



A Curve Approximation Approach Using Bio-inspired Polar Coordinate Bald Eagle Search Algorithm

Yunhui Zhang^{1,2} · Yongquan Zhou^{1,2} · Guo Zhou³ · Qifang Luo^{1,2} · Binwen Zhu^{1,2}

Received: 16 January 2022 / Accepted: 6 April 2022
© The Author(s) 2022

Abstract

In this paper, a curve approximation approach using bio-inspired polar coordinate bald eagle search algorithm (PBES) is proposed. PBES algorithm is inspired by the spiral mechanism of bald eagle during predation. By introducing polar coordinate, the spiral predation process of the bald eagle will become more intuitive, which is more conducive for the algorithm to polar coordinate optimization problems. The initialization stage of PBES algorithm is modified to make the distribution of initialized individuals more uniform and some parameters are introduced to strengthen the exploration and exploitation capabilities of algorithm. The performance of the PBES algorithm is tested in three aspects: polar coordinate transcendental equation, curve approximation and robotic manipulator. The experimental results show that the PBES algorithm is superior to the well-known metaheuristic algorithms as it is effectively applicable for curve approximation problem.

Keywords Polar bald eagle search algorithm · Polar coordinate · Polar transcendental equations · Curve approximation · Robotic manipulator · Metaheuristic algorithm

Abbreviations

PBES	Polar coordinate bald eagle search
PSSA	Polar coordinate salp swarm algorithm
PTLBO	Polar teaching–learning-based optimization
PMPA	Polar marine predators algorithm
GA	Genetic algorithm
DE	Differential evolution algorithm
ED	Evolutionary deduction algorithm
PSO	Particle swarm optimization
ACO	Ant colony optimization
ABC	Artificial bee colony
FA	Firefly algorithm
BA	Bat algorithm
CS	Cuckoo search
DA	Dragonfly algorithm
MFO	Moth-flame optimization

SSA	Salp swarm algorithm
MPA	Marine predators algorithm
SMA	Slime mould algorithm
GSA	Gravity search algorithm
AEFA	Artificial electric field algorithm
EO	Equilibrium optimizer
ACROA	Artificial chemical reaction optimization algorithm
HGSO	Henry gas solubility optimization
SCA	Sine cosine algorithm
HS	Harmony search
TLBO	Teaching-learning-based optimization
GWO	Grey wolf optimization
IBES	Improved bald eagle search
HSSA	Hybrid salp swarm algorithm
BES	Bald eagle search
DBN	Deep belief network
SVM	Support vector machine
SPSO	Style particle swarm optimization
PC-AFSA	Polar-coded version of the artificial fish swarm algorithm
PCFOA	Polar coordinate coding of fruit fly optimization algorithm
CDF	Cumulative density function
IK	Inverse kinematics

✉ Yongquan Zhou
yongquanzhou@126.com

¹ College of Artificial Intelligence, Guangxi University for Nationalities, Nanning 530006, China

² Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis, Nanning 530006, China

³ Department of Science and Technology Teaching, China University of Political Science and Law, Beijing 102249, China

1 Introduction

The polar coordinate systems have applications in many fields such as mathematics, physics, and engineering. In the scene where the angle and distance between two points can be easily expressed, the polar coordinates can have a good representations effect, this kind of scene can only be obtained by trigonometric formula in Cartesian coordinate system. Polar equations are the simplest and easiest representation methods for a great deal of curves [1]. For the solution of some complex mathematical problems, the expression of polar coordinate equation is the most time-saving and labor-saving, such as spirals [2, 3]. In physics, for some special physical phenomena, for instance, objects in a plane physical system moving around a center point, or trajectory motion at a certain point, the use of polar coordinate modeling often simplifies complex problems and is more conducive to understand and solve, and even get unexpected results. Some geometric trajectory problems are simpler to be solved by polar coordinate method than by Cartesian coordinate method. For the expressions of some curves, it is difficult to express in the Cartesian coordinate system, but the use of polar coordinate equations will make the expressions of the curves more intuitive and clearer [4]. Now, the polar coordinate system is widely used in a number of science computation and engineering fields [5–7].

Metaheuristic algorithms are methods to derive the optimal and satisfactory solution of complex problems based on the bio-inspired mechanism [8]. Bio-inspired optimization explains the basic principle of optimization algorithm through the understanding of related behaviors, experiences, rules and action mechanisms in biological, physical, chemical, social and artistic systems, extracts corresponding feature models under the guidance of specific problem characteristics, and designs intelligent iterative search optimization algorithm [9]. Meta heuristic algorithm can solve NP hard problems effectively and search the optimal solution of the problem accurately [10, 11]. In general, the metaheuristic optimization methods are divided into the following four types according to different division criteria: evolution-based [12], swarm intelligence-based [13], physics and chemistry-based [14], and other methods. The first is based on evolutionary algorithms, such as genetic algorithm (GA) [15], differential evolution algorithm (DE) [16], evolutionary deduction algorithm (ED) [17]. The second is based on swarm intelligence and is generally based on the behavior of fauna in nature, such as particle swarm optimization (PSO) [18], ant colony optimization (ACO) [19], artificial bee colony

(ABC) [20], firefly algorithm (FA) [21], bat algorithm (BA) [22], cuckoo search (CS) [23], dragonfly algorithm (DA) [24], moth-flame optimization (MFO) [25], salp swarm algorithm (SSA) [26], marine predators algorithm (MPA) [27] and slime mould algorithm (SMA) [28]. The third type is based on physics and chemistry, such as gravity search algorithm (GSA) [29], artificial electric field algorithm (AEFA) [30], equilibrium optimizer (EO) [31], artificial chemical reaction optimization algorithm (ACROA) [32] and henry gas solubility optimization (HGSO) [33]. The last category is based on some other classical algorithms, such as sine cosine algorithm (SCA) [34], harmony search (HS) [35], teaching–learning-based optimization (TLBO) [36], etc.

In many industrial and practical application fields, curve approximation is a crucial research object [37]. Recently, many scholars are also studying the problem of curve approximation. Li [38] presented an adaptive knot placement algorithm for B-spline curve approximation to dense and noisy data points. The algorithm can be used for initial knot vector estimation for error-bounded curve approximation. The convergence speed of curve approximation iteration is faster than that of traditional methods. However, this method is not easy to understand, and the operation is troublesome. Carlos [37] used the simulated annealing algorithm to solve Bezier curve approximation problem. This study combined the advantages of traditional methods and heuristic algorithms, used Bezier curves to approximate the fitting data, and obtained better experimental results. But, no single simulated annealing pattern generates optimal values for all examples of benchmark. Moreover, different configurations show different effects in different applications, depending on the complexity and noise of the model. In [39], Laube presented a method using deep learning to compute parametrizations for B-spline curve approximation. This paper demonstrated that neural networks can successfully predict the parameters of B-spline curve approximation problems, but are constrained by the training set and limited by segmentation and sampling. Bureick [40] proposed an elitist genetic algorithm that solved the nodal adjustment problem in b-spline curve approximation in a faster and more accurate way than existing methods. However, outliers will appear in this study.

Due to the traditional methods still have problems such as insufficient precision, cumbersome operation, and long time in the curve approximation, the advantages of the metaheuristic algorithms are that it is more efficient than the blind search method, and a search problem can be obtained in a very short time. The optimal solution, for NP problems, can also get a better solution in a shorter time. Therefore,

it is necessary to solve the curve approximation problem through a metaheuristic algorithm. The bald eagle search (BES) algorithm is a novel bio-inspired metaheuristic algorithm proposed by Alsattar et al. [41]. The BES algorithm imitates the hunting strategies and smart social behavior of bald eagles during fishing. Bald eagles achieve their predation through three stages. The three stages are selecting stage, searching stage, and swooping. Taking salmon as an example, bald eagles will first select the search space based on the concentration of individual and population to salmon and fly towards a specific area; second, search the water surface in the selected search space until suitable prey is found, and finally, the bald eagles will gradually change their flight altitude, swoop quickly, and successfully catch salmon and other prey from the water [42]. BES algorithm simulates this hunting method of bald eagles. Moreover, this method has been proved to be an excellent nature-inspired solution [43]. Recently, some improved BES has been proposed as follows: Angayarkanni [43] proposed a new hybrid algorithm, which mixed the grey wolf optimization (GWO) and bald eagle search (BES) algorithm and optimized the parameters of the support vector regression for the prediction of traffic flow. Ramadan applied an improved bald eagle search (IBES) algorithm to PV model parameters [44]. Karthik [45] adopted BES to solve the practical application problem of dual-combustion reactivity control combustion ignition based on engine characteristics by changing low-activity fuel. Sarkar [46] introduced a hybrid salp swarm algorithm (HSSA) and bald eagle search (BES) algorithm to optimize the phase factor of OFDM system by using the PTS method. In [47], Zhou utilized the bald eagle search (BES) algorithm to improve the ELM in predicting carbon price which was used for optimizing and adjusting the values in the experiment. And Palanivel proposed the hybrid deep belief network (DBN) based BES algorithm to optimize and predict the pulsating heat pipe compound in parabolic solar collectors [48]. The bald eagle search algorithm was improved by using levy flight strategy and simulated annealing mechanism to solve simultaneous feature selection optimization problem [42]. Kang [49] proposed a transformer winding fault classification method based on BES algorithm to optimize the kernel parameter g and the penalty coefficient C in the SVM model, denoted as BES-SVM. In biomedicine, Sayed [50] came up with a prediction model for melanoma with imbalanced data based on optimized squeeze net by the BES and so on.

In order to strengthen the population diversity of the algorithm, some of improving the coding methods of algorithm have been proposed [51, 52]. Xiang proposed the polar coordinate salp swarm algorithm (PSSA) [53], which omitted

the transformation process of coordinate system. Different from the traditional coding methods proposed before, PSSA updates position directly in polar coordinate space. However, the convergence speed and optimization accuracy of PSSA need to be improved and it gets easily to fall into local optimization. To solve these problems, the polar coordinate bald eagle search algorithm (PBES) is proposed; its initialization stage is modified to make the distribution of initialized individuals more uniform, and some parameters are introduced to strengthen the exploration and exploitation capabilities of PBES. PBES improves the performance of the algorithm by increasing population diversity and enhancing the exploration ability. This paper introduces a modified PBES algorithm and applied it to three aspects as follows: polar coordinate transcendental equation, curve approximation, and robotic manipulator.

The rest of this paper is structured as follows: Sect. 2 is the preparation of related work: methods of polar coordinate coding and introduction of the BES algorithm. Section 3 will introduce the PBES algorithm. The results and discussion that combine polar transcendental equations and curve approximate design are given in Sect. 4. Section 5 is the inverse kinematics solution of a 7-DOF manipulator. We introduce the conclusion and future work in Sect. 6.

2 Related Work

2.1 Polar Coordinate Coding

Nowadays, there are various coding versions of metaheuristic algorithms have been proposed, such as binary conversion, four-element encoding, complex encoding, quantum coding, polar encoding, etc. They will strengthen the population diversity of the algorithm and enhance the search space by changing the coding modes. Matthysen [51] modified the particle swarm optimization (PSO) algorithm through the appropriate mapping function so that it can search and optimize in the polar coordinate space and put forward the polar coordinate version of PSO algorithm. For some evolutionary algorithms (EAs), corresponding scholars have continuously proposed polar coordinate versions. For instance, the polar evolutionary strategy (Polar ES) [52]. Yang [54] proposed a novel style particle swarm optimization (SPSO) algorithm based on polar coordination system, polar coordinate system and complex number operation, the design of the overlap space searching method and parameter of w . Shan [55] proposed a new selecting strategy in polar coordinates for multi-objective particle swarm optimization; some scholars [56]

designed an improved NSGA-II algorithm based on polar coordinates transform that made pareto optimal solutions distribution more uniform. Al-Sayyed [57] proposed a new optimization algorithm named POLARPSO, which can search more points by using the polar coordinate function, which not only enhanced the performance of PSO algorithm, but also avoided local optimal traps. Recently, some researchers put forward an improved polar-coded version of the artificial fish swarm algorithm (PC-AFSA) [58]. And Wu came up with a polar coordinate coding of fruit fly optimization algorithm (PCFOA) [59]. Lin [60] adopted a local clustering mode based on high-order histogram in polar coordinate; some achievements have been made in face recognition and retrieval problems.

Polar coordinate coding can further enrich the application range of the algorithm and achieve good results. However, the polar coordinate coding all use mapping functions to transform between polar coordinate space and Cartesian space, that is, the encoding and decoding process, which greatly increases the workload and brings distortion in the conversion process. In order to solve this problem, Xiang et al. proposed the PSSA [53]. The proposed PSSA can omit the coordinate system transformation process. PSSA does not need to use the mapping function to convert between the Cartesian system and the polar coordinate system, but directly initializes the individual in the polar coordinate system, and PSSA updates the position of the individual by renewing the polar angles and polar diameters, respectively. Owing to the population initialization, search process and position update are carried out in polar coordinate system directly, so PSSA reduces the computational complexity of the algorithm [53]. But, in PSSA, uneven initialization distribution gets easily to occur in the initialization stage, and the exploration ability of the algorithm needs to be strengthened. Moreover, the convergence speed and search accuracy of PSSA need to be improved as it can easily get stuck in local optimization. So, in this paper, the coding method of PSSA is improved, and its initialization stage is modified to make the distribution of initialized individuals more uniform, and some parameters are introduced to strengthen the exploration and exploitation capabilities of the BES algorithm. During calculating fitness function values, PBES does not need to be converted to Cartesian coordinates. On the contrary, the polar angle and the polar diameter are regarded as two variables, and the individual is judged by calculating the fitness function value. PBES increases the population diversity so that the algorithm can perform better under the same conditions. Accordingly, this makes the exploration ability of the algorithm greatly strengthened.

