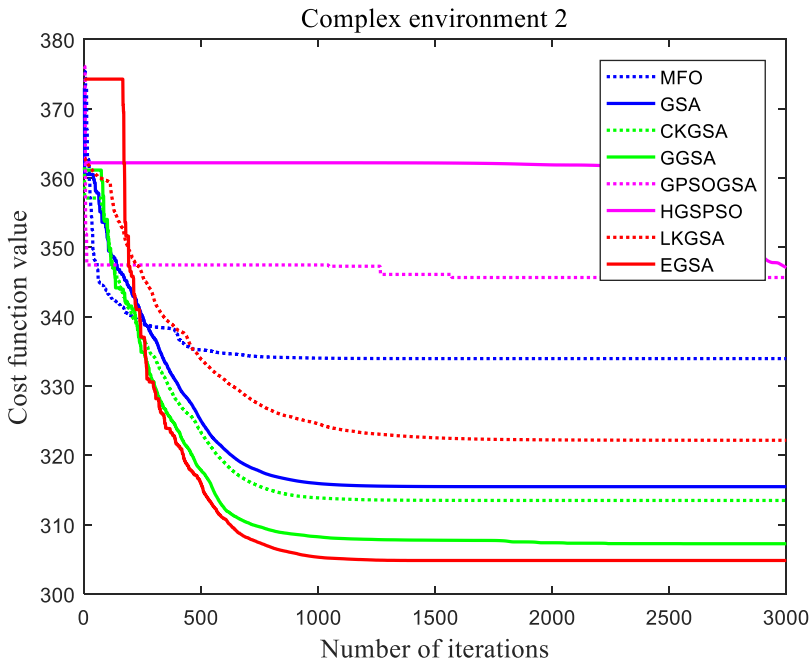
Figure 13. *Convergence curves of eight algorithms in the complex environment 2.*
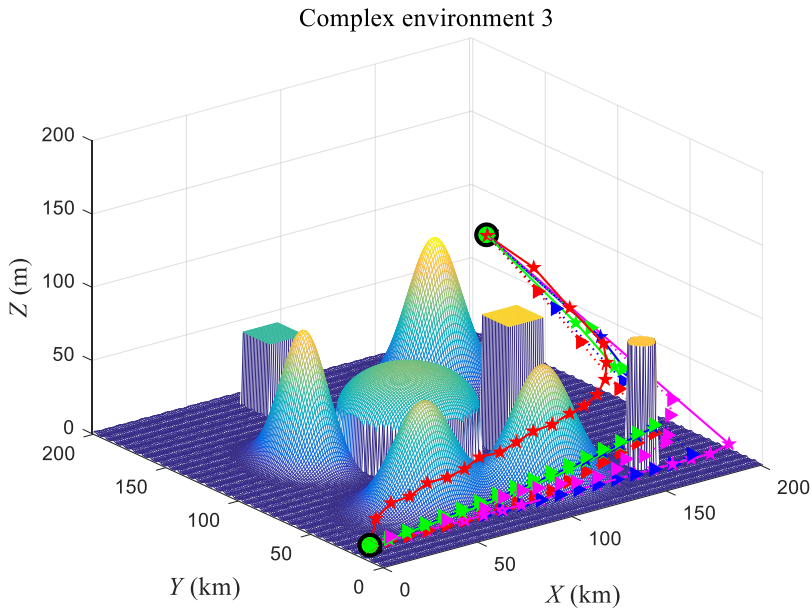
Figure 14. *Path planning results obtained by eight algorithms in the complex environment 3; blue triangle dotted line, blue pentagram solid line, green triangle dotted line, green pentagram solid line, purple triangle dotted line, purple pentagram solid line, red triangle dotted line, and red pentagram solid line are acquired by MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, LKGSA, and EGSA, respectively.*
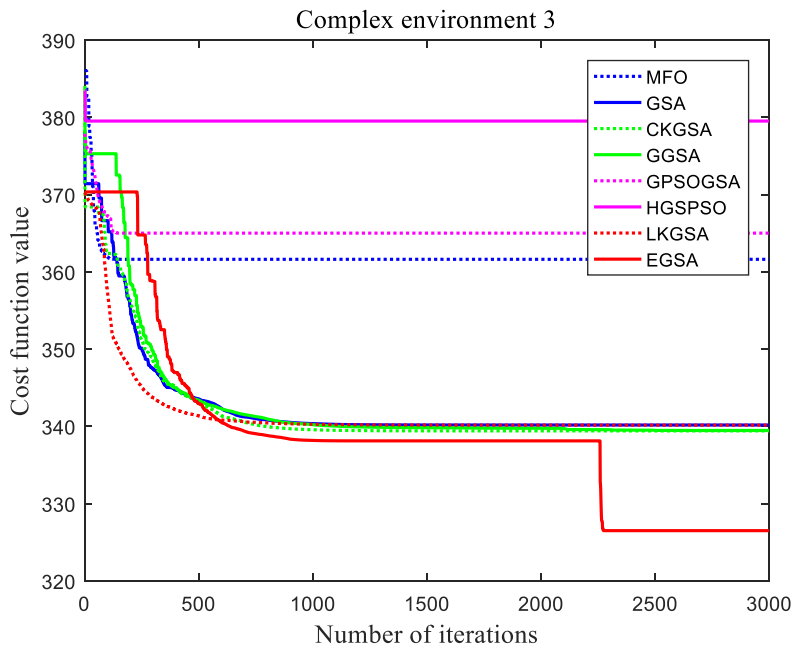
Figure 15. *Convergence curves of eight algorithms in the complex environment 3.*
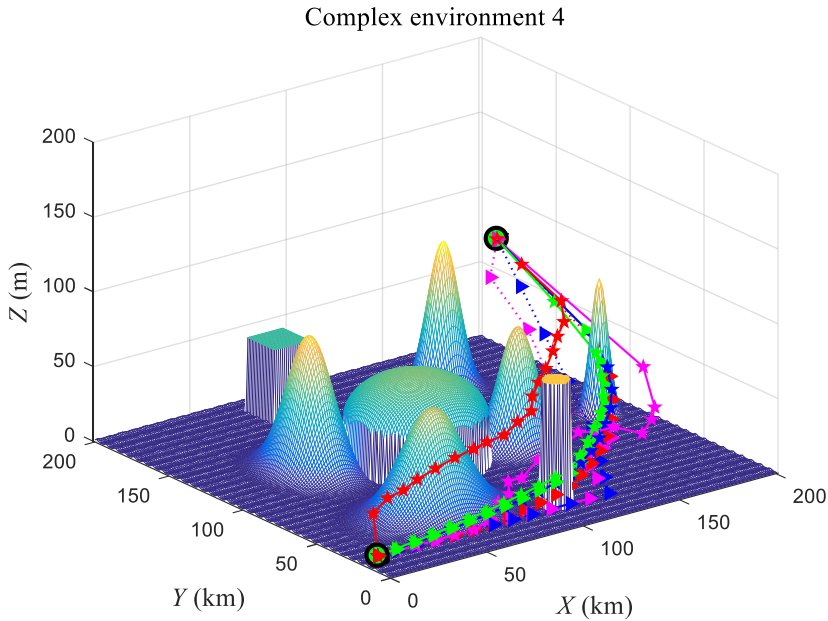
Figure 16. *Path planning results obtained by eight algorithms in the complex environment 4; blue triangle dotted line, blue pentagram solid line, green triangle dotted line, green pentagram solid line, purple triangle dotted line, purple pentagram solid line, red triangle dotted line, and red pentagram solid line are acquired by MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, LKGSA, and EGSA, respectively.*
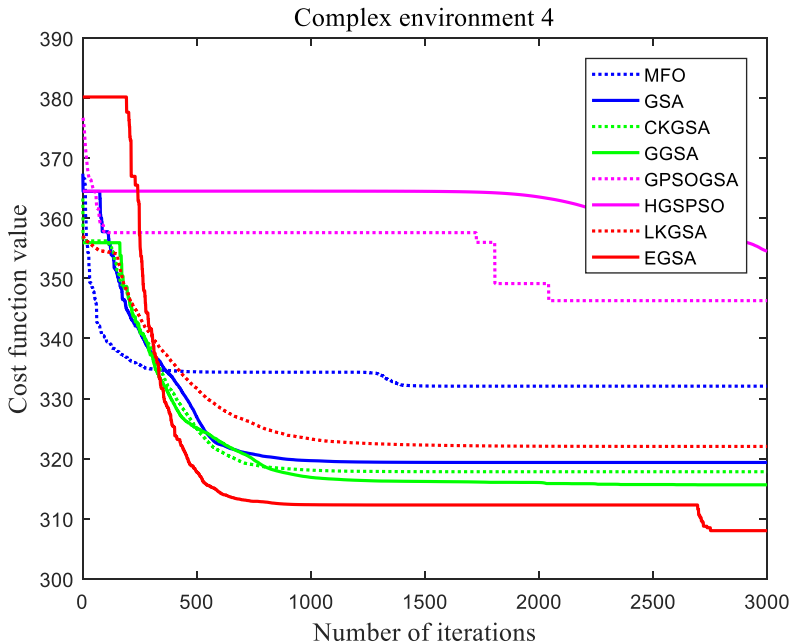
Figure 17. *Convergence curves of eight algorithms in the complex environment 4.*

