Multi-Strategy Equilibrium Optimizer: An Improved Meta-Heuristic Tested On Numerical Optimization And Engineering Problems

Yu Li
Henan University
Xiao Liang
Henan University
Jingsen Liu (ljs@henu.edu.cn)
Henan University
Huan Zhou
Henan University

Research Article

Keywords:

Posted Date: March 10th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1412084/v1

Multi-strategy Equilibrium Optimizer: An improved meta-heuristic tested on numerical optimization and engineering problems

Yu Li [a], Xiao Liang [b], Jingsen Liu [c*] Huan Zhou [d]

a School of Business, Henan University, Kaifeng 475004, China. E-mail: leey@henu.edu.cn
b School of Business, Henan University, Kaifeng 475004, China. E-mail: liangx@henu.edu.cn
c Institute of Intelligent Network Systems, and Software School, Henan University, Kaifeng 475004, China. E-mail: ljs@henu.edu.cn
d School of Business, Henan University, Kaifeng 475004, China. E-mail: zhouhuan@henu.edu.cn
* Corresponding author: ljs@henu.edu.cn

Abstract—In order to solve complex numerical optimization problems and engineering problems more effectively, an Improved Equilibrium Optimizer (IEO) based on multi-strategy optimization is proposed. Firstly, Tent mapping is used to initialize the algorithm instead of randomly generating initial population in basic Equilibrium Optimizer. The diversified initial population lays a good foundation for global search of the algorithm. Moreover, nonlinear time parameter is used to update the position equation, which dynamically balances the exploration and exploitation phases of IEO. Finally, Lens Opposition-based Learning (LOBL) is introduced, which can avoid local optimization by improving the population diversity of the algorithm. Simulation experiments are carried out on 23 classical functions, IEEE CEC2017 problems and IEEE CEC2019 problems, and the stability of the algorithm is further analyzed by Friedman statistical test and box plots. Experimental results show that IEO has good solution accuracy and robustness. In addition, six engineering design problems are solved, and the results show that IEO not only has high optimization efficiency, but also can achieve cost minimization.

1 INTRODUCTION

In recent years, optimization problem has gradually become a very important topic in modern management field. The significance of optimization problem is to provide the optimal solution for the application problems in various fields of society. On the premise of comprehensive consideration of all aspects of constraints, the practical application problems are abstracted as objective functions and solved. Due to the development of science and technology, many optimization problems become more and more complex. Therefore, meta-heuristic algorithms with high flexibility have attracted the attention of a large number of researchers[1]. The meta-heuristic algorithms put forward a set of new research ideas and solutions, whether in the modeling and simulation of complex systems, or in the analysis of complex decision and the solution of optimization problems. Meta-heuristic algorithms are used to solve optimization problems by simulating biological or physical phenomena. Such algorithms are divided into four categories: evolutionary algorithms, physics-based algorithms, human-based algorithms and swarm intelligence algorithms[2].

The evolutionary algorithm mainly realizes the overall progress of the population and solves the optimal solution by simulating the evolution law of survival of the fittest in nature. The representative ones are the Genetic Algorithm (GA) which searches for the optimal solution by simulating the natural evolution process[3], and the Differential Evolution (DE) which simulates the crossover and mutation mechanism in heredity[4]. Similarly, the Population-Based Incremental Learning (PBIL) is proposed based on two strategies: genetic search and competitive learning[5]. Moreover, Biogeography-Based Optimization (BBO) is proposed based on the biogeography theory of species' migration and drift between geographical regions in nature[6].

On the other hand, physics-based algorithms are inspired by the laws of physics in nature. Representative algorithms mainly include Simulated Annealing (SA)[7], the inspiration comes from the annealing process of solid materials. In addition, Big-Bang Big-Crunch (BBBC) is inspired by the Big Bang and contraction theory[8]. Gravitational Search Algorithm (GSA) is based on Newton's law of universal gravity to guide the motion of each particle, so as to search for the optimal solution[9]. Inspired by the theory of physics kinematics, Central Force Optimization (CFO) realizes the update of the optimal solution by updating the acceleration[10].

Human-based algorithms are mainly inspired by human behavior, such as human teaching behavior, social behavior and so on. The representative algorithms are as follows: Tabu Search (TS) is a search process guided by memory. It is a simulation of human intelligence process and a manifestation of artificial intelligence[11]. The Teaching Learning Based Optimization (TLBO) conducts search optimization by simulating the learning methods of human teaching and learning in the learning process[12]. The inspiration of Harmony Search (HS) comes from the process of human musical performance[13], by repeatedly adjusting the solution variables in the memory bank, the algorithm makes the function value continuously converge with the increase of the number of iterations.

The swarm intelligence algorithm comes from the simulation of the behavior process of the biological group. The individuals in the biological group follow the cooperative mode of aggregation, division of labor, collision avoidance, convergence and so on, thus the swarm intelligence emerges. Representative swarm intelligence algorithms mainly include: Particle Swarm Optimization (PSO), which simulates the foraging behavior of birds[14]; Ant Colony Optimization (ACO), which simulates the foraging path of ants by secretion concentration[15]; Artificial Bee Colony (ABC), which simulates the honey gathering behavior of bees[16], and Whale Optimization Algorithm (WOA), which is inspired by the feeding behavior of whales in the ocean[17].

Equilibrium Optimizer(EO) is a physics-based metaheuristic algorithm proposed in 2020[18]，The algorithm was inspired by mass balance equations in physics. The EO algorithm is inspired by the control volume mass balance, and the dynamic state and equilibrium state of the particles can be estimated. In EO, the important parameters are: equilibrium pool ($\vec{C}_{eq,pool}$), exponential term ($\vec{F}$)and generation rate ($\overrightarrow{GCP}$). During the optimization process, the search agent randomly updates its concentration (position) for certain particles called equilibrium pool, eventually reaching an equilibrium state (the best result). The unique update mechanism of EO algorithm makes it have fast convergence speed and precision. Currently, some scholars have studied EO algorithm. Gupta et al.[19] introduced Gaussian variation and new exploratory search mechanism on the basis of EO algorithm to improve the diversity of solutions. Abdel-Basset et al.[20] introduced an equation and Gaussian mutation strategy when EO was performing position update to improve the exploration and exploitation capability of the algorithm. Jia et al.[21] combined EO with thermal exchange optimization (TEO) in order to improve the optimization accuracy of EO algorithm. Fan et al.[1] proposed an improved EO algorithm based on opposition-based learning and new update rules, which improves the exploration ability of the algorithm and avoids falling into local optimal value.

In this paper, an Improved Equilibrium Optimizer (IEO) is proposed, which includes three improvement points: firstly, the random initialization is replaced by Tent chaotic map, so that the particles are evenly distributed in the search space as far as possible, and the quality of the initial solution is improved. Secondly, a dynamic control parameter strategy is proposed to promote the balance between the exploration and exploitation stages of the algorithm through the dynamic changes of parameters. Finally, the Lens Opposition- based Learning (LOBL) strategy is introduced to generate new candidate solutions to expand the search space and avoid the algorithm falling into local optimum. In the simulation experiment, 23 reference functions, IEEE CEC2017 and IEEE CEC2019 three different complexity test sets are optimized, and all the experimental results are compared with the six meta-heuristic algorithms, the results show that the improved algorithm IEO has significant advantages in convergence accuracy and effectiveness. In the statistical test, Friedman test proves that IEO has superior performance when optimizing each test set. In addition, this paper also selected convergence curve and boxplot analysis to show the stability of IEO from different perspectives. Finally, the improved algorithm IEO is applied to six engineering design problems: pressure vessel problem, welded beam problem, tension/compression spring problem, three-bar truss problem, speed reducer problem and optimal design of industrial refrigeration system. Experimental results show that IEO has good optimization efficiency in solving practical application problems.

The remaining structure of this article is as follows: Section 2 introduces the basic Equilibrium Optimizer. Section 3 describes the improved Equilibrium Optimizer. In section 4, three function test suites are selected for simulation experiments, such as 23 classical functions and IEEE CEC2017 and IEEE CEC2019. In section 5, the IEO algorithm is applied to six engineering design problems. Finally, the sixth section summarizes the work of this paper.

2 EQUILIBRIUM OPTIMIZER

Equilibrium Optimizer (EO)[18] is a new intelligent algorithm proposed by Faramarzi et al., which is inspired by the mass balance equation in physics. The mass balance equation reflects the physical process of mass entering, leaving and producing in the control volume. In EO, the concentration of each particle is updated in a random way until it reaches equilibrium. EO algorithm constructs three mathematical models: 1. Initialization phase 2. Equilibrium pool and candidates 3. Updating the concentration. The specific description is as follows：

Step1. Initialization phase

Similar to most meta-heuristic algorithms, EO initiates the optimization process by initializing the population. The initial concentration is constructed by randomly initializing the particles in the D - dimensional search space. The initial concentration of each particle is described below：

$$\vec{C}_i^{initial} = Lb + rand_i(Ub - Lb) \qquad i = 1,2 \ldots \ldots n \tag{1}$$

Where $\vec{C}_i^{initial}$ is the initial concentration of the i-th particle, Ub and Lb represent the maximum and minimum values of particles in the search space, $rand_i$ is a random vector in the range of [0,1], and n represents the number of particles.

Step2. Equilibrium pool and candidates ($\vec{C}_{eq}$)

In order to improve the global search ability of the algorithm and avoid falling into the local optimal solution of low quality, after the initialization phase is completed, the concentration of the generated particles is evaluated and the four particles with the highest fitness value are selected to prepare for the formation of the equilibrium pool.

The equilibrium pool is used to provide candidate solutions during the algorithm optimization process. It consists of four particles with optimal fitness values and one average particle generated during the initialization phase.

The mathematical definition is as follows：

$$\vec{C}_{eq\_ave}=\frac{\vec{C}_{eq1}+\vec{C}_{eq2}+\vec{C}_{eq3}+\vec{C}_{eq4}}{4} \tag{2}$$

$$\vec{C}_{eq,pool}=\{\vec{C}_{eq1},\vec{C}_{eq2},\vec{C}_{eq3},\vec{C}_{eq4},\vec{C}_{eq\_ave}\} \tag{3}$$

Among them，$\vec{C}_{eq1}$~$\vec{C}_{eq4}$ represent the four particles with the highest concentration selected after the initialization of the algorithm, $\vec{C}_{eq\_ave}$ represents the average particle，and $\vec{C}_{eq,pool}$ represents the equilibrium pool. Especially, in the equilibrium pool, the four particles with the highest concentration contribute to the exploration of the algorithm, while the average particle plays an important role in the exploitation phase.