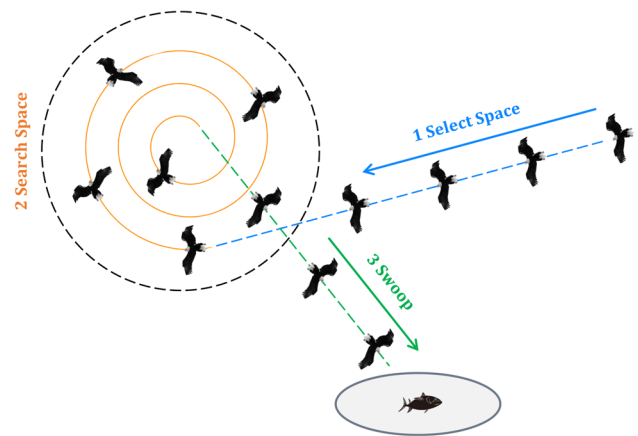


Fig. 1 The three main stages of hunting by BES [41]

2.2 Bald Eagle Search Algorithm (BES)

The bald eagle search algorithm (BES) imitates the hunting strategies and smart social behavior of bald eagles during fishing [41]. BES has three stages during predation, as shown in Fig. 1. In the select stage, bald eagles will select the space with the largest number of prey for search. Then, in the search stage, bald eagles search for prey by moving within the select space in the previous stage. Finally, in the swooping stage, bald eagles move and dive from the best position identified in the search stage and decide the best position for hunting. The swooping stage starts at the best point, to which all other movements are directed.

2.2.1 Select Stage

In the selection stage, bald eagles will first select the search space and then identify and select the best predation area, which is determined based on the number and concentration of prey. And then the bald eagles will prey in this area, as shown in Eq. (1).

$$P_{i,\text{new}} = P_{\text{best}} + \alpha * r(P_{\text{mean}} - P_i), \quad (1)$$

where α is the parameter that controls the position change, the value range is 1.5–2, and r is a random number between 0 and 1. P_{best} represents the determined best location for the bald eagles to pick an area in the previous search. Meanwhile, P_{mean} is the position of the average distribution of bald eagles after the previous search. P_i denotes the position of the i th bald eagles. $P_{i,\text{new}}$ is the new location of the bald eagles.

2.2.2 Search Stage

During the search stage, the bald eagles constantly search for prey in the search space selected in the previous phase, flying in a spiral shape and moving in different directions. The mathematical formula for the location update is as follows:

$$P_{i,\text{new}} = P_i + y(i) * (P_i - P_{i+1}) + x(i) * (P_i - P_{\text{mean}}), \quad (2)$$

$$x(i) = \frac{xr(i)}{\max(|xr|)}, y(i) = \frac{yr(i)}{\max(|yr|)}, \quad (3)$$

$$xr(i) = r(i) * \sin(\theta(i)), yr(i) = r(i) * \cos(\theta(i)), \quad (4)$$

$$\theta(i) = a * \pi * \text{rand} \text{ and } r(i) = \theta(i) + R * \text{rand}, \quad (5)$$

where $\theta(i)$ is the polar angle and $r(i)$ is the polar diameter. a and R are the parameters that control the spiral trajectory, and their variation ranges are 5–10 and 0.5–2, respectively. And rand is a random number between 0 and 1. $x(i)$ and $y(i)$ represent the position of the bald eagle in polar coordinates, both ranging from -1 to 1 . P_{i+1} is the next updated position of the i th bald eagles. $|xr|$ is the absolute value of xr . $|yr|$ means the same as $|xr|$.

2.2.3 Swooping Stage

During the swooping stage, the bald eagles are constantly moving, swooping towards the prey from the best location in the previous search, while other individuals also swoop from the best position to hunt the prey. Equation (6) simulates this behavior of bald eagles as follows:

$$P_{i,\text{new}} = \text{rand} * P_{\text{best}} + x1(i) * (P_i - c_1 * P_{\text{mean}}) + y1(i) * (P_i - c_2 * P_{\text{best}}), \quad (6)$$

$$x1(i) = \frac{xr(i)}{\max(|xr|)}, y1(i) = \frac{yr(i)}{\max(|yr|)} \quad (7)$$

$$xr(i) = r(i) * \sinh(\theta(i)), yr(i) = r(i) * \cosh(\theta(i)), \quad (8)$$

$$\theta(i) = a * \pi * \text{rand} \text{ and } r(i) = \theta(i), \quad (9)$$

where c_1 and c_2 increase the movement intensity of bald eagles towards the best and center points; they all take values between 1 and 2.

The pseudo-code of BES is shown in Algorithm 1 [41].

Algorithm 1 BES algorithm.

Pseudo-code of the BES algorithm

1. Random initialization Point P_i
2. Calculate the fitness values of initial Point: $f(P_i)$
3. **While** (the termination conditions are not met)
- Select space**
4. **For** (each point i in the population)
5. $P_{i,\text{new}} = P_{\text{best}} + \alpha * \text{rand}(P_{\text{mean}} - P_i)$
6. **If** $f(P_{\text{new}}) < f(P_i)$
7. $P_i = P_{\text{new}}$
8. **End If**
9. **If** $f(P_{\text{new}}) < f(P_{\text{best}})$
10. $P_{\text{best}} = P_{\text{new}}$
11. **End If**
12. **End For**
- Search in space**
13. **For** (each point i in the population)
14. $P_{i,\text{new}} = P_i + y(i) * (P_i - P_{i+1}) + x(i) * (P_i - P_{\text{mean}})$
15. **If** $f(P_{\text{new}}) < f(P_i)$
16. $P_i = P_{\text{new}}$
17. **End If**
18. **If** $f(P_{\text{new}}) < f(P_{\text{best}})$
19. $P_{\text{best}} = P_{\text{new}}$
20. **End If**
21. **End For**
- Swoop**
22. **For** (each point i in the population)
23. $P_{i,\text{new}} = \text{rand} * P_{\text{best}} + x1(i) * (P_i - c_1 * P_{\text{mean}}) + y1(i) * (P_i - c_2 * P_{\text{best}})$
24. **If** $f(P_{\text{new}}) < f(P_i)$
25. $P_i = P_{\text{new}}$
26. **End If**
27. **If** $f(P_{\text{new}}) < f(P_{\text{best}})$
28. $P_{\text{best}} = P_{\text{new}}$
29. **End If**
30. **End For**
31. Set $k = k + 1$
32. **End While**

3 Our Proposed PBES Algorithm

3.1 Inspiration

Inspired by Fig. 2, in view of the fact that the trajectory shape of bald eagle searching for preys is spiral and bald eagle dives spirally to capture the preys during the predation phase. Therefore, the BES algorithm can be applied to polar coordinate space. Therefore, the PBES algorithm is proposed. In the polar coordinate system, for each individual can be defined and initialized. In the initialization phase, each individual is defined as a binary

array (ρ_n, θ_n) . This amounts to increasing the population diversity during initialization process. In this way, when calculating the fitness function, PBES omits the transformation process of coordinate system. In polar coordinates, the PBES algorithm updates the polar angle and polar diameter, respectively, and calculates the fitness value directly.

3.2 PBES Mathematical Model

In the polar coordinate system, for the initialization phase of the PBES algorithm, the parameters of the individual, that is, the polar angle and the polar diameter are directly initialized and stored as an array. Furthermore, when the initial points are transformed between polar coordinate space and Cartesian space, they will produce a phenomenon of non-uniform distribution. To avoid the distortion happened in the transformation, we introduce the Archimedes theorem and cumulative density function (CDF) [61]. The CDF is inverted to obtain the initialization formula as follows:

$$\rho = rand * (ub - lb) + lb, \tag{10}$$

$$\theta = \beta * \cos^{-1}(2 * rand - 1), \tag{11}$$

where ub is the upper bound, lb is the lower bound in PBES, $rand$ is a random number in the range of 0–1. β is the disturbance coefficient which takes a value between 0 and 2. The PBES algorithm also needs to control the boundary during initialization, and all individuals can be distributed in the whole search space. Therefore, the value range of polar angle θ is $(0, 2\pi)$. At the same time, polar diameter ρ also needs to be defined using boundaries so that the PBES algorithm does not exceed the boundary in the optimization process.

Next step, the location of each individual is updated. In PBES, the individual position is updated by renewing ρ and θ , respectively. Since BES has three stages, in PBES, the ρ is updated three times, respectively. The update formula of polar diameter ρ is given below as follows:

(1) Select stage

$$\rho_{i,new} = \rho_{best} + \alpha * r(\rho_{mean} - \rho_i), \tag{12}$$

where α is called position change parameter, it is a parameter that controls the position change, the value range is $(1.5, 2)$ and r is a random number with a value between 0 and 1. $\rho_{i,new}$ is the updated polar diameter. ρ_{best} is the current best polar diameter. ρ_{mean} is the average polar diameter after the previous search. ρ_i denotes the i th polar diameter.

(2) Search stage

$$\rho_{i,new} = \rho_i + m_1 * (\rho_i - \rho_{mean}) + n_1 * (\rho_i - \rho_{i+1}), \tag{13}$$

$$m_1 = \frac{\omega_1}{\max(|\omega_1|)}, n_1 = \frac{\eta_1}{\max(|\eta_1|)}, \tag{14}$$

$$\omega_1 = r_1 * \sin(\theta), \eta_2 = r_1 * \cos(\theta), \tag{15}$$

$$r_1 = a * \pi * rand, \tag{16}$$

where a is a parameter that controls the spiral trajectory, and its variation range is 5 to 10. $rand$ is a random number with a value between 0 and 1. r_1 is a control parameter that can control the update range of the polar diameter. m_1 and n_1 represent the normalization of ω_1 and η_1 .

(3) Swooping stage

$$\rho_{i,new} = rand * \rho_{best} + m_2 * (\rho_i - c_1 * \rho_{mean}) + n_2 * (\rho_i - c_2 * \rho_{best}) \tag{17}$$

$$m_2 = \frac{\omega_2}{\max(|\omega_2|)}, n_2 = \frac{\eta_2}{\max(|\eta_2|)} \tag{18}$$

$$\omega_2 = r_1 * \sinh(\theta), \eta_2 = r_1 * \cosh(\theta), \tag{19}$$

where c_1, c_2 is the enhancement coefficient, and they all take the value of 2. The pole diameter ρ is updated in three stages, after each iteration update, the updated polar diameter ρ requires to be contrasted with each other, if it is smaller than the current best polar diameter, the ρ will be changed. The mathematical formula for polar angle θ is expressed as follows:

$$\theta_{i+1} = \beta * \theta_i \pm 2 * \cos^{-1}(2 * rand - 1), \tag{20}$$

where β is the disturbance coefficient which takes a value between 0 and 2. After each iteration update, the updated polar angle θ requires to be contrasted with each other. If it is superior to the current optimal angle, the plus sign operation is performed; otherwise, the minus sign operation is performed.

After updating the values of ρ and θ respectively, the new individual position is determined, and then the fitness function values of all individuals are calculated to find out the current best fitness function value, after that, contrast it with the global optimal fitness value F . There are two situations: one is replacement and the other is retention. If the new fitness function value is found to be superior to the current fitness value, perform the first operation; otherwise perform the second operation [53]. In the PBES, individual positions are updated in the polar coordinate system rather than in the Cartesian coordinate system; by updating ρ and θ respectively, the position of the individual is obtained. This will greatly improve the updating speed of individuals and improve the convergence efficiency.

Through the pseudo code of the PBES algorithm (Algorithm 2) and the flowchart of the PBES algorithm (Fig. 3), the PBES algorithm can be more easily to understand.

Algorithm 2 PBES algorithm.

Pseudo-code of the **PBES algorithm**

Input population size N , maximum iteration number
 Maxiter, dimension dim, upper up and lower bounds lb
Output the optimal solution

1. Initialization and load PBES parameters
2. Let $N =$ population size
3. Each individual $i = 1$ to N
4. Calculate the fitness values of all the individuals
5. **While** (the termination conditions are not met)
6. **for** $i = 1: \text{dim}$ **do**
7. **for** $j = 1: \text{Searchagents}$ **do**
8. Initialize the $S(\rho_n, \theta_n)$ using the Eqs. (10) and (11)
9. Evaluate fitness value of each S
10. Find the best fitness value and set it as best search agent F
11. **For** (each Eagle (x_i)) **do**
12. Update polar diameter through three stages
- Select space**
13. Update polar diameter using Eq. (12)
14. **If** $f(\rho_{\text{new}}) < f(\rho_i)$
15. $\rho_i = \rho_{\text{new}}$
16. **If** $f(\rho_{\text{new}}) < f(\rho_{\text{best}})$
17. $\rho_{\text{best}} = \rho_{\text{new}}$
18. **End If**
19. **End If**
- Search in space**
20. Update polar diameter using Eq. (13)
21. **If** $f(\rho_{\text{new}}) < f(\rho_i)$
22. $\rho_i = \rho_{\text{new}}$
23. **If** $f(\rho_{\text{new}}) < f(\rho_{\text{best}})$
24. $\rho_{\text{best}} = \rho_{\text{new}}$
25. **End If**
26. **End If**
- Swoop**
27. Update polar diameter using Eq. (17)
28. **If** $f(\rho_{\text{new}}) < f(\rho_i)$
29. $\rho_i = \rho_{\text{new}}$
30. **If** $f(\rho_{\text{new}}) < f(\rho_{\text{best}})$
31. $\rho_{\text{best}} = \rho_{\text{new}}$
32. **End If**
33. **End If**
34. Update polar angle using Eq. (20)
35. Each particle $i = 1$ to N
36. Evaluate fitness
37. $\text{Eagle}' = x_j^1$
38. Compare the values of Eagle' and F
 If $F < \text{Eagle}'$, replace the value of F with Eagle'
39. Update the Eagles position based on the upper and lower bounds of variables
40. Termination conditions are met
41. Ultimately, the Eagle' is the optimal solution
42. **End While**

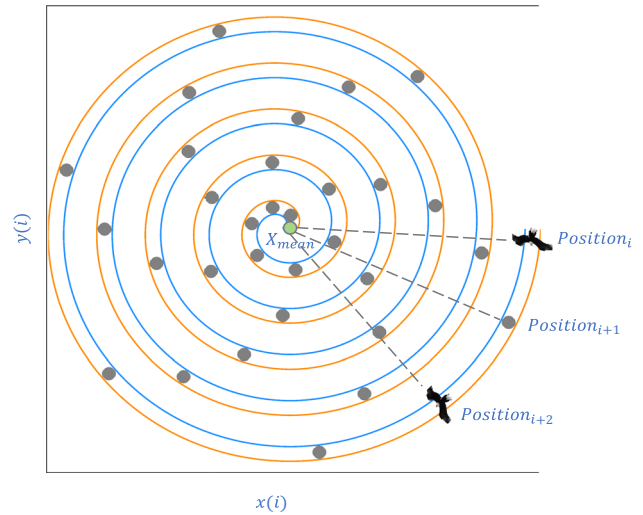


Fig. 2 Bald eagles searching within a spiral space

3.3 PBES Complexity Analysis

In PBES, the population size is N , the dimension of problem is d , and the amount of iterations is t . For time complexity, first, during population initialization and parameter setting, time complexity is $O(N*d)$; then, when updating the polar diameters, since BES has three stages, the algorithm has a three-layer loop, therefore, the time complexity is $O(N*d*t)$; For termination criteria, the relevant time complexity is $O(1)$. Finally, the time complexity of PBES is $O(N*d*t)$. For the space complexity, the search agents of the proposed PBES are N and the problem's dimension is d . Thus, the space complexity of PBES is ultimately obtained: $O(N*d)$.