Table V. *Comparison of simulation results of eight algorithms.*

		MFO	GSA	CKGSA	GGSA	GPSOGSA	HGSPSO	LKGSA	EGSA
Simple environment	Mean	317.607	314.328	313.993	304.477	353.381	357.534	329.291	286.889
	Std	12.723	1.881	2.149	2.158	9.163	9.313	11.752	1.501
	p-value	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	
Complex environment 1	Mean	316.815	312.222	313.152	301.234	372.699	381.986	318.676	292.042
	Std	27.860	8.694	8.191	6.326	16.136	33.695	18.149	8.019
	p-value	7.599E-07 −	4.183E-09 −	1.857E-09 −	6.736E-06 −	3.020E-11 −	3.020E-11 −	4.183E-09 −	
Complex environment 2	Mean	326.938	314.235	314.741	308.466	355.211	361.020	325.264	304.811
	Std	14.128	1.759	1.770	0.979	8.141	12.925	9.519	0.792
	p-value	5.570E-10 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.001E-11 −	3.020E-11 −	
Complex environment 3	Mean	343.907	340.016	340.026	338.584	359.941	369.575	339.464	328.688
	Std	13.259	0.570	0.566	0.321	8.697	9.206	1.096	4.404
	p-value	4.737E-06 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	2.916E-11 −	3.020E-11 −	
Complex environment 4	Mean	322.298	318.991	319.128	315.695	354.011	357.594	325.998	310.476
	Std	17.435	0.904	1.182	0.903	8.555	8.936	6.198	4.075
	p-value	5.800E-03 −	3.020E-11 −	3.020E-11 −	6.121E-10 −	3.020E-11 −	3.020E-11 −	3.020E-11 −	
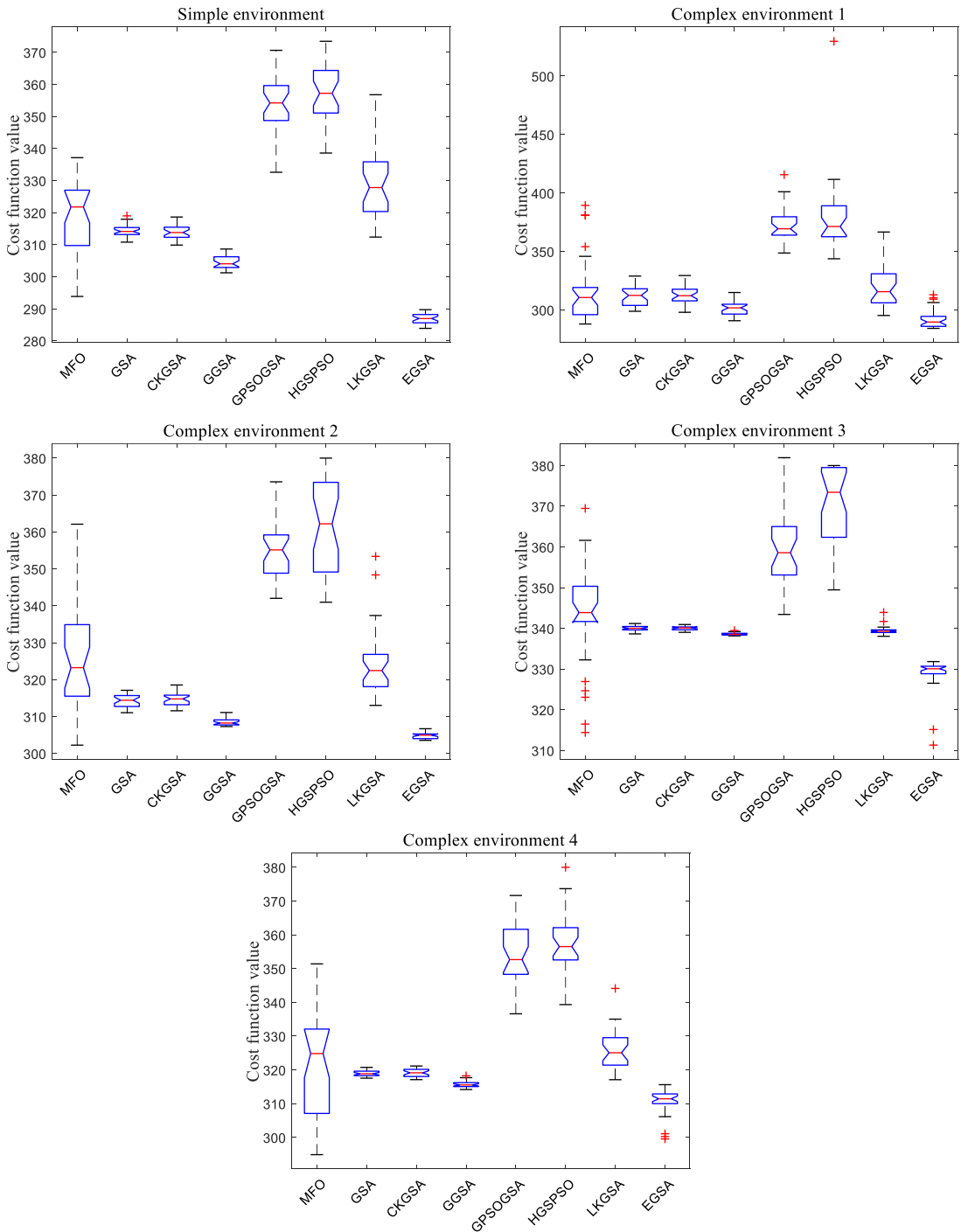
Figure 18. *Boxplots of the results obtained by eight algorithms in different environments.*

MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA. From the Wilcoxon test, our proposed EGSA has significant advantages over other comparison algorithms. Meanwhile, it can be seen that chaotic levy flight could regulate exploration and exploitation compared with GGSA. The data distribution characteristics are presented in Fig. 18. The boxplots of the proposed algorithm, with the

Figure 19. *Average convergence curves of EGSA with different population size in the simple environment.*
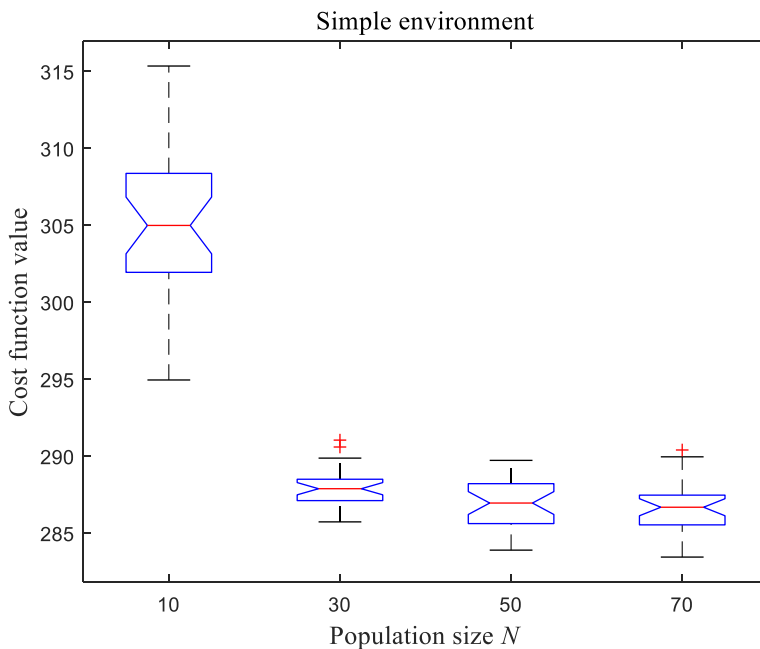
**Figure 20.** *Boxplots of the results obtained by EGSA with different population size in the simple environment.*