In the iterative process of the algorithm, each particle is selected from the five candidate particles in the equilibrium pool with the same probability, which contributes to the generation of the global optimal solution. The mathematical modeling formula is as follows：

$$\vec{C}_{eq}=randi\{\vec{C}_{eq,pool}\} \tag{4}$$

Step3. Updating the concentration

The exponential term $\vec{F}$ is an important indicator to balance the exploration and exploitation capability of EO algorithm. The calculation of F is as follows：

$$\vec{F}=a_1 sign(\vec{r}-0.5)\cdot\left[e^{-\vec{\lambda}t}-1\right] \tag{5}$$

Where a_1 is a constant that controls the exploration ability of the algorithm. $\vec{r}$ and $\vec{\lambda}$ represent vectors within the interval of [0,1], t is the coefficient updated with the number of iterations, which can be calculated as follows：

$$t=\left(1-\frac{Iter}{Max\_iter}\right)^{\left(a_2-\frac{Iter}{Max\_iter}\right)} \tag{6}$$

Where $Iter$ represents the current iteration number of the algorithm, and Max_iter represents the maximum iteration number of the algorithm. a_2 is a constant that can control the exploitation ability of the algorithm. According to the experimental data[18]，when a_1=2 and a_2=1，the performance of algorithm EO is the best. In order to improve the exploitation capability of EO, an equally important indicator is generation rate ($\vec{G}$), which is defined as follows:

$$\vec{G}=\overrightarrow{GCP}\left(\vec{C}_{eq}-\vec{\lambda}\vec{C}\right)\cdot\vec{F} \tag{7}$$

$$\overrightarrow{GCP}=\begin{cases}0.5r_1 & r_2\geq GP\\ 0 & r_2<GP\end{cases} \tag{8}$$

In the formula，$\overrightarrow{GCP}$ represents the control parameter vector of generation rate，$\vec{C}$ represents the current particle concentration，r_1 and r_2 are random vectors within the interval [0,1]，and GP is a constant with value of 0.5. To sum up, after the concentration update phase of EO algorithm, the update formula of each particle is as follows：

$$\vec{C}=\vec{C}_{eq}+\left(\vec{C}-\vec{C}_{eq}\right)\cdot\vec{F}+\frac{\vec{G}}{\vec{\lambda}V}\cdot\left(1-\vec{F}\right) \tag{9}$$

Where V is unit volume.

According to the above description, the updating rule of algorithm EO is to construct the initial concentration of each particle in the initialization phase, select four particles with the highest concentration and form an equilibrium pool with an average particle, which provides candidate solutions for algorithm iteration. Then, the concentration of each particle is calculated using two important indexes: exponential term ($\vec{F}$) and generation rate ($\vec{G}$).

3 THE IMPROVED EQUILIBRIUM OPTIMIZER

In this paper, an improved Equilibrium Optimizer (IEO) algorithm is proposed. In IEO, there are three improved strategies: Firstly, the algorithm is initialized by using Tent chaotic map instead of randomly generating initial population. The uniform initial population generated by Tent mapping improves the quality of final optimization solution. Second, nonlinear dynamic control parameter is introduced to maintain the balance between the exploration and exploitation phases of the algorithm. Third, Lens Opposition-based Learning (LOBL) is used to calculate the relative population of each iteration process to expand the search space of the algorithm and improve the accuracy of the solution.

3.1 Tent chaotic sequence initialization

In the initialization phase, the basic EO algorithm uses the method of random generation to determine the initial solution, which cannot guarantee that the randomly generated initial solution is evenly distributed in the search space. Therefore, in order to improve the quality of the initial solution, Tent chaotic map is introduced[22]. The mathematical expression is as follows：

$$x_{i+1}=\begin{cases}\frac{x_i}{0.7} & x_i<0.7\\ \frac{10}{3}(1-x_i) & x_i\geq 0.7\end{cases} \tag{10}$$

Where x_i shows the chaos variable of i-th particle，$i\in[1,n]$.

Tent chaotic mapping has a rich dynamic space, which is a nonlinear phenomenon between determinism and randomness, and has neither periodicity nor convergence. The randomness and ergodic characteristics of Tent chaotic mapping enable the search individual to experience all states without repetition. In EO algorithm, Tent chaotic mapping is introduced to disperse the population as much as possible in the initialization phase, so as to maintain the diversity of the population and improve the global search ability of the algorithm.

In IEO, the Tent chaotic map is used to replace the random distribution to increase the diversity of the population and accelerate the convergence rate of the algorithm. We set the number of particles as n and the dimension as d. The basic steps of initializing particles by using Tent chaotic map within the search range are as follows:

Step1：In the search range, the number of particles is set as n，and a group of 1×d vectors are randomly generated, which are taken as the position information of the first particle.

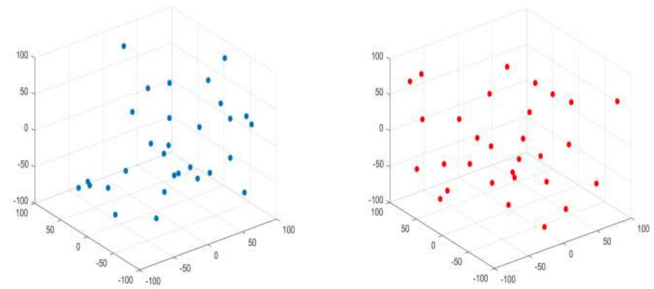
Step2：Using Eq. (10), the position information of the remaining *n-1* particles is calculated to form a chaotic sequence.

Step3：The resulting chaotic sequence is initialized by the Equation (11).

$$\vec{C}_i^{initial}=Lb+x_i(Ub-Lb) \qquad i=1,2\ldots\ldots n \tag{11}$$

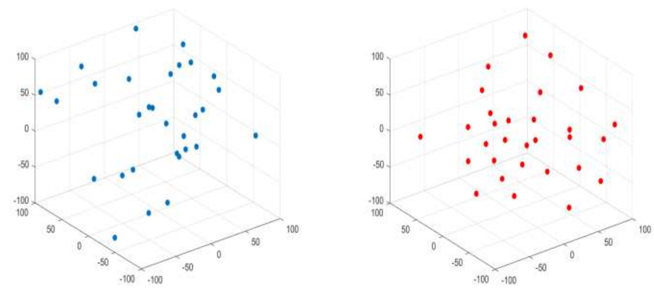
In order to verify the rationality of this method, we compare the Tent mapping chaotic sequence with the random initialization in EO algorithm, set the particle number to 30, and take Sphere function among 23 classical reference functions and F15 function in CEC2017 as examples. The details are shown in Fig.1 and Fig.2.

Each point in the figure represents a search individual. As shown in Fig. 1, in the search space [-100,100] where Sphere function is located, compared with the population generated by random initialization in Figure 1a, the initial population generated by Tent chaotic mapping in Figure 2b is more evenly distributed within the search space. In Fig. 2, the F15 function in the CEC2017 test suite is selected for the experiment. Within the search range of [-100,100], it can be clearly seen that the population initialized through the Tent mapping chaotic sequence has a relatively uniform distribution. This improves IEO's global search capabilities.



(a) random initialization (b) Tent map

Fig. 1. The population distribution of the Sphere function



(a) random initialization (b) Tent map

Fig. 2. The population distribution of the CEC2017 F15

3.2 Dynamic parameter strategy

In EO, the exponential term $\vec{F}$ is an important index that balances the exploration and exploitation capability of EO algorithm. According to Eq. (5), it can be seen that the exponential term $\vec{F}$ is affected by the time parameter t. In addition, According to Eq. (6), the expression of the time parameter t contains the constant a_2, so the change of the parameter t largely determines the performance of EO algorithm. In EO, the time parameter t decreases from 1 to 0, which is a process that produces nonlinear changes as the number of iterations increases.

According to literature[1], the parameter t is redefined and expressed in a new way, as follows:

$$\theta = \frac{\pi}{2} \cdot \frac{Iter}{Max\_iter} \tag{12}$$

$$t = t_{end} + (t_{start} - t_{end})(1 - sin\theta)^{\left(a_2 \frac{Iter}{Max\_iter}\right)} \tag{13}$$

Among them, t_{start}=1; t_{end}=0. $Iter$ indicates the number of iterations, Max_iter represents the maximum number of iterations of the algorithm. The change curve of time parameter t in nonlinear dynamic parameter strategy, original EO algorithm[18], and linear decline strategy[23] is shown in Fig. 3. The dynamic control parameter strategy proposed in this paper reduces slowly in the early stage of algorithm iteration, which avoids premature convergence when the particle is updated, and makes the particle fully search globally in the search space. In addition, in the late iteration of the algorithm, the decreasing speed of parameter t is slowed down, so that the particles can search accurately in the search space, thereby the balanced state can be reached more effectively.

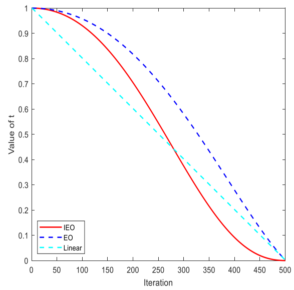


Fig. 3. Comparison of parameters.

3.3 Lens Opposition-based Learning

The original EO algorithm often appears the phenomenon of population aggregation in the late iteration, which makes the algorithm fall into the local extreme value due to the lack of population diversity. In order to strengthen the global search ability of the algorithm and improve the solv-ing accuracy, Lens Opposition-based Learning strategy (LOBL) is applied to EO algorithm. LOBL is used to calculate the opposite solution of candidate solutions in the optimization process of the algorithm. By expanding the opposite region of candidate solutions, the population diversity of the algorithm in the iterative process is enhanced.

The LOBL strategy is a combination of Opposition-based Learning (OBL) strategy and lens imaging principle [24]. When the distance between the object and the convex lens is set to be more than two focal lengths, the process of particles searching for opposite solutions in the search space can be regarded as the process of lens imaging, as shown in Fig. 4.

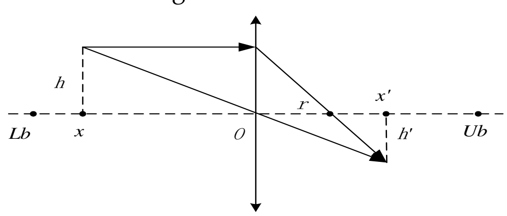


Fig. 4. Lens Opposition-based Learning

In Fig. 4, a convex lens of focal length r is placed on the origin O (this paper takes $(Ub + Lb)/2$). An object of height h is placed x away from the point O and x is two focal lengths away. By the lens imaging principle, an image of height h' is generated at point x' on the other side. In other words, the point x takes O as the basis point to obtain the corresponding reverse point x', and the mathematical relationship is described as follows:

$$\frac{(Lb+Ub)/2-x}{x'-(Lb+Ub)/2}=\frac{h}{h'} \tag{14}$$

In the above equation, $\frac{h}{h'}=k$,and the scaling factor k represents the scaling relationship between the object and the corresponding real image. Therefore, the Eq. (14) can be transformed into the formula to calculate the opposite solution of x':

$$x'=\frac{Lb+Ub}{2}+\frac{Lb+Ub}{2k}-\frac{x}{k} \tag{15}$$

Since the above equation is only applicable to the opposite solution in one dimensional space, when the optimization problem is multi-dimensional, the solution equation of LOBL strategy is as follows:

$$x_i'=\frac{Lb_i+Ub_i}{2}+\frac{Lb_i+Ub_i}{2k}-\frac{x_i}{k} \tag{16}$$

Where x_i' represents the opposite solution generated by LOBL strategy in the i-th dimension, and Lb_i and Ub_i respectively represent the lower bound and upper bound of the i-th dimension in the search range.