4 Results and Discussion

4.1 Experiment Settings

So as to verify the effectiveness of the PBES algorithm, PBES was tested in polar transcendental equations, curve approximation design. All algorithms are run under the environment of MATLAB R2017b and the experiment was performed on an Inter(R) Core (TM) i7-97000 CPU @ 3.00 GHz 64-bit operating system with a 16 GB RAM PC. PBES is compared with the polar teaching-learning-based optimization algorithm (PTLBO), polar coordinate salp swarm algorithm (PSSA) [53], polar marine predators algorithm (PMPA) in the best value, mean value, and standard deviation. The parameter settings of each algorithm are shown in Table 1. In order to avoid accidental results during the experiment, each function is run ten

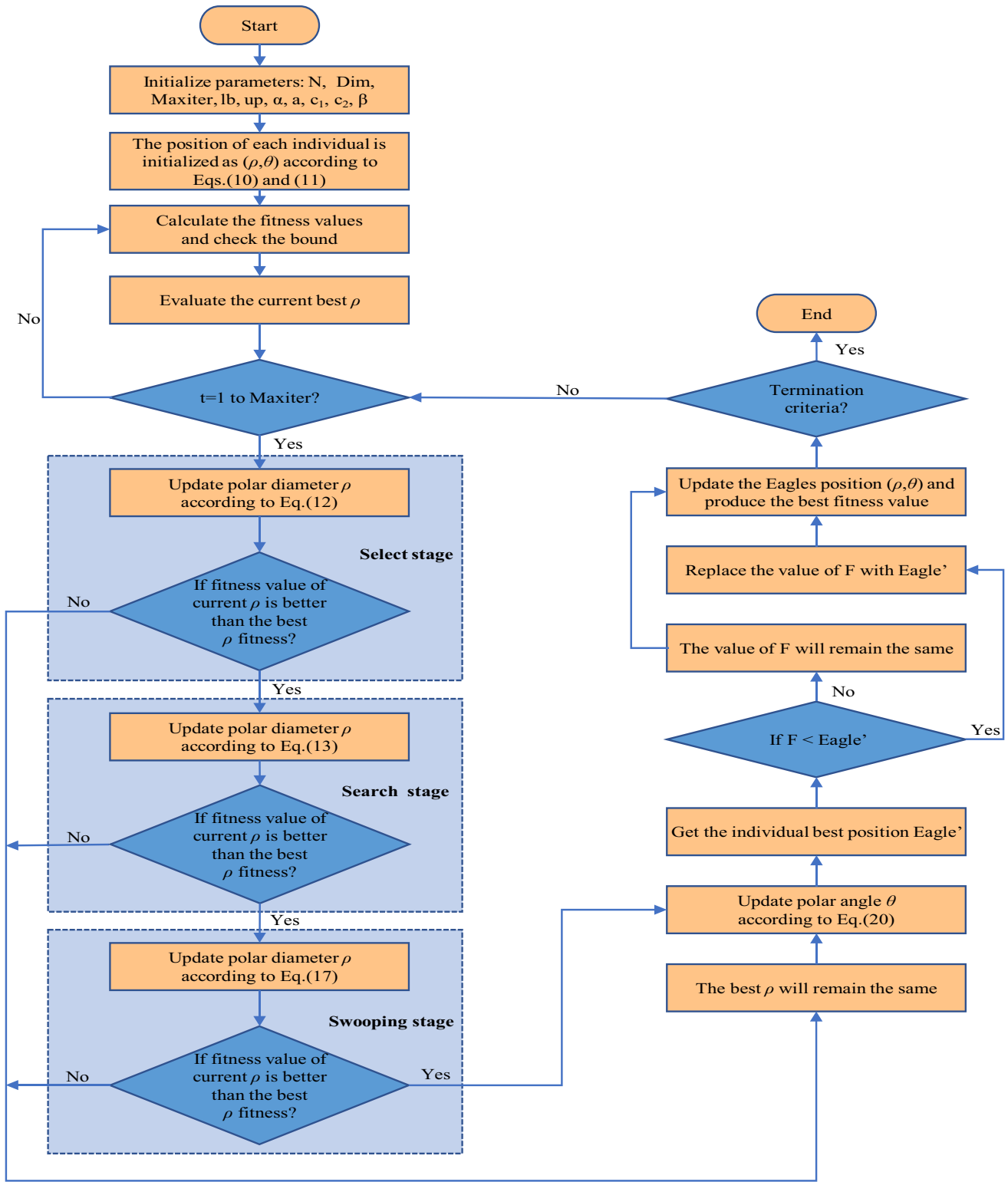


Fig. 3 Flowchart of the PBES

times independently. Furthermore, in this simulation experiment, the population size is set to 200 and the amount of iterations is 500.

4.2 Testing Complex Polar Equations

Transcendental equations are equations that contain transcendental functions, that is, functions that cannot be expressed by polynomials or square roots of independent variables, as opposed to algebraic equations [62]. Transcendental equations cannot be solved by algebraic geometry. There is no general formula for solving most transcendental equations and it is difficult to get analytical solutions [63]. In order to verify the performance of the PBES algorithm, PBES was tested in polar transcendental equations and the transcendental equations are shown in Table 2, where ρ denotes the polar diameter, and θ denotes the polar angle. The data results of the four transcendental equations tested by all algorithms are shown in Table 3, including the best value, the mean value and the standard deviation.

In Table 3, we tested four transcendental equations F1, F2, F3 and F4. For equations F1 and F3, the best value of PBES is 1–2 orders of magnitude better than other algorithms, especially the PTLBO algorithm. For the mean value, PBES also outperforms other comparison algorithms. This shows that PBES is more accurate than other algorithms and can obtain better solutions. The performance of PBES standard deviation is also excellent. So PBES can not only obtain optimal solutions but also be more stable when solving transcendental equations F1 and F3. Towards equation F2, PBES is smaller than other comparison algorithms in the best value and mean value. There is little difference between PBES and PMPA in mean value. But, the standard deviation of PTLBO is better than other algorithms, including PBES. For F4, the performance of PBES is superior to other algorithms, but in the mean value, it is not much different from other algorithms and is inferior to PMPA.

The convergence curve of F1, F2, F3 and F4, is shown in Fig. 4. The first picture of Fig. 4 shows that PBES has better convergence accuracy for F1. The fluctuation of PTLBO and PSSA is relatively large before the F1 converges. Before the convergence of PMPA and PBES, the fluctuations are relatively small, but PBES can find more excellent solution. While, the convergence time of PBES and PSSA is almost the same. They all begin to converge at about 200 generations. For F2, the situation of the PSSA and PBES is basically consistent with that of F1. For the four algorithms, PBES has a slow convergence speed and low accuracy in the early stage, but in the later stage, PBES finds the smallest optimal solution. In the system of equations F3, the convergence effect of PBES is the best, both in terms of convergence speed and accuracy. This is the same as the data in

Table 3. Look at F4; the convergence speed of PBES is the slowest, and it only converges when it is close to 300 generations. Like other equations, the best value obtained by PBES is also better than other algorithms.

To sum up, these results show that PBES is superior to PTLBO, PSSA and PMPA both in convergence precision, average value and standard deviation. These results verify that PBES has better performance in solving polar transcendental equation problems. The exploration capability and exploitation capability of the PBES algorithm is better than other comparison algorithms. Conventional methods are not suitable for solving polar transcendental problems. Although PSSA can also solve polar transcendental problems [53], PBES performs better than PSSA. Compared with other algorithms that can solve transcendental equations, PBES can get better solutions in most aspects.

4.3 Testing of Curve Approximation

4.3.1 The Curve Approximation Design Problem

Intuitively, the curve can be seen as the trace left by the movement of a certain point [64]. This definition of a curve is formalized in modern mathematics as: a curve is a graph of an interval from a continuous function to a topological space [65]. Generally speaking, a curve often refers to a set of consecutive points in Euclidean space. Any continuous line can be called a curve, including straight lines, polylines, line segments, arcs, etc.

The curve approximation design is very similar to curve fitting, but there are some theoretical differences. The mathematical definition of Curve Fitting [66, 67] refers to the use of continuous curves to approximately and it is the process of constructing a simulation curve or mathematical function expression, by fitting a series of data points that may be constrained, and requiring the final effect to achieve the best. Finally, an approximately smooth function curve is constructed to approximate these fitted data [68]. Also, another related topic is regression analysis, but it focuses more on statistical inference problems.

In engineering analysis and scientific research, it is often necessary to obtain approximate function relations between independent variables and dependent variables based on some test data to complete other design calculations [69]. For the function $F(x)$ given in function class A, denoted as $F(x) \in A$, it is required to find the function in another class of simple and calculable functions B so that the error of $P(x)$ and $F(x)$ is minimized in some metric sense. Function class A is usually a continuous function on the interval $[a, b]$, denoted by $C[a, b]$, called function approximation space. And the function B is usually n -degree polynomial, rational function or piecewise low-degree polynomial, etc. [70]. In short, it is to replace the function $F(x)$ with a simple function

Table 1 Parameter setting of the comparison algorithm

Algorithms	Parameter settings
PTLBO	Teaching factor $TF \in \{1, 2\}$, learning step $r_i \in [0, 1]$
PSSA	Convergence factor c_1 , random number $r_1 \in [0, 1]$, integer random number $r_2 \in [0, 50]$, random number $r_3 \in [0, 1]$, random number $r_4 \in [0, 1]$
PMPA	Constant $P = 0.5$, constant $FADs = 0.2$, random number $r \in [0, 1]$
PBES	Position change parameter $\alpha = 1.5$, disturbance coefficient $\beta = 1.2$, random number $r \in [0, 1]$, spiral parameter $a = 10$, enhancement coefficient $c_1, c_2 = 2$, control parameter $r_1 \in [0, 1]$

Table 2 Set of diverse polar equation

Equations [53]
$F1(\rho, \theta) = \log_5^{\rho^2 \times \sin(\theta) + e^{\cot(\theta)}} \times \rho^{\frac{1}{2}} \times \sec(2 \times \theta) + 3 \times \rho^{4 \times e^2} \times \cot(\theta) - \rho^{-\frac{2}{5}} \times e^{\sin(2 \times \theta) + e^{\cos(\theta)}} \times \csc(\theta) + 1$
$F2(\rho, \theta) = \frac{\sin(\cos(2 \times \theta))}{1 + (\rho \times \tan^2(\theta))^2} + e^{1 + \sin^2(\rho \times \theta)} \times \csc(\ln(\cos(\frac{1}{7} \times \theta))) \times \theta - \rho^{\frac{2}{3}} \times (4 - \cot^2(\theta))$
$F3(\rho, \theta) = \begin{cases} \log_2^{\rho^2 \times \cot^2(\theta) + \sin \theta} \times \sec(2 \times \theta) + 2 \\ e^{\rho^8 \times \sin^3(\theta) + \rho^{\frac{2}{3}} \times \cos^{\frac{1}{2}}(2 \times \theta)} + \rho^{\frac{1}{3}} \times \sin(\theta) \times \tan(\theta) - 2 \end{cases}$
$F4(\rho, \theta) = \begin{cases} \sec^3(\frac{\theta}{1 + \rho}) \times e^{\rho^2 - \sin(\theta)} + 2 \\ \frac{\log_3^{\rho^2 \times \cot^2(\theta)}}{1} - 1 \\ \frac{1}{3} \sin^2 \theta \\ \rho^{\frac{2}{2}} \times \tan(\theta) + \cot^5(2 \times \theta) \end{cases}$

Table 3 Experimental results with comparison algorithms

Algorithms		PTLBO	PSSA	PMPA	PBES
F1	Best	6.32E-06	9.25 E-07	4.91 E-07	3.07 E-08
	Mean	4.80 E-05	4.23 E-05	4.51 E-06	1.34 E-06
	Std	1.81 E-04	2.85 E-04	4.60 E-05	8.01 E-06
F2	Best	1.34 E-06	4.16 E-06	2.40 E-07	3.84 E-08
	Mean	3.20 E-06	3.24 E-05	2.10 E-06	1.12 E-06
	Std	7.22 E-06	2.37 E-04	7.38 E-06	4.82 E-05
F3	Best	2.05 E-06	1.32 E-07	3.41 E-07	1.50 E-08
	Mean	3.79 E-05	9.14 E-07	3.47 E-05	6.89 E-07
	Std	1.17 E-04	1.11 E-05	9.87 E-05	4.67 E-06
F4	Best	1.65 E-06	8.11 E-08	1.20 E-07	1.86 E-08
	Mean	1.24 E-05	3.12 E-06	1.42 E-06	2.31 E-06
	Std	6.32 E-05	4.27 E-05	8.78 E-06	7.30 E-06

Optimal values are given in bold

$P(x)$ approximately. This approximate substitution is called function approximation design, which is one of the most basic concepts and methods in computational mathematics and is widely used in image processing, reverse engineering and test data processing [71]. The traditional methods of function approximation include Lagrange interpolation

polynomial, Spline interpolation, Ordinary least squares and Chebyshev approximation; the calculation process is complicated. Nowadays, the latest approximation is the neural network approximation.

4.3.2 Curve Approximation Design

In this section, the PBES algorithm proposed above is applied to the curve approximation design to verify the superiority of the PBES algorithm.