lowest values, are very narrow compared to the results of the other seven algorithms. Hence, the proposed EGSA has much better effectiveness and robustness than MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA for path planning in simple and complex environments.
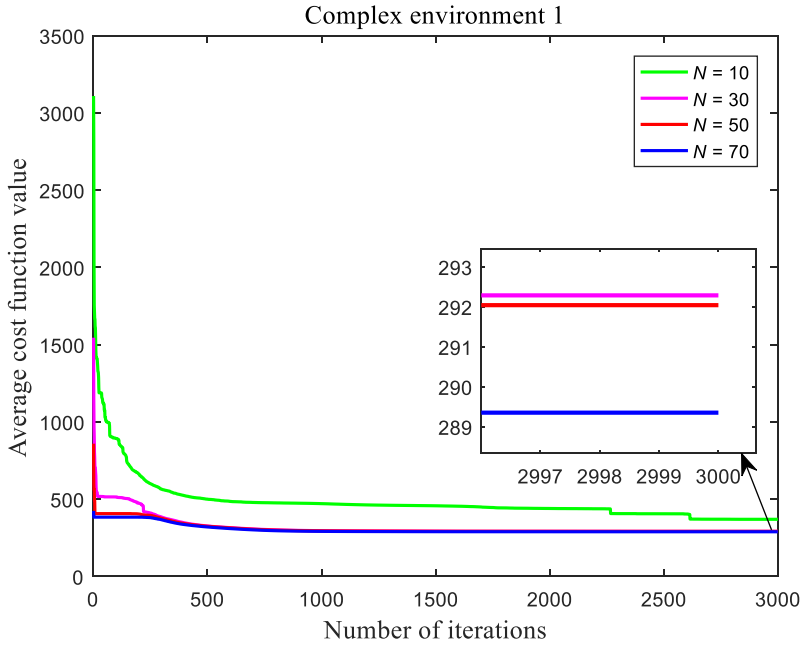
Figure 21. *Average convergence curves of EGSA with different population size in the complex environment 1.*
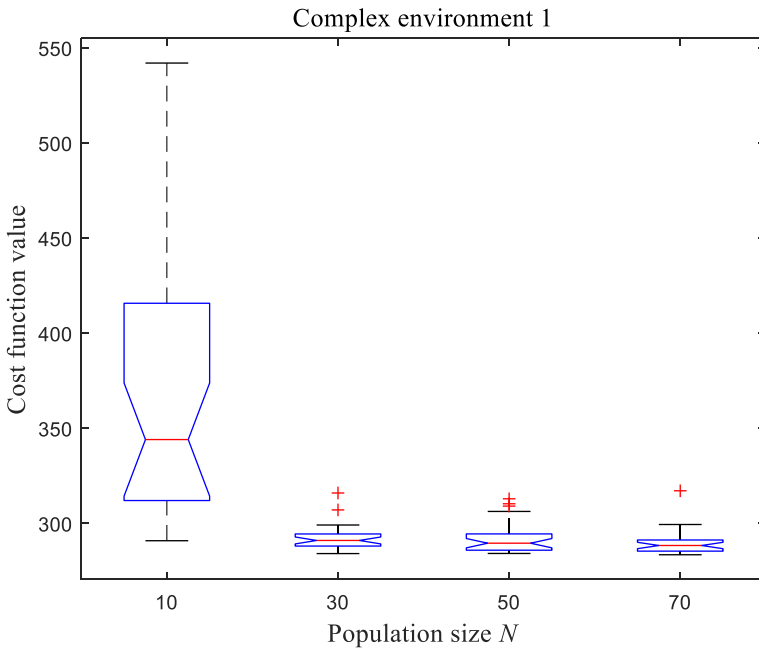
Figure 22. *Boxplots of the results obtained by EGSA with different population size in the complex environment 1.*
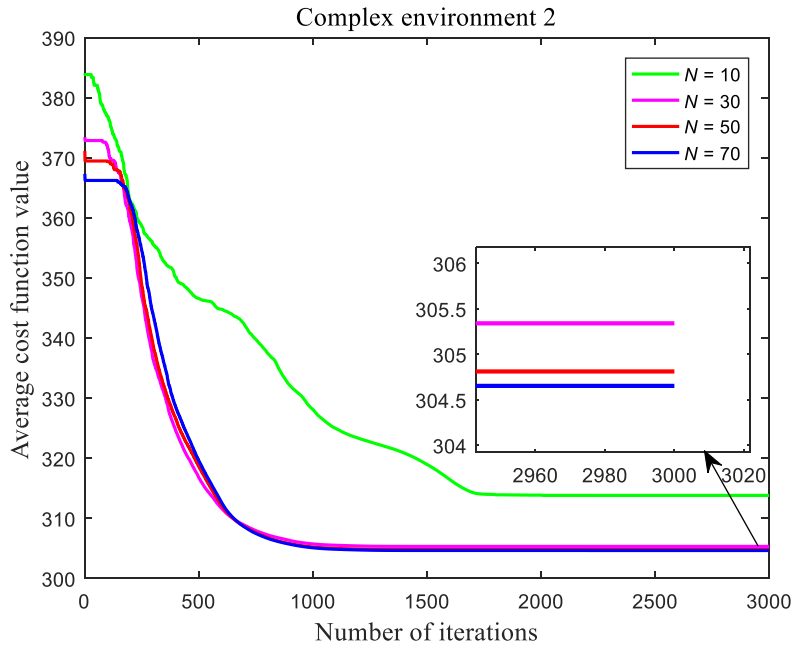
Figure 23. *Average convergence curves of EGSA with different population size in the complex environment 2.*
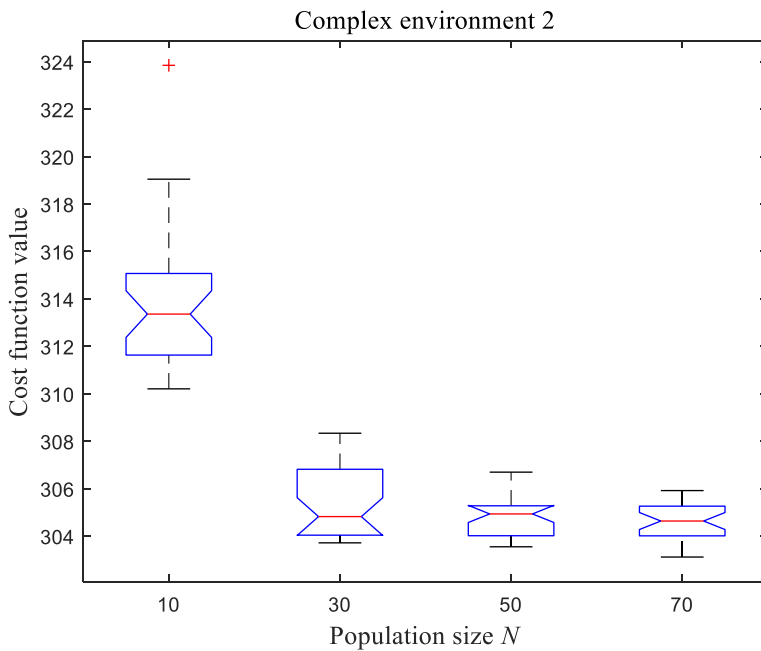
Figure 24. *Boxplots of the results obtained by EGSA with different population size in the complex environment 2.*
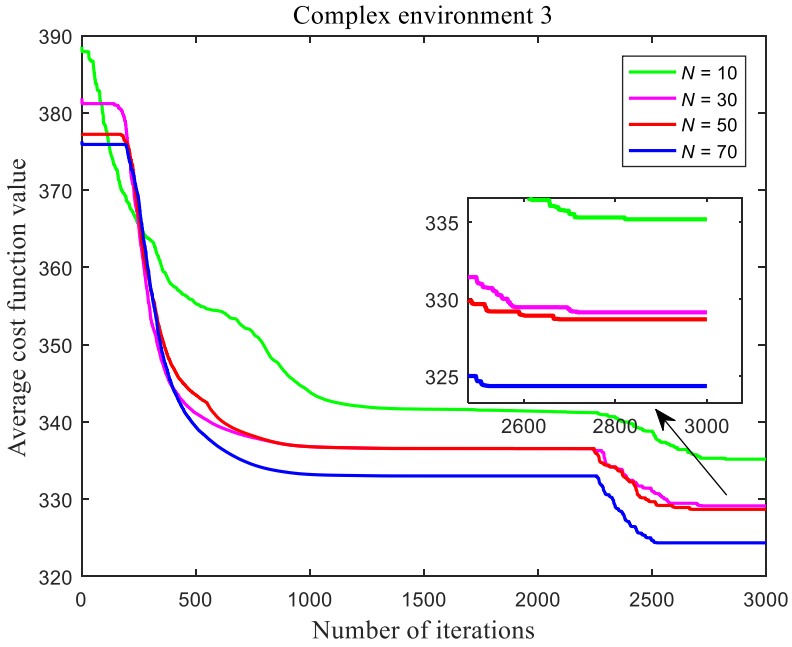
Figure 25. *Average convergence curves of EGSA with different population size in the complex environment 3.*
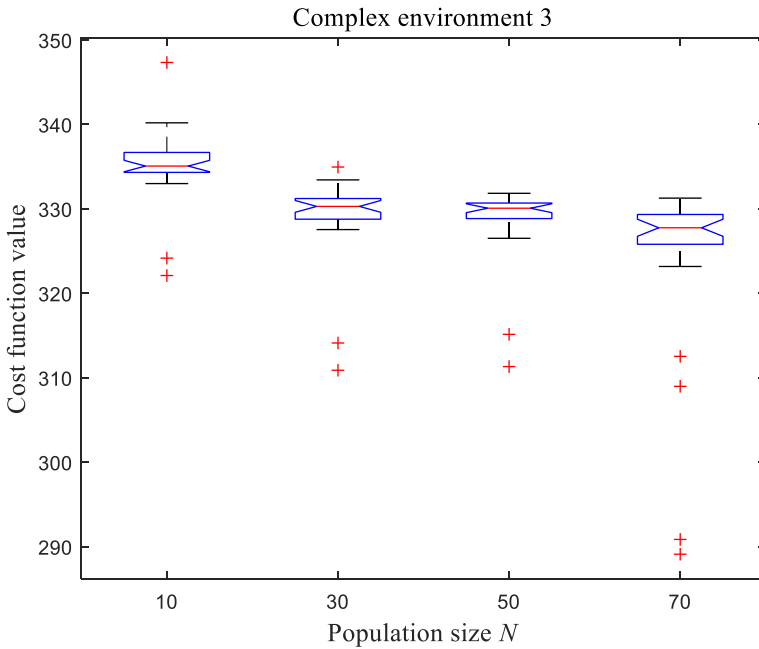
Figure 26. *Boxplots of the results obtained by EGSA with different population size in the complex environment 3.*
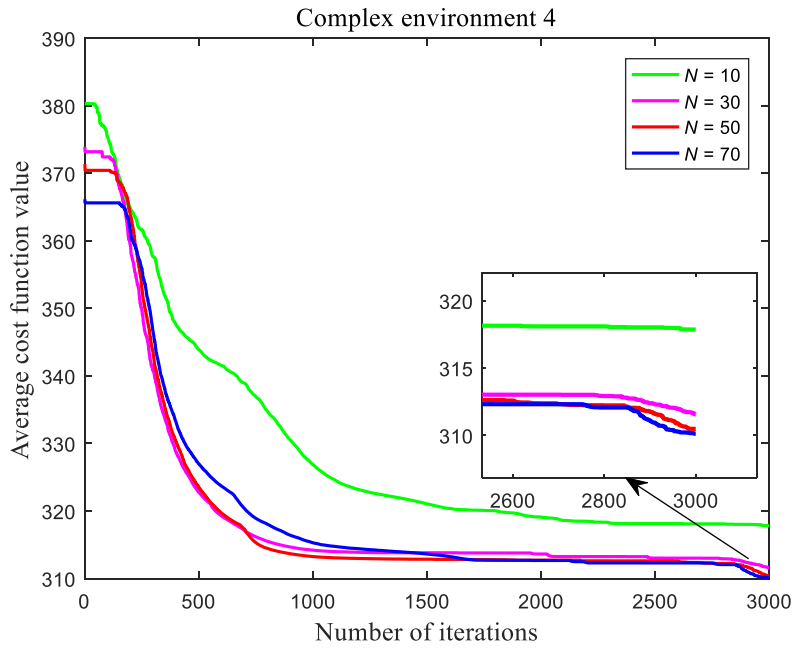
Figure 27. *Average convergence curves of EGSA with different population size in the complex environment 4.*
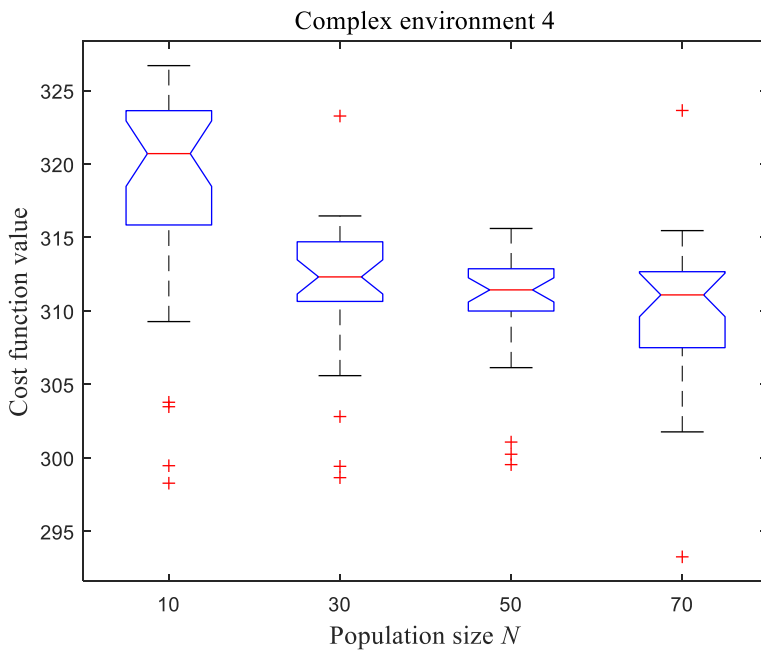
Figure 28. *Boxplots of the results obtained by EGSA with different population size in the complex environment 3.*
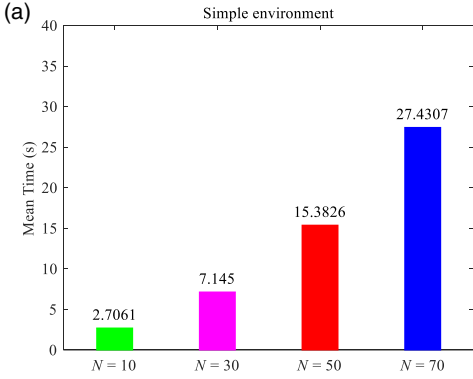
(a) The mean running time of EGSA in the simple environment.
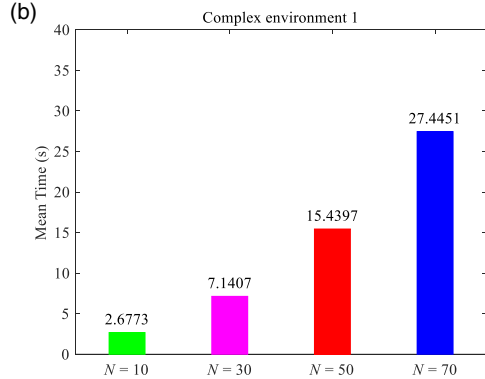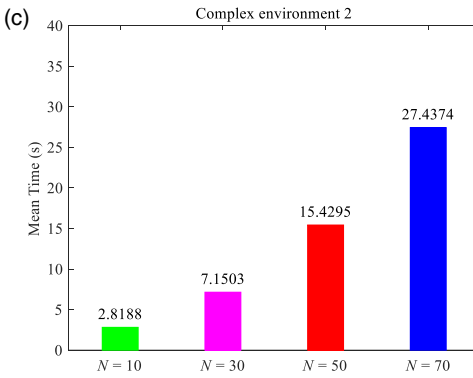
(b) The mean running time of EGSA in the complex environment 1.