Sphere function among the 23 reference functions and F15 function in the IEEE CEC2017 test suite are taken as examples, and the positions of each particle generated by LOBL strategy are shown in Fig. 5 and Fig. 6. In the figure, the blue points represent the positions of particles generated by the original EO algorithm when optimizing the function, and the red points represent the positions of particles generated by LOBL strategy. As shown in Fig. 5, the original EO algorithm falls into local optimum when optimizing Sphere function, and the positions of each particle produce high overlap. The specific magnified part is shown in the lower left corner of Fig. 5. In addition, as can be seen in detail from Fig. 6, the original EO algorithm is easy to fall into the plight of local optimal, resulting in a high degree of location overlap of each particle, and the search space becomes increasingly narrow. By introducing LOBL strategy, the opposite solution is generated, which significantly expands the search space of each particle and avoids the algorithm falling into the local optimal solution.

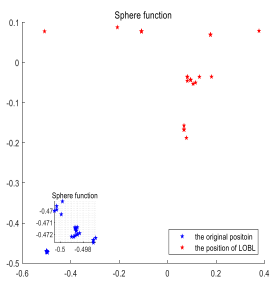


Fig.5. The position of Sphere function generated by LOBL

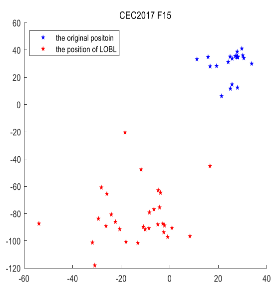


Fig.6. The position of CEC2017 F15 generated by LOBL

In this paper, the LOBL strategy is used to calculate the opposite solution of the candidate solution generated in each iteration. If the fitness value of the opposite solution is better than the candidate solution, the opposite solution is used to replace the candidate solution to become the current optimal solution, and the iterative operation continues. Therefore, the LOBL strategy significantly improves the ability of particles to escape from the extreme region, which effectively avoids the algorithm falling into the local optimal solution and makes the particles reach the balance state better.

In IEO, firstly, the Tent chaotic sequence is used to initialize the particle concentration, so that the initial solution is evenly distributed in the search space as far as possible, and the solving efficiency is improved. Secondly, the new nonlinear dynamic parameter strategy can better balance the exploration and exploitation phases. Finally, the LOBL strategy is used to calculate the opposite solution of the candidate solution generated by each iteration. This strategy avoids falling into the local optimal by increasing the population diversity. These three improved methods can effectively improve the solving speed and accuracy of the algorithm.

4 Numerical Experiment Results and Analysis

In this section, 23 classical benchmark functions and two function test suites IEEE CEC2017 and IEEE CEC2019 with complex changes are selected to carry out simulation experiments. The CEC2017 and CEC2019 test suites contain different functions, which are divided into different categories: basic functions, hybrid functions, and composition functions. Among them, the CEC2017 data suite includes 30 complex composite functions, and the CEC2019 test suite includes 10 functions. These functions have different rotation matrices, each matrix is generated from standard normally distributed entries by Gram-Schmidt ortho-normalization with condition number c that is equal to 1 or 2. Therefore, these functions have stable performance in numerical experiments, which can show the optimization performance of the test algorithm.

In the experiment, the performance of IEO optimization results is analyzed and compared with six famous metaheuristic algorithms, which are Equilibrium Optimizer (EO)[18], Salp Swarm Algorithm (SSA)[25], Sine and Cosine Algorithm (SCA)[23], Butterfly Optimization Algorithm(BOA)[26], the Particle Swarm Optimization (PSO)[14] and Bat Algorithm (BA)[27].

For fair comparison, two standards of mean value (mean) and standard deviation (std) are used as the evaluation indexes. The average value intuitively shows the results of function optimization of each algorithm, and the standard deviation reflects the dispersion degree of optimization data. The smaller the standard deviation is, the higher the stability of the algorithm is. The following three experiments describe in detail the optimization of the improved algorithm IEO for different problems.

4.1 23 classical benchmark functions

In this section, 23 classical benchmark functions are selected for simulation experiments[17]. These functions are divided into unimodal functions and multimodal functions. Among them, F1-F7 is a unimodal function with only one global optimal solution, which is mainly used to test the optimization accuracy of the algorithm. The multimodal functions F8-F23 has multiple optima and is easy to fall into local optima. It is often used to test the exploration ability of the algorithm and the ability to avoid local optima. In addition, F14-F23 belong to fixed dimensional multimodal functions, whose dimensions are lower and fixed, so they have fewer local optimal solutions.

In order to ensure the fairness and comparability of the experiment, this paper sets the population number of the seven algorithms as 30, the maximum number of iterations as 500, and the dimension of the algorithms as 30. Each algorithm runs the test function for 30 times independently and records the mean value (mean) and standard deviation (std) of the experimental data. The optimization results of all the algorithms for the 23 classical functions are shown in Table 1. Note that the bold values in the table represent the best results for each function optimization.

Fig. 7. Mean rank of Friedman test on 23 benchmark functions

As can be seen from Table 1, IEO improves accuracy by several orders of magnitude for most benchmark functions. For unimodal function F1-F7, the optimization efficiency of the algorithm IEO is greatly improved. In particular, for functions F1 and F7, IEO can reach the global optimal solution 0, which indicates that IEO has strong exploitation ability and can find the global optimal solution. Compared with other algorithms, the algorithm IEO has superior performance in optimizing functions F2-F4. For multimodal functions F8-F23, the average value of fitness obtained by IEO in the optimization process of ten functions F9-F12, F15-F19 and F23 is optimal. This shows that the improved IEO has strong exploration ability and robustness when dealing with multimodal functions.

In order to analyze the optimization results of IEO and other algorithms more effectively, the Friedman test is selected as a further evaluation index in the experiment. The order is based on the mean and standard deviation of all the algorithms. If the mean is smaller, the rank obtained by the Friedman test is smaller. In the experiment, the software IBM SPSS24 is used to calculate each algorithm, and the sorting results are shown in Table 1. IEO's average rank is 2.18, ranking first among the seven algorithms, while EO, PSO, BOA, SSA, BA and SCA ranked second to seventh respectively. In addition, in order to view the average rank results of each algorithm more clearly, the Fig. 7 is drawn, which shows that the ranking of IEO is better than other algorithms. The results of Friedman simulation experiment show that IEO has superior performance compared with the original EO algorithm and PSO.

TABLE 1
THE COMPARISON RESULTS OF DIFFERENT ALGORITHMS ON 23 BENCHMARK FUNCTIONS WITH D=30

Function	Result	IEO	EO	SSA	SCA	BOA	PSO	BA
F1	Mean	**0**	2.85E-41	1.48E-07	10.8237	1.31E-11	2.70E-73	3.34E+03
	Std	**0**	5.08E-41	1.25E-07	11.3405	1.14E-12	1.43E-72	5.47E+03
F2	Mean	**1.08E-166**	8.08E-24	2.2539	0.0207	4.32E-09	1.48E-51	33.8185
	Std	**0**	7.82E-24	2.0791	0.0378	1.40E-09	5.78E-51	23.4872
F3	Mean	**2.01E-315**	1.27E-09	1.78E+03	6.87E+03	1.29E-11	4.31E+04	1.91E+04
	Std	**0**	3.54E-09	1.03E+03	4.15E+03	1.12E-12	1.41E+04	8.99E+03
F4	Mean	**8.21E-158**	3.46E-10	11.7559	34.3367	5.99E-09	44.3426	70.3417
	Std	**1.39E-157**	7.56E-10	3.0076	11.9158	4.42E-10	26.9932	8.7013
F5	Mean	25.4623	**25.3233**	281.2543	1.09E+05	28.9402	27.9294	2.68E+06
	Std	0.2692	0.2454	783.4751	2.60E+05	**0.0321**	0.428	1.46E+07
F6	Mean	3.67E-06	7.58E-06	**1.53E-07**	17.3566	6.0152	0.3729	3.01E+03
	Std	2.82E-06	7.00E-06	**2.15E-07**	23.2015	0.5234	0.2593	5.35E+03
F7	Mean	**0**	1.29E-74	2.95E-11	0.0229	1.15E-14	1.51E-111	4.3848
	Std	**0**	3.02E-74	7.31E-11	0.0437	1.33E-15	8.28E-111	12.7958
F8	Mean	-8.82E+03	-8.96E+03	-7.62E+03	-3.77E+03	-3.72E+03	**-9.97E+03**	-8.33E+03
	Std	537.6766	582.6549	729.6419	350.8453	351.4027	**1.70E+03**	792.1073
F9	Mean	**0**	**0**	50.0795	49.4458	2.93E-09	5.68E-15	168.2969
	Std	**0**	**0**	13.9522	39.0467	1.37E-08	2.29E-14	34.0651
F10	Mean	**2.31E-15**	8.35E-15	2.8441	15.8001	6.04E-09	5.15E-15	15.8891
	Std	**1.77E-15**	2.16E-15	1.1462	7.9825	5.53E-10	2.70E-15	6.2066
F11	Mean	**0**	6.56E-04	0.0169	0.9049	4.65E-12	0.0054	27.9399
	Std	**0**	0.0036	0.0126	0.3775	2.55E-12	0.0294	53.6106
F12	Mean	**5.93E-08**	4.70E-07	7.7091	6.24E+04	0.6566	0.0228	12.0731
	Std	**9.98E-08**	4.08E-07	4.1368	2.00E+05	0.1838	0.0138	21.5277
F13	Mean	0.6807	**0.0192**	17.3181	5.79E+05	2.9499	0.5275	1.01E+03
	Std	1.0265	**0.0371**	15.2915	2.27E+06	0.1343	0.2578	4.96E+03
F14	Mean	1.1964	**0.998**	1.1304	1.7264	1.5197	3.5412	3.3945
	Std	0.6054	**1.37E-16**	0.431	0.9718	0.6952	4.099	2.8086
F15	Mean	**3.54E-04**	0.0011	0.0028	0.0011	4.03E-04	6.95E-04	0.0017
	Std	**1.72E-04**	0.0037	0.006	3.51E-04	1.29E-04	4.69E-04	0.0036
F16	Mean	**-1.0316**	-1.0316	-1.0316	-1.0316	-9.94E+02	-1.0316	-1.0316
	Std	6.12E-16	6.12E-16	3.38E-14	4.38E-05	3.08E+03	1.02E-09	**6.78E-16**
F17	Mean	**0.3979**	0.3979	NAN	0.3997	NAN	0.3979	0.3979
	Std	**0**	0	NAN	0.0032	NAN	2.96E-05	0
F18	Mean	**3**	3	3	3.0001	3.3123	3	3
	Std	**1.44E-15**	1.33E-15	2.87E-13	1.33E-04	1.109	1.09E-04	2.00E-15
F19	Mean	**-3.8628**	-3.8628	-3.8628	-3.8539	-4.0035	-3.8565	-3.8628
	Std	**2.49E-15**	2.57E-15	1.81E-11	0.0028	0.3408	0.0086	2.71E-15
F20	Mean	-3.8625	**-3.8628**	-3.8628	-3.8536	-3.7484	-3.8528	-3.8628
	Std	0.0014	**2.54E-15**	8.20E-11	0.0026	3.6326	0.0202	2.71E-15
F21	Mean	-8.5634	**-8.8053**	-7.3911	-2.5161	-4.4844	-8.5282	-5.7239
	Std	2.5078	2.5419	3.3142	1.7151	**0.3622**	2.492	3.3265
F22	Mean	-9.4631	**-10.0031**	-8.6979	-3.6573	-4.2444	-8.0746	-7.5472
	Std	2.4778	1.532	3.1699	1.6085	**0.6247**	3.1563	3.4071
F23	Mean	**-9.9204**	-9.8623	-8.9919	-4.0564	-4.0375	-6.8564	-7.2819
	Std	**1.9106**	2.0874	2.9026	2.0752	0.553	3.2605	3.8045
Average rank		**2.18**	2.25	4.23	5.82	4.18	3.70	5.64
Rank		**1**	2	5	7	4	3	6

