Received 12 November 2023, accepted 27 November 2023, date of publication 30 November 2023,
date of current version 11 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3338160

RESEARCH ARTICLE

Arithmetic Optimization Algorithm Based on Cauchy Mutation Trigonometric Function Search to Solve Combined Economic Emission Dispatch Problem

YUN-PENG LI[1,2], LIN-XUAN ZHANG[1], AND ZHENG-JIE FENG[2]
[1]School of Electrical Engineering, Xinjiang University, Ürümqi, Xinjiang 830049, China
[2]Zhejiang Xingyu Mechanical and Electrical Technology Company Ltd., Jinhua, Zhejiang 311400, China
Corresponding author: Yun-Peng Li (Yunpeng_Li2023@163.com)

This work was supported by the "Tianchi Talent" Youth Doctoral Program of Xinjiang Autonomous Region.

ABSTRACT Nowadays, problems such as greenhouse effect and air pollution become increasingly prominent. The power generation process should pay attention to the fuel cost and reduce the pollution to the environment. In this paper, an arithmetic optimization algorithm (AOA) based on Cauchy mutation trigonometric function search is proposed to solve the combined economic emission dispatch (CEED) problem. The price penalty factor is used to convert multiple objective functions of CEED problem into one objective. The math optimization probability (MOP) parameter is used to control the position update of the AOA. MOP is replaced by the oscillation coefficient and Cauchy mutation, which can not only guarantee the local search accuracy but also enhance the global search capability. 6 kinds of trigonometric function search operators, sine function, hyperbolic sine, inverse hyperbolic sine, hyperbolic tangent, inverse hyperbolic tangent, are used to replace the fixed value control parameters in AOA algorithm. The algorithm still has a certain ability to jump out of the local optimal in the late running. The 23 test functions are used to verify the effect of the improved AOA method. The AOA is compared with 6 kinds of improved AOA, and the improvement method with the best effect is selected, and then compared with other intelligent optimization algorithms to verify the effectiveness of the improved strategy. In addition to improving the algorithm, the constraint of CEED problem is also treated. Five kinds of random disturbance penalty functions are proposed, which are cosine function, hyperbolic sine function, tangent function, hyperbolic tangent function and V-type function. The CEED problem of 6 units was selected for simulation experiment under 4 different load demands. The experimental data show that the arithmetic optimization algorithm based on Cauchy mutation trigonometric function search has a good effect on solving CEED problems, and the random disturbance penalty strategy is more effective in solving quality.

INDEX TERMS Combined economic emission dispatch, Cauchy mutation, random disturbance penalty function, trigonometric function search, arithmetic optimization algorithm.

I. INTRODUCTION

Nowadays, electricity industry is a necessary element in people's work and life. From the past to the present, the power industry has been developing in a highly demanding and competitive environment [1]. Among them, people pay much attention to this problem, how to reduce the cost of power generation has become a key research object in power generation work. The task of obtaining the minimum power generation cost while meeting the power supply demand

The associate editor coordinating the review of this manuscript and approving it for publication was Youngjin Kim.

is called the economic load dispatch (ELD) problem [2]. As environmental problems become more and more serious, people begin to realize the importance of protecting the environment. At present, clean energy related technologies have been vigorously developed, such as wind power generation and hydroelectric power generation are increasingly mature, but thermal plants is still the main way of power generation. The burning of fossil fuels releases a lot of air contaminant, such as SO2, NOx and CO2. These pollutants will cause serious damage to the natural environment and lead to a series of environmental problems. Therefore, it is not only important to consider the fuel cost, but also to reduce the pollution to the environment. When power generation cost and operational constraints are met, the combined economic emission dispatch (CEED) should be considered at the same time as power generation cost and pollutant emission [3]. It is more complicated to solve the CEED problem, which has been widely concerned and studied.

Quadratic function as a way to express the CEED problem [4]. It is found that the high-order polynomial function representing the CEED problem can improve the solution method [5]. However, this makes the solution of CEED problem more difficult and complex. Some methods based on classical mathematical modeling are adopted for solving CEED problem [6], [7], [8], [9], [10], which provides a good solution for solving the CEED problem. However, power generation system is often nonlinear and non-convex in practical work, and there will be many constraints in production work, so it becomes difficult to solve CEED problems by using this kind of method. Therefore, many intelligent optimization methods are used to deal with CEED problems with good results, Particle Swarm Optimization algorithm (PSO) [11], Bacterial Foraging algorithm (BFO) [12], Grasshopper optimization algorithm (GOA) [13], Ant colony optimization (ACO) [14], Ant Lion optimization (ALO) (SMO) [15], Spider Monkey optimization [16], Sine Cosine algorithm (SCA) [17], etc. With the progress of science and technology, the power system has been developed to the cyber-physical power system, which brings convenience but also considers the problem of cyber attacks. Lu et al. propose a three-stage dynamic fake data injection attack (DFDIA) model that considers underlying dynamic behavior. Two constrained differential evolution algorithms are designed to determine the attack location and optimize the attack vector to cooperatively change the meter measurement. Several IEEE bus systems are selected in simulation experiments to prove the effectiveness of the proposed method [18].

Some researchers transform multi-objective CEED problems into single-objective optimization problems for solving. Ziane et al. [19] adopted simulated annealing method to deal with the CEED problem, and used maximum/maximum price penalty factor (PPF) to convert fuel cost and SO2, NOx and CO2 emissions into a single target for optimization. Compared with other methods such as PSO and Lagrange technique, simulated annealing can give better solutions. Aydin et al. proposed the Artificial bee colony with dynamic population size (ABCDP), which uses a similar mechanism defined in the Incremental ABC with local search (IABC-LS) and reduces many of the parameters to be adjusted. In order to test the effectiveness and robustness of the algorithm in CEED problem, the algorithm is applied to the economic emissions joint dispatching problem. The IEEE30 bus test system and 40 generator sets are taken as examples. The results show that ABCDP gives good results in both systems [20]. Gherbi et al. [21] proposed a new method combining firefly algorithm and bat algorithm to deal with CEED problem, which well combined the advantages of the two. Hassan et al. proposed an optimization algorithm based on chaotic artificial ecosystem (CAEO) and used PPF to convert four objective functions into one objective function when solving CEED problem. The proposed improved AEO method is based on chaotic mapping. Instead of using random arguments [22]. It can be seen from the above literature that a multi-objective CEED problem can be transformed into a single-objective optimization problem and solved by an optimization algorithm to get a better performance. The optimization process of this method is simple and the operation time is short. However, the weight setting of each objective function has a great influence on the optimization results, and only one solution can be obtained each time, and the Pareto optimal solution set cannot be obtained.

Using multi-objective optimization algorithm to deal with CEED problem directly is also a very effective method. Kumar et al. [23] proposed a multi-objective directed bee colony optimization algorithm (MODBC), which is well applied to optimize CEED multi-objective optimization problem with equality and inequality constraints. Hybridization enables MODBC to get the high quality and fast solutions that generate better Pareto frontiers for multi-objective problems. Wu et al. [24] proposed a new multi-objective differential brain storm optimization (MDBSO) algorithm to solve the EED problem. Unlike classical BSO, clustering operations are designed in the target space rather than the solution space to improve computational efficiency. In order to keep the diversity of population and improve the convergence speed, differential variation is used to replace Gaussian variation. The effect of the algorithm is tested by simulation experiment. The simulation results show that MDBSO can have better convergence while maintaining the diversity of Pareto optimal solutions. Chen et al. proposed a constrained multi-objective population extremal optimization algorithm (CMOPEO-EED) in order to improve EED performance of renewable energy generating units. Simulation experiments with three versions of improved IEEE 30 bus and 6 generator systems with renewable energy generation are carried out to verify the superiority of the proposed CMOPEO-EED method in solving EED problems [25]. Singh et al. [26] proposed a permutation based multi-objective environment adaptation method (pMOEAM) to solve the EED problem of power systems. The new improved pMOEAM solves the three weaknesses of the original algorithm. A large number of experimental data show that pMOEAM method can get better quality solutions for EED problem. Chen et al. [27] proposed a new multi-objective whale optimization algorithm (MONWOA) to solve the EED problem. In order to explore and develop the balance algorithm, Gaussian mutation operator, differential evolution algorithm mutation process and search mode parameters were used to improve the standard MOWOA. In summary, using the multi-objective optimization algorithm to solve CEED problem directly, a set of Pareto optimal solutions can be obtained in a single run, and the optimal solutions can be selected in the solution set according to demand. However, the calculation process of this method is more complicated, and the performance of the algorithm is higher.

The constraint problem must be considered in the actual production work, and the treatment of the constraint problem is also very important in the process of solving the CEED problem. Penalty function method is usually used to deal with solutions that do not meet constraints. A penalty function proportional to the degree of constraint violation is added to the objective function to discard solutions that do not meet constraints, and the setting of penalty intensity is very important [28]. Literature [29] selects a fixed penalty parameter, which requires repeated trials to find the appropriate penalty parameter and obtain a reasonable solution. For multi-constrained optimization problems, an improved PSO algorithm (IPSO) combined with penalty function is proposed to prevent premature convergence, accelerate the search speed and ensure the feasibility of solution [30]. Haeun Yoo et al. proposed a dynamic penalty (DP) method. In the training process, the penalty factor increases gradually and systematically with the progress of iterative events [31]. Sakthivel et al. introduced a refurbishment strategy to deal with power balance constraints, in which a randomly generated unit is selected to perform the refurbishment process. The advantage of this strategy is that it is easy to implement and quickly updated [32].

Based on the above literature, it can be found that the treatment of constraint problems will have a certain impact on the solution of CEED problems, so it is also very important to deal with constraint problems well while improving the optimization algorithm. This paper chooses to use arithmetic optimization algorithm (AOA) to solve CEED problem, because AOA is a crowd-based meta-heuristic algorithm with simple structure and easy to understand principle. It is realized by arithmetic operators (addition, subtraction, multiplication and division) in mathematics [33], and has a good effect in solving practical engineering problems. So using AOA algorithm can also solve CEED problem well. The AOA based on Cauchy mutation trigonometric function search is designed to solve CEED problem. In addition to improving the algorithm, a random disturbance penalty strategy is proposed to deal with the constraint problem. Firstly, the effectiveness of the improved method is verified on CEC2017 benchmark functions and compared with other optimization

TABLE 1. 5 penalty function expressions.

Function	*RD*
cosine	$1-\cos\left(\frac{\pi}{2}x\right)+x\times rand\cos\left(\frac{\pi}{2}x\right)$
hyperbolic sine	$\lvert a\sinh(1.1x)\rvert+x\times rand\left(1-\lvert a\sinh(1.1x)\rvert\right)$
tangent	$\left\lvert\tan\left(\frac{\pi}{4}x\right)\right\rvert+x\times rand\left(1-\left\lvert\tan\left(\frac{\pi}{4}x\right)\right\rvert\right)$
hyperbolic tangent	$\lvert\tanh(3x)\rvert+x\times rand\left(1-\lvert\tanh(3x)\rvert\right)$
V-type	$\frac{2\sqrt{\lvert x\rvert}}{\lvert x\rvert+1}+x\times rand\left(1-\frac{2\sqrt{\lvert x\rvert}}{\lvert x\rvert+1}\right)$

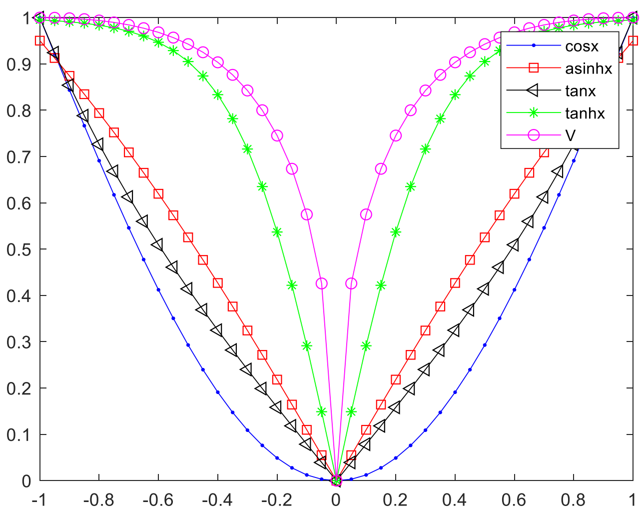


FIGURE 1. The images of 5 penalty functions.

algorithms. Then, the CEED cases with 6 units are selected for simulation. The results show that the arithmetic optimization algorithm based on Cauchy mutation trigonometric function search and the random disturbance penalty strategy have good performance.

The structure of the thesis is as follows: In section II, the expression of CEED problem and the random disturbance penalty function are introduced. Section III describes the basic principle of the algorithm and arithmetic optimization algorithm based on Cauchy mutation trigonometric function search. In Section IV, CEC2017 benchmark functions are used to test the performance of AOA based on Cauchy mutation trigonometric function search. Section V selects CEED cases for simulation experiments; Section VI concludes.

II. PROBLEM DESCRIPTION

A. PROBLEM OBJECTIVE

The task of CEED problem is to satisfy total load demand and minimize fuel cost and pollutant emission under all constraints [1]. Pollutant emission includes three objective functions SO_2, NO_x and CO_2. Therefore, fuel costs and

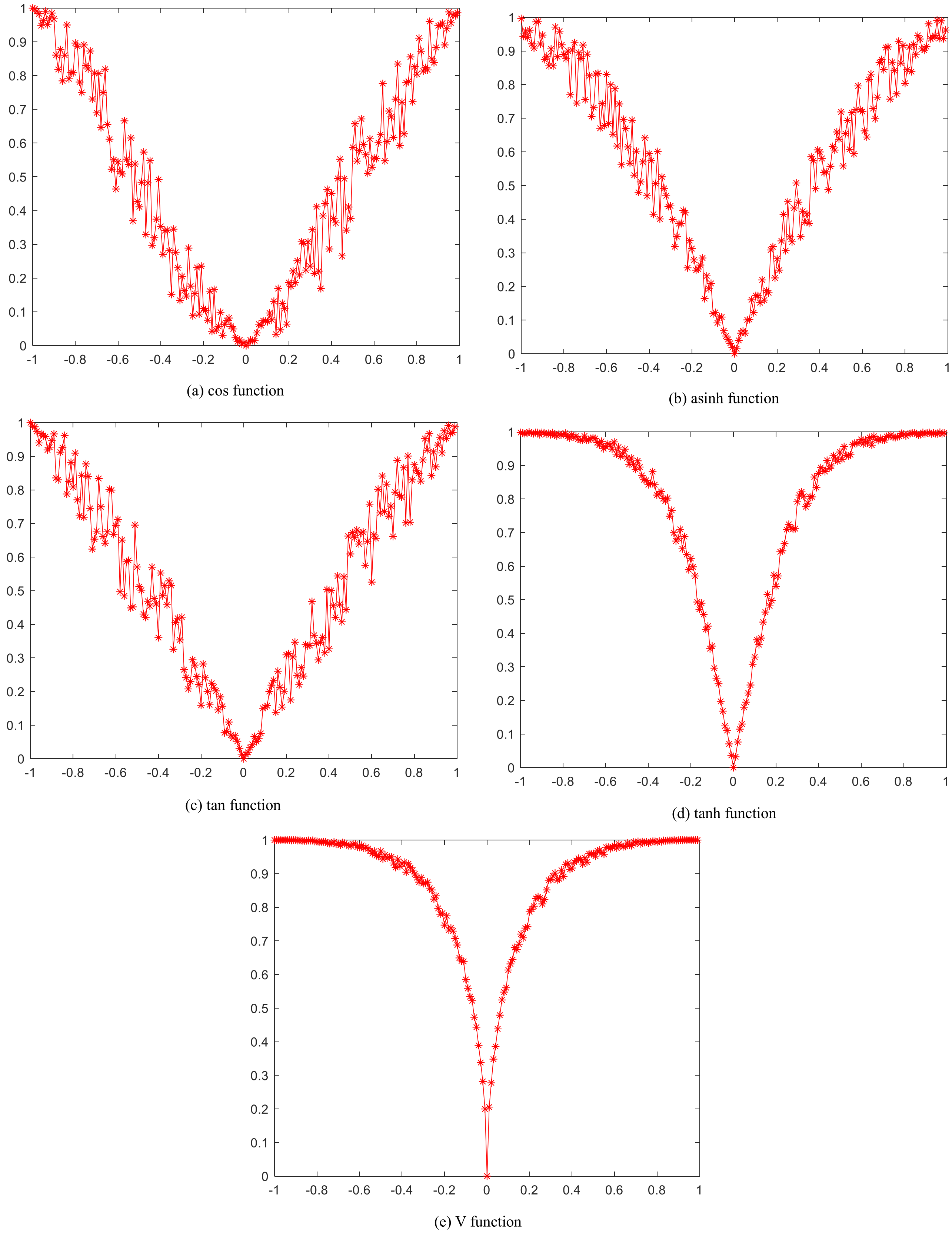


FIGURE 2. The image of 5 penalty functions after adding disturbance.