(c) The mean running time of EGSA in the complex environment 2.
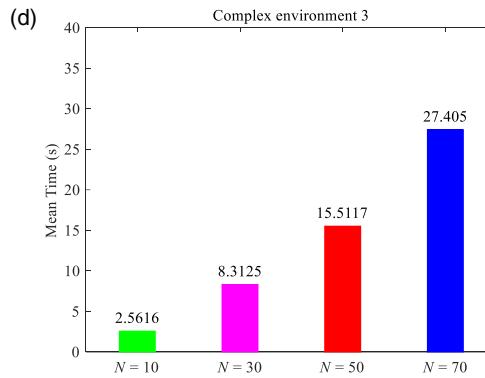
(d) The mean running time of EGSA in the complex environment 3.
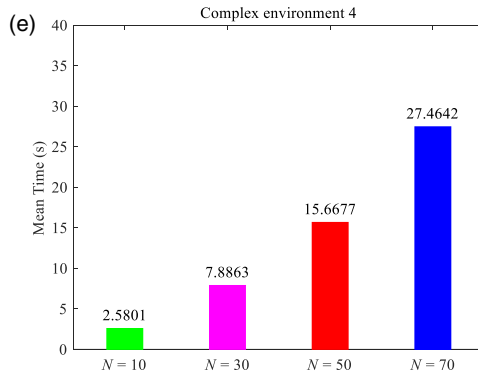
(e) The mean running time of EGSA in the complex environment 4.

