Improved Goose Algorithm Based on Vertical and Horizontal Crossover Strategy and Random Walk to Solve Engineering Optimization Problems

Yu-Liang Qi, Cheng Xing*, Jie-Sheng Wang, Yu-Wei Song, Xin-Yi Guan

Abstract—**The Goose Optimization Algorithm (Goose Optimization Algorithm) is an optimization algorithm based on swarm intelligence, inspired by the behavioral patterns of geese in their natural habitats, this approach aims to enhance the convergence speed and precision of the initial algorithm. It also addresses the issue of the algorithm's tendency to become trapped in local optimal, a goose optimization algorithm based on crossbar strategy and random walk improvement is proposed. Three improvement strategies were proposed in this paper, which introduced the random walk strategy, Lévy flight walk strategy, and crossbar strategy into the development stage, exploration stage, and the later stage of each population iteration of the goose optimization algorithm. These three strategies can enhance the development ability of the algorithm, help the algorithm to escape from the local optimal when it falls into the local optimal, boost the algorithm's capability for global exploration, avoid premature convergence, and enhance the precision of the algorithm's solutions and expedite its convergence rate. The goose optimization algorithm based on crossbar strategy and random walk improvement is abbreviated as CRw-GOOSE. In order to confirm the efficacy and excellence of CRw-GOOSE, 12 benchmark functions in CEC-BC-2022 are adopted. First, simulation experiments are conducted on GOOSE and CRw-GOOSE with three strategies introduced separately. The outcomes of the experiments indicate that these improvement strategies are very effective. Among them, CRw-GOOSE, which combines the three strategies, has the best effect. Then, CRw-GOOSE was compared with seven advanced intelligent optimization algorithms, and the findings from the experiments also demonstrated the superiority and advanced nature of CRw-GOOSE. In conclusion, optimization is carried out on four engineering design problems. The simulation outcomes indicate that the CRw-GOOSE approach is capable of effectively addressing both function optimization and engineering optimization problems.**

Index Terms—*Goose Optimization Algorithm, Crossbar Strategy, Random Walk Strategy, Lévy Flight Migration Strategy, Engineering Optimization*

Manuscript received January 20, 2025; revised March 22, 2025. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJ222410146054) and Postgraduate Education Reform Project of Liaoning Province (Grant No. LNYJG2022137).

Yu-Liang Qi is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: qyl@stu.ustl.edu.cn).

Cheng Xing is a Ph. D candidate in School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114044, P. R. China (Corresponding author, phone: 86-0412-2538355; fax: 86-0412-2538244; e-mail: xingcheng0811@163.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: wjs@ustl.edu.cn).

Yu-Wei Song is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: syw@stu.ustl.edu.cn).

Xin-Yi Guan is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: gxy@stu.ustl.edu.cn).

I. INTRODUCTION

Optimization problem is significant in numerous areas, such as mathematics, engineering, economics, and computer science. It involves finding the optimal value of an objective function under given constraints [1]. The objective function can be cost, profit, time, resource consumption, etc., while the constraints may include resource limitations, technical requirements, laws, and regulations. Optimization problems can be divided into linear optimization, nonlinear optimization, integer optimization, combination optimization, and other types, each type has its unique characteristics and application scenarios. For example, the goal function and limitations of linear optimization problems are linear, which is suitable for problems such as resource allocation [2] and production planning [3]. Nonlinear optimization problems involve nonlinear functions and are often used in engineering design [4], economic modeling [5], and other fields. Integer optimization requires the decision variable to be integer, which is applicable to project selection [6], network design [7], etc.

Combinatorial optimization focuses on selecting the optimal combination from a finite set, such as the traveling salesman problem [8] and the backpack problem [9]. Optimization algorithms are the key tools to solve optimization problems. They use different strategies and methods to find optimal solutions or approximate optimal solutions. The classical optimization algorithms include gradient descent technique, Newton's approach, quasi-Newton technique, etc. These algorithms are mainly suitable for continuous and differentiable optimization problems.

Heuristic algorithm is an important branch of optimization algorithm, whose purpose is to find a feasible solution or an approximate optimal solution to a problem within a limited time [10]. Heuristic algorithms usually explore the solution space through heuristic search strategies based on some characteristics or empirical rules of the problem. The advantage of this kind of algorithm is that it has high computational efficiency and can quickly get a relatively good solution, which is especially suitable for solving large-scale, complex, or difficult to accurately solve optimization problems. A notable feature of the heuristic algorithm is its high efficiency. Heuristic algorithms usually have low computational complexity and can be solved in a short time [11].

In addition, the heuristic algorithm boasts a broad scope of applicability and flexibility. It does not rely on a strict mathematical model of the problem, thus it is applicable to a diverse array of issues, and the heuristic algorithm can be adjusted and improved according to the specific characteristics of the problem to adapt to different application scenarios and needs. Many heuristic algorithms also have strong global search ability and are able to find high-quality solutions in complex solution Spaces, not just locally optimal solutions [12]. For example, by emulating natural selection and genetic processes in biological evolution, genetic algorithms can explore solution space in the global scope and prevent getting trapped in local optimal solutions [13]. Particle swarm optimization (PSO) improves search efficiency and global search ability by mimicking the actions of biological communities and utilizing group collaboration and information sharing [14]. By simulating the positive feedback mechanism of pheromone in the ant foraging process, the ant colony optimization algorithm can find a shorter path in the complex network structure [15]. In addition, over the past few years, numerous researchers have devised a multitude of outstanding heuristic algorithms by imitating some behaviors in nature. For example, Puma Optimizer (PO)[16], Secretary bird optimization algorithm (SBOA)[17], Black eagle optimizer (BEO)[18] and Crested Porcupine Optimizer (CPO)[19].

By introducing horizontal and vertical operations in the exploration process, the strategy enhances the comprehensive search ability and local search ability of the algorithm. Specifically, the horizontal crossover operation resembles the crossover process found in genetic algorithms, which is to cross between the same dimensions of different individuals in the population, which can enhance the variety within the population and prevent the algorithm from falling into the local optimal solution prematurely. Vertical cross operation is to cross different dimensions of the same individual, which helps the individual to escape from the local optimal trap, so as to improve the local search ability of the algorithm to some extent.

For example, Liang et al. introduced lateral and vertical crossover procedures of individuals after the global leader stage of the spider monkey optimization algorithm, aiming to boost the variety of the population, thereby improving the algorithm's comprehensive search capability and its capacity to break free from local optimal. The researchers carried out numerical experiments on 23 test questions, and compared the numerical results from many aspects. The outcomes of the experiments demonstrate that the Spider monkey optimization algorithm with crisscross optimization (CSMO) has improved the precision of solutions and rate of convergence in comparison with the original SMO algorithm [20]. Zhao et al. proposed an improved Bald Eagle Search (BES) algorithm. Gold-SA and Crisscross Bald Eagle Search (GSCBES) is a combination of Gold-SA and Crisscross Bald Eagle Search. The proposed approach addresses the limitations of traditional condor search algorithms, including their susceptibility to becoming trapped in local optima and exhibiting slow convergence rates. Simulation experiments were conducted on 11 benchmark functions as well as CEC2014 functions, with the Wilcoxon rank sum test employed to evaluate the optimization capabilities of the proposed algorithm. The results indicate that the new algorithm demonstrates a faster convergence speed and enhanced optimization performance [21].

Random walk is a simple and effective search strategy, which is commonly employed in optimization algorithms to enhance the ability of global search and local optimization. Typical random walk strategies encompass the Gaussian random walk, Lévy flying random walk, and triangle walk. Among them, the Gaussian random walk is a classical random walk model, which updates the positions of individuals by introducing random perturbations of Gaussian distributions. This strategy can enhance the development ability of the algorithm and assist it in escaping the local optimal when it falls into the local optimal. For example, in the Harris Eagle optimization algorithm, Gaussian random walks are used to perturb the optimal individuals of a population to generate new individuals, thereby accelerating the algorithm's convergence rate. Lévy flight is a long-tailed random walk strategy with step sizes that adhere to the Lévy stable distribution. This strategy can generate a long jump in the search process, which can effectively explore the solution space and boost the algorithm's global search capability. Lévy-flying random walks are often used in particle swarm optimization algorithms to improve their ability to escape from local optimal. The triangle walking strategy is a strategy used to improve an intelligent optimization algorithm, which is designed to improve the algorithm's local optimization and global search capabilities. The basic idea of this strategy is to make the population of the algorithm move around while approaching the best position, so as to increase the randomness and diversity of the algorithm.

Many researchers tend to integrate these wandering strategies into heuristic algorithms in order to improve the performance of the algorithms. For example, Wang et al. proposed an improved algorithm, the Random Walk Gaussian Estimated Distribution Algorithm (RW-GEDA), to solve the precocious convergence problem that the basic Gaussian estimated Distribution algorithm (GEDA) is prone to when solving complex optimization problems. Statistical results show that HW-GEDA is highly competitive in both solving efficiency and accuracy [22]. Cai et al. proposed an improved Strategy, the Triangle-Flipping strategy, to solve the shortcomings of the Bat Algorithm (BA) in global search capability. In this paper, we introduce three distinct triangle-flipping strategies. These design methodologies enable bats to explore the solution space more effectively during the search process by employing various flipping techniques and parameter settings. Consequently, this enhances both the global search capability and the ability to escape from local optima. The effectiveness of the enhanced bat algorithm is evaluated using the CEC2013 benchmark function and compared with that of the standard bat algorithm. The experimental findings indicate that the hybrid triangle-flipping strategy significantly improves the performance of the bat algorithm, making it more efficient and precise in addressing complex optimization problems [23].

The Goose Optimizer is a swarm intelligence-based optimization algorithm, which is inspired by the behavior patterns of geese in their natural environment. This algorithm simulates the cooperative and competitive behavior of geese during foraging, migration and escaping from natural enemies to achieve efficient solution of complex optimization problems [24].

This paper introduces an enhanced goose algorithm incorporating crossbar strategy and random walk. First, the random walk strategy is embedded into two equations in the development phase to discover new goose locations. Secondly, the Lévy flight strategy is introduced into the goose algorithm in the exploration stage to increase the randomness of selecting screaming geese and boost the algorithm's capability for global exploration. Then, after a new goose population is generated at the end of each iteration, a lateral and vertical crossover strategy is used to cross a certain dimension of the two geese in the population, thereby fostering the diversity of the population and preventing the algorithm from prematurely converging to a locally optimal solution. The goose algorithm based on crossbar strategy and random walk improvement is called CRw-GOOSE for short. To confirm the effectiveness and superiority of the improved goose algorithm, 12 reference functions in CEC-BC-2022 are used to verify the effectiveness of the improved goose algorithm by first comparing it with the original goose algorithm, and then comparing it with 8 advanced intelligent heuristic optimization algorithms in recent years to further verify the superiority of the improved goose algorithm. Ultimately, four engineering design problems were optimized, and the enhanced algorithm was capable of effectively addressing these engineering design problems.

II. THE BASIC PRINCIPLE OF GOOSE OPTIMIZATION ALGORITHM

A. Algorithm Initialization Process

The Goose Optimization Algorithm (GOOSE) is an innovative meta-heuristic algorithm inspired by the collective behavior of geese. It is primarily employed to address complex optimization challenges. This algorithm facilitates the exploration and exploitation of the solution space by mimicking the behaviors exhibited by geese during their resting and foraging activities.

Firstly, a specific quantity of geese individuals are randomly generated in the solution space, and each individual represents a candidate solution. The positions of these individuals are randomly distributed at initialization to cover the entire solution space. Since the location of the geese is random, the population is initialized based on the problem's upper and lower bounds, as indicated in Eq. (1).

$$X_{it}(j) = lb_j + rand \times (ub_j - lb_j), i = 1,2,\ldots,\dim \tag{1}$$

where, $X_{it}(i)$ is the j -dimensional position of the i -th goose individual, N is the size of the goose population, $\mathbf{dim}$ is the dimension of the decision variable, $rand$ is the random real number in the interval [0, 1], and ub_j and lb_j are the maximum and minimum limits of the j -dimensional decision variable.

B. Development Phase: Protect and Wake Up the Geese in the Team

During the algorithm's iterative process, a random variable $rand$ is in charge of the distribution between the exploitation and exploration phases. When $rand \geq 0.5$, the algorithm performs the development phase. Geese have a habit of gathering in large groups during rest periods, with one goose standing on one leg. Occasionally, the goose would lift one leg and hold a small stone, so that when he fell asleep and the stone fell again, the goose would wake up. When other geese in the group notice any unexpected noise or activity, the geese will emit a loud call to alert their companions to be safe. In the development phase, the speed of stone fall is first calculated, as shown in Eq. (2).

$$F\_F\_S = T\_o\_A\_O_{it} * \frac{\sqrt[2]{S\_W_{it}}}{9.81} \tag{2}$$

where, F_F_S is calculating the speed at which the stone falls. $T\_o\_A\_O_{it}$ is the time it takes for the rock to fall to the ground, where, $T\_o\_A\_O_{it}$ is a random number vector. S_W_{it} is the mass of the stone stored in the goose's feet, estimated to range from 5 to 25 grams. 9.81 is the acceleration of gravity at the surface of the Earth. Secondly, the distance of sound propagation is calculated, as shown in Eq. (3).

$$D\_S\_T_{it} = S\_S * T\_o\_A\_S_{it} \tag{3}$$

where, $D\_S\_T_{it}$ calculates the distance of sound propagation after the stone falls to the ground. S_S is the speed of sound in the air, here $S\_S = 343.2$. $T\_o\_A\_S_{it}$ is the time it takes for a rock to fall to the ground to make a sound and transmit it to an individual goose within the flock, where $T\_o\_A\_S_{it}$ is a random number vector. 9.81 is the acceleration of gravity at the surface of the Earth. Then you need to find the best individual goose in the population, the sound will travel to the best individual, and when the goose hears the sound, it becomes the guardian goose. The position update equation of the best individual is shown in Eq. (4).

$$X_{(it+1)} = F\_F\_S + D\_G_{it} * T\_A^2 \tag{4}$$

where, F_F_S is calculating the speed at which the stone falls. D_G_{it} is the gap between the guardian goose and another goose that is resting or foraging. Then, take half of the distance $D\_S\_T_{it}$ that the sound travels, because the sound travels back and forth time, and when the sound travels to a certain goose, there is no need to return time. T_A is the average of the total time it takes to spread and reach individual geese within the flock.