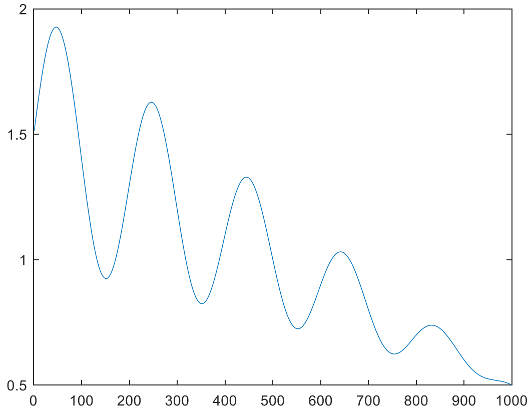


FIGURE 3. Image of oscillation coefficient.

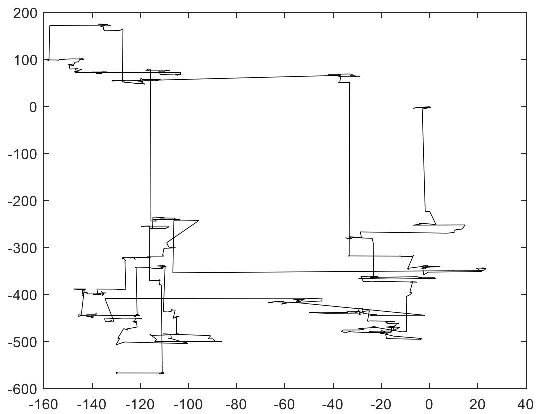


FIGURE 4. The Motion trajectory of oscillating Cauchy mutation.

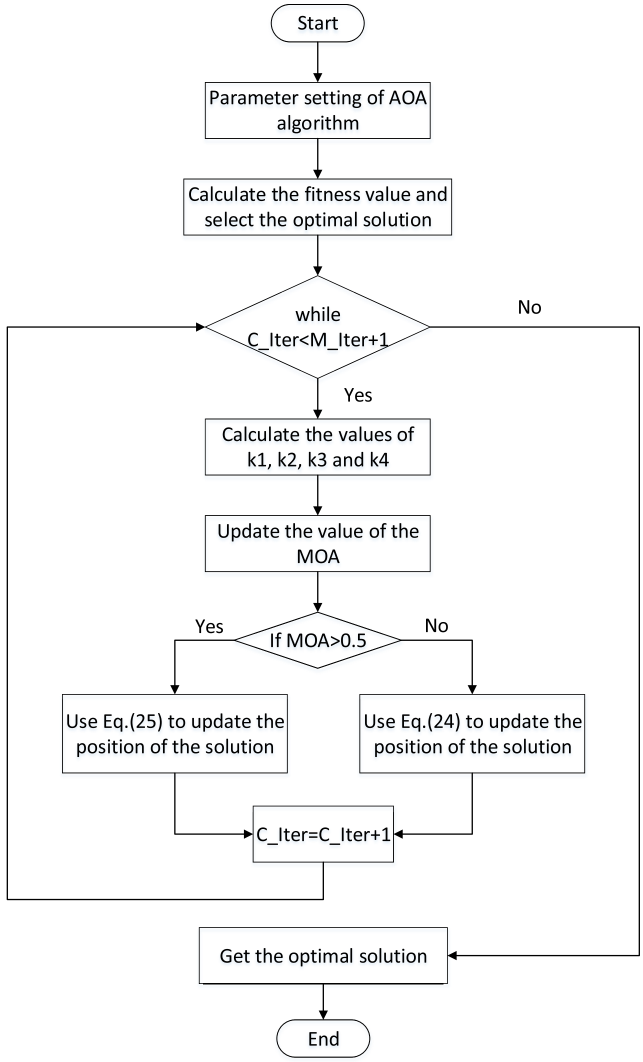


FIGURE 5. Flow chart of improved AOA algorithm.

pollutant emissions need to be considered in four optimization objectives. The fuel cost F(P) is expressed here as a cubic criterion function, and its expression is shown in Eq. (1).

$$F_t(P) = \sum\nolimits_{i=1}^{n} a_i P_i^3 + b_i P_i^2 + c_i P_i + d_i \tag{1}$$

where, a_i, b_i, c_i and d_i is the fuel cost coefficient of the generating set. n is the number of total generating units, P_i is the actual output power of the generating set. Emissions targets of SO_2, NO_x and CO_2 are expressed using cubic polynomial functions respectively.

$$\begin{aligned} E_{SO2}(P) &= \sum\nolimits_{i=1}^{n} e_{SO2i} P_i^3 + f_{SO2i} P_i^2 + g_{SO2i} P_i + h_{SO2i} \\ E_{NOx}(P) &= \sum\nolimits_{i=1}^{n} e_{NOxi} P_i^3 + f_{NOxi} P_i^2 + g_{NOxi} P_i + h_{NOxi} \\ E_{CO2}(P) &= \sum\nolimits_{i=1}^{n} e_{CO2i} P_i^3 + f_{CO2i} P_i^2 + g_{CO2i} P_i + h_{CO2i} \end{aligned} \tag{2}$$

where, $E_{SO2i}, f_{SO2i}, g_{SO2i}, h_{SO2i}, E_{NOxi}, f_{NOxi}, g_{NOxi}, h_{NOxi}, e_{CO2i}, f_{CO2i}, g_{CO2i}$ and h_{CO2i} are the pollutant emission coefficients of the $i-th$ generator set, respectively.

The maximum/maximum penalty factor of a unit direction [19] is used to transform the multi-objective CEED problem into a single-objective problem to solve. The objective function is set to the total cost F_T, and the mathematical expression is as follows:

$$\begin{aligned} OF &= \min(F_T) \\ F_T &= \sum\nolimits_{i=1}^{n} \{F(P_i) + h_{Si} E_{SO2}(P_i) + h_{Ni} E_{NOx}(P_i) \\ &\quad + h_{Ci} E_{CO2}(P_i)\} \\ h_{Si} &= \sum\nolimits_{i=1}^{n} \frac{F(P_{i,\max})}{E_{SO2}(P_{i,\max})} \\ h_{Ni} &= \sum\nolimits_{i=1}^{n} \frac{F(P_{i,\max})}{E_{NOx}(P_{i,\max})} \\ h_{Ci} &= \sum\nolimits_{i=1}^{n} \frac{F(P_{i,\max})}{E_{CO2}(P_{i,\max})} \end{aligned} \tag{3}$$

where, $F(P_{i,max})$, $E_{SO2}(P_{i,max})$, $E_{NOx}(P_{i,max})$ and $E_{CO2}(P_{i,max})$ are respectively the total fuel cost corresponding to the output power of generator set i, the total emissions of SO_2, the total emissions of NO_x and the total emissions of CO_2.

TABLE 2. Parameter settings of the six improvement methods.

Method	k_3	k_4
sinCAOA	$2\times\pi\times rand$	$0.5+0.01\times rand\times\sin(k_3)$
sinhCAOA	$randn$	$0.5+0.01\times rand\times\sinh(k_3)$
asinhCAOA	$randn$	$0.5+0.01\times rand\times a\sinh(k_3)$
tanhCAOA	$randn$	$0.5+0.01\times rand\times\tanh(k_3)$
atanCAOA	$randn$	$0.5+0.01\times rand\times a\tan(k_3)$
atanhCAOA	$2\times rand-1$	$0.5+0.01\times rand\times a\tanh(k_3)$

h_{Si}, h_{Ni} and h_{Ci} are the maximum/maximum penalty factors of SO_2, NO_x and CO_2 emitted by generator set i, respectively.

B. CONSTRAINT PROBLEM

In the process of CEED problem optimization, many constraints will be faced. The concepts of power balance constraints and generator power constraints are mainly introduced below.

1) POWER BALANCE CONSTRAINT

The total output of the generator set must equal the total power required by the load plus the actual power loss of the transmission line, and its expression can be defined as:

$$\sum\nolimits_{i=1}^{n} P_i = P_D + P_L \tag{4}$$

$$P_L = \sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n} P_i B_{ij} P_j + \sum\nolimits_{i=1}^{n} B_{0i} P_i + B_{00} \tag{5}$$

where, P_D is the total actual power demand, and P_L is the loss during transmission. B_{ij} B_{0i} and B_{00} are the loss coefficients.

2) GENERATOR POWER CONSTRAINTS

The generator set must work within the scope of the specification, and its expression is shown as follows:

$$P_i^{\min} \le P_i \le P_i^{\max} \tag{6}$$

where, P_i is the output power output of the $i-th$ generator, $P_i^{\max}$ and $P_i^{\min}$ represent the upper and lower limits of the output power of generator i respectively.

3) PENALTY FUNCTION OF RANDOM DISTURBANCE PENALTY STRATEGY

This paper mainly considers the power balance constraint and generator power constraint. The generator power constraints are inequality constraints, and the following methods are adopted to deal with the generator power constraints of the $i-th$ machine:

$$P_i = \begin{cases} P_i, & \left(P_i^{\min} \le P_i \le P_i^{\max}\right) \\ P_i^{\min}, & \left(P_i < P_i^{\min}\right) \\ P_i^{\max}, & \left(P_i > P_i^{\max}\right) \end{cases} \tag{7}$$

Equality constraint is power balance constraint. When dealing with equality constraint, punishment factor with fixed value is generally used [14]. The definition of objective function is as follows:

$$\min F_t = \min F_t + FP \times \Delta P_D \tag{8}$$

where, FP is the fixed penalty value, which is set to 300 after multiple value attempts in this experiment. ΔP_D is transmission loss plus supply demand minus total generation.

$$\Delta P_D = P_D + P_L - \sum\nolimits_{i=1}^{n} P_i \tag{9}$$

The traditional fixed penalty function is to add the degree of violation of the infeasible solution to the target function as punishment. For the solution that violates the constraint to a large extent, it needs to take a large degree of punishment. However, the solution that violates fewer constraints can be punished less, and the subsequent iteration update may be closer to the optimal solution. In this paper, a random disturbance penalty strategy is proposed to deal with the constraint problem. Five penalty strategies are proposed, including cosine function, hyperbolic sine function, tangent function, hyperbolic tangent function and V-type function. These five punishment strategies can change the degree of punishment with the change of the degree of violation of constraints of the generated solutions. However, the degree of punishment of some functions decreases too fast with the change of the degree of violation of constraints, resulting in too small punishment, and the obtained solutions cannot meet the constraints. Therefore, on the basis of these five penalty functions, perturbations varying with the degree of constraint violation are added, and the degree of punishment is appropriately increased to ensure that the obtained solution satisfies the constraint. The improved penalty factor is shown as follows:

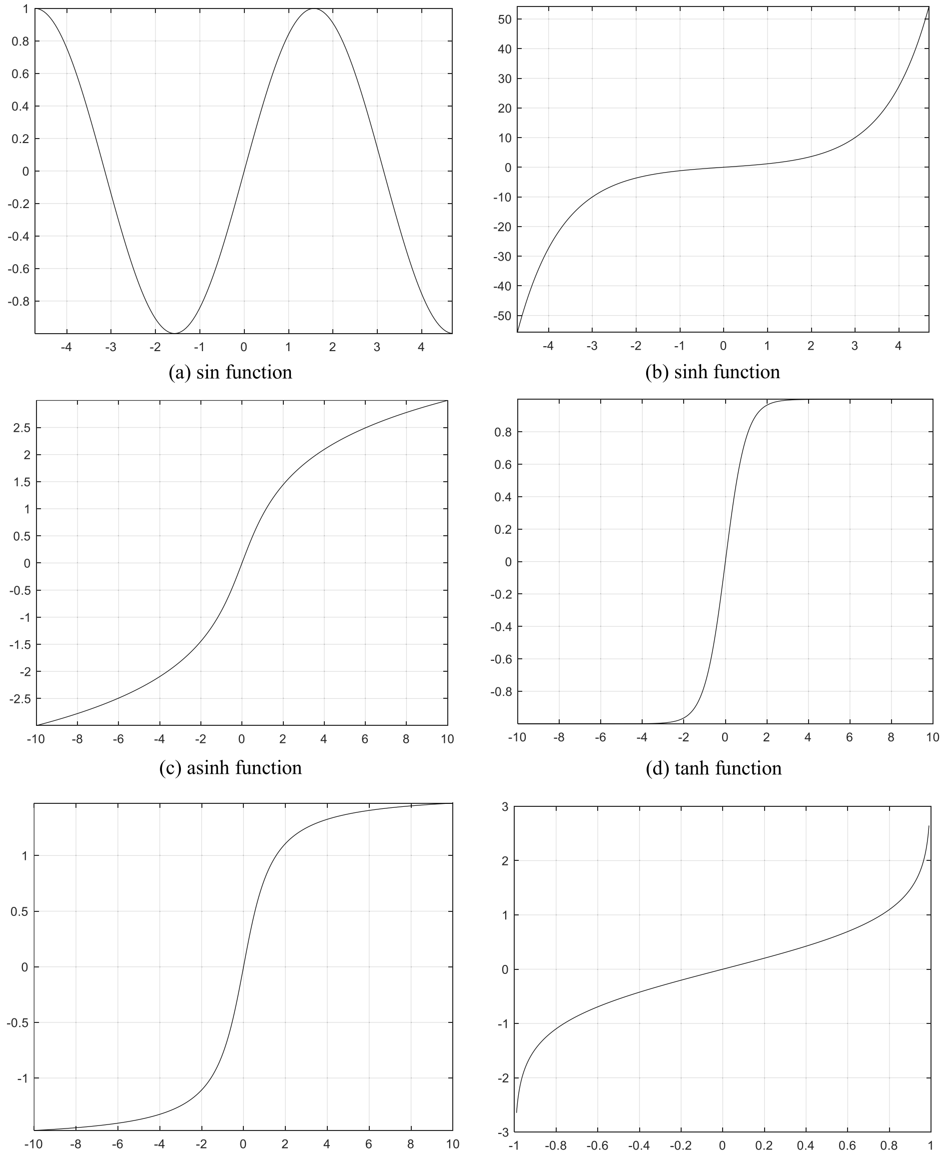
$$RDP = FP \times RD \tag{10}$$

where, RDP is the random disturbance penalty factor, FP is the penalty factor with a fixed value, and Table 1 shows the expression of RD. The x in Table 1 represents the degree to which the resulting solution violates the constraint. Fig. 1 shows an image of the five penalty functions. Fig. 2 is the image of the penalty functions after adding disturbance. The x axis of Fig. 1-2 shows the degree of constraint violation, and the y axis shows the degree of punishment.