Figure 29. *The mean running time of EGSA with different population size N.*

4.6. Parameter analysis

As we know, the population size N can affect the performance of algorithm. If N is too small, it will harm the convergence of the population to the optimal solution. On the contrary, too large N will consume more computing resource. Here, to observe the influence of parameter N on our proposed algorithm, all the parameters in EGSA remain the same except for the population size N, the population N is set to 10, 30, 50, and 70 in simple and complex environments, respectively. Figure 19, Fig. 21, Fig. 23, Fig. 25, and Fig. 27 show the average convergence with iteration in different environments, respectively. The

Table VI. *The simulation results of EGSA with different population size.*

		$N = 10$	$N = 30$	$N = 50$	$N = 70$
Simple environment	Mean	305.633	287.917	286.889	286.792
	Std	4.990	1.331	1.501	1.589
Complex environment 1	Mean	370.056	292.287	292.042	289.351
	Std	72.698	6.695	8.019	6.496
Complex environment 2	Mean	313.825	305.340	304.811	304.653
	Std	2.796	1.501	0.792	0.751
Complex environment 3	Mean	335.184	329.140	328.688	324.354
	Std	4.359	4.825	4.404	10.487
Complex environment 4	Mean	317.860	311.553	310.476	310.120
	Std	7.965	5.022	4.075	5.277

convergence rates are very similar when the populations are 30, 50, and 70, respectively. The boxplots with diverse population sizes for three environments are revealed in Fig. 20, Fig. 22, Fig. 24, Fig. 26, and Fig. 28, respectively. Figure 29 describes the running time for our proposed path planning method in three environments, showing that the running time increases with the population size. Table VI represents the mean and Std of overall cost values for various environments with different population sizes. It is obvious that the performance of EGSA becomes better with the increase of population size N, but in both cases of $N = 50$ and $N = 70$, the performance of EGSA is not very different, the performance improvement is very small for N increasing from 50 to 70. So, a compromise between the obtained path length and the running time is made, and 50 is set as the size of the population.

5. Conclusion

In this work, a UAV path planning algorithm in 3D space based on EGSA is presented. For EGSA, the balance between exploration and exploitation is strengthened, adding the memory and random disturbance ability to each particle in the population. The path can be achieved by connecting these nodes obtained by EGSA, considering the path length, yaw angle, pitch angle, and flight altitude. EGSA is compared with seven peer algorithms on CEC 2020 benchmark functions, such as MFO, GSA, CKGSA, GGSA, GPSOGSA, HGSPSO, and LKGSA, demonstrating that EGSA has superiority over the other algorithms. The experimental results of UAV path planning illustrate that the path planning method based on EGSA is more effective than the other seven methods. The influence of population size on our proposed algorithm is also analyzed and the appropriate population size is given.

In future research, our work will mainly focus on two interesting topics, one is the path planning of multiple UAVs, considering the collaboration and constraints between UAVs, and the other one is to use UAVs to verify the proposed algorithm in reality.

Author contribution. Keming Jiao designed the research, implemented the system, and drafted the paper. Jie Chen, Bin Xin, and Li Li helped organize the paper, revised and finalized the paper. Yifan Zheng and Zhixin Zhao collected the data.

References

[1] O. Kose, T. Oktay and E. Özen, "Simultaneous arm morphing quadcopter and autonomous flight system design," *Aircr Eng Aerosp Tec* **95**(10), 1624–1632 (2023). doi: 10.1108/AEAT-05-2023-0146.