4.2 IEEE CEC2017 functions

The conditions of the CEC2017 test suite are more complex and challenging than unconstrained functions. Therefore, the CEC2017 is selected as the optimization problem to evaluate the algorithm IEO. For the CEC2017 test suite[28], there are four types of problems: they are: F1-F3 are unimodal rotation displacement functions, which are usually used to evaluate the convergence rate and optimization accuracy of the algorithm; F4-F10 are multimodal rotation displacement functions, which are usually used to reflect the ability of the algorithm to avoid local optimality. In addition, F11-F20 are hybrid functions, and F21-F30 are composition functions. It is difficult for most algorithms to reach the global optimal solution of the hybrid functions and the composite functions. Among them, F2 is deleted from the function list due to its instability and is not studied. Therefore, this section conducts simulation experiments on the basis of 29 functions.

In the experiment, we choose the above six meta-heuristic algorithms for comparison. In order to maintain the fairness of experimental data, the population number is uniformly set as 30, the maximum number of iterations is set as 500, and the dimension of the function is 30. All functions are independently run for 30 times and their average value (mean) and standard deviation (std) are recorded. Details of the experimental data are shown in Table 2.

It is clear from Table 2 that the algorithm IEO is superior to other methods. Specifically, the average value of fitness obtained by IEO reaches the minimum when optimizing 18 functions (F3-F5, F10-F12, F14-F19, F21-F24, F26 and F30). In addition, the original EO algorithm achieves the lowest average fitness values for the 10 functions (F6-F9, F13, F20, F25, F27-29). The fitness value obtained by SSA when optimizing function F1 is the smallest. From the experimental data of CEC2017, it can be seen that IEO is more efficient than the original EO algorithm, especially for hybrid functions and composition functions, IEO algorithm has superior performance, which indicates that IEO has better optimization efficiency and convergence speed.

TABLE 2
THE COMPARISON RESULTS OF DIFFERENT ALGORITHMS ON CEC2017 FUNCTIONS WITH D=30

Function	Result	IEO	EO	SSA	SCA	BOA	PSO	BA
F1	Mean	6.64E+04	1.56E+05	**7.67E+03**	2.20E+10	5.61E+10	5.59E+09	1.03E+10
	Std	8.37E+04	3.50E+05	**7.19E+03**	3.36E+09	9.76E+09	1.80E+09	6.70E+09
F3	Mean	**4.26E+04**	5.28E+04	7.56E+04	8.20E+04	7.96E+04	2.57E+05	1.55E+05
	Std	**7.34E+03**	1.25E+04	2.69E+04	1.67E+04	9.67E+03	7.02E+04	5.72E+04
F4	Mean	**511.7805**	516.734	534.4998	3.00E+03	2.04E+04	1.37E+03	1.35E+03
	Std	**18.6095**	24.0721	44.3882	6.87E+02	3.59E+03	473.671	844.6995
F5	Mean	**5.99E+02**	603.0197	6.84E+02	831.9749	9.17E+02	860.098	695.3999
	Std	**2.58E+01**	28.7098	4.07E+01	28.8511	2.84E+01	50.5377	45.4718
F6	Mean	609.3797	**6.02E+02**	6.53E+02	661.6059	6.91E+02	682.8361	637.6466
	Std	4.1784	**1.7633**	1.02E+01	7.1982	7.09E+00	12.6851	10.3965
F7	Mean	872.0349	**8.58E+02**	9.79E+02	1.26E+03	1.40E+03	1.31E+03	1.11E+03
	Std	41.1012	**39.3815**	9.64E+01	60.4739	3.57E+01	102.3825	167.919
F8	Mean	896.4204	**8.91E+02**	9.67E+02	1.09E+03	1.14E+03	1.07E+03	1.01E+03
	Std	25.4733	2.66E+01	3.87E+01	24.2593	**22.8111**	61.2718	51.044
F9	Mean	2.65E+03	**1.24E+03**	5.68E+03	9.01E+03	1.12E+04	1.06E+04	7.58E+03
	Std	1.25E+03	**5.37E+02**	1.79E+03	1.88E+03	1.36E+03	3.26E+03	2.40E+03
F10	Mean	**4.73E+03**	5.47E+03	5.51E+03	8.89E+03	9.11E+03	7.77E+03	5.53E+03
	Std	525.528	9.43E+02	7.86E+02	332.9991	**314.9877**	627.4289	653.9668
F11	Mean	**1.24E+03**	1.25E+03	1.41E+03	3.81E+03	8.68E+03	8.83E+03	5.52E+03
	Std	**37.8988**	4.80E+01	116.0291	717.8026	2.24E+03	4.14E+03	4.92E+03
F12	Mean	**1.30E+06**	1.69E+06	4.19E+07	2.68E+09	1.39E+10	5.31E+08	1.99E+08
	Std	**1.04E+06**	1.33E+06	3.69E+07	7.32E+08	4.34E+09	5.00E+08	5.04E+08
F13	Mean	2.27E+04	**2.16E+04**	1.51E+05	1.14E+09	1.07E+10	1.29E+07	4.33E+07
	Std	**1.57E+04**	2.05E+04	7.86E+04	5.33E+08	5.32E+09	9.94E+06	1.93E+08
F14	Mean	**5.22E+04**	6.84E+04	1.34E+05	9.13E+05	4.52E+06	3.18E+06	8.64E+05
	Std	**5.08E+04**	5.52E+04	1.14E+05	5.05E+05	5.40E+06	4.13E+06	2.13E+06
F15	Mean	**4.49E+03**	8.71E+03	6.69E+04	5.54E+07	6.86E+08	7.71E+06	5.42E+04
	Std	**3.41E+03**	9.31E+03	7.24E+04	4.07E+07	4.38E+08	9.29E+06	3.98E+04
F16	Mean	**2.44E+03**	2.61E+03	3.12E+03	4.12E+03	8.73E+03	4.48E+03	3.02E+03
	Std	284.6801	333.867	3.37E+02	**264.0054**	1.87E+03	720.7158	425.317
F17	Mean	**2.07E+03**	2.11E+03	2.35E+03	2.82E+03	1.76E+04	2.86E+03	2.49E+03
	Std	**178.033**	228.2042	228.2704	199.3174	1.74E+04	293.9794	249.1799
F18	Mean	**3.34E+05**	9.27E+05	2.32E+06	1.47E+07	5.89E+07	1.45E+07	1.20E+07
	Std	**3.40E+05**	9.15E+05	2.60E+06	7.99E+06	6.64E+07	1.78E+07	2.09E+07
F19	Mean	**4.96E+03**	8.14E+03	5.06E+06	1.16E+08	9.74E+08	2.13E+07	2.37E+07
	Std	**1.76E+03**	8.26E+03	3.11E+06	6.30E+07	1.19E+09	1.96E+07	5.13E+07
F20	Mean	2.45E+03	**2.42E+03**	2.59E+03	2.92E+03	3.06E+03	2.94E+03	2.71E+03
	Std	180.8249	**1.52E+02**	1.65E+02	160.5336	148.0879	248.8265	188.2608
F21	Mean	**2.37E+03**	2.38E+03	2.45E+03	2.61E+03	2.70E+03	2.66E+03	2.50E+03
	Std	**15.1045**	2.24E+01	4.26E+01	22.3865	98.8543	48.0492	49.3209
F22	Mean	**2.30E+03**	4.55E+03	5.70E+03	9.09E+03	6.91E+03	8.46E+03	6.09E+03
	Std	**3.4659**	2.24E+03	2.15E+03	2.44E+03	1.29E+03	1.25E+03	1.56E+03
F23	Mean	**2.72E+03**	2.73E+03	2.80E+03	3.09E+03	3.60E+03	3.17E+03	2.83E+03
	Std	**20.9042**	21.8163	3.98E+01	49.963	1.29E+02	110.3316	40.5196
F24	Mean	**2.89E+03**	2.90E+03	2.95E+03	3.25E+03	4.16E+03	3.26E+03	2.99E+03
	Std	**16.6963**	27.0045	3.53E+01	31.1493	251.4123	99.9632	37.488
F25	Mean	2.92E+03	**2.91E+03**	2.95E+03	3.67E+03	6.06E+03	3.21E+03	3.25E+03
	Std	21.9976	**1.48E+01**	3.39E+01	339.7615	5.53E+02	94.6752	321.6041
F26	Mean	**3.90E+03**	4.28E+03	4.96E+03	7.83E+03	1.18E+04	8.55E+03	5.71E+03
	Std	1.17E+03	5.48E+02	1.35E+03	**345.4336**	1.02E+03	937.3668	424.4831
F27	Mean	3.23E+03	**3.22E+03**	3.27E+03	3.55E+03	4.37E+03	3.47E+03	3.26E+03
	Std	13.421	**1.06E+01**	3.41E+01	82.9742	3.31E+02	159.0451	21.3432
F28	Mean	3.28E+03	**3.26E+03**	3.30E+03	4.37E+03	8.23E+03	3.87E+03	4.35E+03
	Std	**20.5387**	2.76E+01	3.59E+01	306.6997	515.4607	219.6228	985.7202
F29	Mean	3.80E+03	**3.77E+03**	4.38E+03	5.19E+03	1.23E+04	5.36E+03	4.26E+03
	Std	**162.1826**	1.81E+02	2.92E+02	265.2462	5.89E+03	478.9612	306.377
F30	Mean	**1.58E+04**	1.90E+04	1.11E+07	2.08E+08	1.88E+09	7.26E+07	4.76E+05
	Std	**8.34E+03**	1.25E+04	1.07E+07	8.79E+07	1.32E+09	6.00E+07	8.50E+05
Average rank		**1.38**	1.69	3.14	5.48	6.79	5.45	4.07
Rank		**1**	2	3	6	7	5	4

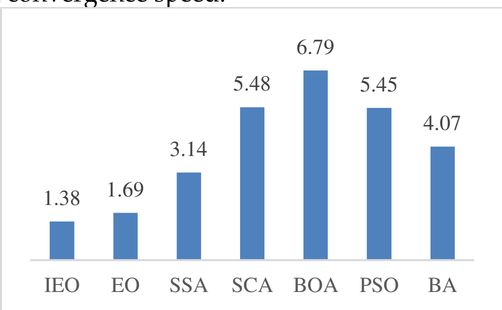


Fig. 8. Mean rank of Friedman test on CEC2017 functions

As shown in Fig. 8, for the Friedman statistical test, the value of IEO's mean rank is 1.38, which is much better than the value of the original EO algorithm. According to the Friedman test results in Fig. 8, IEO ranks best, followed by EO, SSA, BA, PSO, SCA, and BOA. These results also indicate that IEO has higher convergence accuracy when optimizing the CEC2017 test suite.