In this process, if the weight of the stone S_W is continuously less than or equal to 12, and the variable $pro \leq 0.2$ is used, another strategy will be used to find the best location of the geese in the population. At this time, the speed at which the stone falls and the location of the optimal individual are updated in the formula shown in Eq. (5)-(6).

$$F\_F\_S = T\_o\_A\_O_{it} * \frac{S\_W_{it}}{9.81} \quad (5)$$

$$X_{(it+1)} = F\_F\_S + D\_G_{it} * T\_A^2 * Coe \quad (6)$$

where, Coe is a random number less than or equal to 0.17.

C. Exploration Phase: Scream to Protect All Individuals in the Group

When $rand < 0.5$, the algorithm performs the exploration phase. In goose behavior, if one of the geese becomes alert, it begins to scream to safeguard all the members of the flock. During the exploration phase, a variable $alpha$ is set that decreases significantly with each iteration in the cycle, ranging from 2 to 0, as shown in Eq. (7).

$$alpha = \left(2 - \left(\frac{\frac{loop}{Max\_It}}{2}\right)\right) \quad (7)$$

The position update equation is presented in Eq. (8).

$$X_{(it+1)} = Best\_pos + randn(1, \dim) * (M\_T * alpha) \quad (8)$$

where, $Best\_pos$ is the best position in the current population and M_T is always less than or equal to half of the average T_A of the total time.

III. Goose Optimization Algorithm Based on Crossbar Strategy and Random Walk Improvement

A. Random Walk Strategy

Because the goose optimization algorithm is prone to fall into the deficiency of local optimum during the process of finding the guardian goose in the development stage, random walk updating is further used to find the best guardian goose position updating formula. The step length and direction of the random walk strategy based on trigonometric function are determined randomly, which ensures that each step is unpredictable and reflects the randomness of the random walk, thus enhancing the development ability of the algorithm and helping the algorithm to escape from the local optimal when it is trapped in the local optimal. The formula of the random walk strategy is shown in Eq. (9)-(10).

$$theta = 2 * \pi * rand(), lengh = rand() \quad (9)$$

$$Random\_walk = length * \cos(theta) \quad (10)$$

Here Eq. (10) uses $theta$ to generate a random Angle in the range $[0, 2\pi)$, which represents a random direction. The $rand()$ function is used to generate a random step in the range [0, 1), which determines the distance traveled in the selected direction.

In the development stage, when looking for the best position of the guardian goose, the duration needed for the stone to descend to the ground $T\_o\_A\_O_{it}$, the mass of the stone kept in the goose's foot S_W_{it}, and the time for the sound emitted by the stone to be transmitted to a single goose in the flock $T\_o\_A\_S_{it}$ all provide a certain randomness for the location search of the best individual. However, due to the fixity of the position update formula in calculation, The algorithm is prone to getting trapped in a local optimal state. Therefore, the random walk strategy is introduced into two position update formulas in the development stage. The small perturbation provided by the random walk strategy can help the algorithm to conduct fine search around the optimal individual position, so as to find new solution space positions. The position update formula after introducing random walks is shown in Eq. (11)-(12).

$$X_{(it+1)} = F\_F\_S + D\_G_{it} * T\_A^2 + Random\_walk \quad (11)$$

$$X_{(it+1)} = F\_F\_S + D\_G_{it} * T\_A^2 * Coe + Random\_walk \quad (12)$$

The results show that the convergence curve of the goose optimization algorithm with random walk strategy is much better than that of the original goose optimization algorithm.

B. Lévy Flight Migration Strategy

The goose optimization algorithm is prone to fall into the deficiency of local optimal in the process of searching for screeching geese in the exploration stage, and the random walk update is further used to find the best location update formula of guardian geese. Lévy Flight is a particular kind of random walk model, also known as Lévy Walk or Lévy walk, used to describe patterns of movement with a long-tail distribution. In a Lévy flight, an individual or particle moves randomly through space, its step size and direction determined by the Lévy distribution. The Lévy distribution is a kind of heavy tail probability distribution whose probability density function satisfies the power law relation. Lévy Flight can boost the algorithm's global search capability and prevent early convergence. Its formula is shown in Eq. (13)-(14).

$$Levy(\beta) \sim \frac{u}{|v|^{-\beta}} \quad (13)$$

$$u \sim N(o, \sigma_u^2), v \sim N(o, \sigma_u^2),$$
$$\sigma_u = \left[\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta \times 2^{\frac{\beta-1}{2}}}\right]^{\frac{1}{\beta}}, \sigma_v = 1 \quad (14)$$

where, β is the shape parameter of the Lévy distribution, which affects the characteristics of the step distribution. u and v are a set of random numbers sampled from a normal distribution. $|v|^{-\beta}$ is used to scale u so that the step size follows the Lévy distribution. Γ is the gamma function used to calculate the scale parameter of the normal distribution.

In the exploration phase, the optimal location of screeching geese is affected by random number vectors, which are often produced irregularly, which is not conducive to the generation of excellent solutions and may affect the convergence rate of the population. Therefore, Levi's flight wandering strategy is introduced into the position updating formula in the exploration stage, and the generation of new solutions through Levi's flight is beneficial for enhancing the diversity of solution space exploration and boosting the capability of global search. The position update formula after the introduction of the Lévy flight is shown in Eq. (15).

$$X_{(it+1)} = Best\_pos + Levy(\dim) * (M\_T * alphal) \quad (15)$$

The test results indicate that the convergence curve of the goose optimization algorithm, enhanced with a Lévy-flying wandering strategy, significantly outperforms that of the original goose optimization algorithm.

C. Crossbar Strategy

Aiming at the problems of low convergence accuracy and premature convergence of the goose optimization algorithm, the cross-cross strategy is further used to transform the solutions in the population to help the population find new excellent solution space, in order to enhance the algorithm's convergence precision. The crossbar strategy includes horizontal cross and vertical cross, and the crossbar process is a competitive process, and the child generation will be compared with the parent generation to ensure that the update is carried out in a better direction. Horizontal and vertical crossovers are conducted in sequence, and the interaction between the two types of crossover enhances the algorithm's solution accuracy and speeds up convergence. Its formula is shown as Eq. (16)-(17).

$$X\_new(L,j) = r_1 * X(L,j) + (1-r_1) * X(M,j) + c_1 * (X(L,j) - X(M,j)) \quad (16)$$

$$X\_new(M,j) = r_2 * X(M,j) + (1-r_2) * X(L,j) + c_1 * (X(M,j) - X(L,j)) \quad (17)$$

where, M and L respectively represent the position of two geese immediately selected from the population, j represents dimension, $j = 1 \cdots \dim$. r_1 and r_2 are random numbers in the range [0, 1) respectively, and c_1 and c_2 are random numbers in the range [-1, 1) respectively. The two newly generated position vectors will update the population.

In many swarm intelligence search algorithms, premature convergence often stems from certain stagnant dimensions within the population, which can minimize search dead zones and enhance the algorithm's global exploration capability. Longitudinal crossover helps certain stagnant population dimensions break free from premature convergence, enabling the algorithm to escape local optima, and crossing operations enhance the variety within the population. The results show that the convergence curve of the goose optimization algorithm is much better than that of the original goose optimization algorithm. The flow of the improved goose optimization algorithm based on crossbar strategy and random walk is shown in Fig. 1.

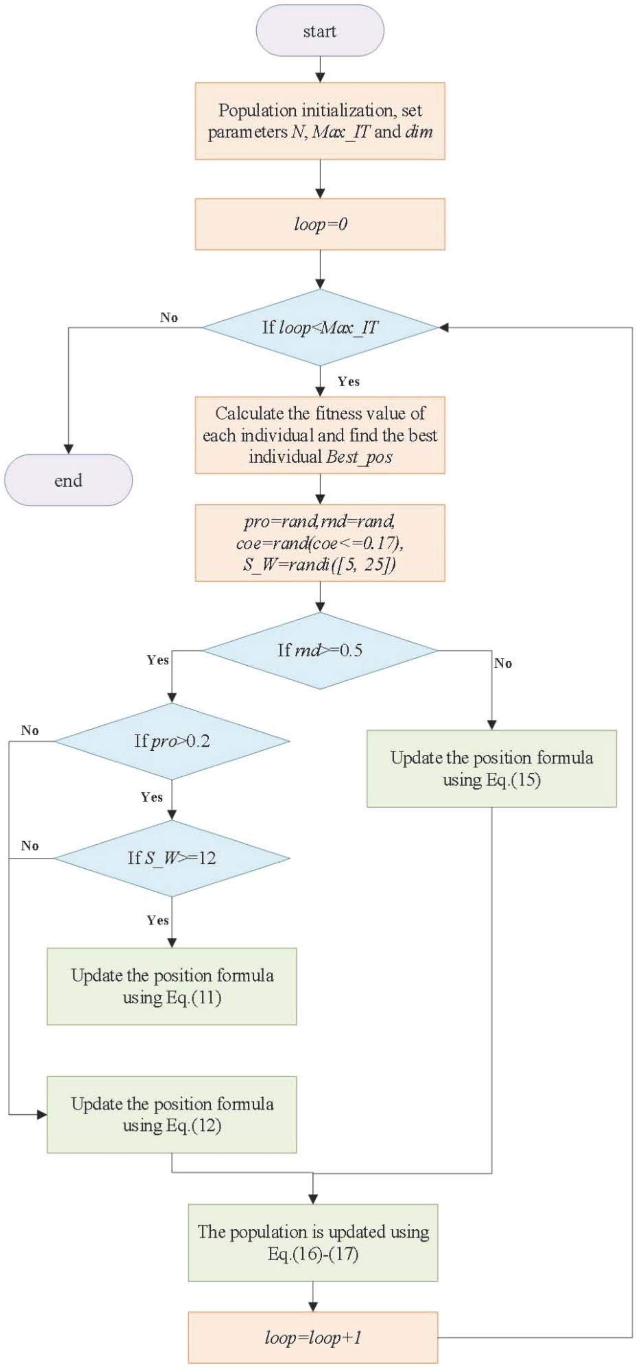


Fig. 1 Flow chart of goose optimization algorithm based on crossbar strategy and random walk improvement.

IV. Simulation Experiment and Result Analysis

A. CEC2022 Test Function

In this paper, we select 12 single-objective test functions with boundary constraints from CEC-BC-2022. All the test functions are designed to address minimization problems.

The effectiveness and superiority of the goose optimization algorithm, which is based on a crossbar strategy and enhanced by random walk improvements, are demonstrated through the optimization results obtained for these 12 test functions. To ensure the impartiality of the experimental evaluation, we set the maximum number of iterations for the goose optimization algorithm utilizing both crossbar strategy and random walk enhancements to 1000 iterations. Additionally, a population size of 30 was established, and all test functions were configured in a 10-dimensional space.

The function expressions of the 12 test functions in CEC-BC-2022 are shown in Table I. The selected functions include four categories: unimodal function f_1 ; Multimodal function $f_2 - f_5$; Mixed function $f_6 - f_8$ and combined function $f_9 - f_{12}$. These functions have different characteristics and can fully evaluate the optimization ability of the algorithm. And the CEC-BC-2022 test function has a boundary constraint, which raises the complexity of the issue and is closer to the problem in real applications.

B. Validity Verification of Goose Optimization Algorithm Based on Crossbar Strategy and Random Walk

In this section, the GOOSE optimization algorithm improved by using only random walk strategy is referred to as Rw-GOOSE, the GOOSE optimization algorithm improved by using only Lévy flying walk strategy is referred to as Lévy-GOOSE, and the GOOSE optimization algorithm improved by using only Crossbar strategy is referred to as Crossbar-GOOSE. To demonstrate the efficacy of CRw-GOOSE, we tested Rw-GOOSE, Lévy-GOOSE, and Crossbar-GOOSE respectively on 12 CEC-BC-2022 test functions. Each algorithm was executed 30 times, with the best solution from these runs being documented. Moreover, mathematical statistical analysis was conducted on the experimental outcomes to aid in comparing the impact of the three strategies on the algorithm improvement respectively, and the effect of the combined three strategies on the jointly improved CRw-GOOSE. The optimal value, average value, and variance results are summarized in Table II . The convergence curve derived from the experiment is illustrated in Fig. 2.