[2] A. A. Saadi, A. Soukane, Y. Meraihi, A. B. Gabis, S. Mirjalili and A. Ramdane-Cherif, "UAV path planning using optimization approaches: A survey," *Arch Computat Methods Eng* **29**(6), 4233–4284 (2022). doi: 10.1007/s11831-022-09742-7.

[3] H. Zhang, B. Xin, L.-H. Dou, J. Chen and K. Hirota, "A review of cooperative path planning of an unmanned aerial vehicle group," *Front Inform Technol Electron Eng* **21**(12), 1671–1694 (2020). doi: 10.1631/FITEE.2000228.

[4] Y. Ding, B. Xin and J. Chen, "A review of recent advances in coordination between unmanned aerial and ground vehicles," *Unmann Syst* **09**(02), 97–117 (2020). doi: 10.1142/S2301385021500084.

[5] G. Xiao, L. Zhang, T. Wu, Y. Han, Y. Ding and C. Han, "FBi-RRT: A path planning algorithm for manipulators with heuristic node expansion," *Robotica* **42**(3), 644–659 (2024). doi: 10.1017/S0263574723001674.

[6] Y. Zhao, Z. Zheng and Y. Liu, "Survey on computational-intelligence-based UAV path planning, knowledge-based systems," *Knowl_Bas Syst* **158**, 54–64 (2018). doi: 10.1016/j.knosys.2018.05.033.

[7] Y. Ding, B. Xin, L. Dou, J. Chen and B. M. Chen, "A memetic algorithm for curvature-constrained path planning of messenger UAV in air-ground coordination," *IEEE Trans Autom Sci Eng* **19**(4), 3735–3749 (2022). doi: 10.1109/TASE.2021.3135044.

[8] W. He, X. Qi and L. Liu, "A novel hybrid particle swarm optimization for multi-UAV cooperate path planning," *Appl Intell* **51**(10), 7350–7364 (2021). doi: 10.1007/s10489-020-02082-8.

[9] Z. Liu, Y. Huang, D. Liu, X. Guo, K. Wang and J. Tan, "Trajectory planning of large redundant manipulator considering kinematic constraints and energy efficiency," *Robotica* **41**(11), 3524–3540 (2023). doi: 10.1017/S0263574723001157.

[10] T. Oktay and F. Sal, "Effect of the simultaneous variation in blade root chord length and blade taper on helicopter flight control effort," *Int J Aerospace Eng* **2017**, 1–8 (2017). doi: 10.1155/2017/6325269.

[11] L. E. Kavraki, M. N. Kolountzakis and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans Robot Autom* **14**(1), 166–171 (1998). doi: 10.1109/70.660866.

[12] T. Chen, G. Zhang, X. Hu and J. Xiao, "Unmanned Aerial Vehicle Route Planning Method Based on a Star Algorithm," **In:** *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, (2018) pp.1510–1514. doi:10.1109/ICIEA.2018.8397948.

[13] C. Zammit and E.-J. van Kampen, "Comparison between A∗ and RRT algorithms for 3D UAV path planning," *Unmann Syst* **10**(02), 129–146 (2022). doi: 10.1142/S2301385022500078.

[14] S. Shao, Y. Peng, C. He and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Transactions* **97**, 415–430 (2020). doi: 10.1016/j.isatra.2019.08.018.

[15] B. C. Shah and S. K. Gupta, "Long-distance path planning for unmanned surface vehicles in complex marine environment," *IEEE J Ocean Eng* **45**(3), 813–830 (2020). doi: 10.1109/JOE.2019.2909508.

[16] W. Lee, G.-H. Choi and T.-W. Kim, "Visibility graph-based path-planning algorithm with quadtree representation," *Appl Ocean Res* **117**, 102887 (2021). doi: 10.1016/j.apor.2021.102887.

[17] B. Di, R. Zhou and H. Duan, "Potential field based receding horizon motion planning for centrality-aware multiple UAV cooperative surveillance," *Aerosp Sci Technol* **46**, 386–397 (2015). doi: 10.1016/j.ast.2015.08.006.

[18] J. Sun, J. Tang and S. Lao, "Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm," *IEEE Access* **5**, 18382–18390 (2017). doi: 10.1109/ACCESS.2017.2746752.

[19] A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Trans Syst Man Cybernet: Syst* **49**(4), 665–676 (2019). doi: 10.1109/TSMC.2017.2702701.

[20] X. Gao, S. Niu, D. Wei, X. Liu, T. Wang, F. Zhu, J. Dong and Q. Sun, "Joint metric learning-based class-specific representation for image set classification," *IEEE Trans Neur Net Lear Syst* **35**(5), 1–15 (2022). doi: 10.1109/TNNLS.2022.3212703.

[21] N. Khlif, K. Nahla and B. Safya, "Reinforcement learning with modified exploration strategy for mobile robot path planning," *Robotica* **41**(9), 2688–2702 (2023). doi: 10.1017/S0263574723000607.

[22] M. Ghafarian Tamizi, H. Honari, A. Nozdryn-Plotnicki and H. Najjaran, "End-to-end deep learning-based framework for path planning and collision checking: Bin-picking application," *Robotica* **42**(4), 1094–1112 (2024). doi: 10.1017/S0263574724000109.

[23] O. Kose and T. Oktay, "Simultaneous design of morphing hexarotor and autopilot system by using deep neural network and SPSA," *Aircr Eng Aerosp Tec* **95**(6), 939–949 (2023). doi: 10.1108/AEAT-07-2022-0178.

[24] Y. Liu, Z. Zheng, F. Qin, X. Zhang and H. Yao, "A residual convolutional neural network based approach for real-time path planning," *Know-Bas Syst* **242**, 108400 (2022). doi: 10.1016/j.knosys.2022.108400.

[25] Y. Liu, H. Wang, J. Fan, J. Wu and T. Wu, "Control-oriented UAV highly feasible trajectory planning: A deep learning method," *Aerosp Sci Technol* **110**, 106435 (2021). doi: 10.1016/j.ast.2020.106435.

[26] A. Sathyan, N. D. Ernest and K. Cohen, "An efficient genetic fuzzy approach to UAV swarm routing," *Unmann Syst* **04**(02), 117–127 (2015). doi: 10.1142/S2301385016500011.

[27] D. Adhikari, E. Kim and H. Reza, "A Fuzzy Adaptive Differential Evolution for Multi-Objective 3D UAV Path Optimization," **In:** *2017 IEEE Congress On Evolutionary Computation (CEC)*, (2017) pp. 2258–2265. doi: 10.1109/CEC.2017.7969578.