III. ARITHMETIC OPTIMIZATION ALGORITHM BASED ON CAUCHY MUTATION TRIGONOMETRIC FUNCTION SEARCH

A. ARITHMETIC OPTIMIZATION ALGORITHM

AOA is a meta-heuristic algorithm newly proposed by Abualigah et al. in 2021 [33]. This algorithm mainly uses four operators of addition, subtraction, multiplication and division to seek and gradually approaches the optimal solution. In the running process of the algorithm, initialization is carried out



(a) sin function (b) sinh function (c) asinh function (d) tanh function (e) atan function (f) atanh function

FIGURE 6. **Trigonometric functions.**

first, and the initial solution X is generated randomly. Its expression is shown as follows.

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,n-1} & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & \cdots & \cdots & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{11}$$

Next, math optimization acceleration function (MOA) is used to control the algorithm selection search phase. MOA is compared with $r1$ ($r1$ is a random number from 0 to 1). When MOA is greater than $r1$, the AOA enters the exploration stage. When MOA is less than $r1$, the AOA enters the exploitation stage. Eq. (12) is the calculation formula of MOA. The math optimization probability function (MOP) is an important parameter to control the algorithm's position update, as shown in Eq. (13).

$$MOA\left(C\_Iter\right) = Min + C\_Iter \times \left(\frac{Max - Min}{M\_Iter}\right) \tag{12}$$

$$MOP\left(C\_Iter\right) = 1 - \frac{C\_Iter^{1/\alpha}}{M\_Iter^{1/\alpha}} \tag{13}$$

where, $MOA\left(C\_Iter\right)$ and $MOP\left(C\_Iter\right)$ indicate the MOA and MOP values respectively. α is a parameter that affects the search accuracy, $\alpha = 5$.

When MOA is greater than $r1$, AOA is optimized using a division or multiplication operator. The exploration stage can be searched extensively. Eq. (14) represents the position update expression of AOA exploration stage. When MOA is less than $r1$, the algorithm enters the exploitation stage. At this time, AOA uses subtraction and addition operators to

TABLE 3. **Benchmark testing functions.**

Type	F	Functions	Global min	Dim
Unimodal Function	F_1	Shifted and Rotated Bent Cigar function	100	[-100,100]
	F_2	Shifted and Rotated sum of Differential Power Function	200	[-100,100]
	F_3	Shifted and Rotated Zakharov Function	300	[-100,100]
Multimodal Functions	F_4	Shifted and Rotated Rosenbrock's Function	400	[-100,100]
	F_5	Shifted and Rotated Rastrigin's Function	500	[-100,100]
	F_6	Shifted and Rotated Expanded Scaffer's Function	600	[-100,100]
	F_7	Shifted and Rotated Lunacek Bi_Rastrigin Function	700	[-100,100]
	F_8	Shifted and Rotated Non-Continuous Rastrigin's Function	800	[-100,100]
	F_9	Shifted and Rotated Levy Function	900	[-100,100]
	F_{10}	Shifted and Rotated Schwefel's Function	1000	[-100,100]
Hybrid Functions	F_{11}	Hybrid Function of Zakharov, Rosenbrock and Rastrigin's	1100	[-100,100]
	F_{12}	Hybrid Function of High Conditioned Elliptic, Modified Schwefel and Bent Cigar	1200	[-100,100]
	F_{13}	Hybrid Function of Bent Ciagr, Rosenbrock and Lunache Bi-Rastrigin	1300	[-100,100]
	F_{14}	Hybrid Function of Bent Ciagr, Rosenbrock and Lunache Bi-Rastrigin	1400	[-100,100]
	F_{15}	Hybrid Function of Eliptic, Ackley, Schaffer and Rastrigin	1500	[-100,100]
	F_{16}	Hybrid Function of Bent Cigar, HGBat, Rastrigin and Rosenbrock	1600	[-100,100]
	F_{17}	Hybrid Function of Expanded Schaffer, HGBat, Rosenbrock and Modified Schwefel	1700	[-100,100]
	F_{18}	Hybrid Function of Katsuura, Ackley, Expanded Griewank plus Rosenbrock, Modifed Schwefel and Rastrigin	1800	[-100,100]
	F_{19}	Hybrid Function of Bent Cigar, Rastrigin, Expanded Grienwank plus Rosenbrock, Weierstrass and expanded Schaffer	1900	[-100,100]
	F_{20}	Hybrid Function of Happycat, Katsuura, Ackley, Rastrigin, Modified Schwefel and Schaffer	2000	[-100,100]
Composition Functions	F_{21}	Composition Function of Rosenbrock, High Conditioned Elliptic and Rastrigin	2100	[-100,100]
	F_{22}	Composition Function of Rastrigin's, Griewank's and Modifed Schwefel's	2200	[-100,100]
	F_{23}	Composition Function of Rosenbrock, Ackley, Modified Schwefel and Rastrigin	2300	[-100,100]
	F_{24}	Composition Function of Rastrigin, Happycat, Ackley, Discus and Rosenbrock	2400	[-100,100]
	F_{25}	Composition Function of Expanded Scaffer, Modified Schwefel, Griewank, Rosenbrock and Rastrigin	2500	[-100,100]
	F_{26}	Composition Function of HGBat, Rastrigin, Modified Schwefel, Bent-Cigar, High Conditioned Elliptic and Expanded Scaffer	2600	[-100,100]
	F_{27}	Composition Function of Ackley, Griewank, Discus, Rosenbrock, HappyCat, Expanded Scaffer	2700	[-100,100]
	F_{28}	Composition Function of shifted and rotated Rastrigin, Expanded Scaffer and Lunacek Bi_Rastrigin	2800	[-100,100]
	F_{29}	Composition Function of shifted and rotated Rastrigin, Expanded Schaffer and Lunacek Bi_Rastrigin	2900	[-100,100]
	F_{30}	Composition Function of shifted and rotated Rastrigin, Non-Continuous Rastrigin and Levy Function	3000	[-100,100]

update the position, which can be searched more accurately and easily approach the optimal solution. Eq. (15) represents the exploitation stage location update formula.

$$x_{i,j}(C\_Iter) = \begin{cases} best(x_j) \div (MOP + \in) \times \left((UB_j - LB_j) \times \mu + LB_j\right), & r2 < 0.5 \\ best(x_j) \times MOP \times \left((UB_j - LB_j) \times \mu + LB_j\right), & otherwise \end{cases} \quad (14)$$

$$x_{i,j}(C\_Iter) = \begin{cases} best(x_j) - MOP \times \left((UB_j - LB_j) \times \mu + LB_j\right), & r3 < 0.5 \\ best(x_j) + MOP \times \left((UB_j - LB_j) \times \mu + LB_j\right), & otherwise \end{cases} \quad (15)$$

where, $best(x_j)$ is the $j-th$ position of the best solution obtained. $x_{i,j}(C\_Iter)$ is the $j-th$ position of the $i-th$ solution. $r2$ and $r3$ are random numbers in the range [0,1]. UB_j and LB_j respectively represent the upper and lower bound values. $\in$ is A small integer, μ is the parameter that controls the search of the algorithm, $\mu = 0.5$.

TABLE 4. Function optimization simulation results.

F	Meteic	AOA	sinCAOA	sinhCAOA	asinhCAOA	tanhCAOA	atanCAOA	atanhCAOA
	Ave	6.3E+09	379672.9	5649652	162086.8	203899.7	292719.7	491981.3
F_1	Std	4.44E+09	424479	3279893	72873.5	171412.5	246180.7	882925.4
	Rank	7	4	6	1	2	3	5
	Ave	5.03E+09	5007.522	50964.64	2870.511	6382.506	4549.397	10166.76
F_2	Std	8.57E+09	7106.149	114673.7	2029.81	9420.396	4586.021	10048.91
	Rank	7	3	6	1	4	2	5
	Ave	7422.805	341.8016	504.0709	369.6911	360.9139	361.2451	359.9522
F_3	Std	2506.139	24.38169	76.24781	44.67225	48.36371	31.78457	29.04602
	Rank	7	1	6	5	3	4	2
	Ave	800.3586	409.7933	408.4191	411.4283	407.3283	405.449	419.3858
F_4	Std	449.6062	14.60431	2.128481	19.5512	1.084275	3.154915	37.80818
	Rank	7	4	3	5	2	1	6
	Ave	544.6486	535.1885	528.9432	527.4528	533.102	536.2961	529.9428
F_5	Std	20.72376	6.800326	7.875667	7.008136	10.13005	11.23671	10.14575
	Rank	7	5	2	1	4	6	3
	Ave	633.5853	617.8828	614.9164	614.6376	620.1421	615.0648	614.8406
F_6	Std	8.845589	10.67271	9.285591	6.785115	10.49224	7.2223	9.297367
	Rank	7	5	3	1	6	4	2
	Ave	799.276	771.6775	765.9595	758.3506	765.0855	772.1387	772.0159
F_7	Std	18.76039	18.99144	15.89493	13.55157	17.01304	16.3987	21.31055
	Rank	7	4	3	1	2	6	5
	Ave	830.0305	822.6346	826.8151	821.5285	823.7998	825.1086	823.1878
F_8	Std	6.15537	7.132221	7.689622	6.640807	8.846827	7.470316	6.637383
	Rank	7	2	6	1	4	5	3
	Ave	1349.055	1001.133	1039.327	1084.942	1031.357	1107.86	1054.754
F_9	Std	192.6892	147.3704	166.734	152.5915	151.9925	147.0071	181.882
	Rank	7	1	3	5	2	6	4
	Ave	2159.813	1734.299	1539.052	1684.131	1748.217	1719.388	1630.652
F_{10}	Std	245.8042	221.9498	186.4666	206.3521	184.3196	311.138	242.6701
	Rank	7	5	1	3	6	4	2
	Ave	1345.435	1123.339	1147.309	1127.162	1125.008	1123.434	1144.239
F_{11}	Std	262.8355	6.491929	55.84425	3.38549	5.932564	7.73986	63.02191
	Rank	7	1	6	4	3	2	5
	Ave	2219827	532599	377079.3	96015.42	1281412	291085.7	1777398
F_{12}	Std	2990782	554003.9	353450.2	92524.76	1750951	266673.1	2634649
	Rank	7	4	3	1	5	2	6
	Ave	14809.29	10739.33	15669.46	10256.94	5684.904	11285.32	17064.15
F_{13}	Std	10990.49	7182.184	13997.44	9141.459	7792.036	8968.808	13340.63
	Rank	5	3	6	2	1	4	7
F_{14}	Ave	13622.24	6126.88	5133.786	4500.298	3651.678	6539.886	6719.908

TABLE 4. ***(Continued.)*** **Function optimization simulation results.**

	Std	9533.483	5040.693	4310.057	3176.965	1349.681	5908.542	8442.825
	Rank	7	4	3	2	1	5	6
F_{15}	Ave	16236.16	5456.366	4024.376	4345.879	3191.567	5998.139	6213.878
	Std	6809.298	5477.71	1897.947	3236.104	878.9692	4415.593	7055.641
	Rank	7	4	2	3	1	5	6
F_{16}	Ave	2011.164	1787.801	1806.226	1772.513	1781.057	1814.413	1765.71
	Std	77.72548	95.48166	154.966	112.4796	96.01	141.3364	146.6365
	Rank	7	4	5	2	3	6	1
F_{17}	Ave	1819.95	1764.685	1755.851	1773.619	1772.167	1795.043	1754.911
	Std	60.52791	19.71587	19.06427	37.18435	19.9957	40.92461	8.55306
	Rank	7	3	2	5	4	6	1
F_{18}	Ave	16872.12	15331.09	21240.96	17049.42	22517.48	20932.76	20481.55
	Std	15035.25	10377.98	14105.46	12928.15	16437.09	12273.19	13274.8
	Rank	2	1	6	3	7	5	4
F_{19}	Ave	15996.99	11202.78	9164.531	9517.06	10226.22	3985.209	6547.563
	Std	21782.91	12237.47	7501.615	8996.282	10415.79	2748.238	5398.001
	Rank	7	6	3	4	5	1	2
F_{20}	Ave	2124.24	2071.755	2083.774	2076.355	2088.761	2096.979	2086.619
	Std	62.19807	53.49242	63.23409	62.70877	54.96277	59.68068	59.76797
	Rank	7	1	3	2	5	6	4
F_{21}	Ave	2310.567	2272.037	2215.85	2262.367	2256.068	2248.269	2262.283
	Std	49.40488	59.08371	6.560978	64.00019	57.20936	56.42777	59.85452
	Rank	7	6	1	5	3	2	4
F_{22}	Ave	2769.713	2309.992	2352.644	2312.558	2307.646	2311.816	2311.59
	Std	168.1829	2.661702	173.9721	6.272488	4.147465	2.446995	4.581923
	Rank	7	2	6	5	1	4	3
F_{23}	Ave	2718.756	2651.948	2642.572	2660.356	2645.958	2648.382	2647.519
	Std	31.59606	14.35437	13.94746	19.21799	12.92127	18.0011	9.629156
	Rank	7	5	1	6	2	4	3
F_{24}	Ave	2828.272	2739.98	2776.398	2734.726	2681.527	2741.459	2780.661
	Std	84.39012	106.3219	11.37628	91.53474	130.3443	90.63813	21.661
	Rank	7	3	5	2	1	4	6
F_{25}	Ave	3136.214	2927.706	2929.876	2925.098	2938.29	2928.798	2884.703
	Std	131.3258	25.15062	23.33947	23.8355	20.61104	24.45904	98.19935
	Rank	7	3	5	2	6	4	1
F_{26}	Ave	3711.105	3129.264	2964.652	3048.847	3171.307	2979.419	3189.776
	Std	253.0218	360.2559	87.29277	197.6423	306.8777	167.8578	272.514
	Rank	7	4	1	3	5	2	5
F_{27}	Ave	3225.538	3096.409	3098.143	3097.573	3100.728	3100.083	3101.81
	Std	46.04964	3.614307	4.542175	3.951869	8.758808	4.947012	15.3375
	Rank	7	1	3	2	5	4	6
F_{28}	Ave	3709.066	3236.078	3346.329	3274.462	3284.228	3293.769	3285.053
	Std	132.0435	94.26914	99.8656	97.44954	106.8563	102.1561	110.4849

