Current Science
CS 4 (5), 754– 760 (2024)
Received 25 **July** 2024 | **Revised 15 August** 2024 | **Accepted 20** Sept. 2024 | **Online Available 21 October** 2024
https://doi.org/10.52845/CS/2024-4-5-9

Current Science
Current Science

OPEN ACCESS

ISSN(O)2795-8639

ORIGINAL ARTICLE

Improved Particle Swarm Optimization using the Cauchy Criterion

Fangyong Chen[1], Fangjian Chen[2], Linchuan Zhang[3], Liping Tan[4*], Hongyan Wang[5*], Xiusheng Xue[6], Yaqi Chen[3], Yunteng Ma[3]

[1]Weichai Lovol Intelligent Agricultural Technology Co., Ltd., Weifang, 261206, China

[2]CATARC Automotive Research and Inspection Center (Tianjin) Co., Ltd., Tianjin 300300, China

[3]Weichai Power Co., Ltd., Weifang 261061, China

[4]Ulsan ship and ocean college ludong university, Yantan, China

[5]College of Electromechanical Engineering, Qingdao University of Science & Technology, Qingdao 266061, China

[6]Shandong Transportation Institute, Shandong Jinan 250031, China

Corresponding Author: Liping Tan, Hongyan Wang

Abstract:
Particle Swarm Optimization (PSO) algorithm is a popular metaheuristic algorithm inspired by the social behavior of bird flocking. Despite its effectiveness, PSO algorithm suffers from premature convergence and limited exploration capability in high-dimensional search spaces. To address these issues, researchers have proposed various enhancements to the standard PSO algorithm. One such enhancement is the utilization of the Cauchy criterion, which introduces heavy-tailed random movements into the particle updates. This review paper provides an overview of the Cauchy-based enhancements in PSO algorithm, highlighting their advantages, implementation strategies, and applications across diverse optimization problems.
Keywords: Metaheuristic algorithm, Particle swarm optimization, Cauchy criterion

Introduction

Particle Swarm Optimization (PSO) is a population-based optimization algorithm that simulates the social behavior of bird flocking. PSO has been widely recognized as an effective technique for various optimization problems due to its simplicity and efficiency [1]. In the field of gene selection, Han et al. (2019) proposed a hybrid gene selection method based on gene scoring strategy and improved PSO, which demonstrated high prediction accuracy for classifiers like extreme learning machine and support vector machine [1]. Similarly, in the context of scientific workflow scheduling in cloud computing, Saeedi et al. (2020) introduced an Improved Many-objective PSO algorithm to address conflicting objectives such as reliability maximization and cost, and energy consumption minimization [2]. Moreover, the application of PSO has been extended to different domains such as quantum computing. Zhang et al. (2020) discussed the use of Quantum-behaved PSO (QPSO) with generalized space transformation search, highlighting its simplicity and effectiveness in numerical optimization [3]. Additionally, Famila et al. (2020) presented an Improved Artificial Bee Colony Optimization-based Clustering Technique (IABCOCT) utilizing the Cauchy operator for optimal clustering in wireless sensor networks [4]. Furthermore, Sun et al. (2020) focused on carbon price prediction using empirical mode decomposition and least squares support vector machine optimized by Improved PSO, emphasizing the importance of understanding and predicting carbon market mechanisms [5]. Li et al. (2020) proposed an adaptive mutation PSO optimized support vector regression combined with improved AdaBoost.RT algorithm for temperature compensation of piezo-resistive pressure sensors, demonstrating its applicability and efficiency for industrial applications [6]. In the realm of swarm intelligence, Ali et al. (2021) introduced a hybrid strategy for multi-unmanned aerial vehicle swarm formation control using PSO with Cauchy mutant operators, aiming to enhance the fitness function of the formation [8]. Additionally, Allioui et al. (2021) integrated particle swarm optimization into a multi-agent system control protocol for medical image segmentation, emphasizing the importance of optimized control for better decision-making and processing quality under medical restrictions [9].

Overall, the literature review showcases the diverse applications and enhancements of PSO in various optimization problems, ranging from gene selection and workflow scheduling to carbon price prediction and swarm formation control [10-13]. The integration of PSO with different algorithms and techniques, such as the Cauchy operator, quantum computing, and adaptive mutation, highlights the continuous efforts to improve the efficiency and effectiveness of optimization algorithms in different domains.