By analyzing the data in Table II, the following conclusions can be drawn: CRw-GOOSE optimizes functions $f_1, f_4, f_6, f_7, f_9, f_{10}, f_{12}$ to achieve the minimum optimal value and average value, optimizes functions f_2, f_{11} to achieve the minimum optimal value, optimizes function f_5 to achieve the minimum average value, and optimizes functions $f_3 - f_7, f_9, f_{11}$ to achieve the minimum variance. It can also be found that in function f_1, the performance of each algorithm is not much different, indicating that the GOOSE algorithm itself has a very good effect in solving unimodal functions. In function f_2, the mean and variance obtained by Lévy-GOOSE are the best. In function f_3, the best and average values obtained by Lévy-GOOSE are the best. In function f_5 , the optimal value obtained by Crossbar-GOOSE is the best. In functions f_6, f_9 , Rw-GOOSE and CRw-GOOSE achieve the best results among the three indexes. In function f_8, the optimal value obtained by Crossbar-GOOSE is the best. In function f_{10}, the variance obtained by Lévy-GOOSE is the smallest. In function f_{11}, the optimal values obtained by all comparison algorithms are the same. The average value obtained by Crossbar-GOOSE is the best, while the variance obtained by Rw-GOOSE is the smallest. In function f_{12}, the variance obtained by Crossbar-GOOSE is the smallest.

As illustrated in Fig. 2, the convergence curve displays the iteration count on the horizontal axis and the corresponding fitness value on the vertical axis. It can be obviously observed that CRw-GOOSE can converge to the lowest level in most of the test functions, and the effect of the other three strategies on the 12 test functions improved by GOOSE alone is far better than that of the original GOOSE. The convergence curve shown in Fig. 2 strongly proves the effectiveness and superiority of integrating three strategies to improve GOOSE.

As shown in Fig. 3, the violin diagram, where the horizontal coordinate represents different algorithms, and the vertical coordinate represents the 30 optimal fitness values counted. It can be obviously observed that CRw-GOOSE performs well in most of the test functions, whether it is the optimal fitness value or the average fitness value. Except for the function, the enhanced GOOSE strategy markedly outperforms the original GOOSE in terms of effectiveness. In addition, the difference in fitness values calculated by CRw-GOOSE each time is smaller than that of other improvement strategies, which reflects the stability of CRw-GOOSE. In summary, the goose optimization algorithm based on crosswalk strategy and random walk improvement is effective for each strategy improvement, and it can well optimize each test function in CEC-BC-2022.

C. Advantages of Goose Optimization Algorithm Based on Crossbar Strategy and Random Walk Improvement Compared with Other Intelligent Algorithms

In order to prove the superiority of the goose optimization algorithm based on crossbar strategy and random walk improvement compared with other intelligent optimization algorithms, 12 CEC-BC-2022 test functions are still simulated. The six intelligent optimization algorithms selected are: Eel and grouper optimizer (EGO) [25], Human Evolutionary Optimization Algorithm (HEOA) [26], Improved Dwarf Mongoose Optimization (IDMO) [27], Hippopotamus Optimization Algorithm (HO) [28], Newton-Raphson-based optimizer (NRBO) [29], Osprey Optimization Algorithm (OOA) [30] and Parrot Optimizer (PO) [31]. The dimensionality of each function is configured to 10 dimensions, and the maximum number of iterations for each algorithm is established at 1000 generations. Each algorithm is executed 30 times, with the optimal solution from these trials being recorded. The results, including the optimal value, average value, and variance obtained, are presented in Table III.

Additionally, the convergence curve generated from the experiment is illustrated in Fig. 4. By analyzing the data in Table III, the following conclusions can be drawn: the optimal value and average value obtained by CRw-GOOSE optimizing the eight test functions $f_1 - f_3, f_7 - f_9, f_{11}, f_{12}$ are the best, the optimal value obtained by optimizing functions f_5, f_{10} is the smallest, and the variance obtained by optimizing functions f_1, f_8, f_9, f_{12} is the smallest. It can also be found that in function f_5, the optimal value obtained by CRw-GOOSE ranks third, next to NRBO and PO. In function f_6, the best value and average value obtained by CRw-GOOSE rank second, next to HO.

As shown in Fig. 4, the convergence curve, where the abscissa denotes the iteration count, and the vertical axis indicates the fitness value. It can be obviously observed that CRw-GOOSE can converge to the lowest in most test functions, especially in the five test functions f_1, f_2, f_3, f_7, f_9 . In the other test functions, except function f_4 , CRw-GOOSE is only slightly less effective than the individual intelligent algorithms, but it can also converge to a good value.

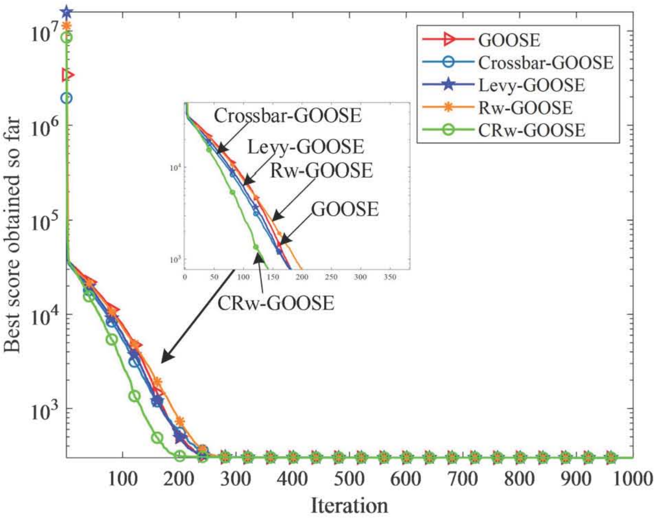
TABLE I. PROPERTIES AND SUMMARY OF THE CEC-BC-2022 TEST FUNCTIONS

No.	Expression	Name	F_i^*
Unimodal Function			
1	$f_1(x)=\sum_{i=1}^{D} x_i^2+\left(\sum_{i=1}^{D} 0.5x_i\right)^2+\left(\sum_{i=1}^{D} 0.5x_i\right)^4$	Zakharov Function	300
Basic Functions			
2	$f_2(x)=\sum_{i=1}^{D-1}\left(100\left(x_i^2-x_{i+1}\right)^2+\left(x_{i+1}-1\right)^2\right)$	Rosenbrock's Function	400
3	$Schaffer's Function: g(x,y)=0.5+\frac{\left(\sin^2\left(\sqrt{x^2+y^2}\right)-0.5\right)}{\left(1+0.001\left(x^2+y^2\right)\right)^2}$ $f_3(x)=g(x_1,x_2)+g(x_2,x_3)+\ldots+g(x_{D-1},x_D)+g(x_D,x_1)$	Expanded Schaffer's Function	600
4	$f_4(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	Rastrigin's Function	800
5	$f_5(x)=\sin^2(\pi w_1)+\sum_{i=1}^{D-1}(w_i-1)^2\left[1+10\sin^2(\pi w_i-1)\right]+(w_D-1)^2\left[1+\sin^2(2\pi w_D)\right]$ $where\ w_i=1+\frac{x_i-1}{4}, \forall i=1,\ldots,D$	Lévy Function	900
Hybrid Functions			
6	$f_6(x)=x_i^2+10^6\sum_{i=2}^{D} x_i^2$	Bent Cigar Function	1800
7	$f_7(x)=\left\|\left(\sum_{i=1}^{D} x_i^2\right)^2-\left(\sum_{i=1}^{D} x_i\right)^2\right\|^{0.5}+\left(0.5\sum_{i=1}^{D} x_i^2+\sum_{i=1}^{D} x_i\right)/D+0.5$	HGBat Function	2000
8	$f_8(x)=\sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}} x_i^2$	High Conditioned Elliptic Function	2200
Composition Functions			
9	$f_9(x)=\frac{10}{D^2}\Pi_{i=1}^{D}\left(1+i\sum_{j=1}^{32}\frac{\left\|2^j x_i-round\left(2^j x_i\right)\right\|}{2^j}\right)^{\frac{10}{D^{1.2}}}-\frac{10}{D^2}$	Katsuura Function	2300
10	$f_{10}(x)=\left\|\sum_{i=1}^{D} x_i^2-D\right\|^{\frac{1}{4}}+\left(0.5\sum_{i=1}^{D} x_i^2+\sum_{i=1}^{D} x_i\right)/D+0.5$	Happycat Function	2400
11	$f_{15}(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\Pi_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$ $f_{11}(x)=f_{15}(f_2(x_1,x_2))+f_{15}(f_2(x_2,x_3))+\ldots+f_{15}(f_2(x_{D-1},x_D))+f_{15}(f_2(x_D,x_1))$	Expanded Rosenbrock's plus Griewangk's Function	2600
12	$f_{12}(x)=418.9829\times D-\sum_{i=1}^{D} g(z_i)$ $z_i=x_i+4.209687462275036E+002$ $g(z_i)=\begin{cases} z_i\sin\left(\|z_i\|^{\frac{1}{2}}\right), if \|z_i\|\le 500 \\ (500-\mathrm{mod}(z_i,500))\sin\left(\sqrt{\|500-\mathrm{mod}(z_i,500)\|}\right)-\frac{(z_i-500)^2}{10000D}, if\ z_i>500 \\ \left(\mathrm{mod}(\|z_i\|,500)-500\right)\sin\left(\sqrt{\|\mathrm{mod}(\|z_i\|,500)-500\|}\right)-\frac{(z_i+500)^2}{10000D}, if\ z_i<-500 \end{cases}$	Modified Schwefel's Function	2700

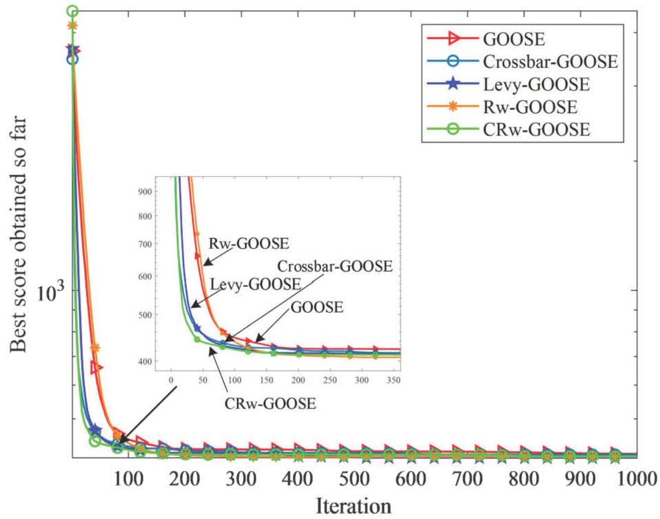
Search range: $[-100,100]^D$, D (Dimensions)=10/20.