4.3 IEEE CEC2019 functions

In this section, the more challenging test suite IEEE CEC2019[29] is selected to evaluate the algorithms. These test functions are minimal and scalable, the functions F1, F2 and F3 are 9-dimensional, 16-dimensional and 18-dimensional problems respectively, and they have different value ranges. Moreover, the functions F4-F10 are all 10-dimensional problems and have the same search scope [-100,100]. In addition, the functions F4-F10 have different rotation matrices. In the experiment, IEO is compared with Equilibrium Optimizer (EO)[18]，salp swarm algorithm(SSA)[25], sine and cosine algorithm (SCA)[23], butterfly optimization algorithm (BOA)[26], the particle swarm optimization (PSO)[14] and bat algorithm (BA)[27]. For all the algorithms, we set the population number of each algorithm as 30, the maximum iteration number as 500, and all the algorithms are independently run on each function for 30 times and recorded the average value (mean) and standard deviation (std) of the fitness value. The results are shown in Table 3.

As shown in Table 3, it is obvious that the IEO obtains the minimum average value of the fitness values of the ten functions F1-F10. In particular, compared with the original EO algorithm, the optimization accuracy of the improved IEO algorithm for the CEC2019 functions has been improved to varying degrees. In other words, IEO is significantly better than other algorithms in optimizing the CEC2019 function. In addition, IEO has a mean rank

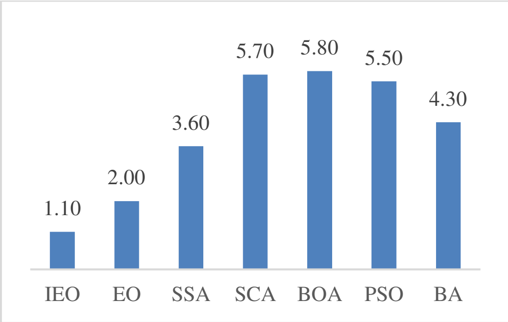


Fig. 9. Mean rank of Friedman test on CEC2019 functions

of 1.1 in the Friedman statistical test, which is smaller than the other six comparison algorithms. The Fig. 9 plots the average rank of each algorithm obtained in Friedman test. It can be clearly seen that IEO ranks first, while EO, SSA, BA, PSO, SCA and BOA rank second to seventh. Overall, it is clear from function optimization and Friedman test that IEO performs well with CEC2019.

4.3 Analysis of Convergence curve

The convergence speed and accuracy of each algorithm can be seen in detail from the convergence curve. The CEC2017 test suite contains functions of different complexity levels, which best represent the optimization performance of the algorithm. Therefore, this section analyzes the convergence status of each algorithm by drawing the convergence curve of the algorithm when optimizing the CEC2017 test suite. The Fig.10 describes the convergence results of IEO, EO, SSA, SCA, BOA, PSO and BA algorithms when the dimension D=30. We randomly select six test functions (unimodal, multimodal, hybrid and composition functions from the CEC2017 function set). In the figure, the horizontal axis represents the maximum iteration times of the algorithm 500 times, and the vertical axis represents the fitness value obtained in the optimization process of the algorithm.

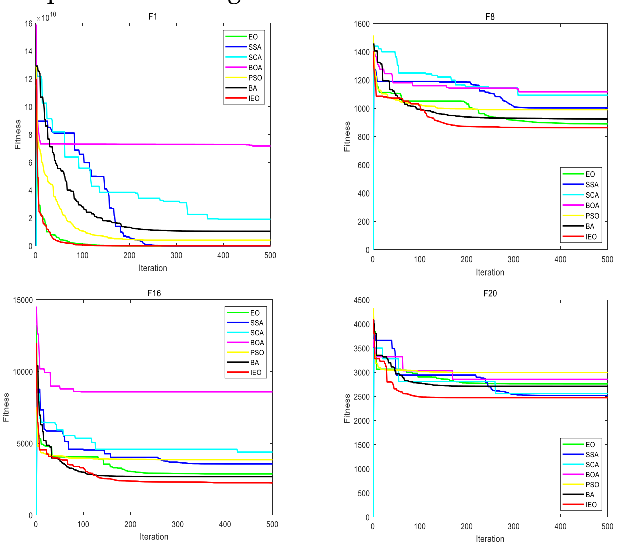


TABLE 3
THE COMPARISON RESULTS OF DIFFERENT ALGORITHMS ON CEC2019

Function	Result	IEO	EO	SSA	SCA	BOA	PSO	BA
F1	Mean	**1**	1	2.22E+06	6.94E+06	1	1.22E+07	1.34E+07
	Std	**0**	0	2.12E+06	7.83E+06	0	1.27E+07	1.82E+07
F2	Mean	**4.4745**	4.53	1.62E+03	3.96E+03	4.991	7.59E+03	1.09E+03
	Std	0.3504	0.3657	993.4936	1.68E+03	**0.0337**	3.51E+03	1.17E+03
F3	Mean	**1.5117**	1.5596	4.2898	9.2874	6.2543	4.8255	7.1059
	Std	**0.3153**	0.4958	1.7503	1.3239	0.8528	1.8222	2.0162
F4	Mean	**14.8968**	15.4703	28.7514	47.5372	85.4029	54.4605	29.7908
	Std	**6.1189**	6.4024	12.1888	8.8151	14.2692	18.5242	11.6568
F5	Mean	**1.0487**	1.055	1.1848	10.1705	108.6633	2.4557	2.617
	Std	**0.0361**	0.0422	0.1081	3.2055	23.6808	0.6545	4.5385
F6	Mean	**1.6811**	1.8179	4.8786	7.8184	9.0551	9.0917	4.8567
	Std	**0.7109**	0.7468	1.9377	1.0821	0.9323	1.9221	1.7535
F7	Mean	**753.474**	858.0009	1.10E+03	1.62E+03	1.94E+03	1.39E+03	995.6335
	Std	258.7518	312.3589	327.5567	234.8356	**185.5797**	375.4894	346.0445
F8	Mean	**3.7105**	3.7587	4.282	4.5317	4.8326	4.6369	4.4854
	Std	0.445	0.4918	0.4283	0.2465	**0.196**	0.3081	0.3704
F9	Mean	**1.1791**	1.187	1.3712	1.6572	4.2831	1.4193	1.3426
	Std	**0.0551**	0.0687	0.1629	0.1551	0.4278	0.1465	0.2017
F10	Mean	**17.5907**	18.0956	20.3658	21.496	21.5022	21.2662	21.1728
	Std	7.6927	7.43	3.6585	0.0898	**0.086**	0.1105	0.143
Average rank		**1.10**	2.00	3.60	5.70	5.80	5.50	4.30
Rank		**1**	2	3	6	7	5	4

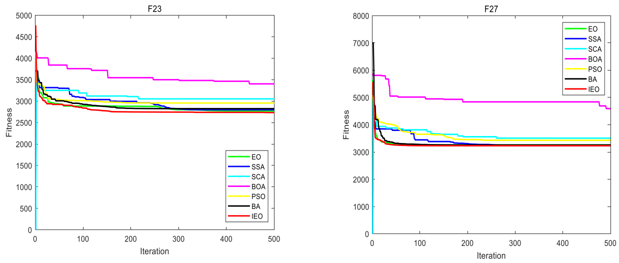


Fig. 10. The convergence curve of seven algorithms on CEC2017 functions

It is clear from the figure that IEO has improved significantly for most functions. For other algorithms, EO has the best convergence, followed by SSA, PSO, BA, SCA and BOA.

For unimodal and multimodal functions (F1, F8), the convergence speed and accuracy of IEO algorithm are obviously better than other algorithms. It can be seen from the convergence curve that IEO converges faster and reaches the minimum fitness value at the early stage of iteration.

For the hybrid functions (F16, F20), although the convergence curve of IEO algorithm differs little from that of other algorithms, the convergence curve of IEO algorithm is relatively smooth and can form fast convergence within 100 iterations, showing good performance in terms of convergence speed.

For the composition functions (F23, F27), the convergence curve of IEO algorithm is better than other algorithms. In addition, the step size of IEO algorithm is obviously smaller than that of other algorithms, and it can always converge to the optimal solution at the beginning of iteration, while other algorithms tend to fall into the local optimal solution, resulting in slower convergence speed and lower convergence accuracy.

In general, the convergence curve shows that the IEO algorithm has good performance in optimizing the CEC2017 functions. In particular, the IEO algorithm has faster convergence speed and higher convergence accuracy when dealing with the hybrid and composition functions in CEC2017. The convergence curve of CEC2017 also verifies the results in Table 2 again.

4.4 Stability analysis

In this section, the boxplot is used to show the data distribution of CEC2017 test suite running for 30 times independently. It describes the stability and optimization performance of experimental data by using five statistics such as the maximum value, minimum value, upper quartile, lower quartile and median in the data. In addition, boxplot can not only reflect the fluctuation degree of data through the height of box, but also show the stability of data through the number of outliers. Similarly, we select six functions from the CEC2017 function set (unimodal, multimodal, hybrid and composition functions respectively). Fig. 11 shows the boxplot when dimension is set to 30, population number is 30, and maximum number of iterations is 500. In the figure, the horizontal axis represents each comparison algorithm, and the vertical axis represents the range of fitness values obtained by the optimization function.

For unimodal and multimodal functions (F1, F8), the differences between the maximum, minimum and median of five indexes of IEO algorithm in the boxplot are the smallest, and there are no outliers, which indicates that IEO algorithm has strong stability in the process of optimizing unimodal and multimodal functions.