In this study, to address the shortcomings of the standard PSO algorithm which is easy to fall into the local optimum, the Cauchy criterion is introduced to obtain the improved PSO algorithm, and the two PSO algorithms are tested separately by using eight commonly used test functions. The results show that the improved PSO algorithm achieves better control effect on the rectification of PID parameters.

2. Standard particle swarm optimization algorithm

The PSO algorithm considers each individual of the biota as a particle, and the biota is not sure of the specific location of the food in the foraging process, but can perceive the location relationship with the food, and the particles keep approaching the food through their own memory and mutual communication until they find the food, and the biota foraging process is a gradually converging process [14-17]. The PSO algorithm can be described in mathematical language as follows: in the D-dimensional space, there are m particles randomly distributed, and the population they form can be denoted as X (x_1, x_2,…, x_m,), the position of the i-th particle at time of k can be expressed as a D-dimensional vector, which is $X_i^k=(x_{i1}^k,x_{i2}^k,\cdots,x_{iD}^k)$[18-22]. The corresponding velocity is expressed as $V_i^k=(v_{i1}^k,v_{i2}^k,\cdots,v_{iD}^k)$. The position of each particle can be regarded as a candidate solution to the objective problem, and the merit of the candidate solution is evaluated by the fitness function. The best position that a single particle swarm can find at moment k is $P_i^k=(p_{i1}^k,p_{i2}^k,\cdots,p_{iD}^k)$, and the best position of the whole population at moment k is $P_g^k=\left(p_{g1}^k,p_{g2}^k,\cdots,p_{gD}^k\right)$, and according to the above rules, the velocity and position of PSO are updated as shown in Eq. (1) and (2).

$$v_{id}^{k+1}=v_{id}^k+c_1\cdot r_1\left(p_{id}^k-x_{id}^k\right)+c_2\cdot r_2\left(p_{gd}^k-x_{gd}^k\right) \tag{1}$$

$$x_{id}^{k+1}=x_{id}^k+v_{id}^{k+1} \tag{2}$$

in which, c_1 is the individual learning factor, which is the coefficient of the particle tracking individual historical optimum, and c_2 is the social learning factor, which indicates the empirical learning of the particle to the particle population. c_1 and c_2 are usually taken as positive real numbers between [0,2]; r_1 and r_2 are random numbers in the range of [0,1]; in order to prevent the particle's speed from going too fast or its position from crossing the boundary, the speed and position of the particle are constrained by setting $x_{id}^k\in[L_d, U_d]$, $v_{id}^k\in[v_{min}, v_{max}]$.

Position update formula remains unchanged and the velocity update formula is modified shown in Eq. (3).

$$v_{id}^{k+1}=\omega v_{id}^k+c_1\cdot r_1\left(p_{id}^k-x_{id}^k\right)+c_2\cdot r_2\left(p_{gd}^k-x_{gd}^k\right) \tag{3}$$

From the Eq. (3), it can be seen that the update of velocity in the PSO algorithm is determined by three aspects. Among them: It is influenced by the inertia weight, which is a balance between the global search ability and the local search ability; $c_1\cdot r_1\left(p_{id}^k-x_{id}^k\right)$ is the exchange of information about the particle itself, which determines the next direction of motion by comparing the current position with the best position of the individual; $c_2\cdot r_2\left(p_{gd}^k-x_{gd}^k\right)$ is the exchange of information between the particle and the global information, which helps the particle to jump out of the local optimum and make it find the global optimum quickly and accurately.

3. Improved PSO algorithm based on Cauchy criterion

The introduction of the Cauchy criterion updates the position of the standard PSO to improve its optimality finding ability, and the Cauchy criterion produces a larger criterion step, giving it a higher probability of escaping the local optimum. The probability density function of the one-dimensional Cauchy distribution is given in Eq. (4).

$$f_t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2} \tag{4}$$

in which, t is a scaling parameter and it is greater than 0.

According to the traditional idea of criterional operators, for individual $X_i = (x_{i1}, x_{i2}, \cdots, x_{in})$, the criterional formula is shown in Eq. (5).

$$x_{ij} = x_{ij} + \eta * C(0,1) \tag{5}$$

in which, j=1, 2, ... , n; η is a constant controlling the criterion step; and C(0,1) denotes the random number generated by the Cauchy distribution function when the scaling parameter t is 1.