TABLE II. PERFORMANCE COMPARISON RESULTS OF CEC-2022 FUNCTION OPTIMIZATION

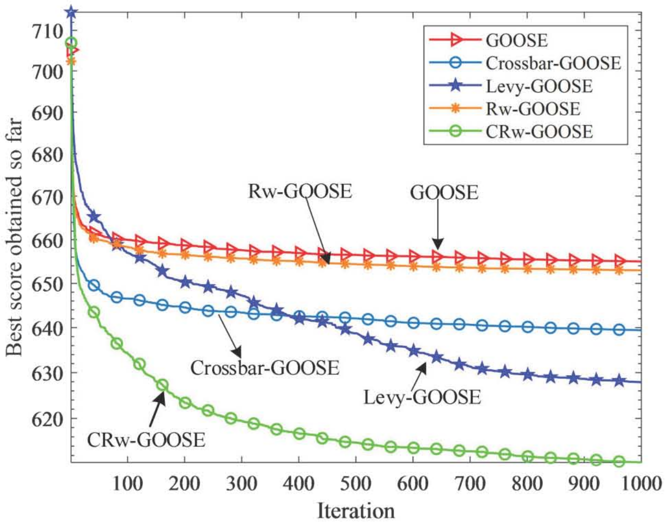
Function		GOOSE	Crossbar-GOOSE	Lévy-GOOSE	Rw-GOOSE	CRw-GOOSE
	Best	**3.0000E+02**	**3.0000E+02**	**3.0000E+02**	**3.0000E+02**	**3.0000E+02**
f_1	Ave	**3.0000E+02**	**3.0000E+02**	**3.0000E+02**	**3.0000E+02**	**3.0000E+02**
	Std	**3.7584E-04**	4.4108E-04	1.2165E-03	4.4815E-04	4.4815E-04
	Best	4.0010E+02	4.0003E+02	4.0001E+02	**4.0000E+02**	**4.0000E+02**
f_2	Ave	4.1401E+02	4.1218E+02	**4.0524E+02**	4.0702E+02	4.0702E+02
	Std	2.1988E+01	1.9852E+01	**3.8214E+00**	1.5506E+01	1.5506E+01
	Best	6.2672E+02	6.0656E+02	**6.0036E+02**	6.2529E+02	6.2329E+02
f_3	Ave	6.5500E+02	6.3942E+02	**6.2788E+02**	6.3299E+02	6.2943E+02
	Std	1.3596E+01	1.8764E+01	1.4999E+01	1.1240E+01	**5.8545E+00**
	Best	8.2288E+02	8.2188E+02	8.2388E+02	8.2587E+02	**8.2187E+02**
f_4	Ave	8.6096E+02	8.4240E+02	8.4980E+02	8.6014E+02	**8.3989E+02**
	Std	2.1784E+01	1.4192E+01	1.7711E+01	1.8688E+01	**9.3873E+00**
	Best	1.5613E+03	**1.2245E+03**	1.5715E+03	1.4912E+03	1.4912E+03
f_5	Ave	2.3657E+03	1.9613E+03	2.4463E+03	2.2379E+03	**1.9519E+03**
	Std	5.8401E+02	4.8638E+02	7.9953E+02	4.9598E+02	**2.8710E+02**
	Best	2.0172E+03	1.9152E+03	2.0637E+03	**1.8635E+03**	**1.8635E+03**
f_6	Ave	4.5515E+03	3.8594E+03	4.0493E+03	**3.6449E+03**	**3.6449E+03**
	Std	2.4321E+03	1.8872E+03	2.1019E+03	**1.7580E+03**	**1.7580E+03**
	Best	2.0596E+03	2.0390E+03	2.0529E+03	2.0529E+03	**2.0130E+03**
f_7	Ave	2.1661E+03	2.0427E+03	2.1593E+03	2.1593E+03	**2.0481E+03**
	Std	7.2397E+01	3.7766E+01	7.3899E+01	7.3899E+01	**3.4191E+01**
	Best	2.2310E+03	**2.2110E+03**	2.2216E+03	2.2310E+03	2.2310E+03
f_8	Ave	2.4184E+03	**2.2506E+03**	2.2557E+03	2.3398E+03	2.3398E+03
	Std	1.3152E+02	**5.0846E+01**	5.3093E+01	1.0452E+02	1.0452E+02
	Best	2.5293E+03	2.5293E+03	2.5293E+03	**2.5150E+03**	**2.5150E+03**
f_9	Ave	2.5293E+03	2.5293E+03	2.5293E+03	**2.5285E+03**	**2.5285E+03**
	Std	3.6846E-02	6.0061E-04	6.1915E-04	**7.8549E+00**	**7.8549E+00**
	Best	2.5004E+03	2.5004E+03	2.4218E+03	2.4915E+03	**2.4101E+03**
f_{10}	Ave	3.1248E+03	2.7433E+03	2.6344E+03	2.7686E+03	**2.6011E+03**
	Std	6.8944E+02	2.8936E+02	**1.3899E+02**	3.5531E+02	2.3282E+02
	Best	**2.6001E+03**	**2.6001E+03**	**2.6001E+03**	**2.6001E+03**	**2.6001E+03**
f_{11}	Ave	2.8070E+03	**2.7303E+03**	2.7603E+03	2.7711E+03	2.7711E+03
	Std	1.6352E+02	1.2503E+02	1.4382E+02	**1.2046E+02**	**1.2046E+02**
	Best	2.8712E+03	2.8718E+03	2.8646E+03	2.8781E+03	**2.8522E+03**
f_{12}	Ave	3.0104E+03	2.8963E+03	2.9002E+03	2.9871E+03	**2.8952E+03**
	Std	1.0588E+02	**1.8417E+01**	3.0355E+01	7.1466E+01	3.1292E+01