Table 4 tests some familiar polar equations of different types of curves, including Conic section, Pascal spiral, Spiral line and Special curve. Table 4 shows the polar coordinate equations of different curves and some necessary parameter settings. Table 5 shows the optimization results and comparison for curves in terms of best value, mean value and standard deviation. From Table 5, we can see that compared with other algorithms, PBES had achieved higher accuracy in most equations. For most curves, PBES shows an excellent effect in terms of maximum accuracy (best value). This indicates that PBES has strong exploitation ability. For the mean value and standard deviation, the four algorithms have their own advantages and disadvantages, but overall, PBES can achieve the best results in the three indicators. Standard deviation can reflect the degree of dispersion of a data set. In

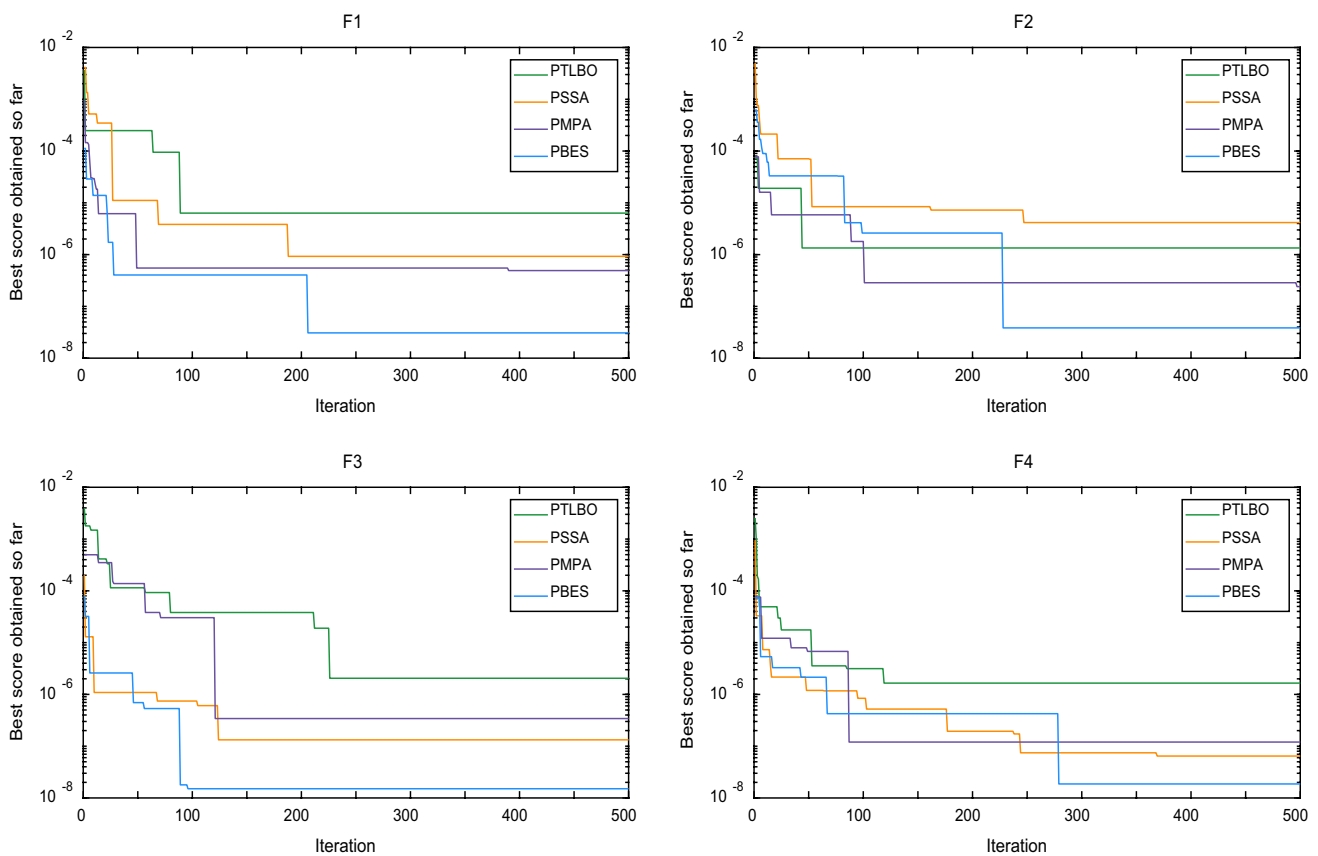


Fig. 4 The convergence curves of F1–F4

Table 5, PBES has the best performance in terms of standard deviation, which shows that the PBES algorithm has less fluctuation and strong stability in solving the curve approximation problem and has the best fitting approximation effect. And PBES can easily solve various curve problems. In Table 6, the original image of the function is obtained by selecting 500 points between 0 and 2π on the equation of the function curve. These uniform points obtained between 0 and 2π constitute the data set of curve approximation. After the iterative approximation of the algorithm, the image of the curve is preserved.

4.3.3 Results and Analysis

Table 6 shows the results of the curve approximation test, including curve approximation diagrams of PTLBO, PSSA, PMPA, PBES and curve’s original image. For conic section, PBES is obviously better than other comparison algorithms in curve approximation of Parabola. For these algorithms, the tails of the Parabola curve cannot fit and approximate well, there are many missing pieces. PSSA and PMPA approximate the upper or lower half of the Ellipse well; correspondingly they lose a lot of detail in the other half.

PTLBO and PBES can approximate the upper and lower parts of the Ellipse greatly. But PTLBO is still lacking in the accuracy of curve approximation. This shows that PSSA and PMPA can easily fall into local optima, while PBES can jump out of local optimum. The points approximated by PBES are more uniform. On the Cardioid, the approximation effect of PBES is more obvious. Like Ellipse, PSSA and PMPA also lost some points on the curve approximation. And PTLBO has the worst performance. For some spiral lines, such as Archimedes spiral, the curve approximation effect of PBES is more competitive than other algorithms and PBES can be far away from the local optimum, so it can better approximate the curve. However, according to the NFL theorem, no single algorithm can be perfectly applied to all problems. For Logarithmic spiral, the approximation effects of PBES and PSSA are almost the same. This indicates that both PBES and PSSA have good approximation effects. For example, in Hyperbolic spiral curve, the effect of PMPA and PBES is similar. On the vast majority of curves, PBES can achieve good results. For Fermat spiral, the accuracy of PTLBO and PMPA still needs to be improved, and there are many noise points around the curve that were not successfully fitted. This situation also exists on other curves.

For most curves, these curves fitted by the PBES algorithm are basically consistent with the original curve image and the effect of PBES is much better than other algorithms. In some special curves like the Water droplets curve, the approximation effects of the PSSA and PBES made no difference. Their fitting effects are better than PTLBO and PMPA. PBES also achieved good results on Lemniscate, Plum blossom curve and Moppet curve. However, in some complex graphs, for instance, Rose curve and Butterfly curve, the approximation graphs of those four algorithms are partially missing, and the pattern of the graphic cannot be fully displayed. The effect of PBES is still superior to others.

In summary, from the curve approximation diagram, we can see that PBES has achieved very effective results in curve approximation. For most of the curves, PBES can approach the curves very well and gets the results with high fit to the original curve. However, like other algorithms, PBES cannot achieve very efficient results on complex curves; they will have the phenomenon of missing graphs in some curve parts. Despite this, the PBES still performs better than others in curve approximation design problem.

4.3.4 Error Analysis

Figures 5 and 6, respectively, show the absolute error diagram and relative error diagram for solving each test curve. Absolute error represents the absolute value of the difference between the measured value and the real value. Relative error refers to the ratio of the absolute error caused by measurement to the measured true value and the value obtained by multiplying by 100%, expressed as a percentage.

In general, relative error is a better indicator of the reliability of the measurement. From Fig. 5, the absolute errors of four algorithm curves are compared, where the yellow circle represents the curve approximation points obtained by the algorithm, and the blue point represents 500 points generated from 0 to 2π on the function curve. When the yellow circle and the blue point are denser and uniform, the error is smaller, and when the algorithm matches the original curve better, the approximation effect of algorithm is better. For relative error, it means the error between the approximation point obtained by the algorithm and the point on the original curve during the iterative process. The relative error is smaller means the approximation effect of algorithm is better, more stable of the algorithm, and the accompanied by smaller fluctuations.

From the absolute error diagram, we can see that PTLBO performs poorly; there are some points that are not fitted, and the error is large. PMPA performs well on some curves, such as Parabola, Logarithmic spiral, Lemniscate, Rose curve. But there is still some gap compared with PBES. The absolute error diagram of PSSA is relatively clean, and there is no large area of noise, which shows that the fitting accuracy of PSSA is high, but the effect is general. For most curves, the relative error of PBES is smaller than others and absolute error performs better, especially in conic section, Lituus, KeNu spiral, Round involute, Water droplet curve, Lemniscate. For the Parabola and Logarithmic spiral, Plum blossom curve, the relative error of PBES is not the best, but the gap with them is very small. And the relative errors of these algorithms are not very different in Butterfly curve. For the Plum blossom curve and Moppet curve, PTLBO,

Table 4 Set of classic spirals

Curve name	Polar equations	Parameters
Parabola	$\rho = \frac{a \times b}{1 - a \times \cos(\theta)}$	$a = 1, b = 1.5$
Ellipse	$\rho = \frac{a \times b}{1 - a \times \cos(\theta)}$	$a = 0.5, b = 1.1$
Cardioid	$\rho = a \times \cos(\theta) + b$	$a = 1, b = 1$
Archimedes spiral	$\rho = a \times \theta$	$a = 0.3$
Fermat spiral	$\rho^2 = a^2 \times \theta$	$a = 0.3$
Logarithmic spiral	$\rho = a \times e^{b \times \theta}$	$a = 0.4, b = 0.3$
Lituus	$\rho^2 \times \theta = a^2$	$a = 3$
Hyperbolic spiral	$\rho \times \theta = a$	$a = 0.2$
KeNu spiral	$\rho^{b \times \theta} = a$	$a = 1.2, b = 2$
Round involute	$\rho = \theta + a$	$a = 1.2$
Water droplets curve	$\rho^2 = -1 - a \times \cos(b \times \theta) - \sin(\theta)$	$a = 2, b = 2$
Lemniscate	$\rho^2 = 2 \times a^2 \times \cos(b \times \theta)$	$a = 1, b = 2$
Rose curve	$\rho = a \times \cos(b \times \theta)$	$a = 2, b = 2$
Plum blossom curve	$\rho = a + (b \times \cos(c \times \theta))^2$	$a = 10, b = 3, c = 2.5$
Moppet curve	$\rho = a \times \sin(b \times \theta) + \sin(c \times \theta) $	$a = 2, b = 2, c = 5$
Butterfly curve	$\rho = e^{\sin \theta} - a \times \cos(b \times \theta) + \sin^5(\frac{2 \times \theta - \pi}{24})$	$a = 2, b = 4$

Table 5 Optimization results and comparison for curves

	Name	PTLBO	PSSA	PMPA	PBES	Name	PTLBO	PSSA	PMPA	PBES
Best	Parabola	4.25E-08	2.23 E-07	2.95 E-08	6.97E-10	KeNu spiral	1.25 E-07	8.97 E-08	1.41 E-08	2.10 E-10
Mean		7.28 E-07	3.76 E-06	1.34 E-06	1.65 E-07		5.70 E-06	2.97 E-06	1.97 E-06	2.51 E-07
Std		9.07 E-07	8.82 E-06	8.46 E-06	4.26 E-07		4.32 E-05	6.99 E-05	1.03 E-05	6.01 E-07
Best	Ellipse	2.01 E-08	5.91 E-07	3.57 E-08	5.09 E-10	Round involute	3.21 E-07	6.55 E-07	6.06 E-08	6.47 E-10
Mean		4.55 E-07	2.76 E-07	5.84 E-07	3.78 E-07		3.39 E-07	3.06 E-07	4.57 E-07	3.39 E-07
Std		1.25 E-06	2.13 E-06	3.97 E-06	2.05 E-06		4.63 E-06	2.33 E-06	1.45 E-06	1.76 E-06
Best	Cardioid	3.23 E-08	2.26 E-07	5.47 E-09	3.03 E-11	Water droplets curve	3.52 E-08	3.08 E-09	1.82 E-07	1.88 E-09
Mean		5.75 E-07	1.24 E-06	6.48 E-07	2.49 E-07		1.17 E-06	6.83 E-07	2.28 E-06	2.84 E-07
Std		3.09 E-06	1.52 E-05	4.12 E-06	2.64 E-06		1.07 E-05	2.11 E-06	4.43 E-05	8.62 E-06
Best	Archimedes spiral	4.64 E-08	3.21 E-09	1.88 E-10	7.97 E-10	Lemniscate	6.40 E-08	4.65 E-09	1.54 E-08	9.23 E-10
Mean		7.57 E-07	4.42 E-07	4.86 E-07	4.02 E-07		1.76 E-06	6.30 E-07	2.39 E-06	2.34 E-07
Std		1.10 E-05	6.12 E-07	2.75 E-06	3.97 E-06	Rose curve	5.27 E-06	9.02 E-06	2.41 E-05	1.78 E-06
Best	Fermat spiral	3.80 E-08	4.30 E-08	6.63 E-09	2.20 E-09		6.25 E-07	1.26 E-09	1.35 E-07	1.57 E-09
Mean		6.63 E-07	1.56 E-06	5.38 E-08	3.53 E-07		1.38 E-06	6.34 E-07	1.01 E-06	5.76 E-07
Std		3.49 E-06	2.24 E-06	2.08 E-07	3.36 E-06		5.55 E-06	4.65 E-06	8.77 E-06	4.34 E-06
Best	Logarithmic spiral	8.42 E-08	2.53 E-07	7.39 E-09	2.74 E-09	Plum blossom curve	3.36 E-07	4.12 E-08	8.02 E-08	2.94 E-08
Mean		9.10 E-06	4.87 E-07	8.15 E-07	2.83 E-07		2.20 E-06	3.19 E-07	2.25 E-06	4.13 E-07
Std		5.76 E-06	1.05 E-06	3.44 E-06	1.29 E-06		8.92 E-06	1.64 E-06	9.38 E-06	1.40 E-06
Best	Lituus	8.64 E-08	4.85 E-08	1.76 E-08	2.62 E-09	Moppet curve	3.19 E-08	7.46 E-09	1.59 E-09	1.18 E-08
Mean		1.07 E-06	5.65 E-07	6.28 E-07	3.22 E-07		8.86 E-07	3.42 E-07	3.51 E-07	3.20 E-07
Std		6.22 E-06	2.45 E-06	6.39 E-06	3.67 E-06		5.36 E-06	1.79 E-06	8.33 E-07	8.80 E-07
Best	Hyperbolic spiral	2.39 E-08	1.09 E-08	2.88 E-08	2.00 E-10	Butterfly curve	4.06 E-08	3.28 E-09	3.12 E-08	1.25 E-09
Mean		1.88 E-06	1.62 E-06	2.42 E-06	4.97 E-07		4.00 E-07	4.06 E-07	4.21 E-07	5.45 E-07
Std		2.07 E-05	1.65 E-05	2.80 E-05	9.99 E-06		3.55 E-06	2.78 E-06	3.65 E-06	1.34 E-06