TABLE 4. *(Continued.)* **Function optimization simulation results.**

	Rank	7	1	6	2	3	5	4
	Ave	3337.745	3238.37	3199.708	3216.709	3221.358	3225.843	3236.021
F_{29}	Std	94.86375	47.17818	23.58181	59.45622	45.1341	37.3858	60.91602
	Rank	7	6	1	2	3	4	5
	Ave	6379844	471604.1	249192.6	234662.5	482956.6	724044.6	466488.8
F_{30}	Std	9599192	465569	295072	517721.2	475021	443630.5	472825.5
	Rank	7	4	2	1	5	6	3
Mean Rank		6.77	3.33	3.63	2.73	3.47	4.07	3.97
Final Ranking		7	2	4	1	3	6	5

B. SINE OPTIMIZATION ALGORITHM

The Sine Optimization algorithm (SOA) was proposed by Mostafa Meshkat in 2017. This algorithm only uses the sine function as the position update strategy, which enables the algorithm to have higher optimization accuracy and faster convergence rate [34]. The search process in an SOA algorithm is divided into two phases, exploration and development, whose location update formula is shown below.

$$x_i^t(j+1) = \begin{cases} X_{rand}^t(j) + r_1 \times \sin(r_2), & r_3 < 0.5 \\ X_{best}^t(j) + r_1 \times \sin(r_2), & r_3 \geq 0.5 \end{cases} \tag{16}$$

where, $X_{rand}^t(j)$ is the position of random search individuals, and $X_{best}^t(j)$ is the position of optimal search individuals. r_1 represents the direction of movement, which can be calculated by equation (17). r_2 is the random number in the range of $[0, 2\pi]$, and r_3 is the random number between $[0, 1]$. a is a constant with the value 2.

$$r_1 = a - t\frac{a}{T} \tag{17}$$

C. INVERSE ACCUMULATION FUNCTION OF CAUCHY DISTRIBUTION

Cauchy distribution is a continuous probability distribution named after Augustine Louis Cauchy and Hendrik Lorentz [35]. The expression of its probability density function is Eq. (18).

$$f(x; a, b) = \frac{1}{\pi}\left[\frac{b}{(x-a)^2 + b^2}\right] \tag{18}$$

The formula for calculating the cumulative distribution function of Cauchy distribution is:

$$f(x; a, b) = \frac{1}{\pi}\arctan\left(\frac{x-a}{b}\right) + \frac{1}{2} \tag{19}$$

The inverse function of the cumulative distribution function of the Cauchy distribution is shown as follows:

$$f^{-1}(p; a, b) = a + b\tan\left(\pi\left(p - \frac{1}{2}\right)\right) \tag{20}$$

$$p = rand(1, \dim) \tag{21}$$

where, p is a random number in the range of [0,1]; dim is the dimension of the function, and the positional parameter a is assigned 0 and the parameter b is assigned 1.

D. ARITHMETIC OPTIMIZATION ALGORITHM BASED ON CAUCHY MUTATION TRIGONOMETRIC FUNCTION SEARCH

MOP is an important parameter that controls the update of AOA position. The original MOP parameter gradually reduced with the progress of iteration, and the MOP value will become very small after the algorithm is run, which makes it difficult for the algorithm to jump out of the local optimal. In this paper, the oscillation coefficient is considered to combine with Cauchy mutation, in which the Cauchy mutation is the inverse accumulation function of Cauchy distribution, enriching its variation trend. Then, the k_2 is generated by the oscillation Cauchy mutation is used to replace MOP, which will change with the oscillation Cauchy mutation in each iteration, thus strengthening the ability to get out of the local optimal and improving the convergence speed of the algorithm. The expression of the oscillating Cauchy mutation operator is shown as follows:

$$k_1 = 1.5 - \frac{C\_iter}{M\_iter} + 0.5\sin\left(10\pi \times \frac{C\_iter}{M\_iter}\right) \times \left(1 - \frac{C\_iter}{M\_iter}\right) \tag{22}$$

$$k_2 = k_1 \times \tan(\pi(rand - 0.5)) \tag{23}$$

where, k_1 is the oscillation coefficient, and k_2 is the parameter generated by oscillation Cauchy mutation. Fig. 3 shows the variation of oscillation coefficient after 1000 iterations, where x axis is the number of iterations and y axis is the size of k_1 value. Fig. 4 shows the motion track of oscillation Cauchy mutation after 1000 iterations, and the two axes are the amount of displacement in both directions.

The value of control parameter μ in the AOA position update formula is 0.5. In this thesis, the k_4 is generated by the sinusoidal search operator is used to replace the original fixed value parameter, which can enhance the search capability of the AOA, so that the AOA still has a certain global exploration capability in the late running. On the basis of introducing sinusoidal search operators, we also put forward some

TABLE 5. Performance comparison results with different intelligent optimization algorithms.

F	Meteic	AOA	asinhCAOA	HHO	CMA-ES	GSA	FDO	GA	SSA
	Ave	4.81E+09	89374.59	3.6428E+05	100.00	296.0	2.1193E+10	9799.7	3396.25
F_1	Std	2.62E+09	50492.39	1.5316E+05	0.000	275.1	1.6865E+09	5942.54	3673.08
	Rank	7	5	6	1	2	8	4	3
	Ave	6971.252	366.0477	301.44	300.00	10829.2	1.5382E+04	8721.4	300.00
F_3	Std	2346.377	36.42204	1.0172	0.000	1620.74	260.47	5900.30	0.00
	Rank	5	4	3	1	7	8	6	1
	Ave	648.9165	404.2003	409.96	400.00	406.6	3458.9	410.71	406.27
F_4	Std	201.9786	2.609539	16.826	0.000	2.92	251.24	18.512	10.07
	Rank	7	2	5	1	4	8	6	3
	Ave	539.4426	529.7581	546.51	530.18	556.7	634.46	516.32	521.82
F_5	Std	20.54336	6.840792	17.142	58.32	8.40	5.6416	6.926	10.50
	Rank	5	3	6	4	7	8	1	2
	Ave	638.5729	614.0404	628.56	682.1	621.6	657.34	600.04	609.77
F_6	Std	6.923598	5.610863	14.009	35.43	9.015	1.3673	0.0668	8.26
	Rank	6	3	5	8	4	7	1	2
	Ave	793.3362	762.8876	786.43	713.4	714.6	808.41	728.32	740.88
F_7	Std	11.1881	21.93916	19.5713	1.63	1.55	8.1592	7.290	16.62
	Rank	7	5	6	1	2	8	3	4
	Ave	831.7577	826.5199	829.08	828.9	820.5	840.41	820.72	823.45
F_8	Std	5.610561	1.624062	9.1952	52.98	4.69	1.7563	8.961	9.95
	Rank	7	4	6	5	1	8	2	3
	Ave	1323.734	970.96	1399.5	4667.3	900.0	1653.1	910.28	944.07
F_9	Std	307.2437	34.83224	247.32	2062.8	6.9E-14	42.672	15.154	104.66
	Rank	5	4	6	8	1	7	2	3
	Ave	1964.968	1520.727	1985.9	2588.1	2694.6	3608.1	1723.3	1858.85
F_{10}	Std	232.7142	194.1624	275.46	414.47	297.62	171.89	252.34	294.50
	Rank	5	1	4	6	7	8	2	3
	Ave	3147.222	1122.435	1153.1	1111.3	1134.7	2.4642E+06	1125.6	1180.5
F_{11}	Std	4310.589	6.09369	44.932	25.44	10.45	3.8727E+06	23.80	59.80
	Rank	7	2	5	1	4	8	3	6
	Ave	572999.5	221494.9	3.3976E+06	1629.6	702723	2.3930E+09	37255	1983166
F_{12}	Std	330137	425765.1	3.8744E+06	198.11	42075.4	5.7508E+08	34792.7	1909901
	Rank	4	3	7	1	5	8	2	6
	Ave	3062.083	2708.721	1553.9	1452.1	7147.5	3.0406E+08	7048.9	1508.94
F_{14}	Std	1361.791	1614.508	146.42	55.98	1489.52	2.8026E+08	8160.08	51.05
	Rank	5	4	3	1	7	8	6	2
	Ave	13981.45	3276.259	3077.3	1509.6	18001	1.0541E+07	9296.2	2236.69
F_{15}	Std	6787.304	1167.299	1286.7	16.43	5498.67	2.0198E+07	8978.18	571.19
	Rank	6	4	3	1	7	8	5	2
F_{16}	Ave	2047.573	1697.204	1882.3	1815.3	2149.7	2400.6	1786.3	1726.26
	Std	120.3839	103.4412	120.58	230.1	105.8	72.087	129.07	126.97

TABLE 5. ***(Continued.)*** **Performance comparison results with different intelligent optimization algorithms.**

	Rank	6	1	5	4	7	8	3	2
	Ave	1901.083	1745.839	1779.7	1830.1	1857.7	1935.0	1746.5	1774.57
F_{17}	Std	81.39328	13.15915	35.434	175.8	108.32	101.78	39.78	34.23
	Rank	7	1	4	5	6	8	2	3
	Ave	17501.28	10677.94	1.4672E+04	1825.9	8720.5	7.0988E+09	15721	23429.1
F_{18}	Std	9161.536	5952.523	1.2135E+04	13.53	5060.1	1.0187E+09	12828	14045.7
	Rank	6	3	4	1	2	8	5	7
	Ave	38413.33	5797.537	8954.2	1920.5	13670	3.2867E+09	9686.5	2916.1
F_{19}	Std	29914.92	5751.063	7966.2	28.68	19168	2.4058E+09	6766.3	1871.2
	Rank	7	3	4	1	6	8	5	2
	Ave	2165.552	2049.754	2126.3	2494.8	2272.3	2237.6	2056.5	2089.3
F_{20}	Std	49.89107	11.22543	60.591	242.65	81.72	20.487	60.01	49.28
	Rank	5	1	4	8	7	6	2	3
	Ave	2307.233	2233.325	2310.4	2324.7	2357.7	2527.9	2303.8	2249.8
F_{21}	Std	47.2523	58.09804	66.374	67.76	28.20	60.44	43.75	60.44
	Rank	4	1	5	6	7	8	3	2
	Ave	2775.807	2310.586	2312.2	3532.4	2300.0	4210.7	2304.6	2301.5
F_{22}	Std	184.4506	3.803339	15.194	847.6	0.072	261.99	2.38	11.80
	Rank	6	3	5	7	1	8	4	2
	Ave	2700.164	2647.702	2659.2	2728.8	2736.5	3332.9	2632.9	2621.7
F_{23}	Std	25.08189	16.46436	24.988	243.1	39.14	166.06	13.42	8.69
	Rank	5	3	4	6	7	8	2	1
	Ave	2828.447	2683.855	2791.9	2704.4	2742.2	3206.8	2758.3	2733.2
F_{24}	Std	57.0816	144.395	85.570	73.42	5.52	37.169	14.92	64.43
	Rank	7	1	6	2	4	8	5	3
	Ave	3189.873	2919.041	2923.4	2932.01	2937.5	3783.6	2947.9	2923.5
F_{25}	Std	52.50918	26.43326	48.503	20.87	15.36	247.31	19.25	23.86
	Rank	7	1	2	4	5	8	6	3
	Ave	3995.715	2996.4	3288.3	3457.7	34407.5	5085.1	3112.1	2900.9
F_{26}	Std	262.758	104.0844	459.24	598.9	628.73	125.07	334.65	36.56
	Rank	6	2	4	5	8	7	3	1
	Ave	3225.731	3096.39	3151.0	3137.5	3259.5	3824.8	3115.1	3092.6
F_{27}	Std	52.75979	1.663944	42.197	21.37	41.66	349.40	19.18	2.78
	Rank	6	2	5	4	7	8	3	1
	Ave	3676.381	3253.608	3389.5	3397.6	3459.4	3973.3	3320.7	3210.5
F_{28}	Std	204.2122	91.84942	128.84	131.3	33.84	270.36	126.34	113.17
	Rank	7	2	4	5	6	8	3	1
	Ave	3363.131	3209.198	3308.7	3213.5	3449.5	5734.4	3253.5	3214.1
F_{29}	Std	89.5474	19.28315	70.303	109.79	171.33	880.61	81.99	51.69
	Rank	6	1	5	2	7	8	4	3
Mean Rank		5.75	2.46	4.5	3.54	4.93	7.54	3.32	2.64
Final Ranking		7	1	5	4	6	8	3	2