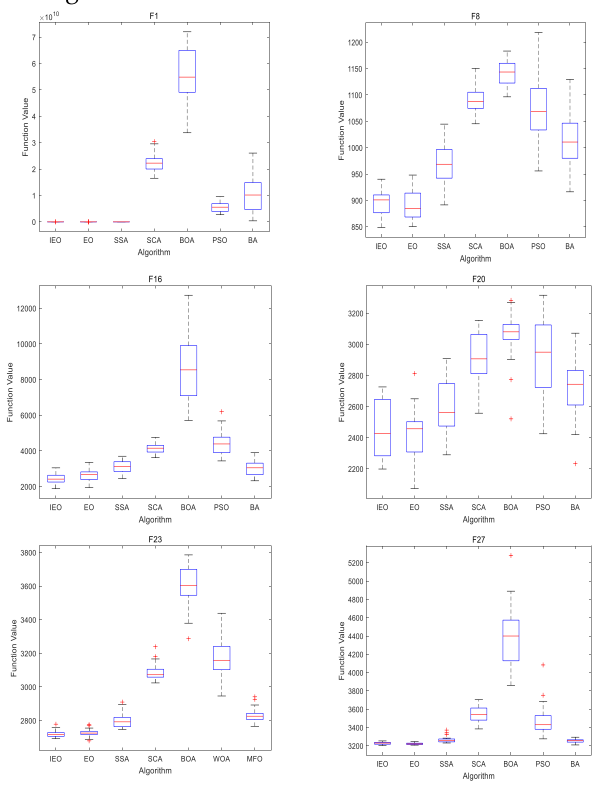


Fig. 11. The boxplot of seven algorithms on CEC2017 functions

By observing the boxplot of hybrid functions (F16, F20), it can be seen that the data difference of IEO algorithm is small, and the number of outliers is also small, which indicates that IEO has strong stability when processing the hybrid functions. For function F20, although the difference between BOA's maximum value and minimum value is the smallest, BOA produces a large number of outliers in the optimization process, while IEO does not produce outliers when processing function F20, and its median value is the smallest among seven algorithms.

Compared with other algorithms, the IEO algorithm is the most stable when optimizing the composition functions (F23, F27). Specifically, in the process of running IEO algorithm for 30 times independently, the data difference is the smallest, and the median value is also the smallest among the seven algorithms, which indicates that IEO algorithm has superior optimization performance.

Overall, the boxplot shows that IEO has better performance. In the initial phase of IEO algorithm, Tent chaotic mapping is introduced to improve the quality of initial solution, and dynamic control parameter strategy is introduced to maintain the balance between exploration and exploitation phase in the iterative process, so that the particles can reach balance state more effectively. In addition, the LOBL strategy is used to calculate the opposite solution of candidate solution for each iteration, which improves the population diversity. Therefore, compared with the original EO algorithm, the stability and optimization performance of the improved IEO algorithm are greatly improved.

5 ENGINEERING DESIGN PROBLEMS

In this section, the efficiency of IEO algorithm in solving practical application problems is tested by solving six engineering problems. The solution of engineering optimization problems is to give the optimal design scheme under the premise of satisfying multiple constraints. In the experiment, the overall size is set as 30, and the maximum number of iterations is 500. IEO is compared with various meta-heuristic algorithms, and the optimal solution of each problem is shown in bold in the table.

5.1 Pressure vessel design problem

The optimization objective of pressure vessel design problem[30] is to minimize the total cost of cylindrical pressure vessels，a schematic diagram of this problem is shown in Fig. 12, where four key optimization variables are involved：thickness of the head (T_h), the thickness of the shell (T_s), the inner radius (R), and the length of the cylindrical section without considering the head(L). The mathematical expression of this problem is as follows:

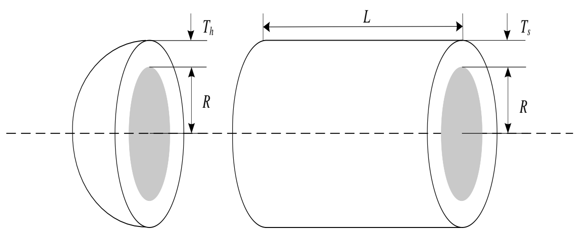


Fig. 12. Pressure vessel design problem

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4] = [T_s\ T_h\ \mathrm{R}\ \mathrm{L}]$

Minimize

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0$$
$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0$$
$$g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$
$$g_4(\vec{x}) = x_4 - 240 \le 0$$

Variable range $0 \le x_1 \le 99$, $0 \le x_2 \le 99$, $10 \le x_3 \le 200$, $10 \le x_4 \le 200$

Four key variables of the pressure vessel problem are optimized by IEO algorithm and the optimal values are obtained, and the results are compared with the data of 12 algorithms that have solved the problem. Table 4 shows the values of the lowest costs and associated variables derived from each algorithm.

As can be seen from Table 4, IEO algorithm can obtain the minimum cost when solving the pressure vessel problem, and the values of four related parameters are relatively good, which effectively saves the engineering design cost. When the values of the four variables T_s, T_h, R and L are respectively 0.7790748, 0.3850971, 40.36657 and 199.3474, the lowest cost obtained is 5886.8835.

5.2 Welded beam design problem

The objective of welded beam design problem is to minimize the manufacturing cost of welded beam design[38], as shown in Fig. 13. The following constraints must be met during the optimization of welded beam problem： height (t), thickness of weld (h), length (l), and thickness (b) of the bar.

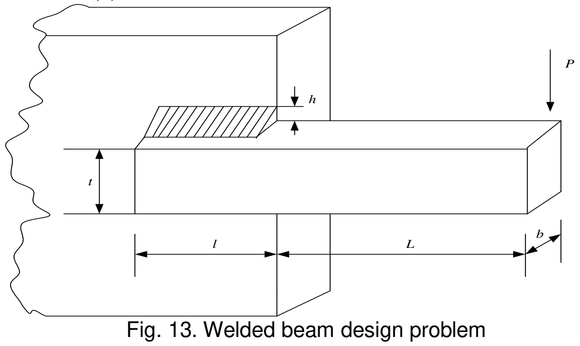


Fig. 13. Welded beam design problem

The mathematical model of this problem is expressed as follows：

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4] = [h\ l\ t\ b]$

Minimize $f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Subject to

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0;$$
$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0$$
$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0$$
$$g_4(\vec{x}) = x_1 - x_4 \le 0$$
$$g_5(\vec{x}) = p - p_c(\vec{x}) \le 0$$
$$g_6(\vec{x}) = 0.125 - x_1 \le 0$$

TABLE 4

COMPARISON OF RESULT ON PRESSURE VESSEL DESIGN PROBLEM.

Algorithm	Optimal values for variables				Optimal cost
	T_s	T_h	R	L	
IEO	**0.7790748**	**0.3850971**	**40.36657**	**199.3474**	**5886.8835**
EO	0.8257301	0.4081611	42.78394	168.3235	5971.7095
m-EO[19]	0.8125	0.4375	42.0984	76.6366	6059.7144
ESSAWOA[31]	0.7817639	0.3864301	40.5056956	197.4631899	5892.3546036
WOA[17]	0.812500	0.437500	42.0982699	176.638998	6059.7410
hHHO-SCA[32]	0.945909	0.447138	46.8513	125.4684	6393.092794
GWO[33]	0.8125	0.4345	42.0892	176.7587	6051.564
DDSCA[34]	0.7782114	0.3855657	40.31989	200	5888.3366
ES[35]	0.8125	0.4375	42.098087	176.640518	6059.74560
AFA[36]	0.8125	0.4375	42.0984	176.6366	6059.7143
LSA-SM[37]	0.8103764	0.4005695	41.98842	178.0048	5942.6966
HHO[2]	0.8175838	0.4072927	42.09174576	176.7196352	6000.46259
GSA[9]	1.125	0.625	55.9886598	84.4542025	8538.8359

TABLE 5

COMPARISON OF RESULT ON WELDED BEAM DESIGN PROBLEM.

Algorithm	Optimal values for variables				Optimal cost
	h	l	t	b	
IEO	**0.20573**	**3.4703**	**9.0372**	**0.20573**	**1.7249**
EO	0.20593	3.4681	9.0322	0.20594	1.7257
CPSO[39]	0.202369	3.544214	9.048210	0.205723	1.72802
Random[40]	0.4575	4.7313	5.0853	0.6600	4.1185
CDE[41]	0.20317	3.542998	9.033498	0.206179	1.733462
IACO[42]	0.205700	3.471131	9.036683	0.205731	1.724918
RO[43]	0.203687	3.528467	9.004263	0.207241	1.735344
HHO[2]	0.204039	3.531061	9.027463	0.206147	1.7319906
ESSAWOA[31]	0.2055051	3.4753160	9.0366562	0.2057295	1.7251597
EHOI[44]	0.205377	3.472652	9.050768	0.205659	1.726501
BBO[45]	0.2287	3.2003	8.5666	8.9985	1.8077
IPSO[46]	0.2444	6.2175	8.2915	0.2444	2.3810
MTSA[47]	0.2442	6.2231	8.2956	0.2444	2.3824
DDSCA[34]	0.20516	3.4759	9.0797	0.20552	1.7305
NOSA[48]	0.2444	6.2175	8.2915	0.2444	2.3810

$$g_7(\vec{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0$$

Variable range $0.1 \le x_1 \le 2$, $0.1 \le x_2 \le 10$, $0.1 \le x_3 \le 10$, $0.1 \le x_4 \le 2$

where $\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$, $\tau' = \frac{p}{\sqrt{2}x_1x_2}$, $\tau'' = \frac{MR}{J}$ $M = p\left(L + \frac{x_2}{2}\right)$, $\mathrm{R} = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2}$

$\mathrm{J} = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$, $\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}$, $\delta(\vec{x}) = \frac{4PL^3}{Ex_3^3x_4}$

$P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$ $p = 6000\mathrm{lb}$, $\mathrm{L} = 14$ in., $\delta_{max} = 0.25\ in.$ $\mathrm{E} = 30 \times 10^6\mathrm{psi}$, $\mathrm{G} = 12 \times 10^6\mathrm{psi}$, $\tau_{max} = 13{,}600psi$, $\sigma_{max} = 30{,}000psi$

The welded beam problem is optimized by IEO and the original EO algorithm, and the experimental results are compared with 13 algorithms in other literatures. Table 5 shows the results of different algorithms and the values of four related parameters.

It can be seen from Table 5 that the IEO algorithm has the smallest manufacturing cost for welded beam problem. In other words, when the value of four key parameters are 0.20573, 3.4703, 9.0372, 0.20573, the manufacturing cost of welded beam is 1.7249. This shows that the performance of IEO is better than other algorithms. It not only improves the optimization efficiency, but also reduces the cost of solving welded beam problem.

5.3 Tension/compression spring design problem

The tension/compression spring problem is a classic structural engineering design problem[49]，whose purpose is to minimize the weight of tension/compression spring. To solve the problem, three core variables are needed：wire diameter (*d*), mean coil diameter (*D*), and number of active coils (*N*). The details of the spring and the three parameters are shown in Fig. 14.

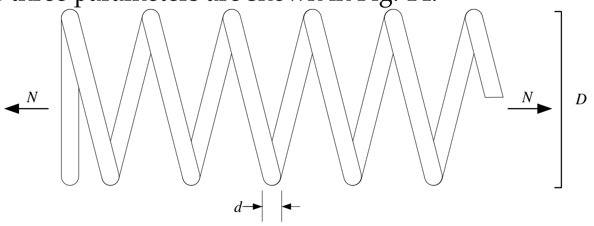


Fig. 14. Tension/compression spring design problem

The mathematical model of this problem is as follows：

Consider $\vec{x} = [x_1\ x_2\ x_3] = [d\ D\ N]$,

Minimize $f(\vec{x}) = (x_3 + 2)x_2x_1^2$,

Subject to $g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0$,

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 ,$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0 ,$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 ,$$

Variable range $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.30, 2 \leq x_3 \leq 15$,

On the basis of IEO and EO, the tension/compression spring problem is optimized and the values of relevant parameters are obtained. The optimization results are compared with 10 algorithms in other literatures. The detailed information is shown in Table 6.