Optimal values are given in bold

Table 6 Curve approximation design effect diagram

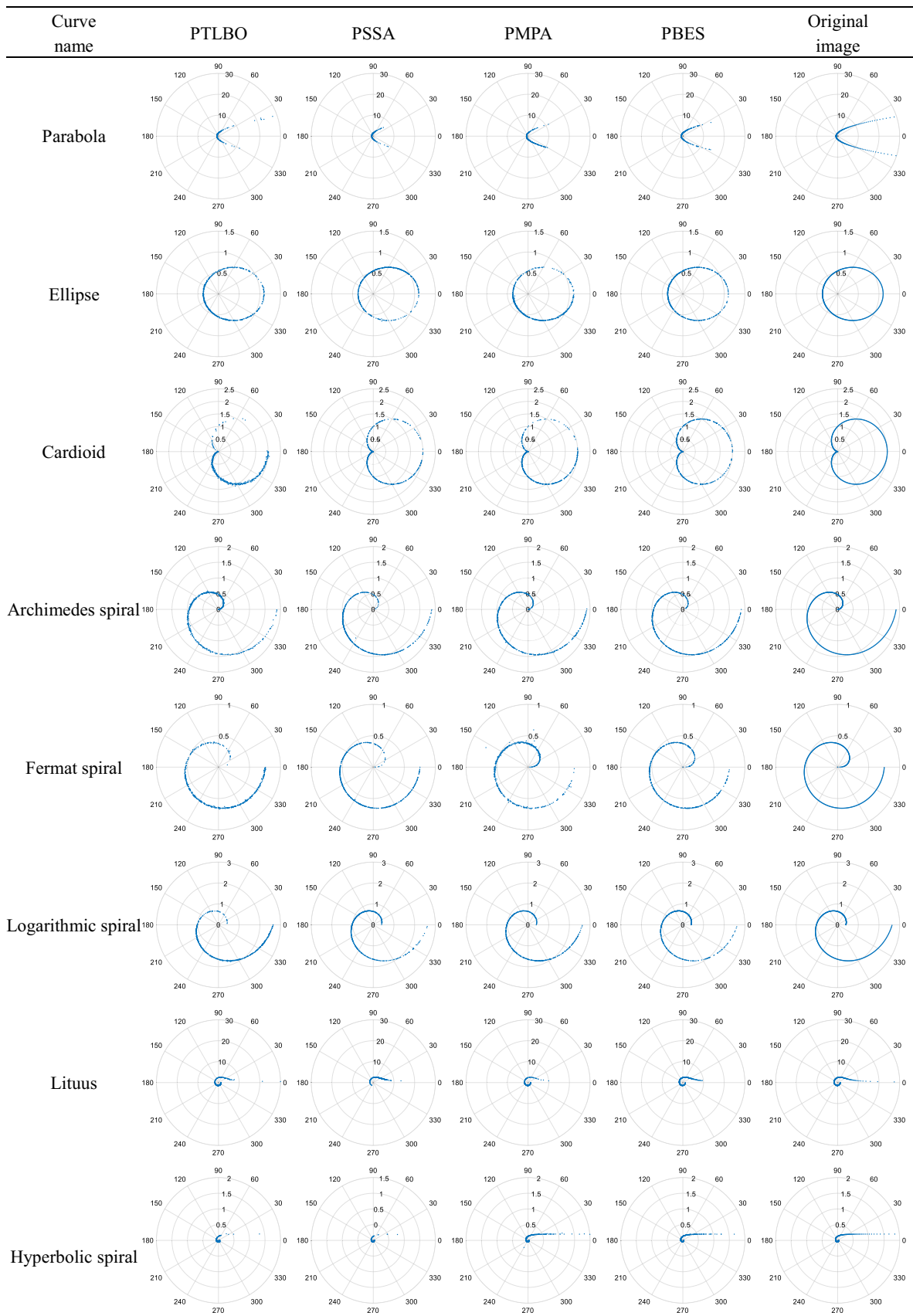
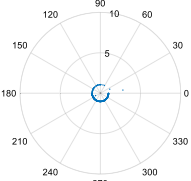
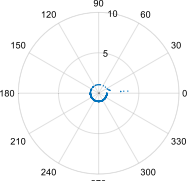
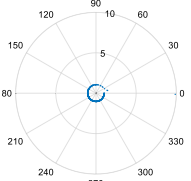
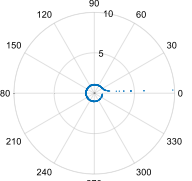
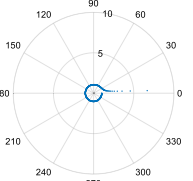
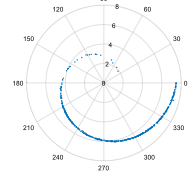
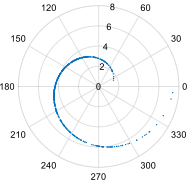
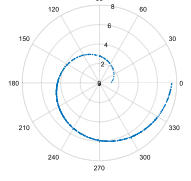
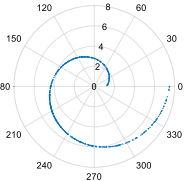
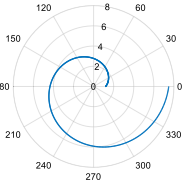
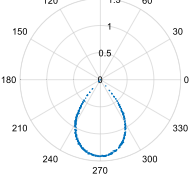
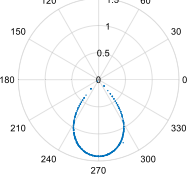
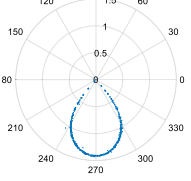
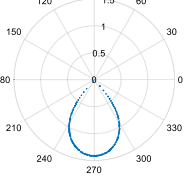
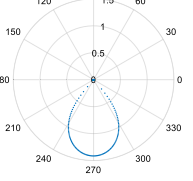
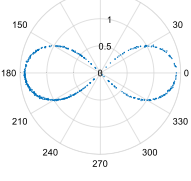
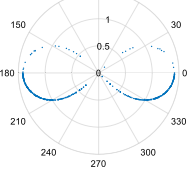
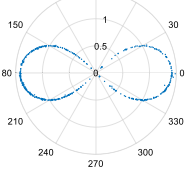
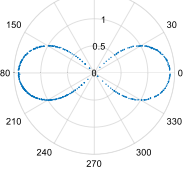
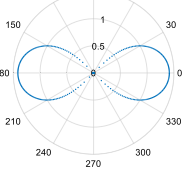
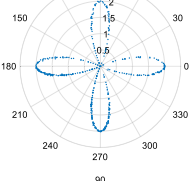
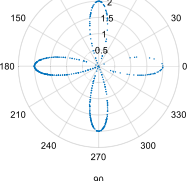
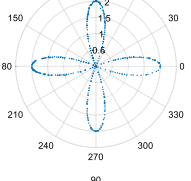
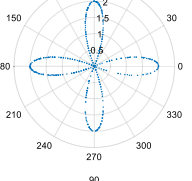
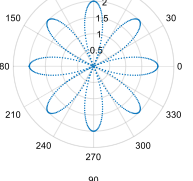
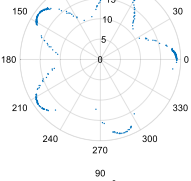
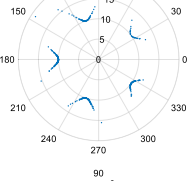
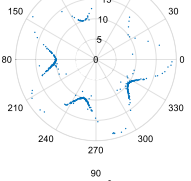
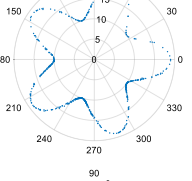
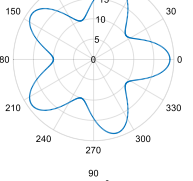
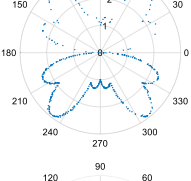
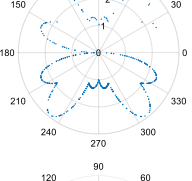
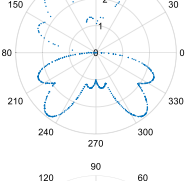
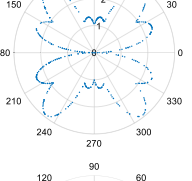
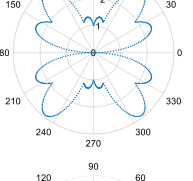
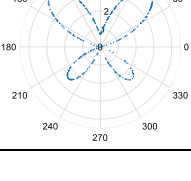
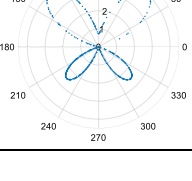
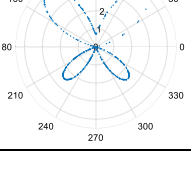
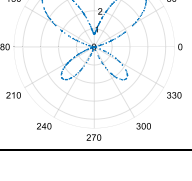
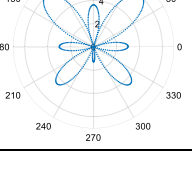


Table 6 (continued)

Curve name	PTLBO	PSSA	PMPA	PBES	Original image
KeNu spiral					
Round involute					
Water droplets curve					
Lemniscate					
Rose curve					
Plum blossom curve					
Moppet curve					
Butterfly curve					

PSSA and PMPA show poor performance. Furthermore, for some complex curves, such as Rose curve, Butterfly curve, the relative error difference is small, but PBES still performs better in absolute error.

Through testing the absolute error and relative error, it can be proved that PBES has better performance in curve approximation problem, and it can have strong stability while approximating the curve. In this section, by solving the curve approximation problem, it is proved that the PBES algorithm has excellent performance and strong exploration and exploitation capabilities. This is due to the introduction of polar coordinates and the adjustment of some parameters. In addition, polar angle and polar diameter are updated directly in polar coordinate space, which simplifies the position updating formula and reduces the calculation process of the algorithm.

4.3.5 Wilcoxon Rank Sum Test

To verify the effect of PBES on statistical test, we performed nonparametric Wilcoxon rank sum test analysis. On account of the heuristic algorithms have great contingency and randomness, similar statistical tests are needed to ensure the reliability and effectiveness of the experimental data. This will facilitate meaningful comparison between the proposed PBES and PTLBO, PSSA and PMPA. Generally, Wilcoxon's p -value test [72] is a method to verify whether there are significant differences between the two groups of data and it is widely used in statistical hypothesis testing. The smaller the p value, the greater the difference between the data. When $p < 0.05$, it indicates that there is a significant difference between the data considered within the 95% confidence interval.

Figure 7 shows the p -value comparison results among PBES and PTLBO, PSSA and PMPA. It records the comparison results of 16 curves between PBES and other three algorithms. It can be seen that the p value of each curve is much lower than 0.05, and all are below $10E-10$. This indicates that there are significant differences between PBES and other three algorithms in all curves. Especially for the Archimedes spiral, between PBES and PSSA, the p value is the smallest of the 16 curves and it reached $10E-46$, indicating that the difference between PBES and PSSA is the largest for the Archimedes spiral. For PMPA, the p value of Butterfly curve spiral is the highest, but it is much lower than 0.05. This indicates that there are significant differences in Butterfly curve. Therefore, the p -value test proved that in all curves there were significant differences between PBES and PTLBO, PSSA and PMPA. As well as, the performance of the PBES algorithm is more excellent than other algorithms.

4.3.6 Friedman Rank Test

Friedman's test is tested with SPSS software, and the following three parts are obtained: variables, average ranking size, and average ranking size [73]. The smaller the average ranking, the better the results obtained by the algorithm. We use Friedman rank test to further verify the performance of the PBES algorithm. Figure 8 presents the mean ranks obtained by Friedman rank test from the mean performances of the algorithms for each curve. The value obtained by PBES is the smallest, which proves that the PBES algorithm has better performance in solving curve approximation problems.

5 Inverse Kinematics Solution of a 7-DOF Robotic Manipulator

5.1 Background of the Inverse Kinematics

Inverse kinematics (IK) is a fundamental discipline of robotics and is applied to many practical problems; it plays a vital role in some scenarios such as trajectory planning, dynamics analysis, and motion control [74]. Inverse kinematics is used for solving the angle of each joint when the end position of the manipulator is known. The IK problem requires moving the end effector to the desired location, and our job is to move the end of the robotic manipulator to the desired location as precisely as possible and take the least time. When the joint arrangement of the manipulator meets Pieper's criterion: the adjacent three axes are parallel or intersect at one point, the IK problem has an analytical solution [74]. The equations of inverse kinematics are complex, highly nonlinear, and coupled, so it is difficult to solve them quickly and accurately by general methods [74]. For the IK problem, there are some traditional solutions including analytical algorithms and numerical iterative algorithms. It is difficult for traditional algorithms to obtain accurate solutions, and it is time-consuming, with unsatisfactory stability and real-time performance. In contrast, intelligent algorithms have great advantages. First, there is no need to establish a complex kinematic model. Second, the use of random or meta-heuristic search strategies can effectively solve the global optimization problem and obtain the optimal solution in a shorter time [74]. Owing to that the intelligent algorithms can solve the inverse kinematics problem effectively, it is necessary to use the intelligent algorithms, for example, the PBES algorithm.