[28] Y. Dai, S. Li, X. Chen, X. Nie, X. Rui and Q. Zhang, "Three-dimensional truss path planning of cellular robots based on improved sparrow algorithm," *Robotica* **42**(2), 347–366 (2023). doi: 10.1017/S0263574723001480.

[29] B. Xin, J. Chen, J. Zhang, H. Fang and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Trans Syst Man Cybernet Part C (Appl Rev)* **42**(5), 744–767 (2012). doi: 10.1109/TSMCC.2011.2160941.

[30] S. Aslan and T. Oktay, "Path planning of an unmanned combat aerial vehicle with an extended-treatment-approach-based immune plasma algorithm," *Aerospace* **10**(5), 487 (2023). doi: 10.3390/aerospace10050487.

[31] J. F. S. and S. R., "Self-adaptive learning particle swarm optimization-based path planning of mobile robot using 2D Lidar environment," *Robotica* **42**(4), 977–1000 (2024). doi: 10.1017/S0263574723001819.

[32] Z. Qadir, M. H. Zafar, S. K. R. Moosavi, K. N. Le and M. A. P. Mahmud, "Autonomous UAV path-planning optimization using metaheuristic approach for predisaster assessment," *IEEE Inter Things J* **9**(14), 12505–12514 (2022). doi: 10.1109/JIOT.2021.3137331.

[33] Z. Yu, Z. Si, X. Li, D. Wang and H. Song, "A novel hybrid particle swarm optimization algorithm for path planning of UAVs," *IEEE Inter Things J* **9**(22), 22547–22558 (2022). doi: 10.1109/JIOT.2022.3182798.

[34] Y. Dai, J. Yu, C. Zhang, B. Zhan and X. Zheng, "A novel whale optimization algorithm of path planning strategy for mobile robots," *Appl Intell* **53**(9), 10843–10857 (2022). doi: 10.1007/s10489-022-04030-0.

[35] Z. W. Xie, Q. Zhang, Z. N. Jiang and H. Liu, "Robot learning from demonstration for path planning: A review," *Sci China Technol Sci* **63**(8), 1325–1334 (2020). doi: 10.1007/s11431-020-1648-4.

[36] P. Li and H. B. Duan, "Path planning of unmanned aerial vehicle based on improved gravitational search algorithm," *Sci China Technol Sci* **55**(10), 2712–2719 (2012). doi: 10.1007/s11431-012-4890-x.

[37] Z. Su and H. Wang, "A novel robust hybrid gravitational search algorithm for reusable launch vehicle approach and landing trajectory optimization," *Neurocomputing* **162**, 116–127 (2015). doi: 10.1016/j.neucom.2015.03.063.

[38] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "GSA: A gravitational search algorithm," *Inform Sci*, **179**(13), 2232–2248 (2009). doi: 10.1016/j.ins.2009.03.004.

[39] K. Jiao, J. Chen, B. Xin, L. Li, Y. Zheng and Z. Zhao, "Three Dimensional Path Planning for UAV Based on Chaotic Gravitational Search Algorithm," **In:** *The 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII2021)*, (Beijing, China, 2021) pp. 1–6.

[40] Y. Yang, J. Liu, Q. Wang and S. Yang, "Dynamic Path Planning for AGV Based on Tent Chaotic Sparrow Search Algorithm," **In:** *2021 International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT)*, (2021) pp.100–104. doi: 10.1109/ICEERT53919.2021.00029.

[41] J. Chen, B. Xin, Z. Peng, L. Dou and J. Zhang, "Optimal contraction theorem for exploration-exploitation trade-off in search and optimization," *IEEE Trans Syst Man Cybernet - Part A: Syst Humans* **39**(3), 680–691 (2009). doi: 10.1109/TSMCA.2009.2012436.

[42] M. Črepinšek, S.-H. Liu and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput Surv (CSUR)* **45**(3), 1–33 (2013).

[43] Vikas and D. R. Parhi, "Single and multiple humanoid path planning using Hill valley approach applied to gravitational drift in gravitational search algorithm," *Robotica* **41**(12), 3627–3648 (2023). doi: 10.1017/S0263574723001182.

[44] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, knowledge-based systems," *Knowl-Bas Syst* **89**, 228–249 (2015). doi: 10.1016/j.knosys.2015.07.006.

[45] H. Mittal, R. Pal, A. Kulhari and M. Saraswat, "Chaotic kbest Gravitational Search Algorithm (CKGSA)," In:*2016 Ninth International Conference on Contemporary Computing (IC3)*, (2016). pp.1–6. doi: 10.1109/IC3.2016.7880252.

[46] S. Mirjalili and A. Lewis, "Adaptive gbest-guided gravitational search algorithm," *Neural Comput & Applic* **25**(7-8), 1569–1584 (2014). doi: 10.1007/s00521-014-1640-y.

[47] S. Kumar and B. John, "A novel gaussian based particle swarm optimization gravitational search algorithm for feature selection and classification," *Neural Comput & Applic* **33**(19), 12301–12315 (2021). doi: 10.1007/s00521-021-05830-0.

[48] T. A. Khan and S. H. Ling, "A novel hybrid gravitational search particle swarm optimization algorithm," *Eng Appl Artif Intel* **102**, 104263 (2021). doi: 10.1016/j.engappai.2021.104263.

[49] H. Mittal and M. Saraswat, "An image segmentation method using logarithmic kbest gravitational search algorithm based superpixel clustering," *Evol Intell* **14**(3), 1293–1305 (2021). doi: 10.1007/s12065-018-0192-y.

[50] C. Yue, K. Price, P. N. Suganthan, J. Liang, M. Z. Ali, B. Qu, N. H. Awad and P. P. Biswas, "Problem Definitions and Evaluation Criteria for the CEC. 2020 Special Session and Competition On Single Objective Bound Constrained Numerical optimization, Comput. Intell. Lab.," Lab., Zhengzhou Univ., Zhengzhou, China, (2019).Tech. Rep 201911

[51] D. Li, W. Guo, A. Lerch, Y. Li, L. Wang and Q. Wu, "An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization," *Swarm Evol Comput* **60**, 100789 (2021). doi: 10.1016/j.swevo.2020.100789.

[52] J. Derrac, S. García, D. Molina and 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**(1), 3–18 (2011). doi: 10.1016/j.swevo.2011.02.002.