As can be seen from Table 6, compared with other algorithms, the spring weight obtained by IEO algorithm is the smallest. When the values of *d, D* and *N* are 0.0516704，0.356269 and 11.3153 respectively，the weight of tension/compression spring problem is 0.012665. In general, IEO algorithm can effectively obtain the optimal solution of engineering problems and get the best parameter values.

5.4 Three-bar truss design problem

The problem of three-bar truss is a common application problem in civil engineering field [55]，and its optimization purpose is to minimize the weight of the three-bar truss. This engineering problem includes two core parameters：A_1 and A_2. The details are shown in Fig. 15.

The mathematical formula of this problem is expressed as follows：

TABLE 6

COMPARISON OF RESULT ON TENSION/COMPRESSION SPRING DESIGN PROBLEM.

Algorithm	Optimal values for variables			Optimal cost
	d	*D*	*N*	
IEO	**0.0516704**	**0.356269**	**11.3153**	**0.012665**
EO	0.0528	0.38404	9.8503	0.012687
BGRA[50]	0.0516747	0.3563726	1.309229	0.012665237
NM-PSO[51]	0.051620	0.355498	11.3333272	0.0126706
ES[35]	0.051643	0.355360	11.397926	0.012698
IHS[52]	0.0511543	0.3498711	12.0764321	0.0126706
AFA[36]	0.0516674837	0.3561976945	11.3195613646	0.0126653049
LSA-SM[37]	0.05170453	0.3570899	11.26718	0.01266524
EHO[44]	0.053666	0.406156	8.887284	0.012736
BWOA[53]	0.051602	0.357488	11. 24 41198	0.0126654
GSA[54]	0.050276	0.323680	13.525410	0.0127022
HS[52]	0.051609	0.354714	11.410831	0.0126702

TABLE 7

COMPARISON OF RESULT ON THREE-BAR TRUSS DESIGN PROBLEM.

Algorithm	Optimal values for variables		Optimal cost
	A_1	A_2	
IEO	**0.78868**	**0.40822**	**263.8958**
EO	0.7896	0.40563	263.8965
m-SCA[56]	0.81915	0.36956	263.8972
CS[57]	0.78867	0.40902	263.9716
Tsai[58]	0.788	0.408	263.68
Ray and Sain[55]	0.795	0.395	264.3
BWOA[53]	0.788666327	0.408273202	263.8958435
m-EO[1]	0.78834565	0.40918256	263.89607783
OBTLBO[19]	0.78909	0.40706	263.89600

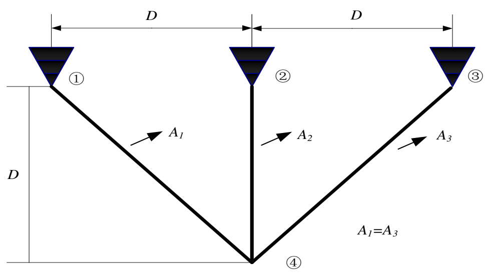


Fig. 15. Three-bar truss design problem

Consider $\vec{x} = [x_1\ x_2\ ] = [A_1\ A_2]$,

Minimize $f(\vec{x}) = \left(2\sqrt{2}x_1 + x_2\right) * l$

Subject to $g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0$,

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 ,$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0 ,$$

Variable range $0 \leq x_1, x_2 \leq 1$

$l = 100cm, P = 2\ km/cm^2 , \sigma = 2\ km/cm^2$.

The IEO algorithm is applied to the three-bar truss problem, and the experimental results are compared with eight other algorithms in other literatures. The results are shown in Table 7. Compared with other algorithms, it is obvious that IEO obtains the optimal solution for solving three-bar truss problem. When the parameters are set as A_1 =0.78868 and A_2 =0.40822， the minimum value of 263.8958 is obtained by IEO when optimizing the three-bar truss problem. This shows that IEO also has superior performance in solving practical engineering problems.

5.5 Speed reducer design problem

Speed reducer problem is an engineering problem with complex constraints, and its optimization purpose is to minimize the weight of speed reducer itself. The constraint variables are shown in Fig.16.

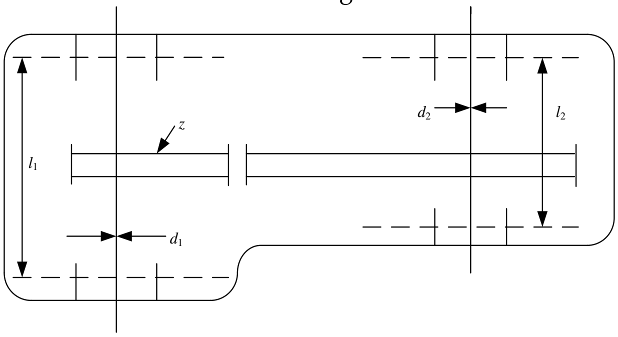


Fig. 16. Speed reducer design problem

The mathematical model of speed reducer is as follows:

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7] = [b\ m\ z\ l_1\ l_2\ d_1\ d_2]$

Minimize $f(\vec{x}) = 0.7894x_2^2x_1(14.9334x_3 - 43.0934 + 3.3333x_3^2) + 0.7854(x_5x_7^2 + x_4x_6^2) - 1.508x_1(x_7^2 + x_6^2) + 7.477(x_7^3 + x_6^3)$

Subject to $g_1(\vec{x}) = -x_1x_2^2x_3 + 27 \le 0$

$g_2(\vec{x}) = -x_1x_2^2x_3^2 + 397.5 \le 0$

$g_3(\vec{x}) = -x_2x_6^4x_3x_4^{-3} + 1.93 \le 0$

$g_4(\vec{x}) = -x_2x_7^4x_3x_5^{-3} + 1.93 \le 0$

$g_5(\vec{x}) = 10x_6^{-3}\sqrt{16.91 \times 10^6 + (745x_4x_2^{-1}x_3^{-1})^2} - 1100 \le 0$

$g_6(\vec{x}) = 10x_7^{-3}\sqrt{157.5 \times 10^6 + (745x_5x_2^{-1}x_3^{-1})^2} - 850 \le 0$

$g_7(\vec{x}) = x_2x_3 - 40 \le 0$

$g_8(\vec{x}) = -x_1x_2^{-1} + 5 \le 0$

$g_9(\vec{x}) = x_1x_2^{-1} - 12 \le 0$

$g_{10}(\vec{x}) = 1.5x_6 - x_4 + 1.9 \le 0$

$g_{11}(\vec{x}) = 1.1x_7 - x_5 + 1.9 \le 0$

Variable range $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4, x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5 \le x_7 \le 5.5$

On the basis of improved algorithm IEO, the speed reducer problem is optimized and the values of relevant parameters are obtained. The optimization results are compared with seven algorithms in other literatures. The details are shown in Table 8.

Compared with other algorithms, the improved algorithm IEO in this paper has higher accuracy in dealing with speed reducer engineering problem. In other words, The IEO algorithm find the best values for seven design variables to minimize the weight of speed reducer.

TABLE 8
COMPARISON OF RESULT ON SPEED REDUCER DESIGN PROBLEM.

Algorithm	Optimal values for variables							Optimal cost
	b	m	z	l_1	l_2	d_1	d_2	
IEO	**3.5000**	**0.7000**	**17.0000**	**7.30008**	**7.71532**	**3.35054**	**5.28665**	**2994.4254**
EO	3.5000	0.7000	17.0000	7.30366	7.71532	3.35055	5.28665	2994.4586
m-EO [19]	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2996.3482
SCA [19]	3.5198	0.7000	17.0000	7.3000	8.3000	3.4131	5.2919	3034.7970
OBSCA [19]	3.1507	0.7716	19.9472	7.7174	8.2332	3.5060	5.2938	3027.3263
TLBO [19]	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2996.3482
MDE [32]	3.50001	0.7000	17.0000	7.300156	7.800027	3.350221	5.286685	2996.35669
ABC [59]	3.5000	0.7000	17.0000	7.3000	7.715319	3.350214	5.286654	2994.47106

5.6 Optimal design of industrial refrigeration system

At present, energy saving and emission reduction work has become the focus of various fields. Industrial refrigeration system accounts for a large proportion of energy consumption, so it is necessary to optimize and control the industrial refrigeration system. Optimal design of industrial refrigeration system is an extremely complex engineering design problem, which has fourteen design variables and fifteen constraints. Its mathematical model is shown as follows：

Consider

$\vec{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}\ x_{13}\ x_{14}]$

Minimize $f(\vec{x}) = 63098.88x_2x_4x_{12} + 5441.5x_2^2x_{12} + 115055.5x_2^{1.664}x_6 + 6172.27x_2^2x_6 + 63098.88x_1x_3x_{11} + 5441.5x_1^2x_{11} + 115055.5x_1^{1.664}x_5 + 6172.27x_1^2x_5 + 140.53x_1x_{11} + 281.29x_3x_{11} + 70.26x_1^2 + 281.29x_1x_3 + 281.29x_3^2 + 14437x_8^{1.8812}x_{12}^{0.3424}x_{10}x_{14}^{-1}x_7x_9^{-1} + 20470x_7^{2.893}x_{11}^{0.316}x_1^2$

Subject to $g_1(\vec{x}) = 1.524x_7^{-1} \le 1$

$g_2(\vec{x}) = 1.524x_8^{-1} \le 1$

$g_3(\vec{x}) = 0.07789x_1 - 2x_7^{-3}x_9 - 1 \le 0$

$g_4(\vec{x}) = 7.05305x_9^{-1}x_1^2x_{10}x_8^{-1}x_2^{-1}x_{14}^{-1} - 1 \le 0$

$g_5(\vec{x}) = 0.0833x_{13}^{-1}x_{14} - 1 \le 0$

$g_6(\vec{x}) = 47.136x_2^{0.333}x_{10}^{-1}x_{12} - 1.333x_8x_{13}^{2.1195} + 62.08x_{13}^{2.1195}x_{12}^{-1}x_8^{0.2}x_{10}^{-1} - 1 \le 0$

$g_7(\vec{x}) = 0.04771x_{10}x_8^{1.8812}x_{12}^{0.3424} - 1 \le 0$

$g_8(\vec{x}) = 0.0488x_9x_7^{1.893}x_{11}^{0.316} - 1 \le 0$

$g_9(\vec{x}) = 0.0099x_1x_3^{-1} - 1 \le 0$

$g_{10}(\vec{x}) = 0.0193x_2x_4^{-1} - 1 \le 0$

$g_{11}(\vec{x}) = 0.0298x_1x_5^{-1} - 1 \le 0$

$g_{12}(\vec{x}) = 0.056x_2x_6^{-1} - 1 \le 0$

$g_{13}(\vec{x}) = 2x_9^{-1} - 1 \le 0$

$g_{14}(\vec{x}) = 2x_{10}^{-1} - 1 \le 0$

$g_{15}(\vec{x}) = x_{12}x_{11}^{-1} - 1 \le 0$

Variable range $0.001 \le x_i \le 5$, $i = 1, \ldots, 14$

Fourteen key variables of optimal design of industrial refrigeration system are optimized by IEO algorithm and the optimal values are obtained, and the results are compared with other meta-heuristic algorithms. Table 9 shows the lowest cost of each algorithm and the values of related variables.