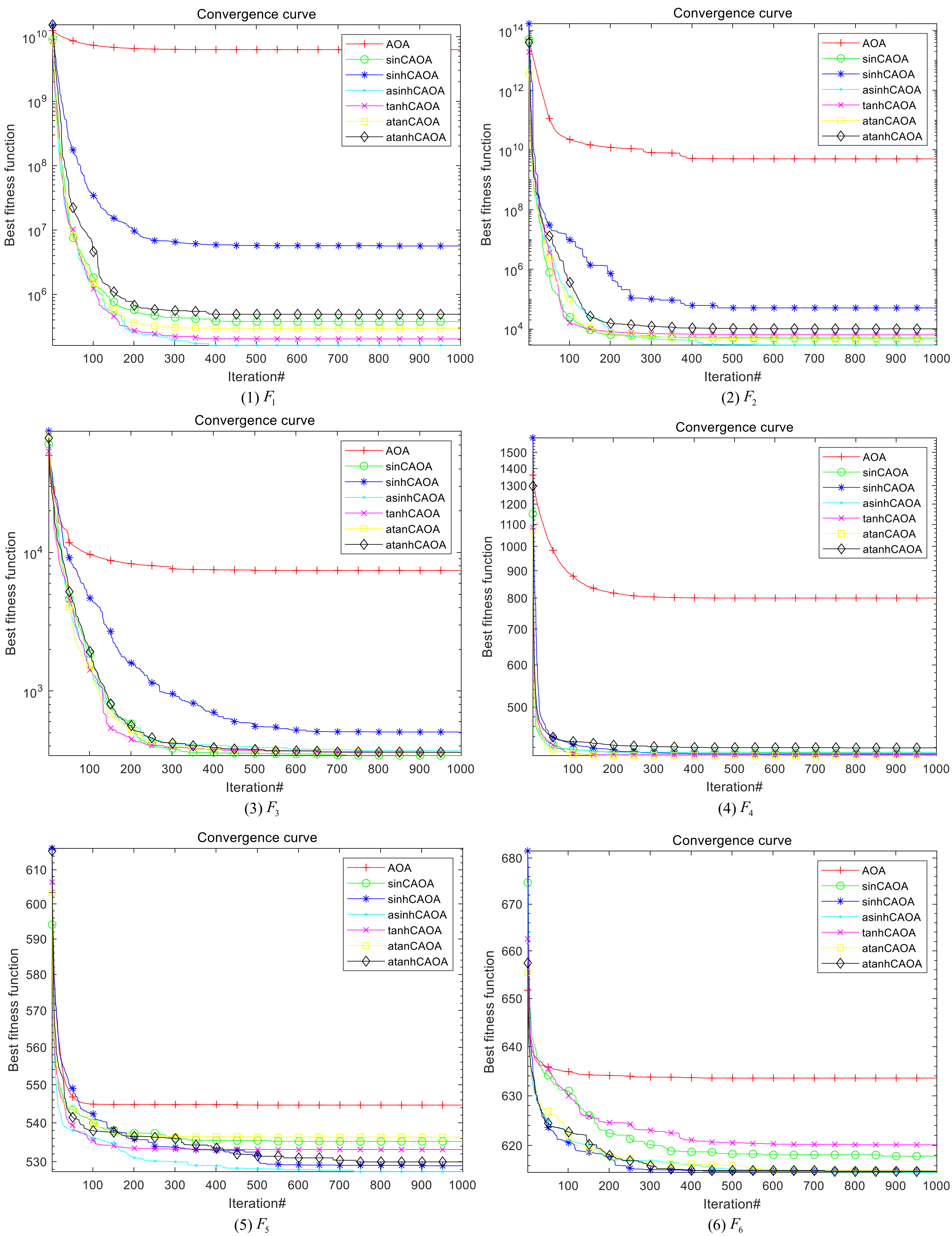


FIGURE 7. **Results of function convergence curves.**

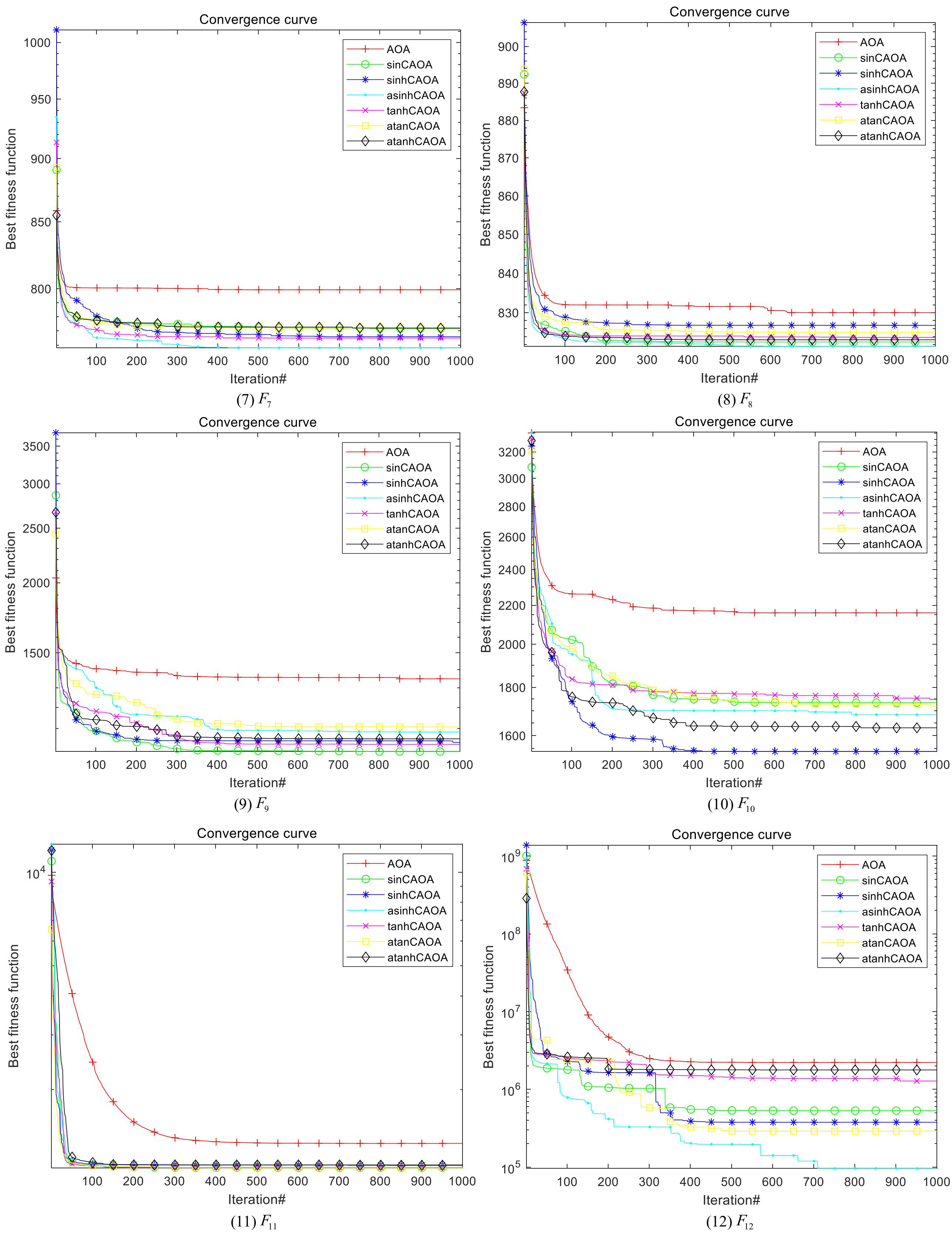
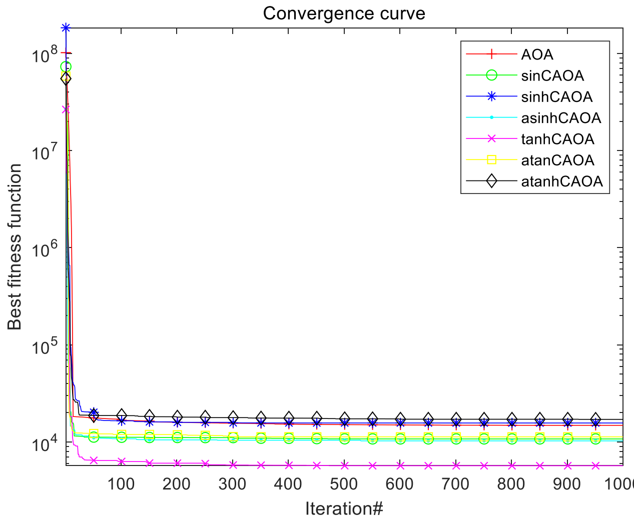
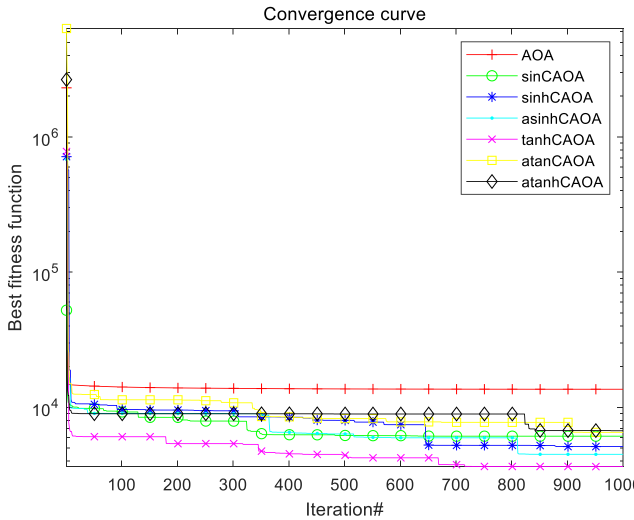


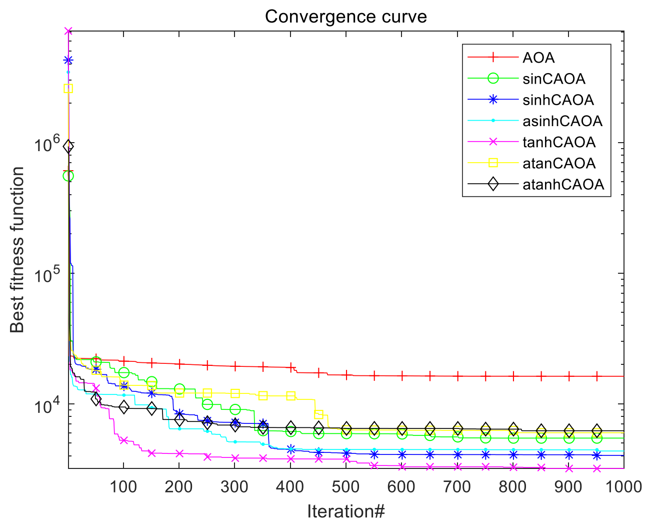
FIGURE 7. ***(Continued.)*** **Results of function convergence curves.**



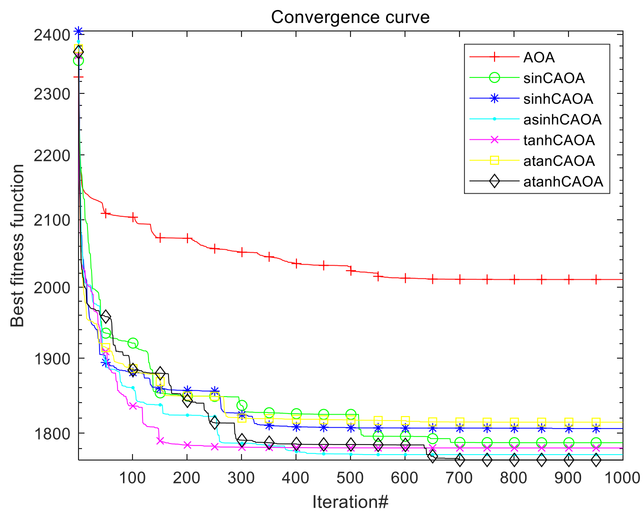
(13) F_{13}



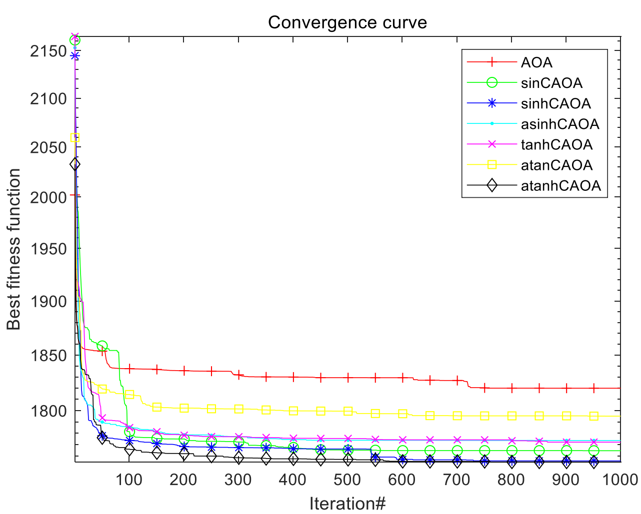
(14) F_{14}



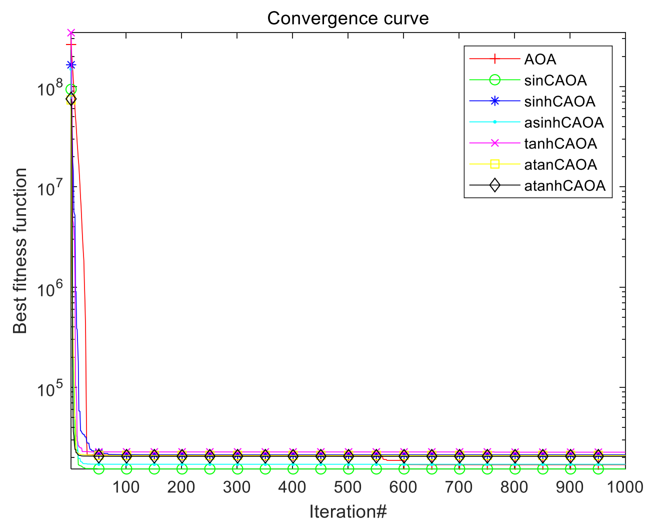
(15) F_{15}



(16) F_{16}



(17) F_{17}



(18) F_{18}

FIGURE 7. ***(Continued.)*** **Results of function convergence curves.**

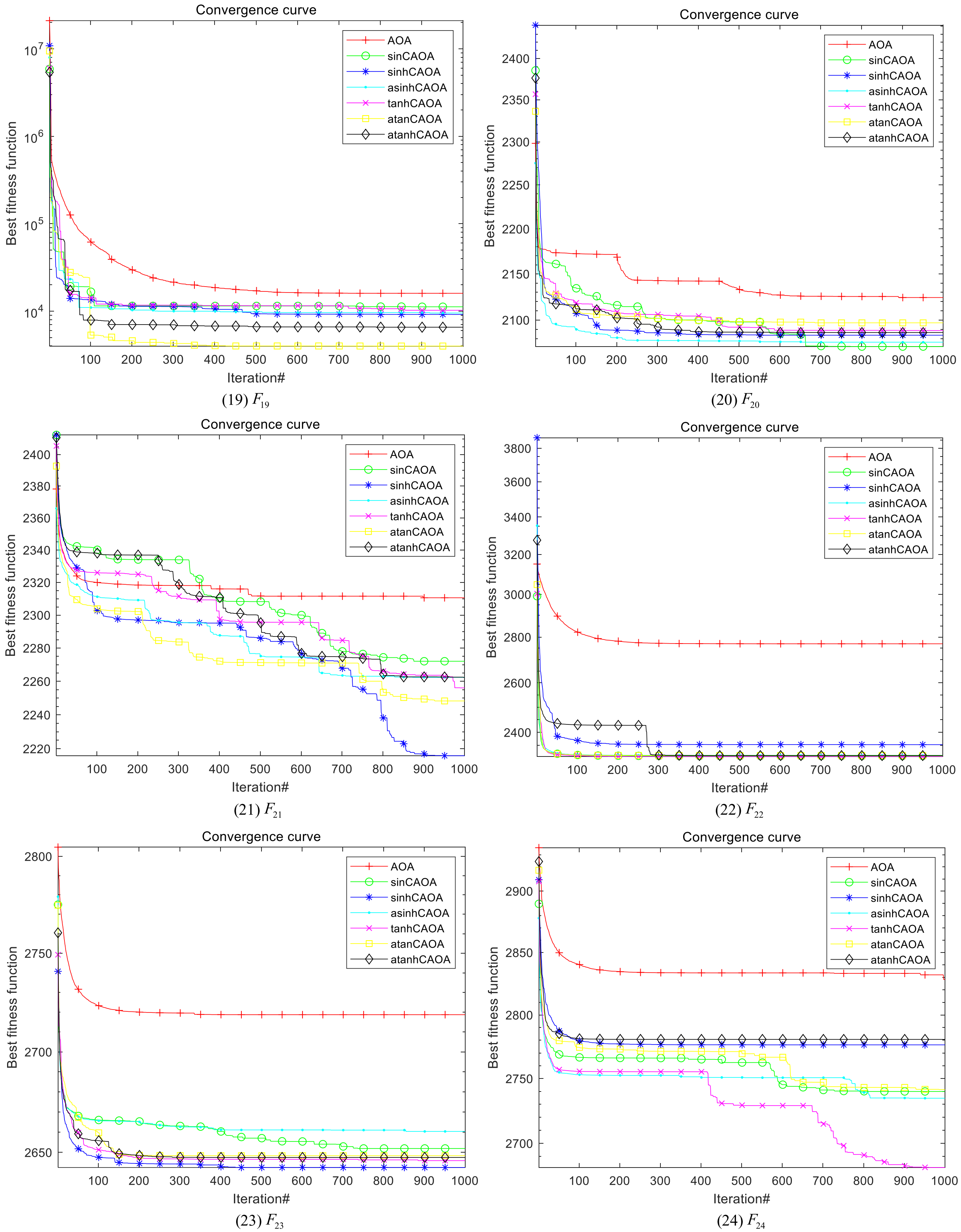
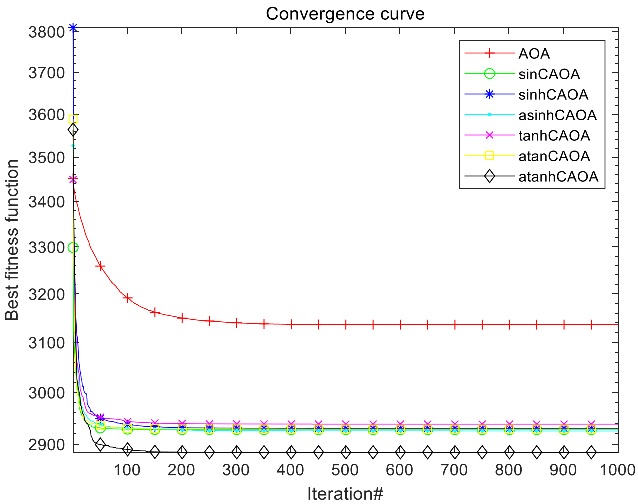
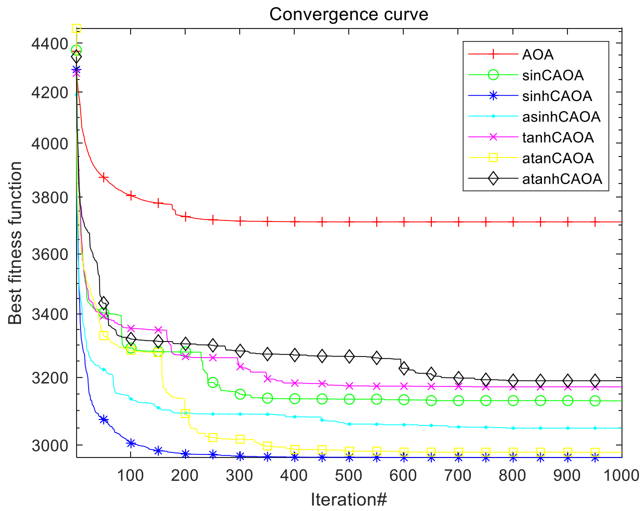