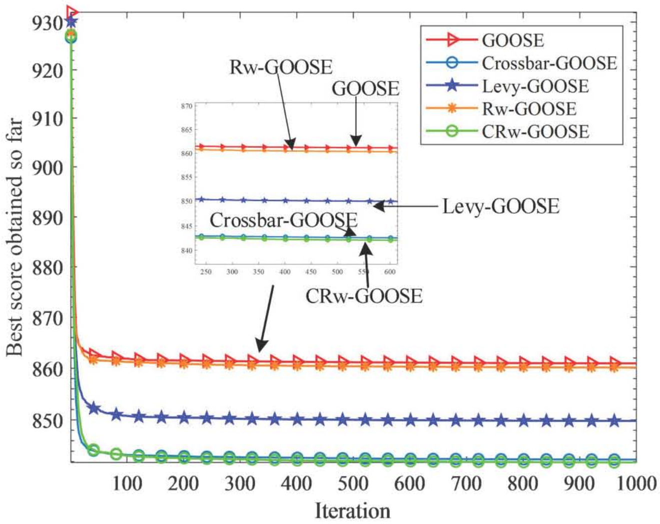
(1) f_1



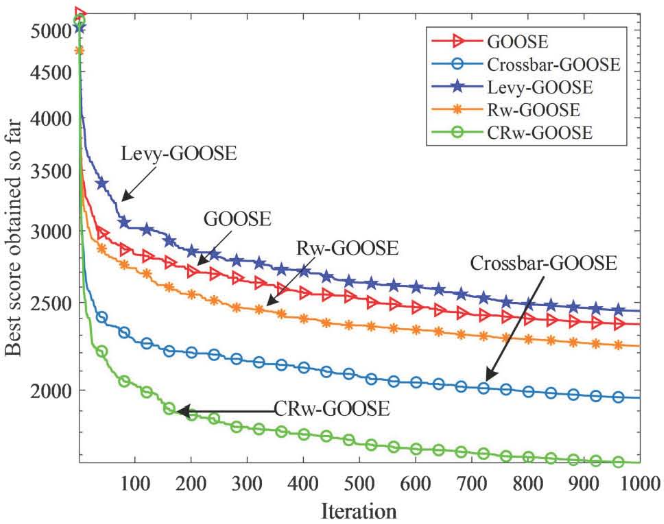
(2) f_2



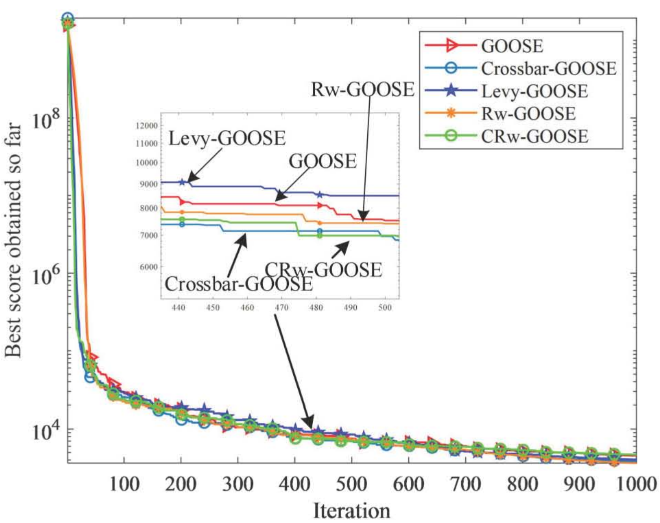
(3) f_3



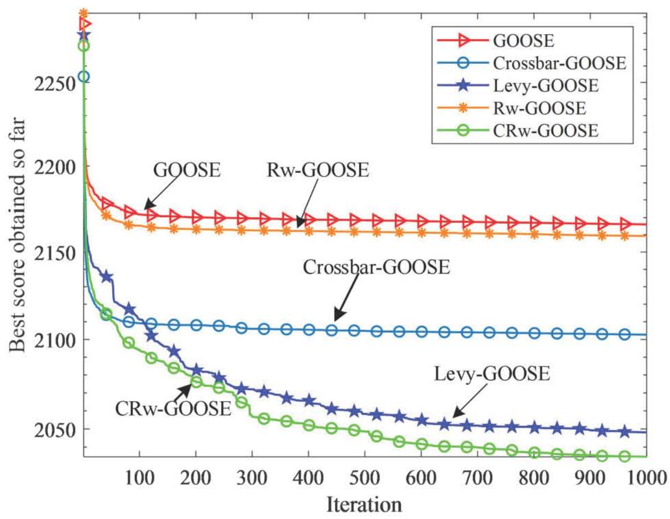
(4) f_4



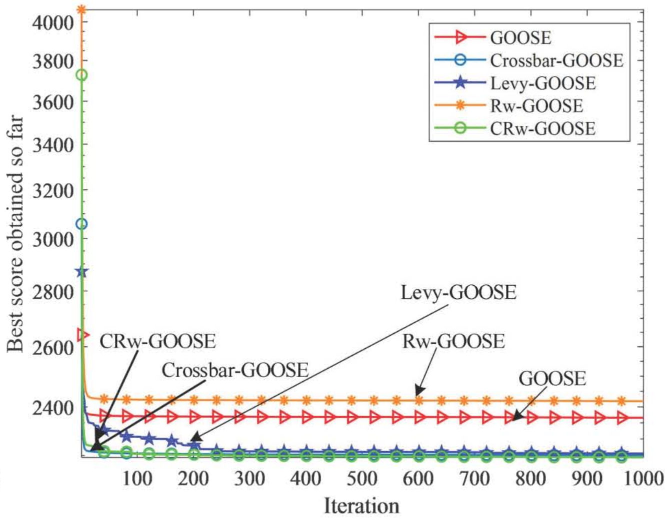
(5) f_5



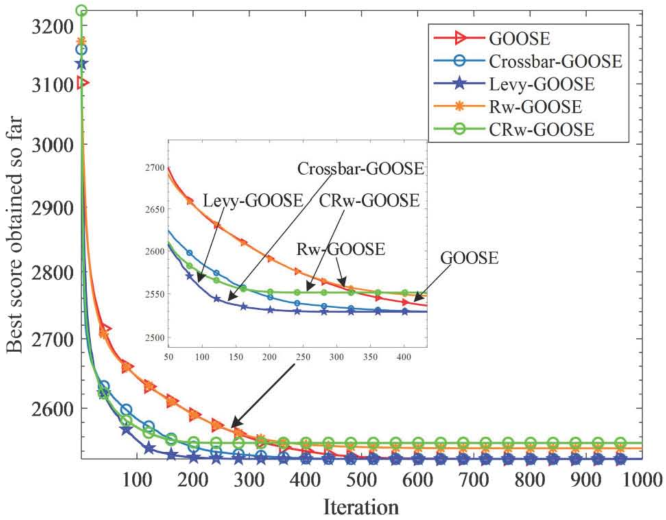
(6) f_6



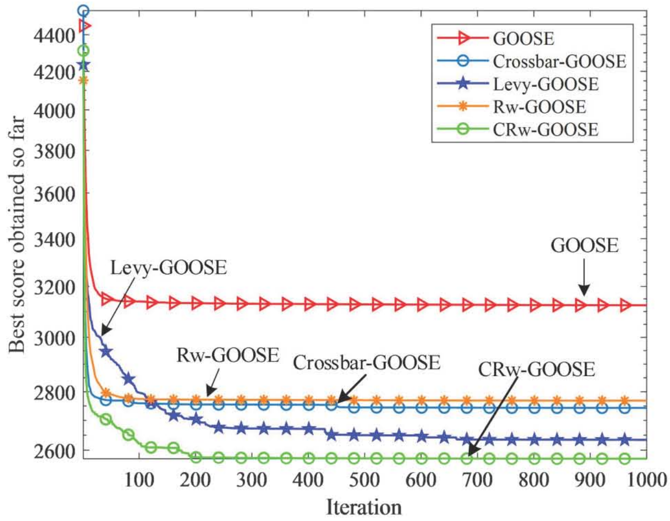
(7) f_7



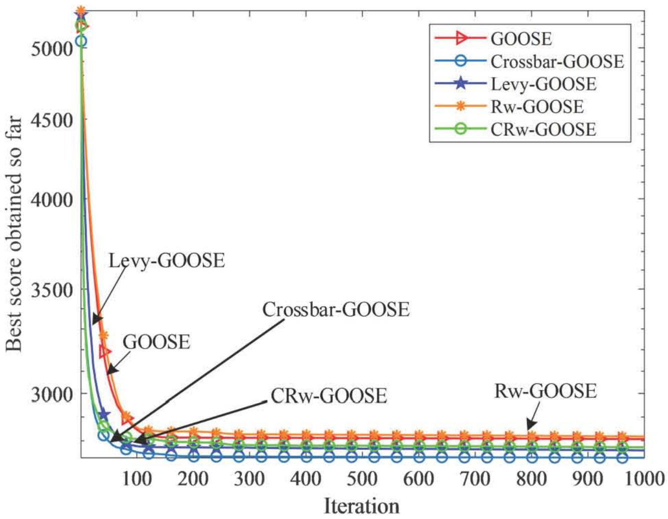
(8) f_8



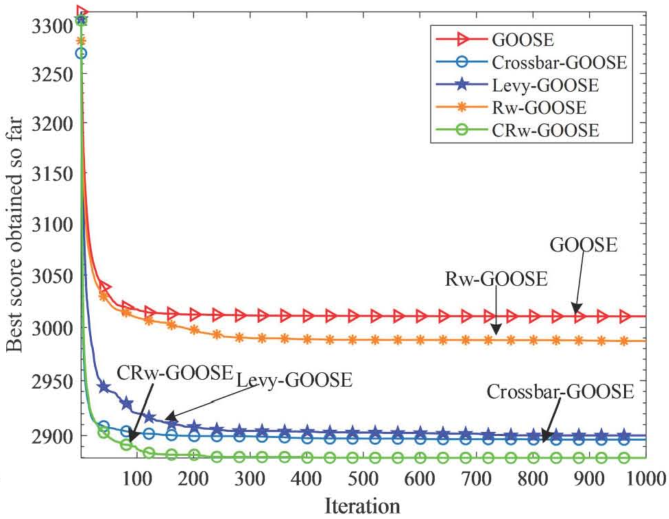
(9) f_9



(10) f_{10}

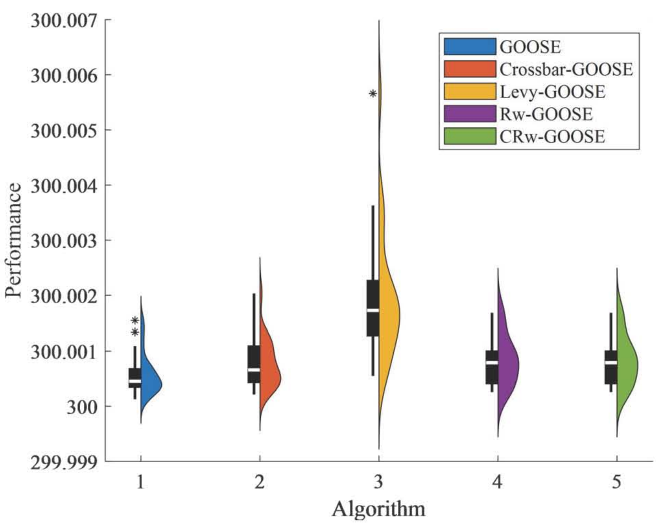


(11) f_{11}

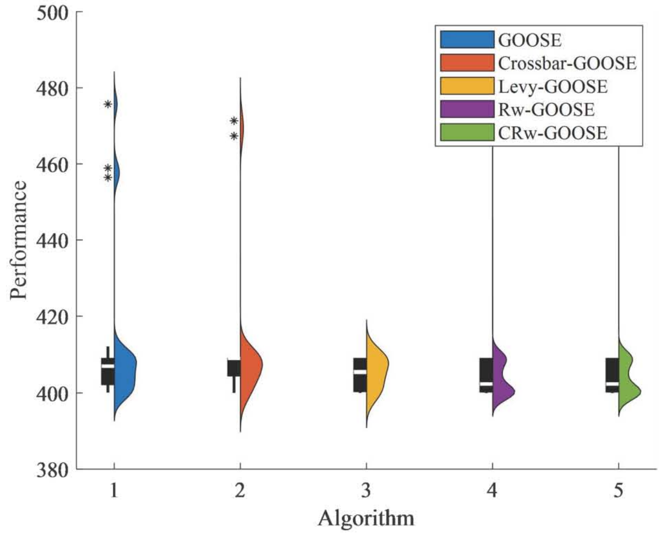


(12) f_{12}

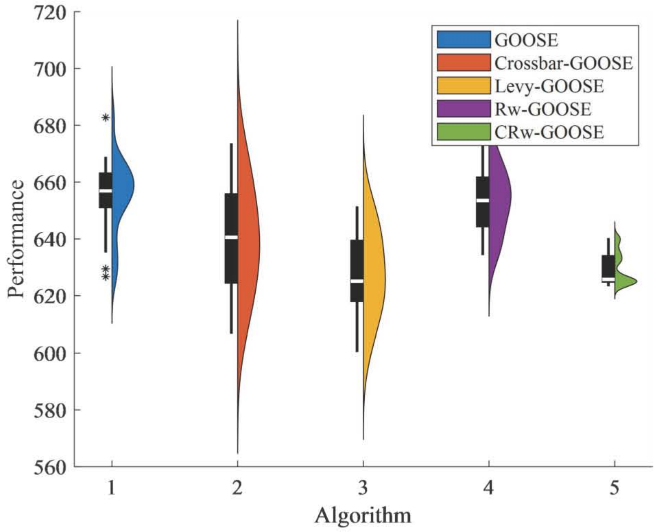
Fig. 2 The convergence graph of CEC-BC-2022 test function is solved based on crossbar strategy and random walk improved goose optimization algorithm.