5.2 Mathematical Model of the Robotic Manipulator

At present, the most popular ones in the field of robotics are the 6-DOF and 7-DOF robotic manipulators, which arouse the interest of related researchers and scholars [75]. This

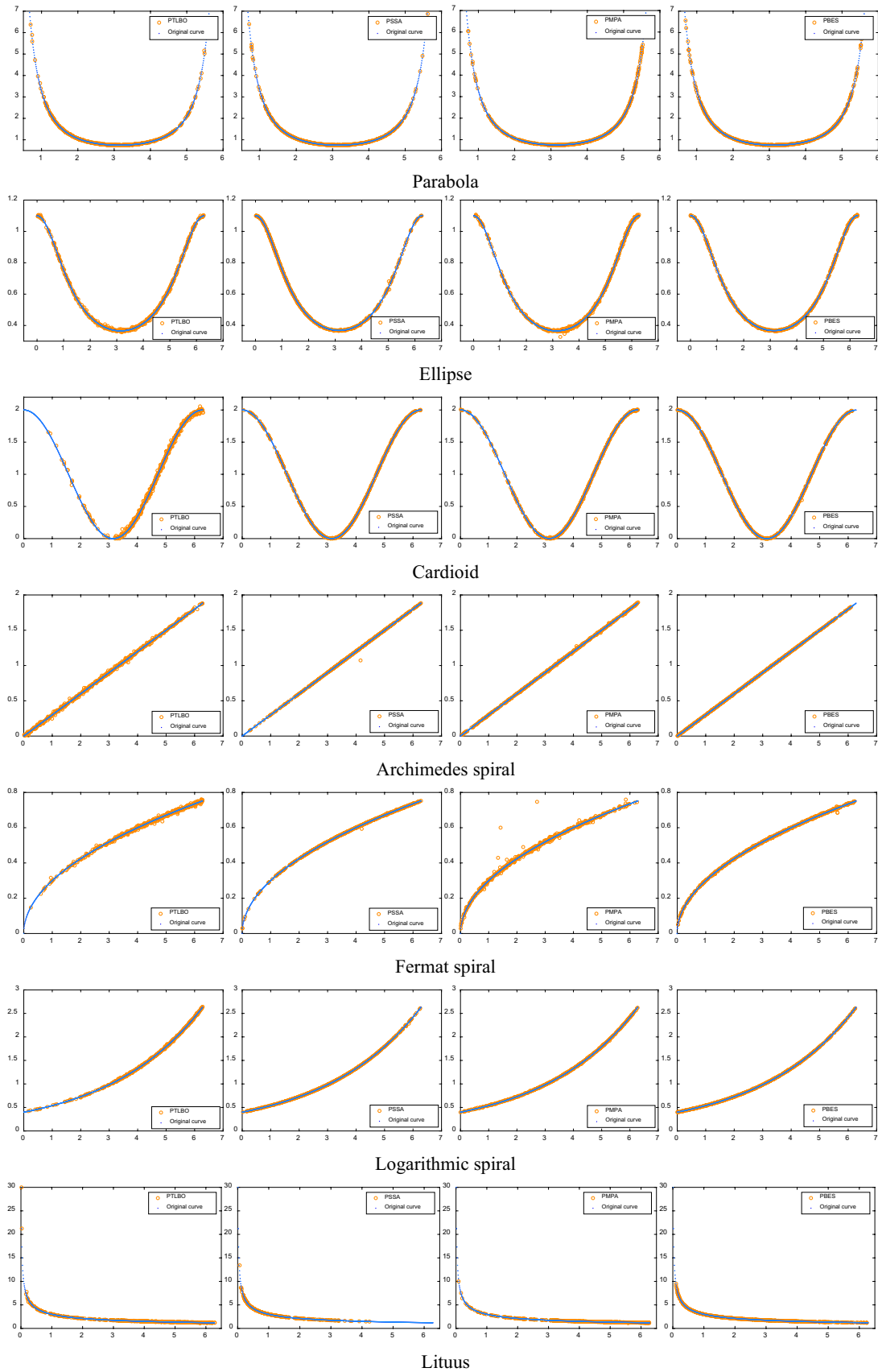


Fig. 5 Absolute error diagram of all curves

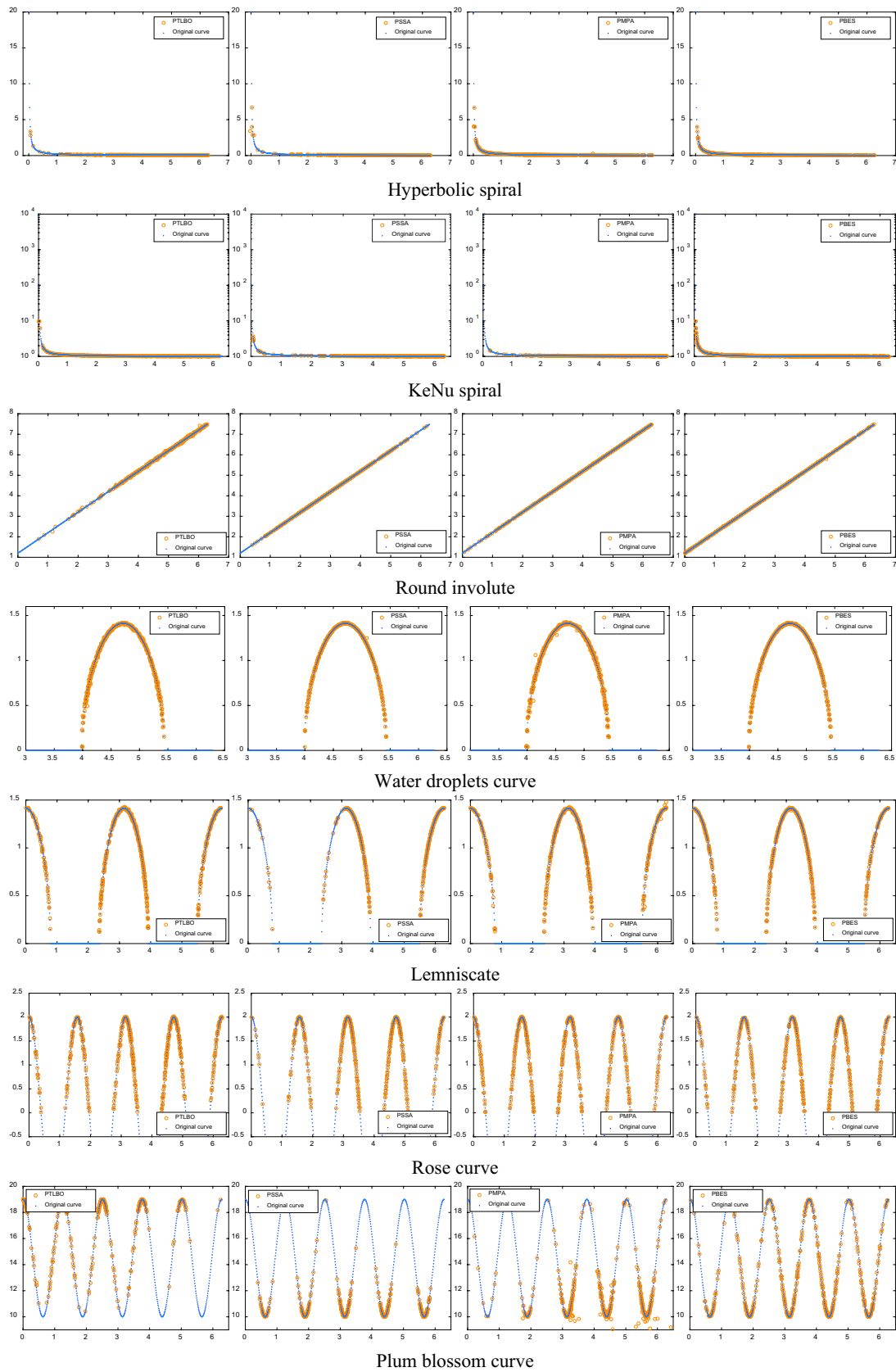


Fig. 5 (continued)

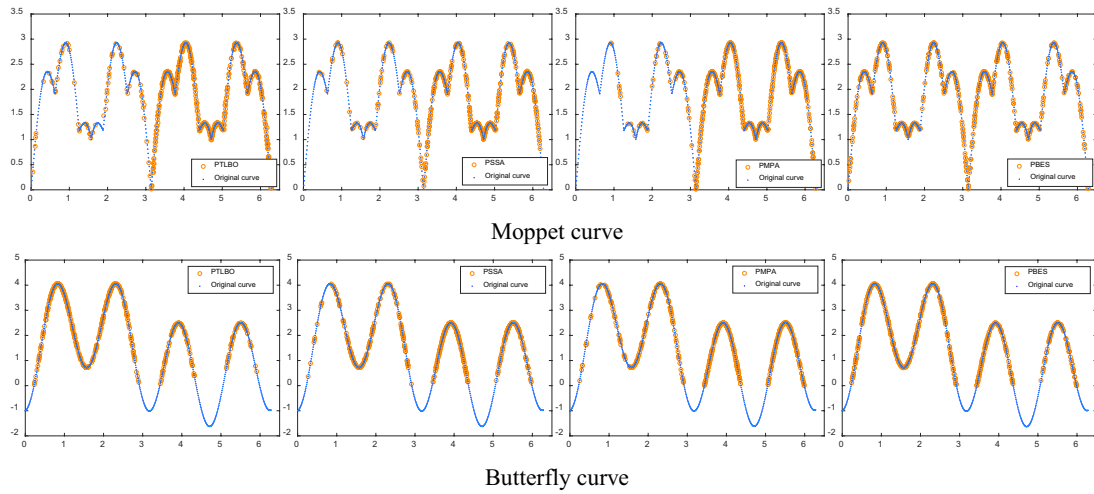


Fig. 5 (continued)

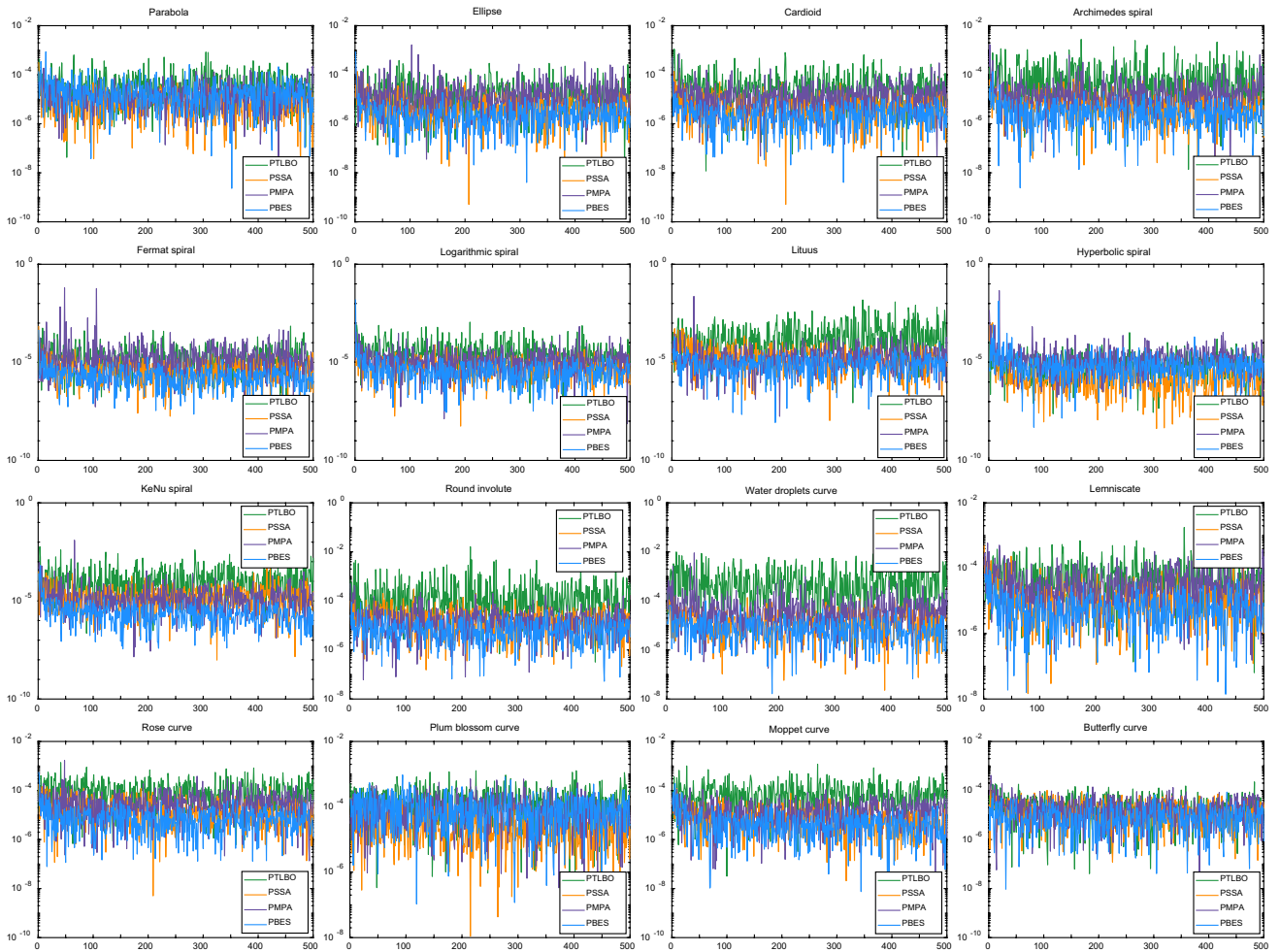


Fig. 6 Relative error diagram of all curves

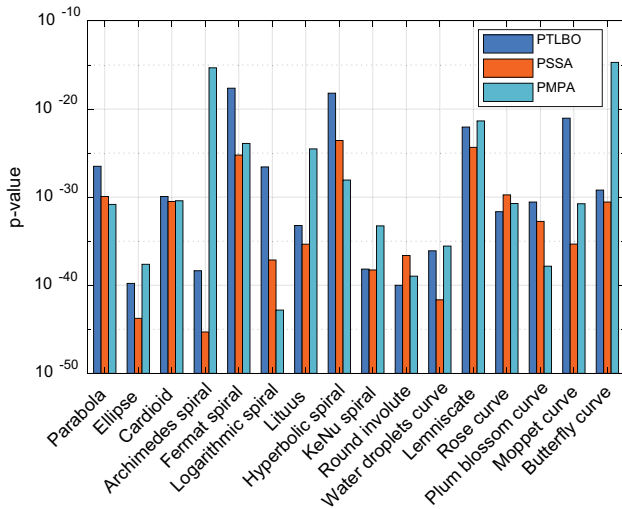


Fig. 7 The histogram of p -value

kind of manipulator has many characteristics, such as flexible movement, easy to avoid obstacles, large working space, and low latency. In general, the higher the degree of freedom of the manipulator, the more complex the structure, and these manipulators also have the problem of complex structure and difficult operation [76]. We use a 7-DOF robotic manipulator to conduct experimental simulations to verify the performance of the PBES algorithm. As shown in Fig. 9, the manipulator has seven revolute joints.