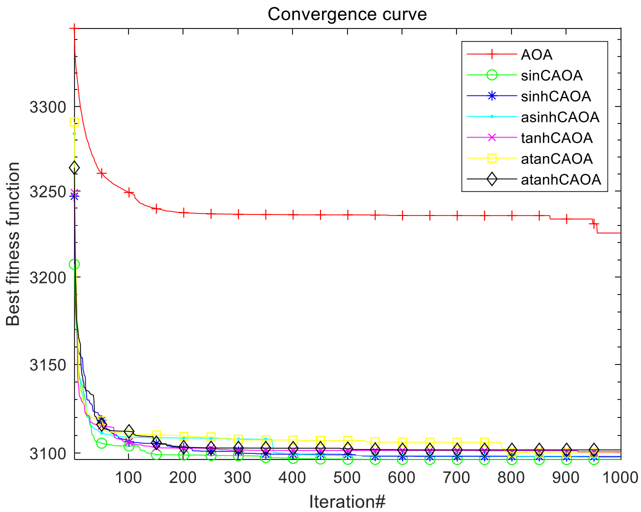
FIGURE 7. ***(Continued.)*** **Results of function convergence curves.**



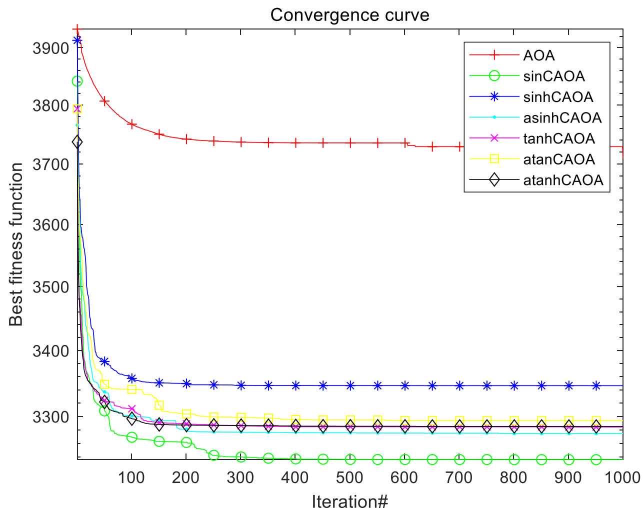
(25) F_{25}



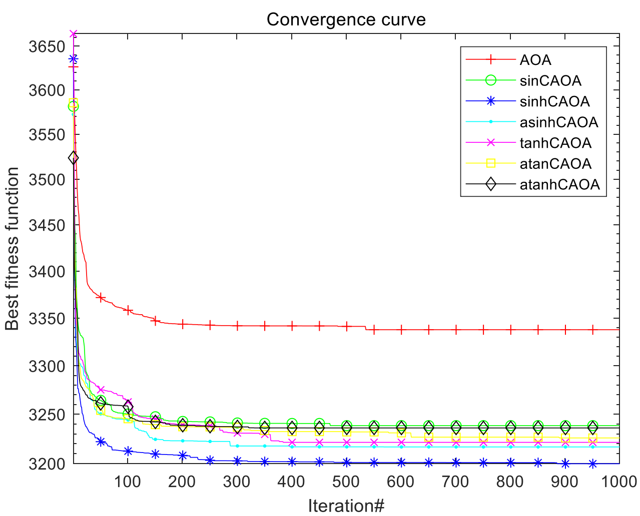
(26) F_{26}



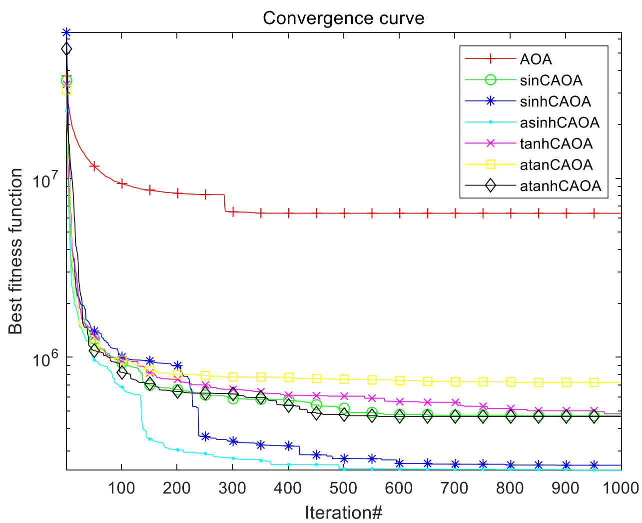
(27) F_{27}



(28) F_{28}



(29) F_{29}



(30) F_{30}

FIGURE 7. ***(Continued.)*** **Results of function convergence curves.**

improved methods similar to sinusoidal search operators, these methods are hyperbolic sine, inverse hyperbolic sine, hyperbolic tangent, arctangent, inverse hyperbolic tangent search operators. The parameter changes generated by them have different effects on the algorithm effect. Fig. 6 shows the images of six trigonometric functions, and Table 2 shows the parameter Settings of six improved methods. Combined with the above two improved position update expressions are as follows ($r2$ and $r3$ are random numbers in the range [0,1]):

$$x_{i,j}\left(C\_Iter\right) = \begin{cases} best\left(x_j\right) \div (k_2 + \in) \times \left(\left(UB_j - LB_j\right) \times k_4 + LB_j\right), & r2 < 0.5 \\ best\left(x_j\right) \times k_2 \times \left(\left(UB_j - LB_j\right) \times k_4 + LB_j\right), & otherwise \end{cases} \quad (24)$$

$$x_{i,j}\left(C\_Iter\right) = \begin{cases} best\left(x_j\right) - k_2 \times \left(\left(UB_j - LB_j\right) \times k_4 + LB_j\right), & r3 < 0.5 \\ best\left(x_j\right) + k_2 \times \left(\left(UB_j - LB_j\right) \times k_4 + LB_j\right), & otherwise \end{cases} \quad (25)$$

IV. FUNCTION OPTIMIZATION TEST SIMULATION

A. BENCHMARK TEST FUNCTIONS

In this section, the CEC2017 test functions are used to check the performance of the AOA based on Cauchy mutation trigonometric function search. In order to observe the effect of the six improved methods, the improved schemes that add sine function, hyperbolic sine, inverse hyperbolic sine, hyperbolic tangent, arctan and inverse hyperbolic tangent search operators are denoted as sinCAOA, sinhCAOA, asinhCAOA, tanhCAOA, atanCAOA and atanhCAOA, respectively. The relevant information of CEC2017 benchmark Functions is shown in Table 3, including Unimodal, Multimodal, Hybrid and Composition Functions. The two parameters α and μ in AOA are $\alpha = 5$, $\mu = 0.5$. In terms of ensuring the fairness of the experiment, each algorithm runs 10 times independently when optimizing the benchmark function, averages the optimal value obtained after optimization, and compares the effect of the improved method with the calculated average value. At the same time, the population of each method is 30, and the number of iterations is 1000.

B. SIMULATION RESULTS

Table 4 shows the result data of the six improved methods and the original AOA optimization of CEC2017 benchmark test functions. From the comparison of the average value obtained from the optimized test function in the table, it can be found that the six improved methods sinCAOA, sinhCAOA, asinhCAOA, tanhCAOA, atanCAOA and atanhCAOA are all better than the optimization results of AOA algorithm, which proves that the improved method is effective. Among the six improvement methods, asinhCAOA method has the best comprehensive effect. Next, AOA and asinhCAOA were used

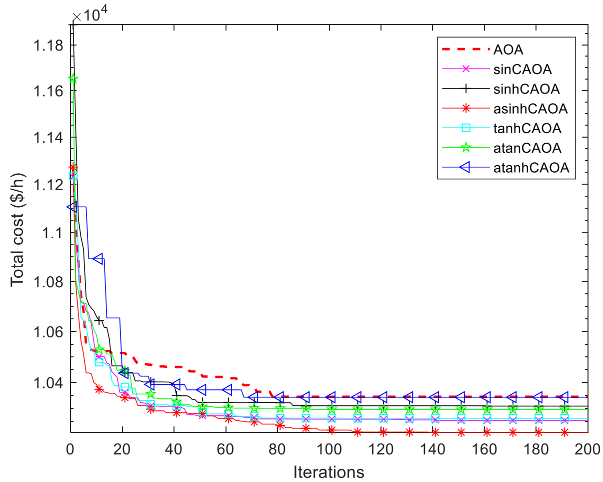


FIGURE 8. The convergence curve of solving the 150MW demand CEED problem.

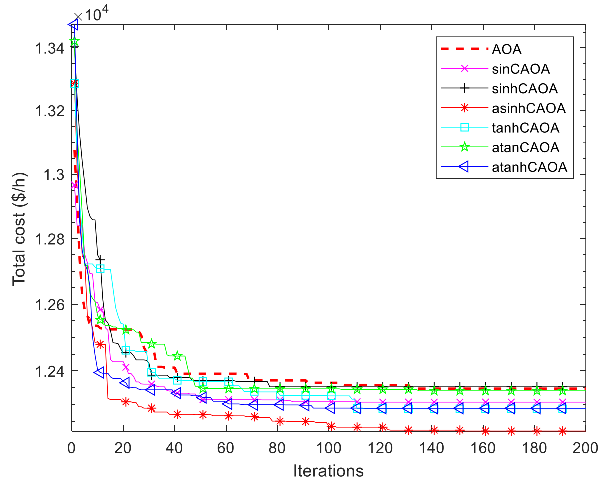


FIGURE 9. The convergence curve of solving the 175MW demand CEED problem.

to compare with Harris hawk optimizer (HHO) [36], Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [37], Gravitational Search Algorithm (GSA) [38], Fitness-dependent Optimizer (FDO) [39], Genetic Algorithm (GA) [40] and Salp Swarm Algorithm (SSA) [41], and the same CEC2017 test functions as above are used for comparison. The performance data of the test functions optimized by HHO, CMA-ES, GSA, FDO, GA and SSA algorithms are quoted from literature [42] and [43]. In order to ensure the fairness of the experiment, the number of populations and iterations set by AOA and asinhCAOA are the same as those of the optimization algorithms used for comparison, which are 50 and 1000 respectively. Each group of experiments is run independently for 10 times, and the minimum value of each optimized test function is averaged, and the value is compared to judge the performance of the algorithm. The comparison data of the simulation experiment is shown in Table 5. It can be clearly seen from the data in Table 5 that

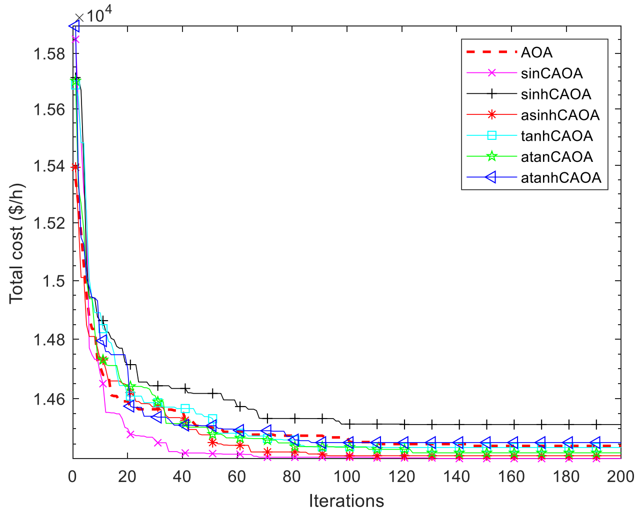


FIGURE 10. The convergence curve of solving the 200MW demand CEED problem.

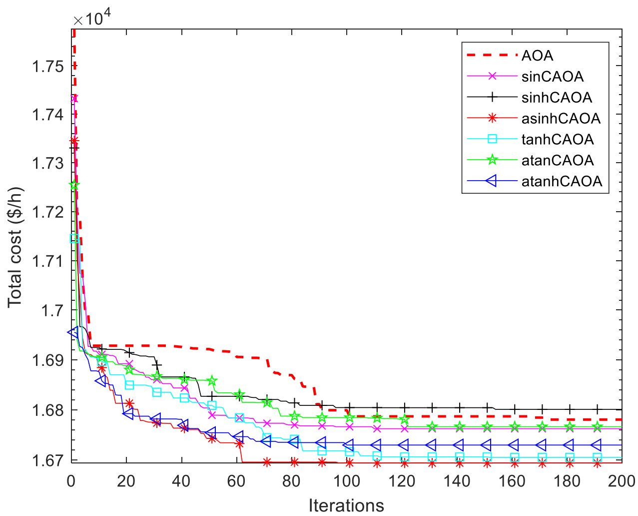


FIGURE 11. The convergence curve of solving the 225MW demand CEED problem.

asinhCAOA obtains the least optimal value when optimizing the F_{10}, F_{16}-F_{17}, F_{20}-F_{21}, F_{24}-F_{25}, F_{29} benchmark functions, and also has a better performance when optimizing other functions. The improved method of asinhCAOA can obtain a smaller optimal value by comprehensively comparing the algorithms of HHO, CMA-ES, GSA, FDO, GA and SSA, which better verifies the effectiveness of the improved method.

V. SIMULATION EXPERIMENT OF CEED PROBLEM

In this section, the experiment is divided into two parts. First, it is verified that the AOA based on Cauchy mutation trigonometric function search can achieve better optimization effect in solving CEED problems compared with the AOA algorithm, and the method with the best improvement effect is selected by comparing the experimental data results. Next, verify whether the effect of using the random disturbance penalty function is improved compared to the fixed-value penalty strategy. The CEED problem cases of 6 units is used, and the load demands is set as 150 MW, 175 MW, 200 MW and 225 MW for experiments. In the following experiment, the arguments of AOA algorithm are set as $\alpha = 5$, $\mu = 0.5$.