Introducing the Cauchy criterion and taking the criterion step η as the square root of the absolute value of the position, the position update formula of the IPSO is obtained shown in Eq. (6).

$$x_{id}^{k+1} = x_{id}^{k+1} + \left|x_{id}^{k+1}\right|^{0.5} * C(0,1) \tag{6}$$

The steps of the IPSO algorithm are described as shown in Fig 1.

The steps of the improved particle swarm optimization algorithm	
Step1：	Set the particle swarm size, problem solving dimension Dim, maximum iteration MaxIter, inertia weight ω, individual experience learning factor c_1, social experience learning factor c_2, velocity and position boundaries, and initialize the particles and their velocities;
Step2：	Evaluating the first generation of particle fitness to obtain the individual optimum $P1$ and the global optimum Pg;
Step3：	When the number of iterations exceeds the maximum number of iterations (Iter>MaxIter), stop the search, and output the extreme point at the same time, otherwise go to Step4;
Step4：	IPSO does a global search;
Step5：	Iter=Iter+1， return to Step3.

Fig 1 The steps of the improved PSO algorithm

4. Experimental verification

To verify the effectiveness of the improved PSO (IPSO) algorithm and evaluate its performance in terms of merit-seeking ability and convergence speed, eight test functions as shown in Table 1 are introduced for testing, among which F1~F4 are single-peak test functions and F5~F8 are multi-peak test functions.

Table1 List of test functions

Name of function	Function expressions	Superiority-seeking interval	Extremum
Dixon-Price	$F_1 = (x_1 - 1)^2 + \sum_{i=2}^{d} i\left(2x_i^2 - x_{i-1}\right)^2$	$X \in [-10,10]^d$	$F_{min} = 0$
Exponential	$F_2 = -\exp\left(-0.5\sum_{i=1}^{d} x_i^2\right)$	$X \in [-1,1]^d$	$F_{min} = -1$
Rosenbrock	$F_3 = \sum_{i=1}^{d-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	$X \in [-30,30]^d$	$F_{min} = 0$
Sphere	$F_4 = \sum_{i=1}^{d} x_i^2$	$X \in [-100,100]^d$	$F_{min} = 0$

Griewank	$F_5 = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$X \in [-600{,}600]^d$	$F_{min} = 0$
Rastrigin	$F_6 = \sum_{i=1}^{d} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$X \in [-5.12{,}5.12]^d$	$F_{min} = 0$
Salomon	$F_7 = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{d} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{d} x_i^2}$	$X \in [-100{,}100]^d$	$F_{min} = 0$
Ackley	$F_8 = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + exp(1)$	$X \in [-100{,}100]^d$	$F_{min} = 0$

The simulation results of the standard PSO and the IPSO for solving the test functions F1 to F8 are shown in Table 2.

Table2 Simulation test result

Function	Optimum value		Average value		Standard deviation	
No.	PSO	IPSO	PSO	IPSO	PSO	IPSO
F1	1.62E-01	9.73E-02	7.32E-01	2.34E-01	2.57E-01	2.97E-02
F2	1.00E+00	1.00E+00	0.999966	1.00E+00	8.14E-06	0.00E+00
F3	5.60E-01	7.82E-08	1.50E+02	1.20E-02	5.96E+02	2.38E-02
F4	1.70E-04	3.72E-09	5.23E-02	4.32E-04	1.22E-01	1.10E-03
F5	3.97E-02	2.54E-08	3.18E-01	4.52E-02	2.09E-01	1.39E-01
F6	3.99E+00	3.45E-07	1.83E+01	8.46E-04	7.79E+00	1.30E-03
F7	3.00E-01	1.07E-04	7.48E-01	2.57E-02	3.09E-01	3.72E-02
F8	6.79E-02	1.73E-04	2.69E+00	6.61E-03	9.64E-01	7.08E-03

Through comparison, it can be found that the solution performance of the IPSO is substantially improved. In order to compare the convergence speed of PSO and IPSO, the eight test functions were solved 50 times respectively, as shown in Fig 2-9.