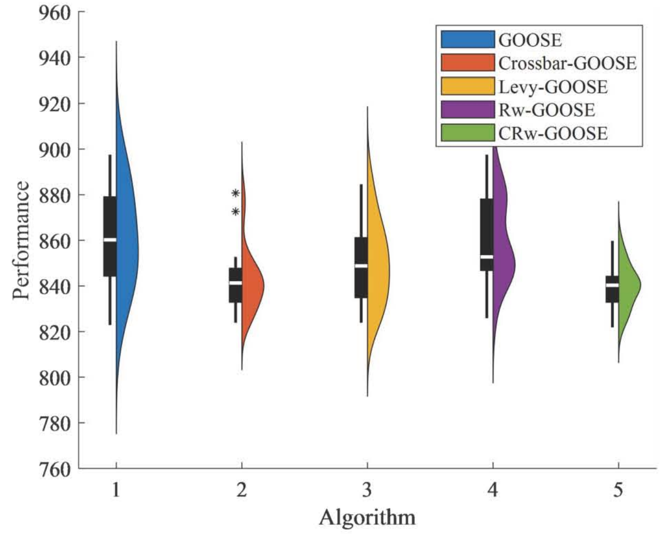
(1) f_1



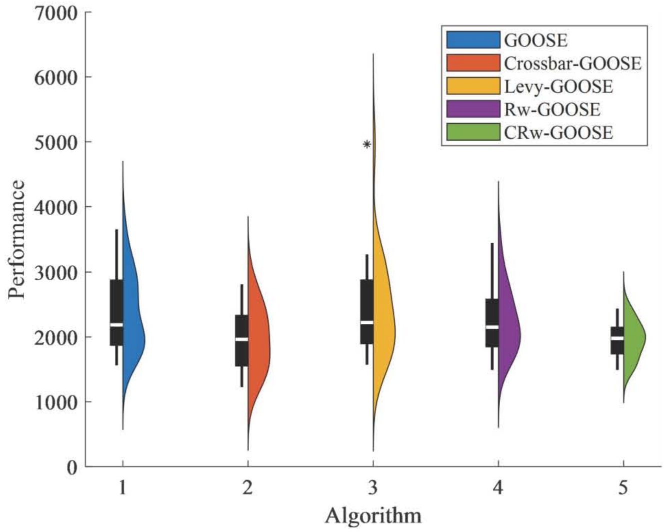
(2) f_2



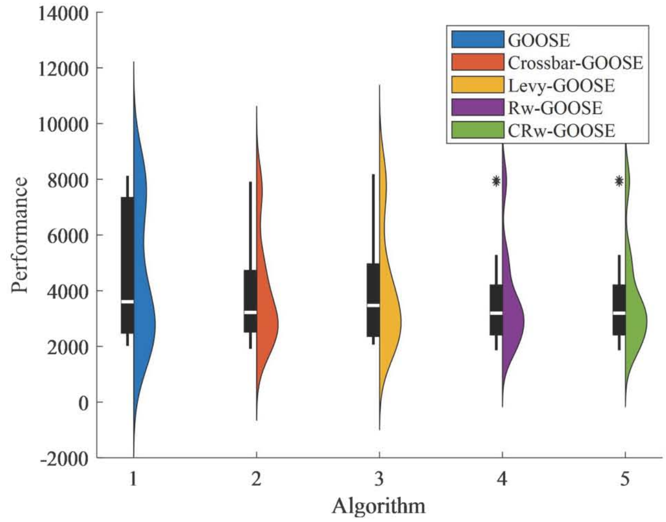
(3) f_3



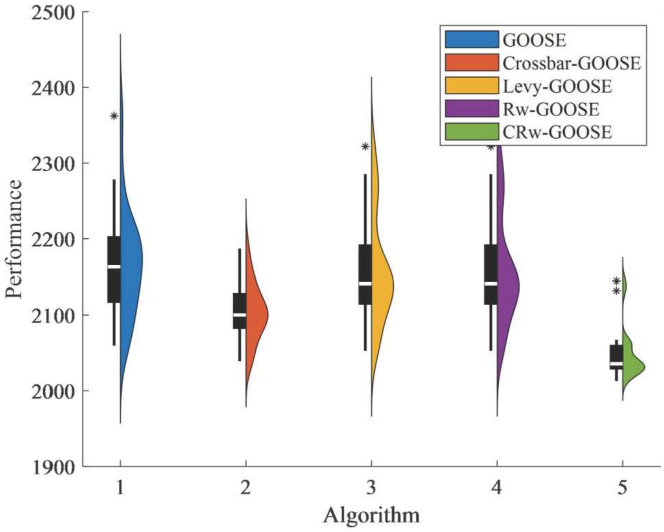
(4) f_4



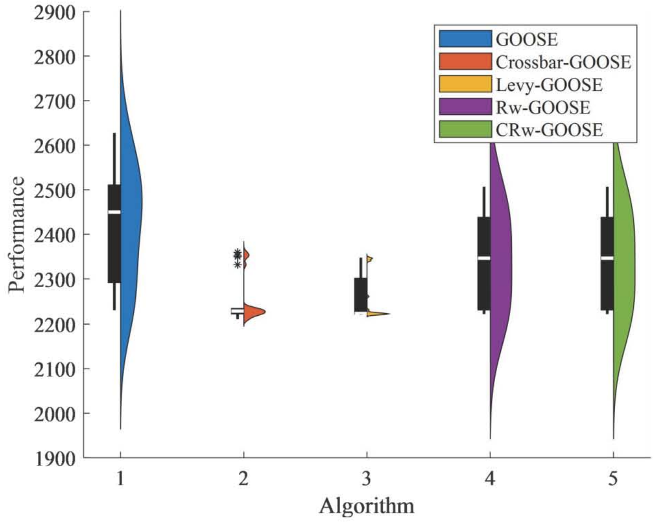
(5) f_5



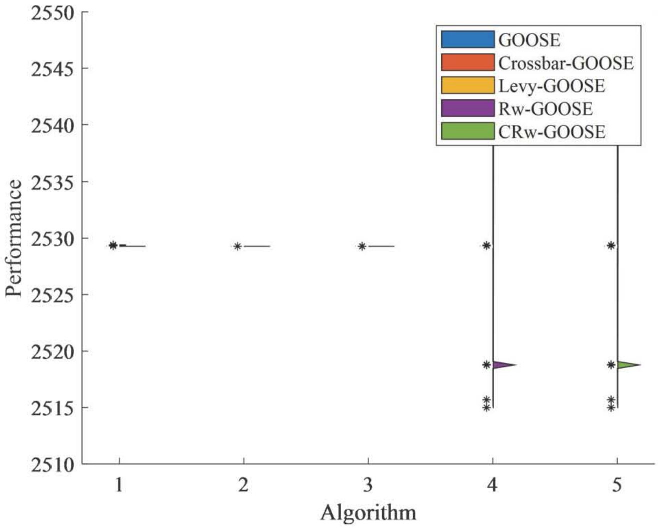
(6) f_6



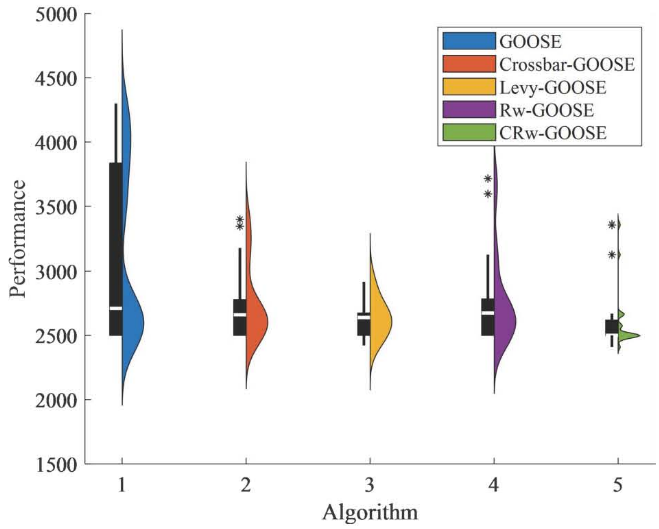
(7) f_7



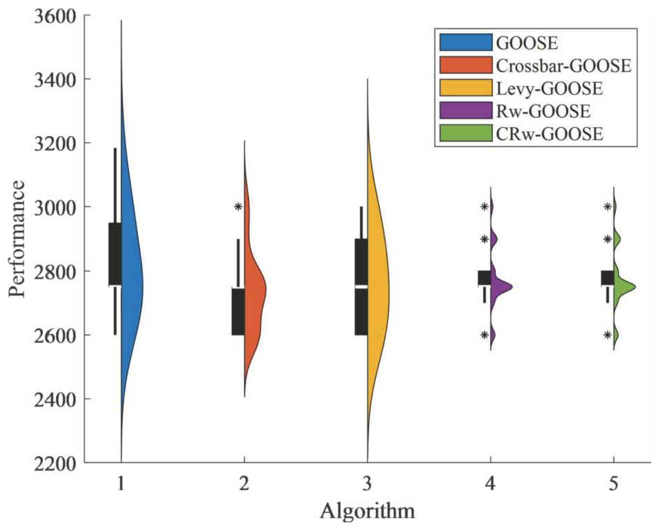
(8) f_8



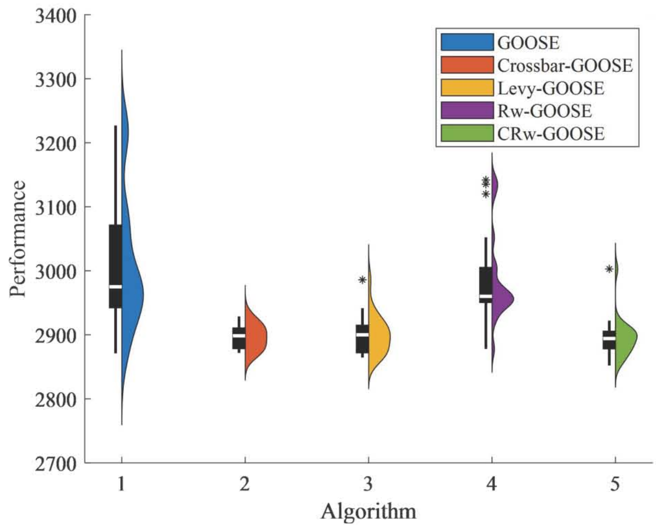
(9) f_9



(10) f_{10}



(11) f_{11}

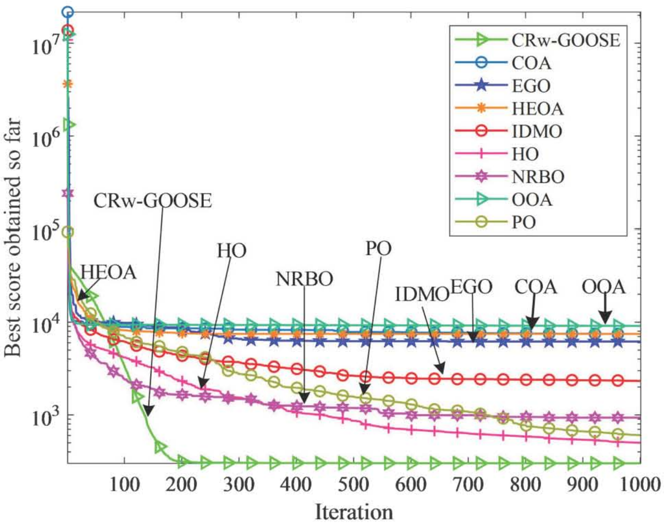


(12) f_{12}

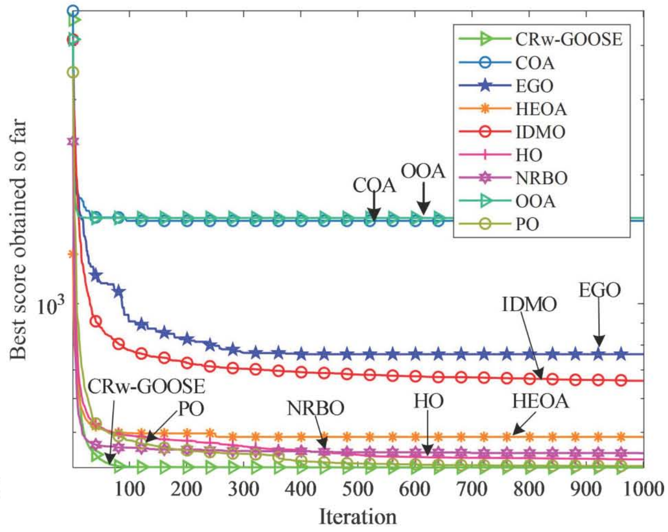
Fig. 3 The violin graph of CEC-BC-2022 test function is solved based on crossbar strategy and random walk improved goose optimization algorithm.

TABLE III. PERFORMANCE COMPARISON RESULTS OF CEC-2022 FUNCTION OPTIMIZATION

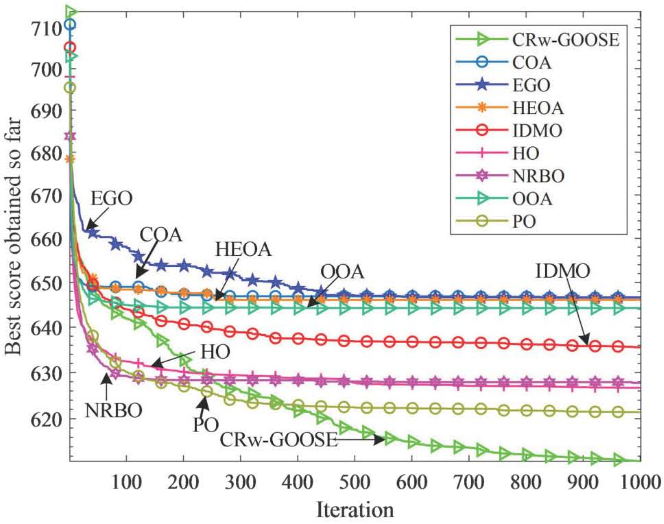
Function		CRw-GOOSE	COA	EGO	HEOA	IDMO	HO	NRBO	OOA	PO
	Best	**3.0000E+02**	2.8989E+03	3.5887E+03	4.1957E+03	1.0075E+03	3.4010E+02	4.6728E+02	4.5303E+03	3.5257E+02
f_1	Ave	**3.0000E+02**	7.4632E+03	6.1858E+03	7.4799E+03	2.3220E+03	5.0052E+02	9.3386E+02	9.1136E+03	6.0731E+02
	Std	**1.3433E-03**	2.1089E+03	2.1823E+03	1.5102E+03	7.3168E+02	1.2066E+02	7.0000E+00	2.6958E+03	3.7772E+02
	Best	**4.0001E+02**	7.5895E+02	5.4443E+02	4.2524E+02	4.2021E+02	4.0000E+02	4.1323E+02	7.9708E+02	4.0205E+02
f_2	Ave	**4.1360E+02**	1.5621E+03	7.6167E+02	4.8801E+02	6.6005E+02	4.3236E+02	4.4715E+02	1.5863E+03	4.1719E+02
	Std	2.5192E+01	7.3372E+02	1.1392E+02	3.6851E+01	2.9388E+02	3.2563E+01	**7.0000E+00**	5.6249E+02	2.2295E+01
	Best	**6.0002E+02**	6.2357E+02	6.3436E+02	6.1407E+02	6.1092E+02	6.0947E+02	6.1519E+02	6.3331E+02	6.1387E+02
f_3	Ave	**6.1091E+02**	6.4655E+02	6.4663E+02	6.4607E+02	6.3550E+02	6.2669E+02	6.2775E+02	6.4424E+02	6.2141E+02
	Std	1.3105E+01	9.3126E+00	**6.2967E+00**	1.3898E+01	1.6837E+01	9.9517E+00	7.0000E+00	7.3802E+00	7.1245E+00
	Best	8.2370E+02	8.3358E+02	8.3835E+02	8.2218E+02	**8.0916E+02**	8.1393E+02	8.1851E+02	8.2969E+02	8.0929E+02
f_4	Ave	8.4055E+02	8.5358E+02	8.5028E+02	8.4514E+02	8.2668E+02	**8.2244E+02**	8.3225E+02	8.4627E+02	8.2432E+02
	Std	1.2490E+01	9.5249E+00	6.6173E+00	1.2066E+01	7.1154E+00	**4.1024E+00**	7.0000E+00	8.4915E+00	6.5850E+00
	Best	**9.0000E+02**	1.1098E+03	1.2211E+03	9.8941E+02	9.8649E+02	9.2149E+02	9.2294E+02	1.1446E+03	9.1113E+02
f_5	Ave	1.0894E+03	1.3943E+03	1.4407E+03	1.5268E+03	1.3759E+03	1.6358E+03	**1.0402E+03**	1.4111E+03	1.0561E+03
	Std	1.0704E+02	1.6211E+02	1.5790E+02	3.2133E+02	1.5534E+02	5.6271E+02	**7.0000E+00**	1.8260E+02	1.1157E+02
	Best	1.9167E+03	8.7500E+05	2.7902E+06	3.1897E+04	3.2392E+03	**1.8436E+03**	1.9532E+03	2.5632E+03	2.2573E+03
f_6	Ave	3.7818E+03	2.0753E+07	2.3240E+07	5.0981E+05	1.0099E+04	**1.8756E+03**	4.6421E+03	4.9417E+06	5.4248E+03
	Std	1.9283E+03	2.9546E+07	1.5181E+07	7.0924E+05	5.7355E+03	1.8798E+01	**7.0000E+00**	1.2295E+07	2.0935E+03
	Best	**2.0050E+03**	2.0545E+03	2.0851E+03	2.0365E+03	2.0403E+03	2.0275E+03	2.0266E+03	2.0543E+03	2.0311E+03
f_7	Ave	**2.0405E+03**	2.0894E+03	2.1024E+03	2.1206E+03	2.0945E+03	2.0459E+03	2.0552E+03	2.0871E+03	2.0483E+03
	Std	3.7812E+01	1.2803E+01	1.2864E+01	3.8153E+01	3.8693E+01	1.1647E+01	**7.0000E+00**	1.8868E+01	1.2515E+01
	Best	**2.2024E+03**	2.2326E+03	2.2259E+03	2.2271E+03	2.2229E+03	2.2085E+03	2.2225E+03	2.2256E+03	2.2218E+03
f_8	Ave	**2.2241E+03**	2.2405E+03	2.2323E+03	2.2358E+03	2.2576E+03	2.2271E+03	2.2420E+03	2.2317E+03	2.2308E+03
	Std	**3.7559E+00**	1.0725E+01	3.8552E+00	4.6272E+00	5.3525E+01	6.2348E+00	7.0000E+00	4.8473E+00	4.2580E+00
	Best	**2.5293E+03**	2.6685E+03	2.6208E+03	2.6125E+03	2.5968E+03	2.5299E+03	2.5324E+03	2.6247E+03	2.5321E+03
f_9	Ave	**2.5293E+03**	2.7418E+03	2.6766E+03	2.6571E+03	2.6628E+03	2.5493E+03	2.5681E+03	2.7414E+03	2.5725E+03
	Std	**6.4604E-04**	4.3379E+01	2.8766E+01	3.1362E+01	4.0128E+01	3.7708E+01	7.0000E+00	3.8706E+01	4.4755E+01
	Best	**2.5003E+03**	2.5129E+03	2.5169E+03	2.6296E+03	2.5005E+03	2.5005E+03	2.5005E+03	2.5138E+03	2.5004E+03
f_{10}	Ave	2.5994E+03	2.7099E+03	2.5925E+03	2.6563E+03	2.6704E+03	2.5438E+03	2.5241E+03	2.7193E+03	**2.5126E+03**
	Std	1.0814E+02	1.4187E+02	7.4668E+01	2.1778E+01	2.2495E+02	6.0063E+01	**7.0000E+00**	2.3503E+02	3.5931E+01
	Best	**2.6001E+03**	3.0530E+03	2.8398E+03	2.7127E+03	2.6536E+03	**2.6001E+03**	2.7357E+03	2.9383E+03	2.6061E+03
f_{11}	Ave	**2.7228E+03**	3.7955E+03	2.9919E+03	2.7443E+03	3.1970E+03	2.8239E+03	2.8186E+03	3.5076E+03	2.7470E+03
	Std	1.9819E+02	4.0851E+02	8.1383E+01	1.5353E+01	4.4501E+02	2.1609E+02	**7.0000E+00**	4.6449E+02	1.6186E+02
	Best	**2.8623E+03**	2.8827E+03	2.8682E+03	2.8691E+03	2.8711E+03	2.8641E+03	2.8631E+03	2.9514E+03	2.8639E+03
f_{12}	Ave	**2.8664E+03**	2.9617E+03	2.8957E+03	2.9272E+03	2.9325E+03	2.8809E+03	2.8677E+03	3.0616E+03	2.8856E+03
	Std	**2.1967E+00**	6.3909E+01	1.0071E+01	5.5136E+01	5.1356E+01	2.7942E+01	7.0000E+00	7.6293E+01	3.1223E+01