TABLE 6. Power limits and fuel cost coefficient of generators.

Unit	a_i (10^{-3})	b_i	c_i (10^2)	d_i (10^3)	P_{min}	P_{max}
1	0.1000	0.0920	0.1450	-0.1360	50	200
2	0.4000	0.0250	0.2200	-0.0035	20	80
3	0.6000	0.0750	0.2300	-0.0810	15	50
4	0.2000	0.1000	0.1350	-0.0145	10	50
5	0.1300	0.1200	0.1150	-0.0098	10	50
6	0.4000	0.0840	0.1250	0.0756	12	40

TABLE 7. The coefficient of no_x emission and max/max penalty factor.

Unit	e_{NOxi}	f_{NOxi}	g_{NOxi}	h_{NOxi}	h_N
1	0.0012	0.0520	18.50	-26.00	0.9407
2	0.0004	0.0450	12.00	-35.00	1.4962
3	0.0016	0.0500	13.00	-15.00	1.3870
4	0.0012	0.0700	17.50	-74.00	0.8308
5	0.0003	0.0400	8.50	-89.00	2.1705
6	0.0014	0.0240	15.50	-75.00	1.0930

TABLE 8. The coefficient of co_2 emission and max/max penalty factor.

Unit	e_{CO2i}	f_{CO2i}	g_{CO2i}	h_{CO2i}	h_C
1	0.0015	0.0920	14.00	-16.00	0.7823
2	0.0014	0.0250	12.50	-93.50	1.1895
3	0.0016	0.0550	13.50	-85.00	1.4356
4	0.0012	0.0100	13.50	-24.50	1.1333
5	0.0023	0.0400	21.00	-59.00	0.7456
6	0.0014	0.0800	22.00	-70.00	0.7158

TABLE 9. The coefficient of so_2 emission and max/max penalty factor.

Unit	e_{SO2i}	f_{SO2i}	g_{SO2i}	h_{SO2i}	h_S
1	0.0005	0.150	17.00	-90.00	1.0852
2	0.0014	0.055	12.00	-30.50	1.0616
3	0.0010	0.035	10.00	-80.00	2.1051
4	0.0020	0.070	23.50	-34.50	0.5976
5	0.0013	0.120	21.50	-19.75	0.6772
6	0.0021	0.080	22.50	25.60	0.6192

TABLE 10. Experimental data for solving the 150mw demand ceed problem.

	AOA	sinCAOA	sinhCAOA	asinhCAOA	tanhCAOA	atanCAOA	atanhCAOA
P_1	50	50	50	50	50	50	50
P_2	20	20	20	20	20.19944	20	20
P_3	15	15	15	15	15	15	15
P_4	11.86382	28.03583	23.59578	10.20696	19.72084	32.59998	24.22102
P_5	36.57987	12.86178	12.91201	24.33985	13.71956	16.14233	12.12747
P_6	16.9582	24.25655	28.59542	30.49611	31.39614	16.4383	28.92076
F_{Cost} (\$/h)	2615.708	2603.32	2596.553	2587.416	2595.12	2616.161	2602.732
E_{CO2} (kg/h)	2761.741	2556.066	2610.873	2754.108	2657.809	2501.843	2612.574
E_{NOx} (kg/h)	2239.041	2462.6	2444.17	2320.669	2426.787	2466.71	2457.548
E_{SO2} (kg/h)	3210.268	3172.695	3170.877	3175.35	3173.151	3187.189	3180.616
F_{Total} (\$/h)	10344.46	10258.25	10303.57	10237.21	10293.27	10282.66	10301.89

TABLE 11. Experimental data for solving the 175mw demand ceed problem.

	AOA	sinCAOA	sinhCAOA	asinhCAOA	tanhCAOA	atanCAOA	atanhCAOA
P_1	50	50	50	50	50	50	50
P_2	25.95509	31.144	20.79557	20	20	20	20.56601
P_3	15	16.94124	15	15	15	15	17.81973
P_4	23.52634	37.22361	29.56719	34.27872	28.97005	19.27156	38.21895
P_5	20.98749	11.69035	25.08684	20.02265	21.14093	44.68826	13.52578
P_6	40	28.20268	34.56364	36.25066	39.90974	27.16255	35.04209
F_{Cost} (\$/h)	3088.071	3164.701	3040.895	3065.758	3051.409	3088.872	3111.318
E_{CO2} (kg/h)	3271.519	3026.012	3214.129	3208.132	3267.641	3441.698	3140.42
E_{NOx} (kg/h)	2864.287	2966.072	2850.258	2943.067	2904.983	2688.977	3010.858
E_{SO2} (kg/h)	3888.754	3802.055	3908.622	3970.997	3960.302	4014.721	3940.619
F_{Total} (\$/h)	12347.66	12307.66	12352.99	12222.56	12287.73	12341.23	12289.58

A. ARITHMETIC OPTIMIZATION ALGORITHM BASED ON CAUCHY MUTATION TRIGONOMETRIC FUNCTION SEARCH TO SOLVE CEED PROBLEM

The fuel cost coefficient and pollutant emission coefficient of generator are from [44]. Their data are shown in Table 6-9, and the transmission losses are not considered in this case. Firstly, an AOA based on Cauchy mutation trigonometric function search is applied to solve CEED problem of 6 units, and its effect is tested. Since there are 4 objective functions in this problem, the generating cost and pollutant emissions need to be converted into one optimization objective, which is called the total cost (\$/h). The original AOA algorithm and six improved AOA methods are used for experiments and their results are compared. The following parameters are set in the following experiments. The population considered by the algorithm is 50, and the number of iterations is 200. Each methods are run for 20 times, and the average value of the experimental data is calculated. Table 10-13 shows the simulation results of 4 different loads CEED problems, where the recorded data is the average value, and the simulation figures are shown in Fig. 8-11.

Table 10-Table 13 and Fig. 8-11 show the total cost obtained by six improved methods in solving this case under different load requirements. Most of the results of the six improved methods are better than the original AOA algorithm. According to the data in Table 10 and the simulation diagram in Figure 8, it can be found that when the load demand is set to 150MW, the total cost obtained by asinhCAOA is 10237.21(\$/h), which is the minimum value. When the load demand is 175MW, it is shown in Table 11 and Fig. 9 that the total cost obtained by asinhCAOA to solve this problem is 12222.56 (\$/h), which has the best

TABLE 12. Experimental data for solving the 200mw demand ceed problem.

	AOA	sinCAOA	sinhCAOA	asinhCAOA	tanhCAOA	atanCAOA	atanhCAOA
P_1	50	50	50	50	50	50	50
P_2	28.15645	33.59214	23.37567	27.35535	20.596	27.72021	39.64162
P_3	20.30235	15	17.19603	15	15.38744	15	15.09997
P_4	35.17356	43.66716	47.97644	41.24658	49.64244	50	38.36619
P_5	30.88494	31.97632	23.71293	27.49407	30.67354	31.79019	17.82748
P_6	35.75812	25.77442	37.75451	39.01588	33.80951	25.76836	40
F_{Cost} ($/h)	3602.837	3621.001	3607.163	3580.873	3583.992	3625.588	3681.462
E_{CO2} (kg/h)	3729.58	3631.206	3724.103	3755.822	3755.508	3661.2	3711.572
E_{NOx} (kg/h)	3284.234	3331.195	3503.698	3392.487	3479.099	3437.971	3431.702
E_{SO2} (kg/h)	4516.768	4585.746	4754.774	4679.618	4833.869	4739.695	4578.901
F_{Total} ($/h)	14443.38	14401.8	14514.13	14410.88	14438.18	14420.52	14454.37

TABLE 13. Experimental data for solving the 225mw demand ceed problem.

	AOA	sinCAOA	sinhCAOA	asinhCAOA	tanhCAOA	atanCAOA	atanhCAOA
P_1	53.42939	50	50	50	50	50	50
P_2	27.43515	27.8035	31.4233	41.04481	34.2461	20	25.09994
P_3	15.8581	15	15	15	15	22.99286	23.94985
P_4	50	50	40.83785	45.64204	50	49.67611	46.94303
P_5	39.03363	42.21536	50	33.32814	35.76609	42.62369	41.74513
P_6	40	40	37.98438	40	40	40	37.34563
F_{Cost} ($/h)	4149.555	4120.963	4131.538	4159.893	4141.813	4147.499	4152.972
E_{CO2} (kg/h)	4481.433	4465.18	4565.715	4319.922	4358.488	4485.514	4397.664
E_{NOx} (kg/h)	3960.953	3880.362	3712.318	3844.14	3899.351	3892.369	3820.39
E_{SO2} (kg/h)	5616.29	5607.539	5529.734	5370.548	5495.641	5574.931	5416.401
F_{Total} ($/h)	16780.34	16762.16	16800.73	16693.98	16705.12	16766.2	16729.69

effect and fast convergence speed. However, when the load requirement is set to 200MW, the data in Table 12 and the simulation diagram in Fig. 10 show that sinCAOA has the lowest total cost compared with other methods, and the effect of asinhCAOA is second only to sinCAOA. Table 13 and Fig. 11 show the simulation results when the load demand is set to 225MW. It can be found that asinhCAOA has the best effect among the AOA and the improved method, and the total cost obtained is 16693.98 ($/h). Through data comparison and simulation graph, it can be found that asinhCAOA have better solution quality and faster convergence rate than other improved methods.

Next, the total cost obtained by asinhCAOA method is compared with LM [5], SA [19], QBA [44], MBO [45] and SCA [17]. The total cost data obtained by LM [5], SA [19], QBA [44], MBO [45] and SCA [17] are obtained from literature [17] and [22]. The table records the total cost obtained by solving CEED problem under four different load demands by different algorithms. The data in Table 14 prove that the total cost obtained by asinhCAOA is smaller than that of other algorithms cited under the four load demands, which better proves the superiority of the improved method of asinhCAOA.

TABLE 14. Comparison of results of ceed problem solved by different algorithms.

Load(MW)	Total cost ($/h)					
	asinhCAOA	LM[5]	SA[19]	QBA[44]	MBO[45]	SCA[17]
150	10237.21	10264.5667	10261.4905	10,255.28	10255.21	10255.208
175	12222.56	13251.5166	12280.0437	12,241.74	12241.67	12241.668
200	14410.88	16077.4089	14421.3044	14,413.88	14413.71	14413.708
225	16693.98	19661.3278	16790.6906	16,783.91	16784.34	16783.781

TABLE 15. Experimental data for solving the 150mw demand ceed problem.

	P_1	P_2	P_3	P_4	P_5	P_6	F_T ($/h)
F_CAOA	50	20	15	33.22108	10	21.82467	10219.86
cos_CAOA	50	20.49368	15	27.42418	10	27.11462	10233.31
asinh_CAOA	50	20	15	32.23516	13.92087	18.80403	10205.68
tan_CAOA	50	20.62971	15	30.51903	10	23.94141	10234.56
tanh_CAOA	50	20	15	21.58681	20.89678	22.77248	10231.77
V_AOA	50	20	15	25.86455	10	29.14367	10241.62

TABLE 16. Experimental data for solving the 175mw demand ceed problem.

	P_1	P_2	P_3	P_4	P_5	P_6	F_T ($/h)
F_CAOA	50	20	15	28.4786	22.51101	40	12308.28
cos_CAOA	50	20	15.85351	49.08973	11.53187	28.58829	12222.66
asinh_CAOA	50	21.46713	15	23.08005	26.22622	39.61858	12206.51
tan_CAOA	50	20	15	31.32886	19.86718	38.77133	12253.51
tanh_CAOA	50	20	15	28.98013	23.93548	37.07699	12292.03
V_AOA	50	20	15	36.01674	14.23861	40	12281.11

TABLE 17. Experimental data for solving the 200mw demand ceed problem.

	P_1	P_2	P_3	P_4	P_5	P_6	F_T ($/h)
F_CAOA	50	37.76003	15	34.72443	33.52886	29.89622	14426.68
cos_CAOA	50	25.27359	18.42867	34.96584	34.40803	37.25707	14381.96
asinh_CAOA	50	23.40233	16.74399	28.47119	41.61599	40	14375.87
tan_CAOA	50	22.11805	17.75074	37.99802	34.35786	37.79941	14364.74
tanh_CAOA	50	20	15	44.43245	38.7487	32.11241	14399.58
V_AOA	50	29.303	15	37.20131	29.28335	40	14406.82

TABLE 18. Experimental data for solving the 225mw demand ceed problem.

	P_1	P_2	P_3	P_4	P_5	P_6	F_T ($/h)
F_CAOA	50	20	16.45088	48.65911	49.92478	40	16733.89
cos_CAOA	50	23.97154	22.22258	47.90962	43.03837	37.93525	16783.59
asinh_CAOA	50	35.54562	18.33049	50	31.42747	40	16686.75
tan_CAOA	50	20	17.14695	50	48.01926	40	16727.3
tanh_CAOA	50	36.77825	15	50	33.54483	39.66798	16706.77
V_AOA	50	37.38017	15	46.31213	36.33713	40	16738.66

B. RANDOM DISTURBANCE PENALTY FUNCTION TEST

The above experiment verifies the performance of arithmetic optimization algorithm based on Cauchy mutation trigonometric function search in solving CEED problems. After comprehensive comparison of the total cost of each method, asinhCAOA with the best effect is selected among the six improved methods. The next step of the experiment is to test the performance of the random disturbance penalty function. The case of 6 units is simulated and solved by asinhCAOA, which is the same as the previous section. The proposed random disturbance penalty function selects five different functions, which are cosine function, hyperbolic sine function, tangent function, hyperbolic tangent function and V-type function. For the convenience of recording, asinhCAOA is abbreviated as CAOA. The five punishment strategies are denoted as cos_CAOA, asinh_CAOA, tan_CAOA, tanh_CAOA and V_CAOA; The fixed penalty strategy is denoted as F_CAOA. In this experiment, the population number is set to 50, the maximum number of iterations is 200, and each methods is run for 20 times to take the average. The experimental results are shown in Table 15-18 and Fig. 12-15. The data recorded in Table 15-18 are average values.

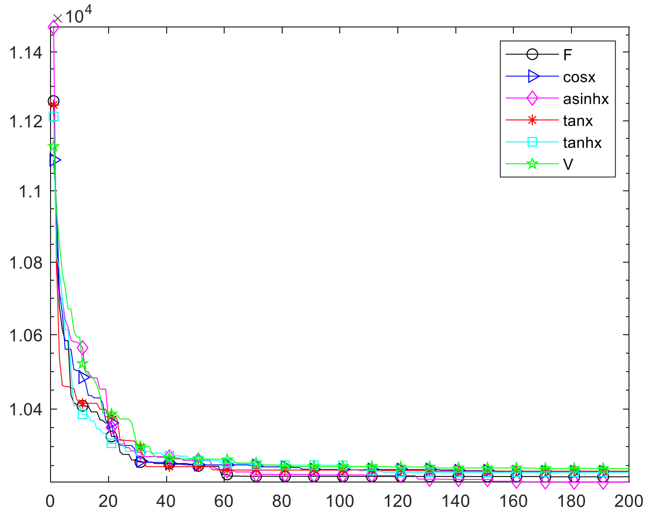


FIGURE 12. The convergence curve of solving the 150MW demand CEED problem.