In this subsection, the PBES algorithm presented in this paper is applied to the computational simulation experiment of inverse kinematics is carried out on the 7-DOF robot

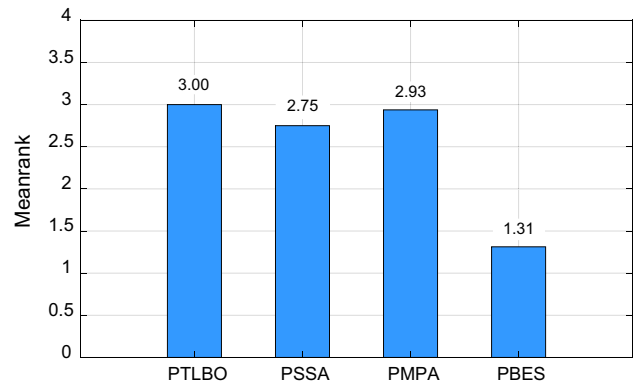


Fig. 8 Friedman rank test for the mean performances obtained

manipulator to verify its accuracy and efficiency. In the inverse kinematics problem, DH parameters [77] are indispensable in the inverse kinematics calculation and analysis process of 7-DOF robot manipulator. They use four parameters to represent the relationship between the two joints as shown in Table 7.

$${}^i_{i-1}T = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i & a_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

$$A_{\text{End - Effector}} = {}^7T = {}^0T \cdot {}^1T \cdot {}^2T \cdot {}^3T \cdot {}^4T \cdot {}^5T \cdot {}^6T \cdot {}^7T = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

$$\begin{aligned} {}^1_0T &= \begin{bmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^2_1T = \begin{bmatrix} c\theta_2 & 0 & s\theta_2 & l_2c\theta_2 \\ s\theta_2 & 0 & -c\theta_2 & l_2s\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^3_2T = \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 & l_3c\theta_3 \\ s\theta_3 & 0 & c\theta_3 & l_3s\theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\ {}^4_3T &= \begin{bmatrix} c\theta_4 & 0 & s\theta_4 & l_4c\theta_4 \\ s\theta_4 & 0 & -c\theta_4 & l_4s\theta_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^5_4T = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 & l_5c\theta_5 \\ s\theta_5 & 0 & c\theta_5 & l_5s\theta_5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\ {}^6_5T &= \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & l_6c\theta_6 \\ s\theta_6 & c\theta_6 & 0 & l_6s\theta_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^7_6T = \begin{bmatrix} c\theta_7 & -s\theta_7 & 0 & l_7c\theta_7 \\ s\theta_7 & c\theta_7 & 0 & l_7s\theta_7 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{23}$$

$$\begin{aligned}
 P_x &= (c\theta_1 c\theta_2 c\theta_3 c\theta_4 - s\theta_1 s\theta_3 c\theta_4 - c\theta_1 s\theta_2 s\theta_4) \\
 &\quad (c\theta_5 c\theta_6 l_7 c\theta_7 - c\theta_5 s\theta_6 l_7 s\theta_7 - s\theta_5 d_7 + c\theta_5 l_6 c\theta_6 + l_5 c\theta_5) \\
 &\quad (-c\theta_1 c\theta_2 s\theta_3 - s\theta_1 c\theta_3) (s\theta_5 c\theta_6 l_7 c\theta_7 - s\theta_5 s\theta_6 l_7 s\theta_7 + c\theta_5 d_7 + s\theta_5 c\theta_6 l_6 + l_5 s\theta_5) \\
 &\quad (c\theta_1 c\theta_2 c\theta_3 s\theta_4 s\theta_1 s\theta_3 s\theta_4 c\theta_1 c\theta_4 s\theta_2) (-s\theta_6 l_7 c\theta_7 - c\theta_6 l_7 s\theta_7 - l_6 s\theta_6) \\
 &\quad + c\theta_1 c\theta_2 (c\theta_3 c\theta_4 l_4 + l_3 c\theta_3) - s\theta_1 (s\theta_3 c\theta_4 l_4 + l_3 s\theta_3) - c\theta_1 s\theta_2 l_4 s\theta_4 + c\theta_1 c\theta_2 l_2 \\
 P_y &= \theta_1 c\theta_2 c\theta_3 c\theta_4 + c\theta_1 s\theta_3 c\theta_4 - s\theta_1 s\theta_2 s\theta_4 \\
 &\quad (c\theta_5 c\theta_6 l_7 c\theta_7 - c\theta_5 s\theta_6 l_7 s\theta_7 - s\theta_5 d_7 + c\theta_5 c\theta_6 l_6 + l_5 c\theta_5) \\
 &\quad + (-s\theta_1 c\theta_2 s\theta_3 + c\theta_1 c\theta_3) (s\theta_5 c\theta_6 l_7 c\theta_7 - s\theta_5 s\theta_6 l_7 s\theta_7 + c\theta_5 d_7 + s\theta_5 c\theta_6 l_6 + l_5 s\theta_5) \\
 &\quad + (s\theta_1 c\theta_2 c\theta_3 s\theta_4 + c\theta_1 s\theta_3 s\theta_4 + s\theta_1 s\theta_2 c\theta_4) \\
 &\quad (-s\theta_6 l_7 c\theta_7 - c\theta_6 l_7 s\theta_7 - l_6 s\theta_6) + s\theta_1 c\theta_2 (c\theta_3 c\theta_4 l_4 + l_3 c\theta_3) \\
 &\quad + c\theta_1 (s\theta_3 c\theta_4 l_4 + l_3 s\theta_3) - s\theta_1 s\theta_2 s\theta_4 l_4 + s\theta_1 c\theta_2 l_2 \\
 P_z &= (-s\theta_2 c\theta_3 c\theta_4 - c\theta_2 s\theta_4) (c\theta_5 c\theta_6 l_7 c\theta_7 - c\theta_5 s\theta_6 l_7 s\theta_7 - s\theta_5 d_7 + c\theta_5 c\theta_6 l_6 + l_5 c\theta_5) \\
 &\quad + s\theta_2 s\theta_3 (s\theta_5 c\theta_6 l_7 c\theta_7 - s\theta_5 s\theta_6 l_7 s\theta_7 + c\theta_5 d_7 + s\theta_5 c\theta_6 l_6 + s\theta_5 l_5) \\
 &\quad + (-s\theta_2 c\theta_3 s\theta_4 + c\theta_2 c\theta_4) (-s\theta_6 l_7 c\theta_7 - c\theta_6 l_7 s\theta_7 - s\theta_6 l_6) \\
 &\quad - s\theta_2 (c\theta_3 c\theta_4 l_4 + l_3 c\theta_3) - c\theta_2 s\theta_4 l_4 - s\theta_2 l_2 + l_1
 \end{aligned} \tag{24}$$

where T_{i-1}^i is the transformation matrix from joint $i - 1$ to joint i as found by using Eq. (21) and the position of the robot manipulator end controller can be calculated using formula in Eq. (22). In Eq. (22) P_x, P_y, P_z denote the specific components of the position vector of the end-effector. $n_x, n_y, n_z, s_x, s_y, s_z, a_x, a_y, a_z$, represents the rotation components of the transformation matrix $A_{\text{End-Effector}}$ represents the homogeneous transformation matrix of the end-effector. The transformation matrices Eq. (23) of the 7-DOF

robot manipulator are continuously multiplied to obtain $A_{\text{End-Effector}}$. s represents the sine function, and c is the cosine function.

5.3 Fitness Function for 7-Dof Robot Manipulator

The FK formula is used to calculate the position of joint angle and the corresponding pose. In the experiment, the position of the robotic manipulator is obtained by using PBES, that is, the predicted position. By using PBES, making the predicted position is as close as possible to the desired position. For a desired position, the mathematical formula of the fitness function is as follows:

$$\text{Error} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{25}$$

Suppose $P_1 = (x_1, y_1, z_1)$ is the predicted point calculated by PBES, $P_2 = (x_2, y_2, z_2)$ is the desired point derived from forward kinematics formula. The position error is the Euclidean distance between P_1 and P_2 , which is obtained by Euclidean equation (Eq. 25). The smaller of the position error, the algorithm's effect will be better. Our goal is to obtain each joint angle and optimize the position error under the condition of minimum time-consuming.

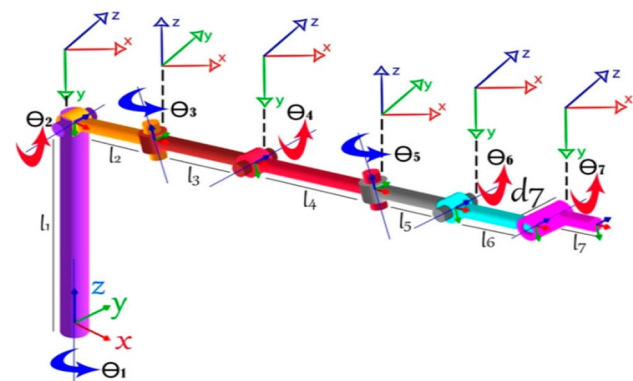


Fig. 9 The 7-DOF robotic manipulator [76]

Table 7 DH parameters for 7-DOF robot manipulator [78]

i	$a_i(\text{m})$	$\alpha_i(^{\circ})$	$d_i(\text{m})$	$\theta_i(^{\circ})$ range
1	0	-90	$l_1 = 0.5$	$-180 < \theta_1 < 180$
2	$l_2 = 0.2$	90	0	$-90 < \theta_2 < 30$
3	$l_3 = 0.25$	-90	0	$-90 < \theta_3 < 120$
4	$l_4 = 0.3$	90	0	$-90 < \theta_4 < 90$
5	$l_5 = 0.2$	-90	0	$-90 < \theta_5 < 90$
6	$l_6 = 0.2$	0	0	$-90 < \theta_6 < 90$
7	$l_7 = 0.1$	0	$d_7 = 0.05$	$-30 < \theta_7 < 90$

5.4 Simulation Results and Analysis

In this experiment, PBES algorithm is compared with other four intelligent algorithms: PSO, ABC, FA and PSSA, by solving the inverse kinematics problem. Each algorithm runs independently for 10 times with 150 individual populations and 500 iterations. The comparison results of end effector position points and position errors are shown in Table 8. PSO, ABC, FA and PSSA have been produced effective

Table 8 Comparison of PBES with positions of end effector

	Target position	PSO [78]	ABC [78]	FA [78]	PSSA	PBES
P_x (cm)	-24.7487	-24.6665	-24.7149	-24.7402	-24.7488	-24.7487
P_y (cm)	100.9619	100.3896	101.0105	100.9723	100.9619	100.9620
P_z (cm)	50	49.6557	50.0104	49.9989	49.9999	50
Error (cm)	NA	6.72E-03	5.45E-04	6.53E-05	1.01E-07	3.64E-10

Optimal values are given in bold

results in the IK problem, but PBES produced better results in the positions of end effector and error.

As shown in Table 9, the population size of PSO, ABC and FA are 300, 100 and 50, respectively, the population size of PSSA and PBES is set to 150. The maximum number of iterations of each algorithm is 500. Solution time is also a standard factor for comparison of robotic manipulator. It is critical for robotic manipulator. Only when the manipulator reaches the target position in the shortest time, can the algorithm have good competitiveness in solving the IK problem. In Fig. 10, they were all able to reach their positions of end effector in a relatively short time. But, compared with PBES, the performance of PSO, ABC and FA is not very good. There are still differences between PBES and PSSA, PBES took less time than PSSA, whereas, under the same conditions, it took the least time for PBES to reach the target position among these algorithms.

In fact, each algorithm has a certain accuracy and short solution time in solving the IK problem, but the PBES performs best. And the desired point is almost the same as the target position by using PBES. Compared with PSO, ABC, FA and PSSA, its position error is several orders of magnitude smaller. In addition, the PBES took the shortest solution time in solving the IK problem. From the above results, it is proved that the PBES algorithm can be applied to the IK problem of a 7-DOF robotic manipulator successfully.

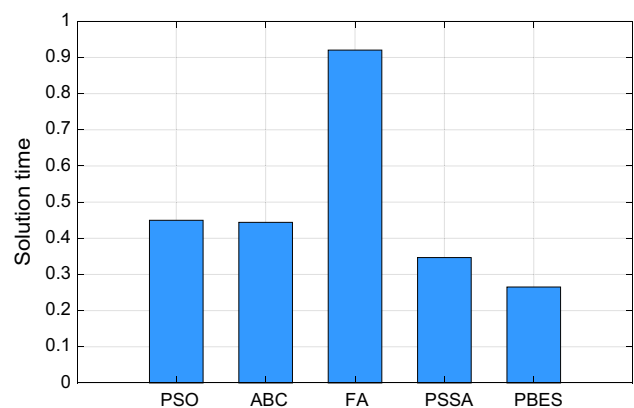
6 Conclusions and Future Work

This paper presents a bio-inspired PBES algorithm to solve the different problems of polar coordinate. Firstly, the polar transcendental equations of four different types of polar coordinates are tested. PBES is superior to PSSA in both the convergence speed and accuracy, which reflects the excellent performance of the PBES algorithm. Then, PBES is applied to the curve approximation design problem. The results show that PBES can approach the curve well and the approximation effect is satisfactory. It is superior to PSSA in both precision and stability of curve design problem. In addition, the error of PBES is kept small while good effect is achieved. Furthermore, Wilcoxon rank sum test was performed on PBES, the statistical results showed that PBES and PSSA were significantly different, besides, PBES has

Table 9 Comparison of PBES with other techniques

	Particle size	Max_iteration	Solution time (s)
PSO [78]	300	500	0.4498
ABC [78]	100	500	0.4441
FA [78]	50	500	0.9204
PSSA	150	500	0.3467
PBES	150	500	0.2653

Optimal values are given in bold

**Fig. 10** The histogram of solution time

better performance. Finally, PBES is used to solve the robotic manipulator problem. Compared with other algorithms, PBES performs best and has strong competitiveness. At present, PBES has only been applied in polar transcendental equations, curve approximation inverse kinematic manipulator problems. In the future, PBES can be used to solve some more complex curve approximation problems. Of course, we can also try to use the PBES algorithm for target detection [79], face recognition [80] and the current popular auto-driving [81] problems. In addition, we can establish different polar coordinate mathematical models according to different practical application problems, improve PBES

more effectively, and make PBES more widely used in polar coordinate problems.