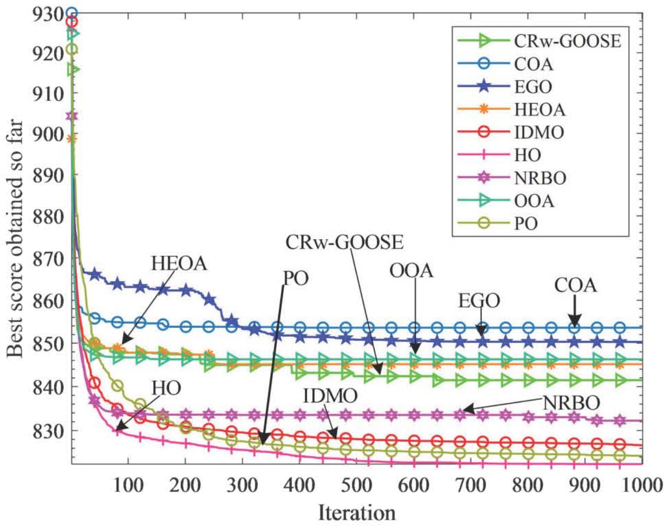
(1) f_1



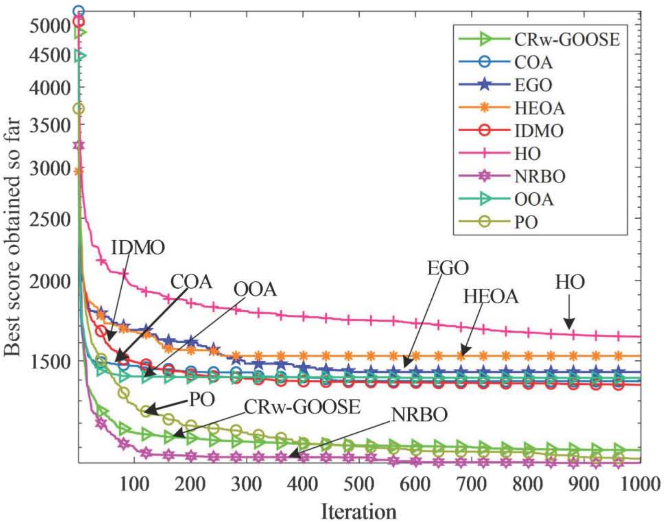
(2) f_2



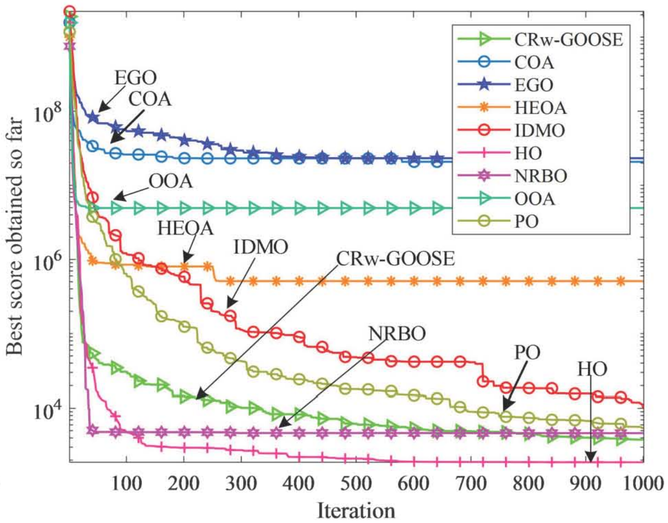
(3) f_3



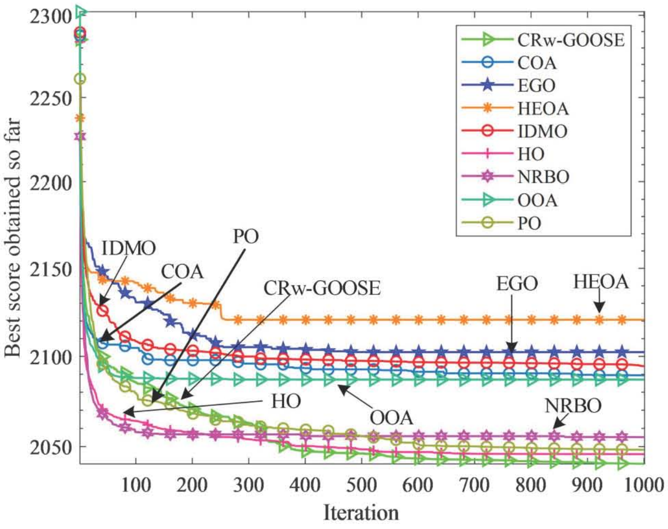
(4) f_4



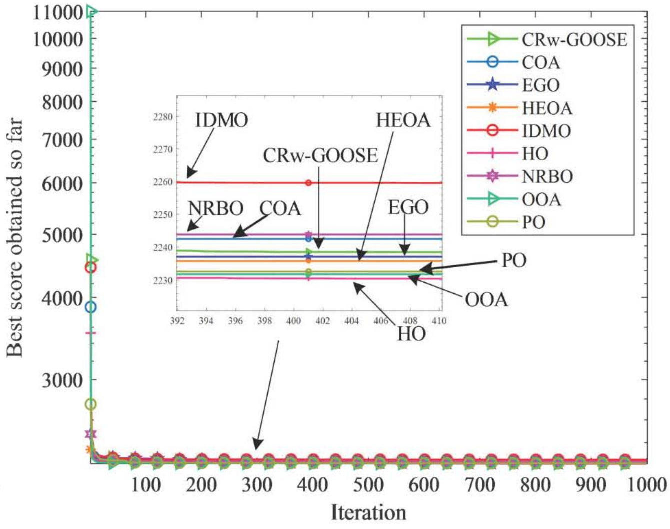
(5) f_5



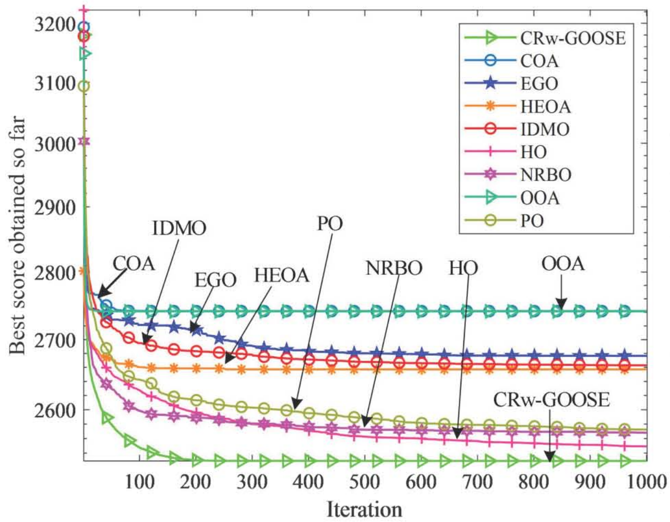
(6) f_6



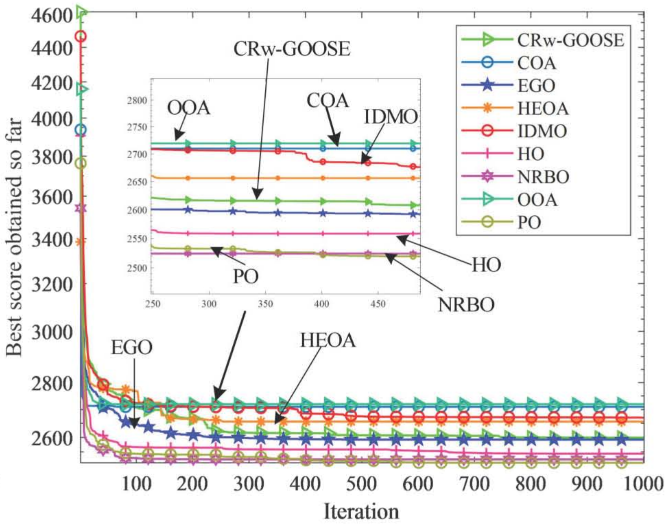
(7) f_7



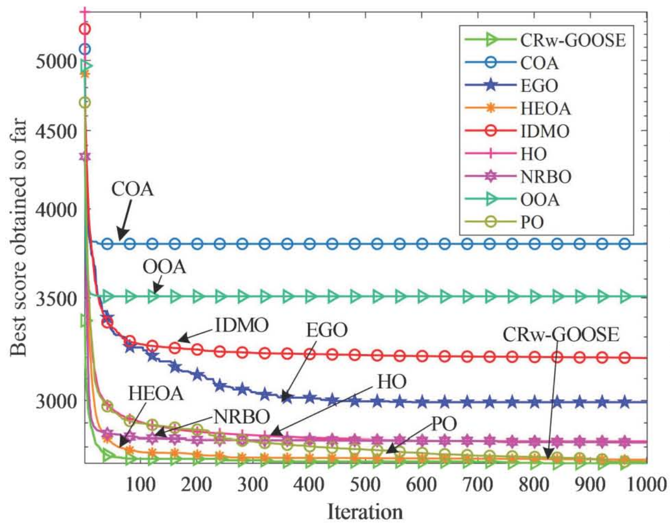
(8) f_8



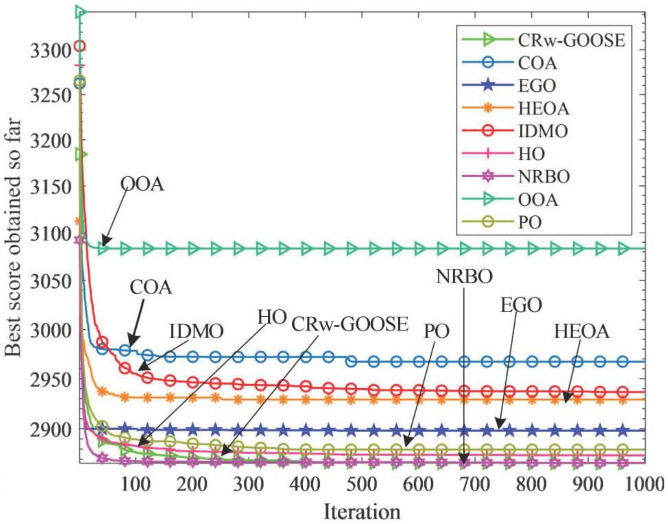
(9) f_9



(10) f_{10}

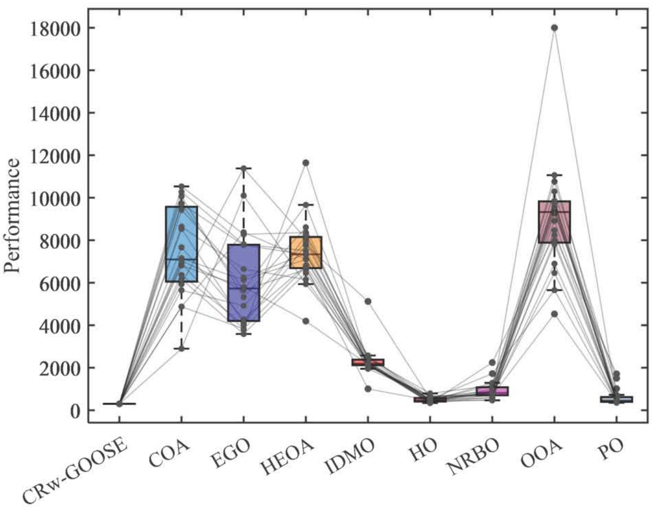


(11) f_{11}

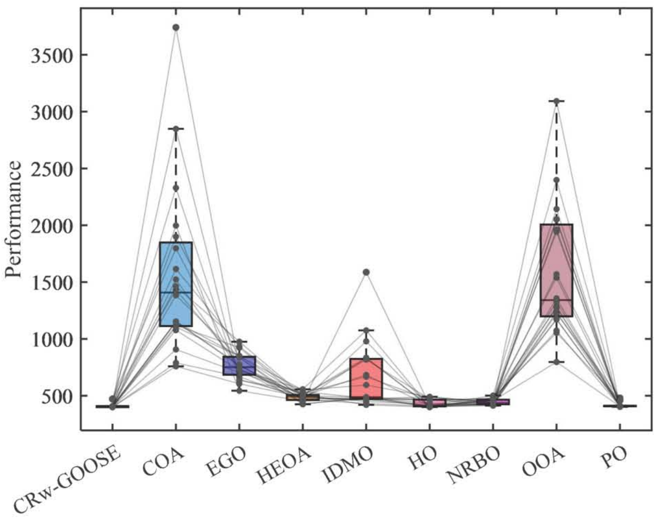


(12) f_{12}

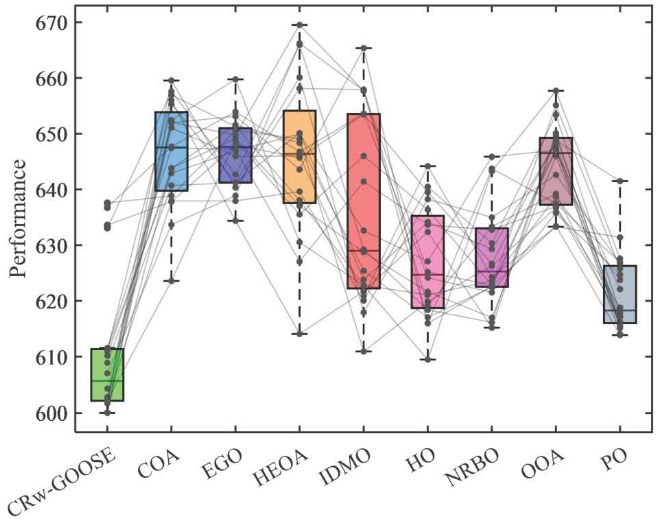
Fig. 4 The convergence graph of CEC-BC-2022 test function was solved by the improved goose optimization algorithm based on the vertical and horizontal crossing strategy and random walk and other intelligent optimization algorithms.