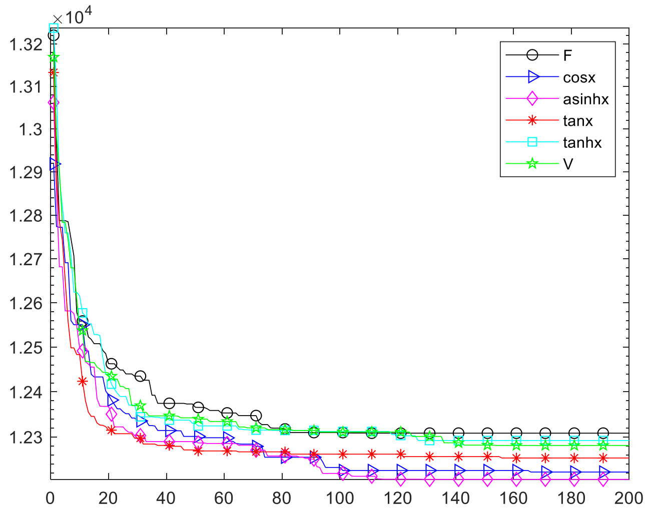


FIGURE 13. The convergence curve of solving the 175MW demand CEED problem.

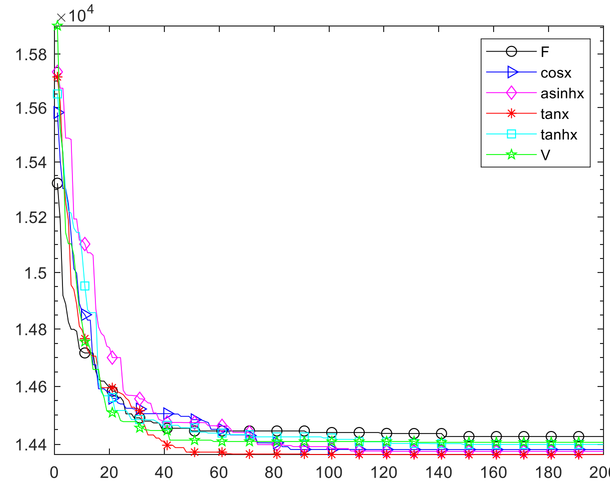


FIGURE 14. **The convergence curve of solving the 200MW demand CEED problem.**

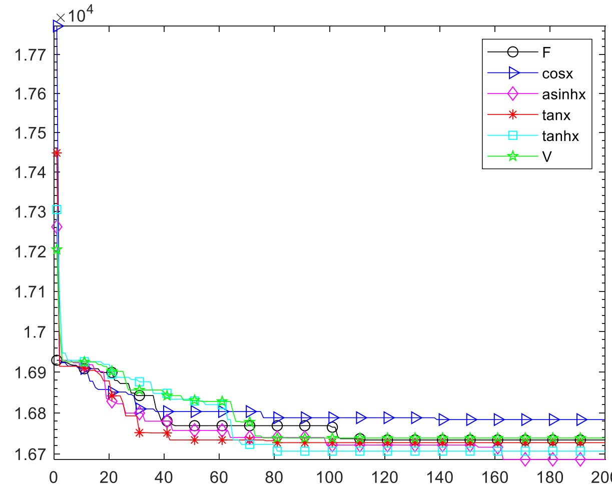


FIGURE 15. **The convergence curve of solving the 225MW demand CEED problem.**

By observing Table 15-18 and Figure 12-15, it can be concluded that Compared with F_CAOA, cos_CAOA, tan_CAOA, tanh_CAOA and V_CAOA, the total cost obtained by using asinh_CAOA penalty strategy is 10205.68($/h), 12,206.51 ($/h) and 16,686.75 ($/h) when solving 150MW, 175MW and 225MW load demands, respectively. When the load demand is 200MW, tan_CAOA works best, and the total cost is 14364.74($/h). When the load demand is 150MW, 175MW, 200MW and 225MW, the total cost of asinh_CAOA is reduced by 0.14%, 0.83%, 0.35% and 0.28% respectively compared with the fixed penalty strategy. However, in general, the stability of the random disturbance penalty strategy needs to be improved.

VI. CONCLUSION

In this thesis, an arithmetic optimization algorithm based on Cauchy mutation trigonometric function search is proposed. MOP is replaced by oscillating factor and Cauchy mutation, and 6 kinds of trigonometric function search are used to replace fixed value parameters in AOA position update formula. In order to test the optimization performance of arithmetic optimization algorithm based on Cauchy mutation trigonometric function search, 23 benchmark functions are applied to test the optimization effect, and asinhCAOA with the best optimization effect is selected, and then the performance of other algorithms is compared. Then the AOA based on Cauchy mutation trigonometric function search is applied to solve CEED problem and compared with other methods. The experimental data show that the improved AOA algorithm is effective in solving CEED problem. Finally, it is verified by experiments that the effect of random disturbance penalty function is obviously better than that of fixed penalty function, but the stability of this penalty strategy still needs to be improved.

REFERENCES

[1] M. H. Hassan, S. Kamel, L. Abualigah, and A. Eid, ''Development and application of slime mould algorithm for optimal economic emission dispatch,'' *Expert Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115205.

[2] R. P. Parouha and K. N. Das, ''A novel hybrid optimizer for solving economic load dispatch problem,'' *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 108–126, Jun. 2016.

[3] T. C. Bora, V. C. Mariani, and L. D. S. Coelho, ''Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm,'' *Appl. Thermal Eng.*, vol. 146, pp. 688–700, Jan. 2019.

[4] M. A. Abido, ''Multiobjective particle swarm optimization for environmental/economic dispatch problem,'' *Electr. Power Syst. Res.*, vol. 79, no. 7, pp. 1105–1113, Jul. 2009.

[5] S. Krishnamurthy and R. Tzoneva, ''Impact of price penalty factors on the solution of the combined economic emission dispatch problem using cubic criterion functions,'' in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2012, pp. 1–9.

[6] J. Nanda, D. P. Kothari, and K. S. Lingamurthy, ''Economic-emission load dispatch through goal programming techniques,'' *IEEE Trans. Energy Convers.*, vol. 3, no. 1, pp. 26–32, Mar. 1988.

[7] G. P. Granelli, M. Montagna, G. L. Pasini, and P. Marannino, ''Emission constrained dynamic dispatch,'' *Electr. Power Syst. Res.*, vol. 24, no. 1, pp. 55–64, Jul. 1992.

[8] J. S. Dhillon, S. C. Parti, and D. P. Kothari, ''Stochastic economic emission load dispatch,'' *Electr. Power Syst. Res.*, vol. 26, no. 3, pp. 179–186, Apr. 1993.

[9] A. A. El-Keib and H. Ding, ''Environmentally constrained economic dispatch using linear programming,'' *Electr. Power Syst. Res.*, vol. 29, no. 3, pp. 155–159, May 1994.

[10] J.-F. Chen and S.-D. Chen, ''Multiobjective power dispatch with line flow constraints using the fast Newton–Raphson method,'' *IEEE Trans. Energy Convers.*, vol. 12, no. 1, pp. 86–93, Mar. 1997.

[11] A. I. S. Kumar, K. Dhanushkodi, J. J. Kumar, and C. K. C. Paul, ''Particle swarm optimization solution to emission and economic dispatch problem,'' in *Proc. Conf. Convergent Technol. Asia–Pacific Region (TENCON)*, Oct. 2003, pp. 435–439.

[12] P. K. Hota, A. K. Barisal, and R. Chakrabarti, ''Economic emission load dispatch through fuzzy based bacterial foraging algorithm,'' *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 7, pp. 794–803, Sep. 2010.

[13] R. Karthikeyan, ''Combined economic emission dispatch using grasshopper optimization algorithm,'' *Mater. Today, Proc.*, vol. 33, pp. 3378–3382, Jan. 2020.

[14] I. Karakonstantis and A. Vlachos, ''Ant colony optimization for continuous domains applied to emission and economic dispatch problems,'' *J. Inf. Optim. Sci.*, vol. 36, nos. 1–2, pp. 23–42, Mar. 2015.

[15] H. Hardiansyah, ''Multi-objective ant lion optimizer for solving environmental/economic dispatch,'' *Przeglad Elektrotechniczny*, vol. 1, no. 2, pp. 155–160, Feb. 2021.

[16] A. S. Tomar, H. M. Dubey, and M. Pandit, ''Combined economic emission dispatch using spider monkey optimization,'' in *Advanced Engineering Optimization Through Intelligent Techniques*. Singapore: Springer, 2020, pp. 809–818.

[17] D. Gonidakis and A. Vlachos, "A new sine cosine algorithm for economic and emission dispatch problems with price penalty factors," *J. Inf. Optim. Sci.*, vol. 40, no. 3, pp. 679–697, Apr. 2019.
[18] K.-D. Lu, Z.-G. Wu, and T. Huang, "Differential evolution-based three stage dynamic cyber-attack of cyber-physical power systems," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 2, pp. 1137–1148, Apr. 2023.
[19] I. Ziane, F. Benhamida, and A. Graa, "Simulated annealing algorithm for combined economic and emission power dispatch using max/max price penalty factor," *Neural Comput. Appl.*, vol. 28, no. 1, pp. 197–205, Dec. 2017.
[20] D. Aydin, S. Özyön, C. Yaşar, and T. Liao, "Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem," *Int. J. Electr. Power Energy Syst.*, vol. 54, pp. 144–153, Jan. 2014.
[21] Y. A. Gherbi, H. Bouzeboudja, and F. Z. Gherbi, "The combined economic environmental dispatch using new hybrid metaheuristic," *Energy*, vol. 115, pp. 468–477, Nov. 2016.
[22] M. H. Hassan, S. Kamel, S. Q. Salih, T. Khurshaid, and M. Ebeed, "Developing chaotic artificial ecosystem-based optimization algorithm for combined economic emission dispatch," *IEEE Access*, vol. 9, pp. 51146–51165, 2021.
[23] R. Kumar, A. Sadu, R. Kumar, and S. K. Panda, "A novel multi-objective directed bee colony optimization algorithm for multi-objective emission constrained economic power dispatch," *Int. J. Electr. Power Energy Syst.*, vol. 43, no. 1, pp. 1241–1250, Dec. 2012.
[24] Y. Wu, X. Wang, Y. Xu, and Y. Fu, "Multi-objective differential-based brain storm optimization for environmental economic dispatch problem," in *Brain Storm Optimization Algorithms: Concepts, Principles and Applications*. Cham, Switzerland: Springer, 2019, pp. 79–104.
[25] M.-R. Chen, G.-Q. Zeng, and K.-D. Lu, "Constrained multi-objective population extremal optimization based economic-emission dispatch incorporating renewable energy resources," *Renew. Energy*, vol. 143, pp. 277–294, Dec. 2019.
[26] T. Singh, K. K. Mishra, and Ranvijay, "Multiobjective environmental adaptation method for solving environmental/economic dispatch problem," *Evol. Intell.*, vol. 12, no. 2, pp. 305–319, Jun. 2019.
[27] G. Chen, X. Man, Y. Long, and Z. Zhang, "Application of multi-objective new whale optimization algorithm for environment economic power dispatch problem," *Eng. Lett.*, vol. 29, no. 1, pp. 68–81, Feb. 2021.
[28] K. Deb and R. Datta, "A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach," *Eng. Optim.*, vol. 45, no. 5, pp. 503–527, May 2013.
[29] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA J.*, vol. 29, no. 11, pp. 2013–2015, Nov. 1991.
[30] Y. Li, L. Han, and C. Fu, "Dynamic economic dispatch based on improved particle swarm optimization and penalty function," in *Proc. IEEE Conf. Expo Transp. Electrific. Asia–Pacific (ITEC Asia–Pacific)*, Aug. 2014, pp. 1–4.
[31] H. Yoo, V. M. Zavala, and J. H. Lee, "A dynamic penalty function approach for constraint-handling in reinforcement learning," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 487–491, 2021.
[32] V. P. Sakthivel, M. Suman, and P. D. Sathya, "Combined economic and emission power dispatch problems through multi-objective squirrel search algorithm," *Appl. Soft Comput.*, vol. 100, Mar. 2021, Art. no. 106950.
[33] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 376, Apr. 2021, Art. no. 113609.
[34] M. Meshkat and M. Parhizgar, "Sine Optimization Algorithm (SOA): A novel optimization algorithm by change update position strategy of search agent in sine cosine algorithm," in *Proc. 3rd Iranian Conf. Intell. Syst. Signal Process. (ICSPIS)*, Shahrood, Iran, Dec. 2017, pp. 11–16.
[35] K. Wang, G. K. Rohde, and J. Xiao, "Edge feature enhancement approach using Hilbert transform of Cauchy distribution and its applications," *IET Image Process.*, vol. 15, no. 12, pp. 2891–2909, Oct. 2021.
[36] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[37] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, Mar. 2003.
[38] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
[39] J. M. Abdullah and T. Ahmed, "Fitness dependent optimizer: Inspired by the bee swarming reproductive process," *IEEE Access*, vol. 7, pp. 43473–43486, 2019.
[40] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, Jul. 1992.
[41] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
[42] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.
[43] I. Naruei and F. Keynia, "Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems," *Eng. Comput.*, vol. 38, no. 4, pp. 3025–3056, Oct. 2022.
[44] F. P. Mahdi, P. Vasant, M. Abdullah-Al-Wadud, V. Kallimani, and J. Watada, "Quantum-behaved bat algorithm for many-objective combined economic emission dispatch problem using cubic criterion function," *Neural Comput. Appl.*, vol. 31, no. 10, pp. 5857–5869, Oct. 2019.
[45] R. Ramachandaran and P. Avirajamanjula, "Modified biogeography based optimization algorithm for power dispatch using max/max price penalty factor," *Int. J. Pure Appl. Math.*, vol. 118, no. 18, pp. 3813–3818, 2018.

YUN-PENG LI received the B.Sc. degree in electronic information engineering from the University of Science and Technology Liaoning, China, in 2008, and the M.Sc. and Ph.D. degrees in electrical engineering from Gyeongsang National University, South Korea, in 2010 and 2013, respectively. He is currently a Postdoctoral Fellow with the School of Electrical Engineering, Xinjiang University. His main research interests include modeling of complex industry process, intelligent control, and computer integrated manufacturing.

LIN-XUAN ZHANG received the Ph.D. degree in engineering. He is currently a Professor and a Ph.D. Supervisor with Xinjiang University, the Dean of the School of Electrical Engineering, and the Deputy Director of the Institute of Industrial Intelligence and Systems, Department of Automation, Tsinghua University/National Computer Integrated Manufacturing System Engineering Technology Research Center. He is a Senior Member of the Chinese Mechanical Engineering Society and a member of the Academic Committee of the China Electrotechnical Society's University Electrical and Electronic Engineering Innovation Competition. He is also the Director and the Deputy Secretary-General of the China Simulation Society and the Chairman of the Education and Training Committee.

ZHENG-JIE FENG is currently a General Manager and a Senior Engineer of Zhejiang Xingyu Mechanical and Electrical Technology Company Ltd.