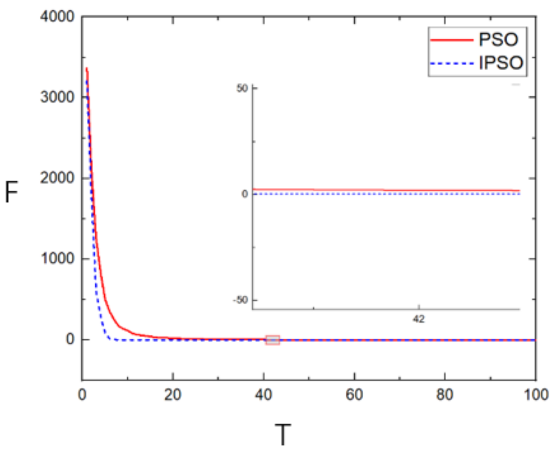

Figure 2 Fitness curve of PSO and IPSO optimizing function F1

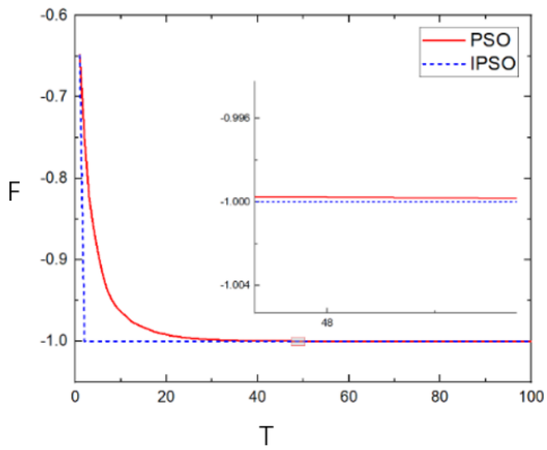

Figure 3 Fitness curve of PSO and IPSO optimizing function F2

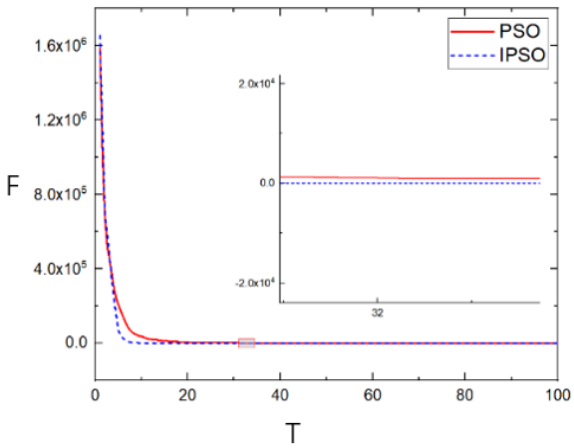

Figure 4 Fitness curve of PSO and IPSO optimizing function F3

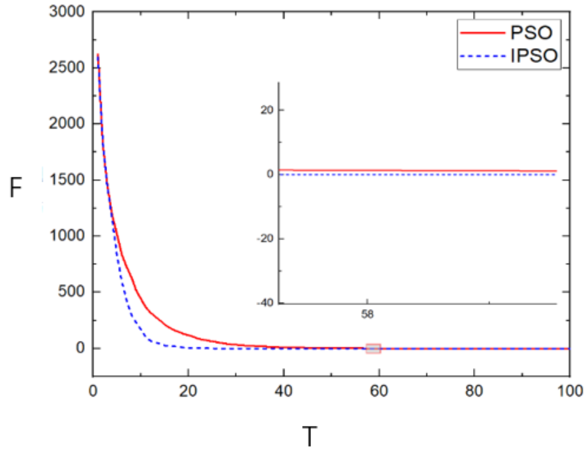

Figure 5 Fitness curve of PSO and IPSO optimizing function F4

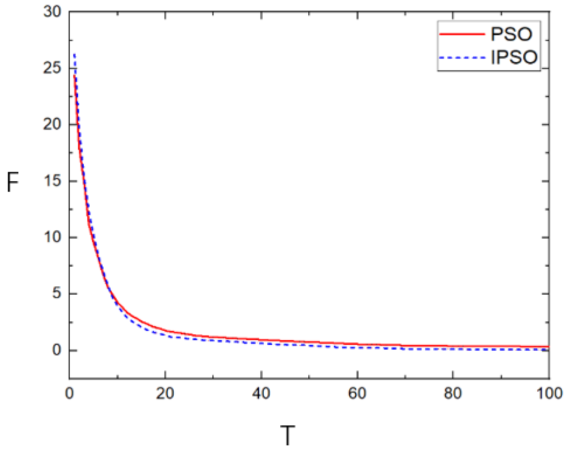

Figure 6 Fitness curve of PSO and IPSO optimizing function F5

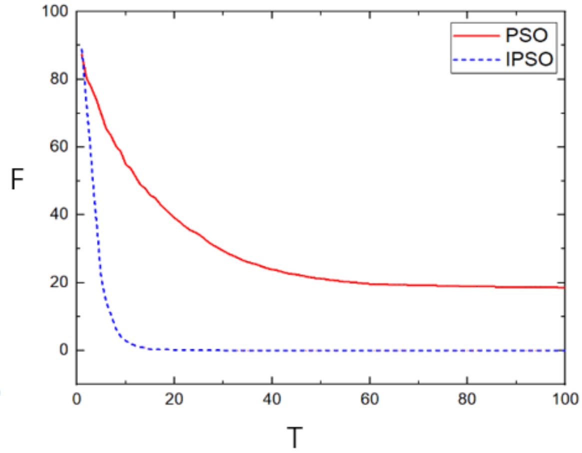

Figure 7 Fitness curve of PSO and IPSO optimizing function F6

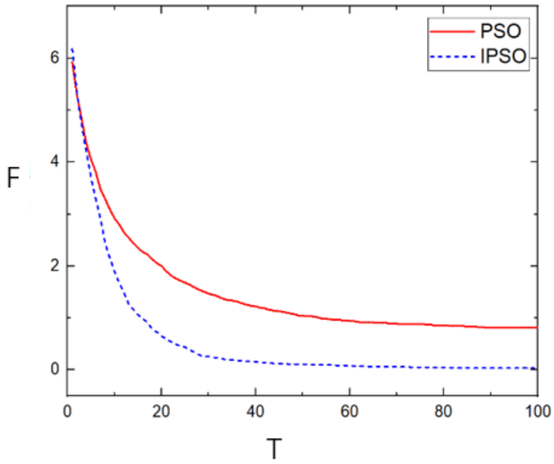

Figure 8 Fitness curve of PSO and IPSO optimizing function F7

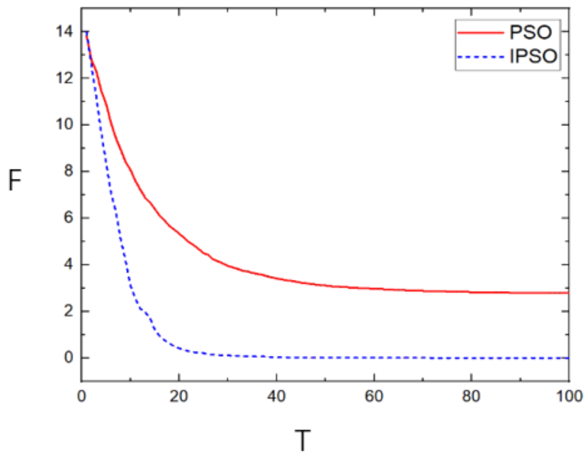

Figure 9 Fitness curve of PSO and IPSO optimizing function F8

Through the above simulation experiments, it can be seen that the improved PSO has significantly improved its optimization finding accuracy and convergence speed when solving single-peak and multi-peak problems.

Conclusion

In this paper, the integration of the Cauchy criterion into PSO algorithms offers a promising avenue for enhancing their exploration-exploitation balance and overcoming premature convergence issues. Through comprehensive experimentation and real-world applications, Cauchy-based PSO variants have shown significant improvements in optimization performance. Future research directions may focus on further refining the Cauchy-based strategies and exploring their applicability to complex and dynamic optimization problems.

Declaration of competing interest

The authors declare no conflict of interest.

Data Availability

The datasets used during the current study available from the corresponding author on reasonable request.