TABLE 9
COMPARISON OF RESULT ON OPTIMAL DESIGN OF INDUSTRIAL REFRIGERATION SYSTEM.

variable	algorithm				
	IEO	EO	WOA	SCA	GWO
x_1	**0.001**	0.001	0.0010002	0.001	0.001
x_2	**0.0010005**	0.0010003	0.0010079	0.0028508	0.0010782
x_3	**0.0010001**	0.0010005	0.0010205	0.0044385	0.0010242
x_4	**0.0010121**	0.0012186	0.0058931	0.0016035	0.0012434
x_5	**0.0010002**	0.0011102	0.001	0.0015727	0.0019951
x_6	**0.0010009**	0.0013208	0.0010007	0.001235	0.0010718
x_7	**1.524**	1.5245	1.5246	1.7023	1.5243
x_8	**1.524**	1.524	1.524	1.7463	1.5244
x_9	**5**	5	4.9963	4.6685	4.9977
x_{10}	**2**	3.0708	2.0001	2.3346	2.0099
x_{11}	**0.0010001**	0.029739	0.0010077	0.0037313	0.0067319
x_{12}	**0.001**	0.029569	0.0010007	0.002211	0.0066506
x_{13}	**0.0072837**	0.043223	0.0050226	0.0046903	0.017752
x_{14}	**0.087436**	0.51759	0.060171	0.052574	0.21302
optimal value	**0.032247**	0.060632	0.042151	0.12256	0.03917

As can be seen from the optimization results in Table 9, IEO algorithm still has good performance in dealing with highly complex engineering design problems. Compared with the original EO algorithm, the optimal value obtained by IEO algorithm is improved by several orders of magnitude. In general, for engineering problems with fourteen objective variables and constraints, IEO algorithm can also get the optimal value.

In this section, six engineering optimization problems of pressure vessel, welded beam, tension/compression spring, three-bar truss, speed reducer and optimal design of industrial refrigeration system with constraint state are solved by IEO algorithm, and the design scheme given by IEO is compared with the scheme proposed by the algorithms in the existing literature. The comparison results show that the design cost of IEO is much lower than the original EO algorithm and other comparison algorithms, and it is an algorithm that can effectively solve engineering optimization problems. At the same time, the good optimization results also show that IEO has better optimization efficiency and performance in practical application.

7 CONCLUSION

In this paper, a multi-strategy improved Equilibrium Optimizer (IEO) is proposed to solve numerical optimization and engineering problems. Tent mapping is used to initialize the population and produce the initial solution with rich diversity, which lays a good foundation for the global search of the search population in space. At the same time, a nonlinear time parameter strategy is introduced into the update equation of the algorithm, which dynamically coordinates the exploration and exploitation phase of IEO algorithm. Lens Opposition- based Learning (LOBL) strategy is adopted in the late iteration of the algorithm to improve the diversity of the population and avoid the algorithm falling into local optimal. Simulation experiments are carried out by using 23 classical functions, IEEE CEC2017 and IEEE CEC2019. The experimental results show that compared with the other six meta-heuristic algorithms, the improved IEO algorithm has obvious improvement in solving accuracy and convergence speed. In addition, the stability and effectiveness of IEO are proved from different perspectives by Friedman statistical test and boxplot analysis. Finally, IEO is applied to six engineering design problems: pressure vessel problem, welded beam problem, tension /compression spring problem, three-bar truss problem, speed reducer problem and optimal design of industrial refrigeration system. The research results show that the improved IEO algorithm has good optimization efficiency in solving practical application problems. In the future, we will further study how IEO can be applied to multi-objective problems and more complex practical engineering problems.

DATA AVAILABILITY

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

REFERENCES

[1] Fan, Q., Huang, H., Yang, K., Zhang, S., Yao, L., & Xiong, Q. (2021). A modified equilibrium optimizer using opposition-based learning and novel update rules. *Expert Systems with Applications*, *170*, 114575.

[2] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, *97*, 849-872.

[3] Metawa, N., Hassan, M. K., & Elhoseny, M. (2017). Genetic algorithm based model for optimizing bank lending decisions. *Expert Systems with Applications*, *80*, 75-82.

[4] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341-359.

[5] Yang, S., & Yao, X. (2005). Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft computing*, *9*(11), 815-834.

[6] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, *12*(6), 702-713.

[7] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

[8] Erol, O. K., & Eksin, I. (2006). A new optimization method: big bang–big crunch. *Advances in Engineering Software*, *37*(2), 106-111.

[9] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*,*179*(13), 2232-2248

[10] Formato, R. A. (2007). Central force optimization. *Prog Electromagn Res*, *77*(1), 425-491.

[11] Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*, *1*(3), 190-206.

[12] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences*, *183*(1), 1-15.

[13] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, *76*(2), 60-68.

[14] Kennedy, J., & Eberhart, R. C. (1997, October). A discrete binary version of the particle swarm algorithm. In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, 5, 4104-4108. IEEE.

[15] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *26*(1), 29-41.

[16] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, *39*(3), 459-471.
[17] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67.
[18] Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, *191*, 105190.
[19] Gupta, S., Deep, K., & Mirjalili, S. (2020). An efficient equilibrium optimizer with mutation strategy for numerical optimization. *Applied Soft Computing*, *96*, 106542.
[20] Abdel-Basset, M., Mohamed, R., Mirjalili, S., Chakrabortty, R. K., & Ryan, M. J. (2021). MOEO-EED: A multi-objective equilibrium optimizer with exploration–exploitation dominance strategy. *Knowledge-Based Systems*, *214(6), 106717.*
[21] Jia, H., & Peng, X. (2021). High equilibrium optimizer for global optimization. *Journal of Intelligent & Fuzzy Systems*, (Preprint).
[22] Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, *18*(1), 89-98.
[23] Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, *96*, 120-133.
[24] Yu, F., Li, Y. X., Wei, B., Xu, X., & ZHAO, Z. Y. (2014). The application of a novel OBL based on lens imaging principle in PSO. *ACTA ELECTONICA SINICA*, *42*(2), 230.
[25] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163-191.
[26] Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, *23*(3), 715-734.
[27] Yang, X. S., & He, X. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-inspired computation*, *5*(3), 141-149.
[28] Gupta, S., & Deep, K. (2019). Improved sine cosine algorithm with crossover scheme for global optimization. *Knowledge-Based Systems*, *165*, 374-406.
[29] Chakraborty, S., Saha, A. K., Sharma, S., Mirjalili, S., & Chakraborty, R. (2021). A novel enhanced whale optimization algorithm for global optimization. *Computers & Industrial Engineering*, *153*, 107086.
[30] Van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE transactions on evolutionary computation*, *8*(3), 225-239.
[31] Fan, Q., Chen, Z., Zhang, W., & Fang, X. (2020). ESSAWOA: enhanced whale optimization algorithm integrated with salp swarm algorithm for global optimization. *Engineering with Computers*, 2020:1-18.
[32] Kamboj, V. K., Nandi, A., Bhadoria, A., & Sehgal, S. (2020). An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Applied Soft Computing*, *89*, 106018.
[33] Hameed, I. A., Bye, R. T., & Osen, O. L. (2016, December). Grey wolf optimizer (GWO) for automated offshore crane design. In *2016 IEEE symposium series on computational intelligence ((IEEE SSCI '16)* 2016:1-6.
[34] Li, Y., Zhao, Y., & Liu, J. (2021). Dimension by dimension dynamic sine cosine algorithm for global optimization problems. *Applied Soft Computing*, *98*, 106933.
[35] Mezura-Montes, E., & Coello, C. A. C. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems*, *37*(4), 443-473.
[36] Baykasoğlu, A., & Ozsoydan, F. B. (2015). Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Applied soft computing*, *36*, 152-164.
[37] Lu, Y., Zhou, Y., & Wu, X. (2017). A hybrid lightning search algorithm-simplex method for global optimization. *Discrete Dynamics in Nature and Society*, *2017*, 8342694.
[38] Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, *41*(2), 113-127.
[39] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61.
[40] Ragsdell, K. M., & Phillips, D. T. (1976). Optimal design of a class of welded structures using geometric programming. *Journal of Engineering for Industry, 98*(3), 97-97.
[41] Huang, F. Z., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and computation*, *186*(1), 340-356.
[42] Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computations*, *27*(1-2), 155-182.
[43] Kaveh, A.,&Khayatazad,M.(2012). A new meta-heuristic method: ray optimization. *Computers & structures*, *112*, 283-294.
[44] Muthusamy, H., Ravindran, S., Yaacob, S., & Polat, K. (2021). An improved elephant herding optimization using sine–cosine mechanism and opposition based learning for global optimization problems. *Expert Systems with Applications*, *172*, 114607.
[45] Guo, W., Li, W., Zhang, Q., Wang, L., Wu, Q., & Ren, H. (2014). Biogeography-based particle swarm optimization with fuzzy elitism and its applications to constrained engineering problems. *Engineering Optimization*, *46*(11), 1465-1484.
[46] He, S., Prempain, E., & Wu, Q. H. (2004). An improved particle swarm optimizer for mechanical design optimization problems. *Engineering optimization*, *36*(5), 585-605.
[47] Babalik, A., Cinar, A. C., & Kiran, M. S. (2018). A modification of tree-seed algorithm using Deb's rules for constrained optimization. *Applied Soft Computing*, *63*, 289-305.
[48] Liu, J. L. (2005). Novel orthogonal simulated annealing with fractional factorial analysis to solve global optimization problems. *Engineering Optimization*, *37*(5), 499-519.
[49] Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, *13*(5), 2592-2612.
[50] Hsieh, T. J. (2014). A bacterial gene recombination algorithm for solving constrained optimization problems. *Applied Mathematics and Computation*, *231*, 187-204.
[51] Zahara, E., & Kao, Y. T. (2009). Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Systems with Applications*, *36*(2), 3880-3886.
[52] Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, *188*(2), 1567-1579.
[53] Chen, H., Xu, Y., Wang, M., & Zhao, X. (2019). A balanced whale optimization algorithm for constrained engineering design problems. *Applied Mathematical Modelling*, *71*, 45-59.
[54] Arora, J. (2004). *Introduction to optimum design*. Elsevier.
[55] Ray, T., & Saini, P. (2001). Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization*, *33*(6), 735-748.
[56] Gupta, S., & Deep, K. (2019). A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Systems with Applications*, *119*, 210-230.
[57] Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, *29*(1), 17-35.
[58] Tsai, J. F. (2005). Global optimization of nonlinear fractional programming problems in engineering design. *Engineering Optimization*, *37*(4), 399-409.
[59] Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of intelligent manufacturing*, *23*(4), 1001-1014.