Acknowledgements The authors express great thanks to the financial support from the National Natural Science Foundation of China.

Author Contributions YZ carried out the PBES algorithm studies, participated in the drafted the manuscript. YZ carried out the review & editing; GZ carried out the original draft and review & editing; QL carried out the design PBES algorithm model; BZ carried out the results analysis. All authors read and approved the final manuscript.

Funding This research is funded by the National Natural Science Foundation of China, Grant number U21A20464, 62066005; and Supported by Program for Young Innovative Research Team in China University of Political Science and Law, under Grant No. 21CXTD02, and Project of the Guangxi Science and Technology under Grant No. AD21196006.

Availability of Data and Material Not applicable.

Declarations

Conflict of interest The authors declare no conflicts of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for Publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Grossman, S.I.: Polar coordinates. In: *Calculus of One Variable*, vol. 12, 2nd edn., pp. 608–640. Elsevier, Cambridge (1984)
- Azzam, R.M.A.: Polar curves for transmission ellipsometry. *Opt. Commun.* **14**(1), 145–147 (1975)
- Moore, K.C., Paoletti, T., Musgrave, S.: Complexities in students' construction of the polar coordinate system. *J. Math. Behav.* **36**, 135–149 (2014)
- Rong, J., Wen, L., Xiao, J.: Efficiency improvement of the polar coordinate transformation for evaluating BEM singular integrals on curved elements. *Eng. Anal. Bound. Elem.* **38**, 83–93 (2014)
- Mohseni, K., Colonius, T.: Numerical treatment of polar coordinate singularities. *J. Comput Phys* **157**(2), 787–795 (2000)
- Ahmad, T., Hu, J., Wang, S.: Pair-polar coordinate-based cancelable fingerprint templates. *Pattern Recogn.* **44**(10), 2555–2564 (2011)
- Lei, X., et al.: Roundness error evaluation algorithm based on polar coordinate transform. *Measurement* **44**(2), 345–350 (2011)
- Kennedy, J.: Swarm intelligence. In: Zomaya, A.Y. (ed.) *Handbook of Nature-Inspired and Innovative Computing*, pp. 187–219. Springer, Berlin (2006)
- Hashim, F.A., Hussain, K., Houssein, E.H., et al.: Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **51**, 1531–1551 (2020)
- Neumann, F., Witt, C.: Combinatorial optimization and computational complexity. In: *Bioinspired Computation in Combinatorial Optimization*. Natural Computing Series. Springer, Berlin (2010)
- Stanovov, V., Akhmedova, S., Semenkin, E.: LSHADE Algorithm with Rank-Based Selective Pressure Strategy for Solving CEC2017 Benchmark Problems. *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Rio de Janeiro (2018)
- Galletly, J.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. *Kybernetes* **27**(8), 979–980 (1998)
- Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems. In: Dario, P., Sandini, G., Aebischer, P. (eds.) *Robots and Biological Systems: Towards a New Bionics? NATO ASI Series (Series F: Computer and Systems Sciences)*, vol. 102. Springer, Berlin (1993)
- Webster, B., Bernhard, P. J.: A local search optimization algorithm based on natural principles of gravitation. In: *Proceedings of the International Conference on Information and Knowledge Engineering*. IKE'03, June 23–26, Las Vegas, Nevada, USA, Vol 1 (2003).
- Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. *Mach. Learn.* **3**, 95–99 (1988)
- Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**, 341–359 (1997)
- Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **3**, 82–102 (1999)
- Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. MHS'95. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE, pp. 39–43 (2002)
- Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2006)
- Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *J. Glob. Optim.* **39**, 459–471 (2007)
- Yang, X.S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) *Stochastic Algorithms: Foundations and Applications*. Lecture Notes in Computer Science, pp. 169–178. Springer, Berlin (2009)
- Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) *Nature Inspired Cooperative Strategies for Optimization, NICO 2010*, pp. 65–74. Springer, Berlin (2010)
- Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **29**, 17–35 (2013)
- Mirjalili, S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **27**, 1053–1073 (2016)
- Mirjalili, S.: Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **89**, 228–249 (2015)
- Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114**, 163–191 (2017)

27. Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H.: Marine Predators algorithm: a nature-inspired metaheuristic. *Expert Syst. Appl.* **152**, 113377 (2020)
28. Li, S., Chen, H., Wang, M., et al.: Slime mould algorithm: A new method for stochastic optimization. *Futur. Gener. Comput. Syst.* **111**(1), 300–323 (2020)
29. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**, 2232–2248 (2009)
30. Anita, Y.A.: AEFA: Artificial electric field algorithm for global optimization. *Swarm Evol. Comput.* **48**, 93–108 (2019)
31. Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S.: Equilibrium optimizer: A novel optimization algorithm. *Knowl. Based Syst.* **191**, 105190 (2020)
32. Alatas, B.: ACROA: artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* **38**, 13170–13180 (2011)
33. Hashim, F.A., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W., Mirjalili, S.: Henry gas solubility optimization: a novel physics-based algorithm. *Future Gener. Comput. Syst.* **101**, 646–667 (2019)
34. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **96**, 120–133 (2016)
35. Kang, S.L., Zong, W.G.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **194**(36/38), 3902–3933 (2005)
36. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–Learning–Based Optimization: an optimization method for continuous non-linear large scale problems. *Inf. Sci.* **183**(1), 1–15 (2012)
37. Loucera, C., Gálvez, A., Iglesias, A.: Simulated annealing algorithm for Bezier curve approximation. In *International Conference on Cyberworlds*, pp. 182–189 (2014)
38. Li, W., Xu, S., Zhao, G., Goh, L.: Adaptive knot placement in B-spline curve approximation. *Comput. Aided Des.* **37**(8), 791–797 (2005)
39. Laube, P., Franz, M., Umlauf, G.: Deep Learning parametrization for B-Spline curve approximation. In: *International Conference on 3D Vision (3DV)*, pp. 691–699 (2018)
40. Bureick, J., Alkhatib, H., Neumann, I.: Fast converging elitist genetic algorithm for knot adjustment in B-spline curve approximation. *J. Appl. Geodesy* **13**(4), 317–328 (2019)
41. Alsattar, H.A., Zaidan, A.A., Zaidan, B.B.: Novel meta-heuristic bald eagle search optimisation algorithm. *Artif. Intell. Rev.* **53**, 2237–2264 (2020)
42. Jia, H.M., Jiang, Z.C., Li, Y.: Simultaneous feature selection optimization based on improved bald eagle search algorithm. *Control Dec.* (2021). <https://doi.org/10.13195/j.kzyjc.2020.1025>
43. Angayarkanni, S.A., Sivakumar, R., Ramana Rao, Y.V.: Hybrid Grey Wolf: Bald Eagle search optimized support vector regression for traffic flow forecasting. *J. Ambient Intell. Hum. Comput.* **12**, 1293–1304 (2021)
44. Ramadan, A., Kamel, S., Hassan, M.H., Khurshaid, T., Rahmann, C.: An improved bald eagle search algorithm for parameter estimation of different photovoltaic models. *Processes* **9**, 1127 (2021)
45. Karthik, A.G., Saravanakumar, R., Vijayakumar, P.: Bald eagle search optimization on dual fueled reactivity controlled combustion ignition based engine characteristics by altering low reactive fuels. *Environ. Prog. Sustain. Energy* **40**(6), e13683 (2021)
46. Sarkar, M., Kumar, A., Maji, B.: PAPR reduction using twin symbol hybrid optimization-based PTS and multi-chaotic-DFT sequence-based encryption in CP-OFDM system. *Photon Netw. Commun.* **41**, 148–162 (2021)
47. Zhou, J., Wang, Q.: Forecasting carbon price with secondary decomposition algorithm and optimized extreme learning machine. *Sustainability* **13**, 1–17 (2021)
48. Palanivel, V., Govindasamy, K., Arunachalam, G.K.: Optimization and prediction of pulsating heat pipe compound parabolic solar collector performances by hybrid deep belief network based bald eagle search optimizer. *Environ. Prog. Sustain. Energy* (2021). <https://doi.org/10.1002/ep.13740>
49. Kang, Z., Ren, F., Zhang, H., Lu, X., Li, Q.: Diagnosis method of transformer winding fault based on bald eagle search optimizing support vector machines. In: *IEEE 4th International Electrical and Energy Conference (CIEEC)*, pp 1–5 (2021)
50. Sayed, G.I., Soliman, M.M., Hassanien, A.E.: A novel melanoma prediction model for imbalanced data using optimized squeeze net by bald eagle search optimization. *Comput. Biol. Med.* **136**, 104712 (2021)
51. Matthysen, W., Engelbrecht, A.P.: A polar coordinate particle swarm optimiser. *Appl. Soft Comput.* **11**(1), 1322–1399 (2010)
52. Sierra, A., Echeverra, A.: The polar evolution strategy. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2301–2306 (2006)
53. Xiang, Z., Zhou, Y., Luo, Q., et al.: PSSA: polar coordinate salp swarm algorithm for curve design problems. *Neural Process Lett.* **52**, 615–645 (2020)
54. Liu, S., Zhang, X., Sun, D., Lin, M., Wen, S.: New-style Particle Swarm Optimization Algorithm based on polar coordination system and complex number operation. *J. Syst. Simul.* **18**, 1794–1798 (2006)
55. Pang, S., Zou, H., Su, Q.: Improved multi-objective particle swarm optimization in polar coordinates. *J. Syst. Simul.* **24**, 1576–1581 (2012)
56. Liu, J., Wei, J.: An improved NSGA-II algorithm based on polar coordinate. *Transformation* **29**(2), 34–37 (2016)
57. Al-Sayyed, R.M.H., Fakhouri, H.N., Rodan, A., Pattinson, C.: Polar particle swarm algorithm for solving cloud data migration optimization problem. *Mod. Appl. Sci.* **11**, 98–117 (2017)
58. Wei, S., Sun, Y., Jia, X.: Improved artificial fish swarm algorithm based on polar coordinate coding. *Syst. Eng. Electron.* **32**, 2248–2251 (2010)
59. Wu, Y., Zhou, Y.: A drosophila optimization algorithm based on polar coding. *Math. Pract. Cognit.* **47**, 161–170 (2017)
60. Lin, C.W., Hong, S.: High-order histogram-based local clustering patterns in polar coordinate for facial recognition and retrieval. *Vis. Comput.* (2021). <https://doi.org/10.1007/s00371-021-02102-9>
61. Shatnawi, M., Nasrudin, M.F., Sahran, S.: A new initialization technique in polar coordinates for particle swarm optimization and polar PSO. *Int. J. Adv. Sci. Eng. Inf. Technol.* **7**, 242–249 (2017)
62. Thota, S., Srivastav, V.K.: Quadratically convergent algorithm for computing real root of non-linear transcendental equations. *BMC Res. Notes* **11**, 909 (2018)
63. Srivastav, V.K., Thota, S., Kumar, M.: A new trigonometrical algorithm for computing real root of non-linear transcendental equations. *Int. J. Appl. Comput. Math.* **5**, 44 (2019)
64. Böhm, W., Farin, G., Kahmann, J.: A survey of curve and surface methods in CAGD. *Comput. Aided Geom. Des.* **1**, 1–60 (1984)
65. Kreeger, K.: The learning curve. *Nat. Biotechnol.* **21**, 951–952 (2003)
66. Cobb, J.R.: The problem of the primary curve. *J. Bone Jt. Surg.* **42**(8), 1413–1425 (1960)
67. Levy, E.C.: Complex-curve fitting. *IRE Trans. Autom. Control.* **4**(1), 37–43 (1959)
68. Wolfson, H.J.: On curve matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(5), 483–489 (1990)
69. Holladay, J.C.: A smoothest curve approximation. *Math. Tables Other Aids Comput.* **11**(60), 233–243 (1957)
70. Farah, J.R., Pesce, D.W., Johnson, M.D., et al.: On the approximation of the black hole shadow with a simple polar curve. *ApJ* **900**, 77 (2020)

71. Ueda, E.K., Sato, A.K., Martins, T.C., et al.: Curve approximation by adaptive neighborhood simulated annealing and piecewise Bézier curves. *Soft Comput.* **24**, 18821–18839 (2020)
72. Zimmerman, D.W., Zumbo, B.D.: Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks. *J. Exp. Educ.* **62**, 75–86 (1993)
73. Tang, C., Zhou, Y., Tang, Z.: Teaching-learning-based pathfinder algorithm for function and engineering optimization problems. *Appl. Intell.* **51**, 5040–5066 (2021)
74. Liu, F., Huang, H., Li, B., et al.: A parallel learning particle swarm optimizer for inverse kinematics of robotic manipulator. *Int. J. Intell. Syst.* **36**(10), 6101–6132 (2021)
75. Dereli, S., Köker, R., Öylek, İ., Ay, M.: A comprehensive research on the use of swarm algorithms in the inverse kinematics solution. *Politeknik Dergisi* **22**(1), 75–79 (2019)
76. Dereli, S., Köker, R.: A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm. *Artif. Intell. Rev.* **53**, 949–964 (2020)
77. Sun, J., Cao, G., Li, W., Liang, Y., Huang, S.: Analytical inverse kinematic solution using the D–H method for a 6-DOF Robot. In: *International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 714–716 (2017)
78. Dereli, S., Köker, R.: Calculation of the inverse kinematics solution of the 7-DOF redundant robot manipulator by the firefly algorithm and statistical analysis of the results in terms of speed and accuracy. *Inverse Probl. Sci. Eng.* **28**(5), 601–613 (2020)
79. Lei, Y.: Research on microvideo character perception and recognition based on target detection technology. *J. Comput. Cognit. Eng.* **1**(1), 1–5 (2022)
80. Yang, Y., Song, X.: Research on face intelligent perception technology integrating deep learning under different illumination intensities. *J. Comput. Cognit. Eng.* **1**(1), 32–36 (2022)
81. Yang, M.: Research on vehicle automatic driving target perception technology based on improved MSRPN algorithm. *J. Comput. Cognit. Eng.* (2022). <https://doi.org/10.47852/bonviewJCC E20514>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.