References

1. Han F, Tang D, Sun Y W T, et al. A hybrid gene selection method based on gene scoring strategy and improved particle swarm optimization[J]. BMC bioinformatics, 2019, 20: 1-13.
2. Saeedi S, Khorsand R, Bidgoli S G, et al. Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing[J]. Computers & Industrial Engineering, 2020, 147: 106649.
3. Zhang Y, Jin Z. Quantum-behaved particle swarm optimization with generalized space transformation search[J]. Soft Computing, 20 20, 24(19): 14981-14997.
4. Famila S, Jawahar A. Improved artificial bee Colony optimization-based clustering technique for WSNs[J]. Wireless Personal Communications, 2020, 110(4): 2195-2212.
5. Sun W, Wang Y W. Factor analysis and carbon price prediction based on empirical mode decomposition and least squares support vector machine optimized by improved particle swarm optimization[J]. Carbon Management, 2020, 11(3): 315-329.
6. Li J, Zhang C, Zhang X, et al. Temperature compensation of piezo-resistive pressure sensor utilizing ensemble AMPSO-SVR based on improved AdaBoost. RT[J]. IEEE Access, 2020, 8: 12413-12425.
7. Zhang J, Wang J S. Improved salp swarm algorithm based on levy flight and sine cosine operator[J]. Ieee Access, 2020, 8: 99740-99 771.
8. Ali Z A, Zhangang H. Multi-unmanned aerial vehicle swarm formation control using hybrid strategy[J]. Transactions of the Institute of Measurement and Control, 2021, 43(12): 268 9-2701.
9. Optimized Control for Medical Image Segmentation: Improved Multi-agent Systems Agreements Using Particle Swarm Optimization
10. Zhou L, Chen K, Dong H, et al. An improved beetle swarm optimization algorithm for the intelligent navigation control of autonomous sailing robots[J]. IEEE Access, 2020, 9: 5296-5311.
11. Wu L, Yang Y, Maheshwari M. Strain prediction for critical positions of FPSO under different loading of stored oil using GAIFOA-BP neural network[J]. Marine Structures, 20 20, 72: 102762.
12. Wu L, Mei J, Zhao S. Pipeline damage identification based on an optimized back-propagation neural network improved by whale optimization algorithm[J]. Applied Intelligence, 2023, 53(10): 12937-12954.
13. Liu, C., Wu, L., Huang, X.D., Xiao, W.S. Improved dynamic adaptive ant colony optimization algorithm to solve pipe routing design [J]. Knowledge-based systems, 2022, 237:107846.
14. Liu, C., Wu, L., Xiao, W.S., et al. An improved heuristic mechanism ant colony optimization algorithm for solving path planning[J]. Knowledge-Based Systems, 2023, 271: 110540.
15. Liu, C., Wu, L., Li, G.X., H, Zhang, Xiao, W.S, Xu, D.P., Guo, J.J. Improved multi-search strategy A* algorithm to solve three-dimensional pipe routing design [J]. Expert Systems with Applications, 240 (2024), Article 122313.
16. Wu L, Zhang L, Xiao W S, et al. A novel heuristic algorithm for two-dimensional rectangle packing area minimization problem with central rectangle[J]. Computers & Industrial Engineering, 2016, 102: 208-218.
17. Cui J, Wu L, Huang X, et al. Multi-strategy adaptable ant colony optimization algorithm and its application in robot path planning[J]. Knowledge-Based Systems, 2024,288:111459.
18. Xiao W, Wu L, Tian X, et al. Applying a new adaptive genetic algorithm to study the layout of drilling equipment in semisubmersible

drilling platforms[J]. Mathematical Problems in Engineering, 2015, 2015(1): 146902.
19. Wu L, Mei J, Zhao S. Pipeline damage identification based on an optimized back-propagation neural network improved by whale optimization algorithm[J]. Applied Intelligence, 2023, 53(10): 12937-12954.
20. Li G.X, Liu C., Xiao W.S, et al. A mixing algorithm of ACO and ABC for solving path planning mobile robot [J]. Applied Soft Computing Journal，2023, 148: 110868.
21. Xiao W.S, Li G.X., Liu C., et al. A novel chaotic and neighborhood search-based artificial bee colony algorithm for solving optimization problems [J]. Scientific Reports, 2023, 13(1): 20496.
22. Liu C, Wu L, Li G, et al. AI-based 3D pipe automation layout with enhanced ant colony optimization algorithm[J]. Automation in Construction, 2024, 167: 105689.