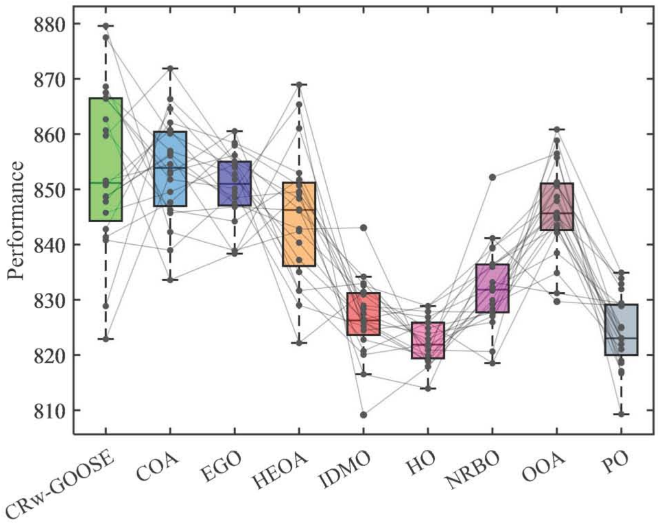
(1) f_1



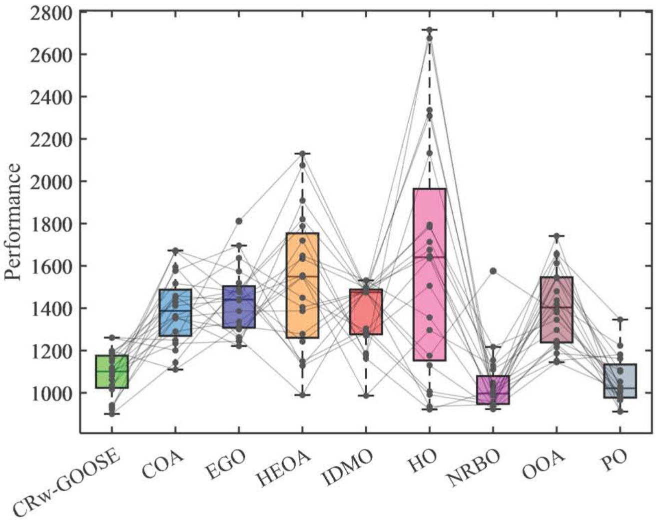
(2) f_2



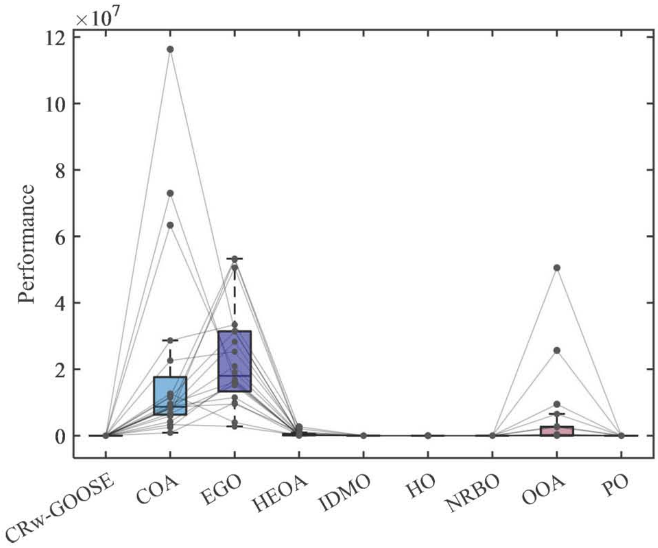
(3) f_3



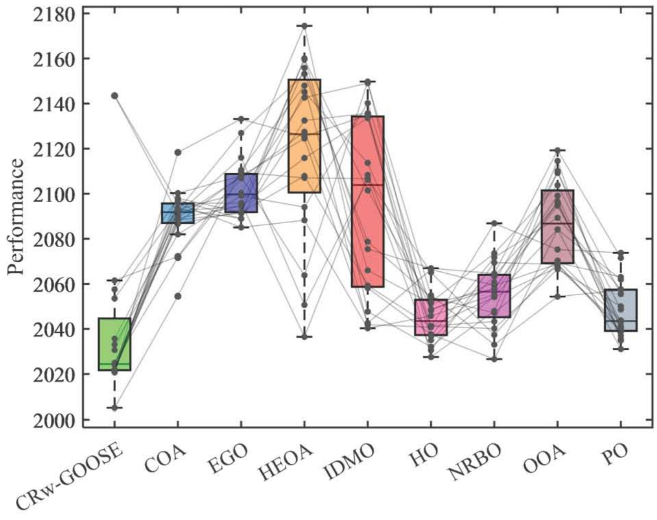
(4) f_4



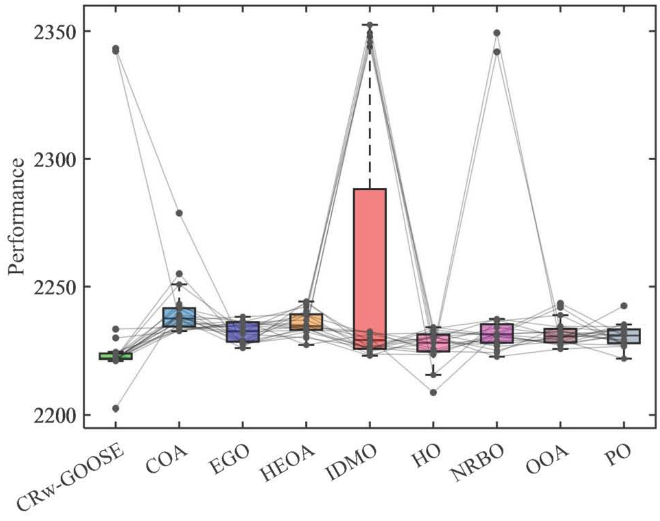
(5) f_5



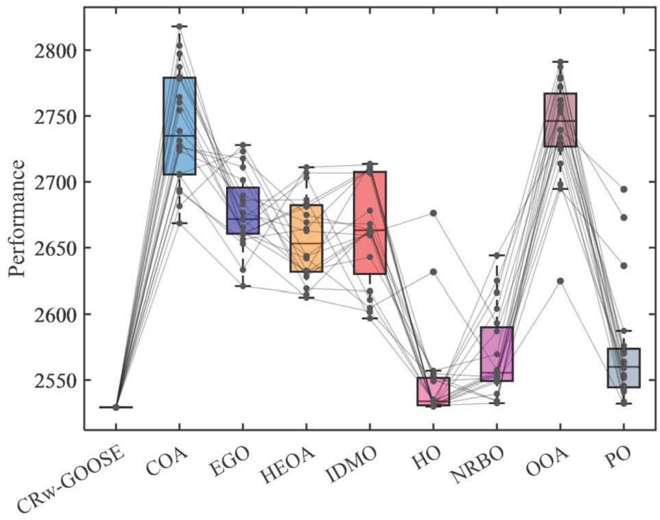
(6) f_6



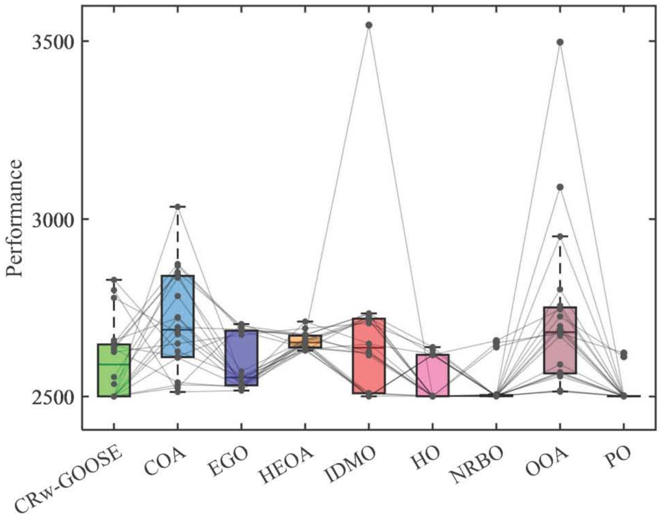
(7) f_7



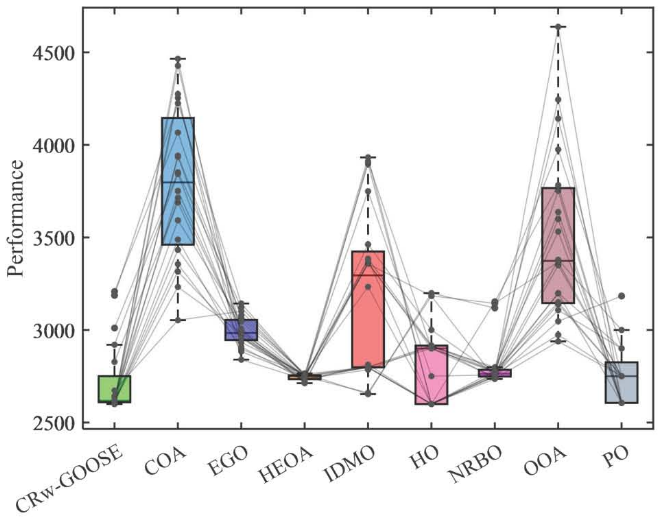
(8) f_8



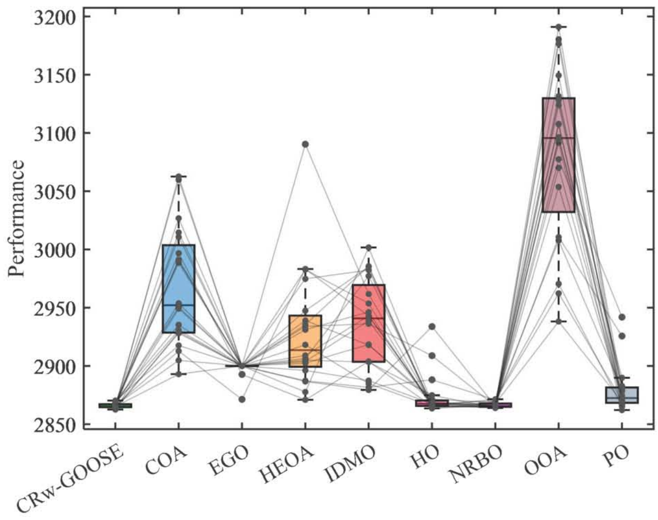
(9) f_9



(10) f_{10}



(11) f_{11}



(12) f_{12}

Fig. 5 The boxplot of CC-BC-2022 test function is solved by the improved goose optimization algorithm based on crosswalk strategy and random walk and other intelligent optimization algorithms.

V.

VI.

It can be seen that CRw-GOOSE, which integrates the three strategies to improve, is better than other advanced intelligent optimization algorithms in most cases in recent years, which proves the superiority and advancement of CRw-GOOSE and the feasibility of the algorithm improvement strategy. As shown in the box diagram in Fig 5, the horizontal coordinate represents different algorithms, and the vertical coordinate represents the 30 optimal fitness values collected. It can be obviously observed that CRw-GOOSE performs very well in most test functions, whether it is the optimal fitness value or the average fitness value. In addition to the function, the average value and optimal value obtained by CRw-GOOSE can reach the minimum when compared with other 7 advanced intelligent optimization algorithms. It can be observed from the figure that the box length of CRw-GOOSE in each test function species is relatively small, which indicates that there are few extreme values of CRw-GOOSE, and also indicates the stability of CRw-GOOSE. In summary, compared with the goose optimization algorithm improved by random walk, the advanced intelligent optimization algorithm based on crossbar strategy has excellent performance, and it can well optimize each test function in CEC-BC-2022.

VII. ENGINEERING OPTIMIZATION DESIGN PROBLEM

A. Three-bar Truss Design Problem

The design of a three-bar truss represents a classic engineering optimization challenge, particularly within the realm of structural engineering. The primary objective is to enhance the performance and cost-effectiveness of the structure while adhering to specified constraints by adjusting parameters such as the size, shape, and connection methods of the truss members. Due to its straightforward configuration and remarkable efficiency, the three-bar truss has found widespread application in bridges, buildings, and various mechanical devices. In optimizing three-bar truss designs, key objectives encompass structural strength and stiffness, overall weight reduction, stability considerations, and economic viability. A schematic representation of the three-bar truss design problem is illustrated in Fig. 6. The objective function along with the associated constraints for this design problem is delineated as follows:

Objective function: $f(X) = \left(2\sqrt{2}X_1 + X_2\right) * l$

Constraints: $g_1(X) = \dfrac{\sqrt{2}X_1 + X_2}{\sqrt{2}X_1 + 2X_1X_2}P - \sigma \le 0$

$$g_2(X) = \frac{X_1}{\sqrt{2}X_1 + 2X_1X_2}P - \sigma \le 0$$

$$g_3(X) = \frac{1}{\sqrt{2}X_2 + X_1}P - \sigma \le 0$$

Boundary conditions: $0 \le X_1, X_2 \le 1$

where, $l = 100cm, P = 2KN/cm, \sigma = 2KN/cm$.

Fig. 7 illustrates the convergence plots of CRw-GOOSE alongside seven other state-of-the-art intelligent optimization algorithms applied to the three-bar truss design problem. The optimization outcomes for the optimal solution of this design challenge are summarized in Table IV. To facilitate a comparative analysis between the performance of the enhanced algorithm and other optimization techniques, each algorithm was constrained to a maximum iteration limit of 1000 generations, with simulations conducted over 30 trials. The optimal value, average value, and variance were calculated and recorded in Table IV, with the optimal experimental data highlighted in bold. As indicated in Table IV, CRw-GOOSE's average results for optimizing the three-bar truss design closely match those achieved by six other intelligent optimization algorithms, excluding EGO.

However, it is noteworthy that CRw-GOOSE's identified optimal value aligns only with those obtained by IDMO and NRBO. Furthermore, CRw-GOOSE exhibits the smallest variance among all methods evaluated, demonstrating its stability when addressing the three-bar truss design problem. In conclusion, both Fig. 7 and Table IV confirm that CRw-GOOSE delivers commendable overall performance in optimizing the three-bar truss design.

B. Cantilever Beam Design Problem

Cantilever beam design is a classic and challenging problem in the field of engineering optimization. It relates to a key component in structural engineering - cantilever beams, which are fixed at one end and extend freely at the other and are widely used in structures such as Bridges, buildings, and mechanical components. The central goal of cantilever beam design is to optimize the use of materials, reduce costs, and improve the economic efficiency of the structure while satisfying the criteria of strength, stiffness, and stability.

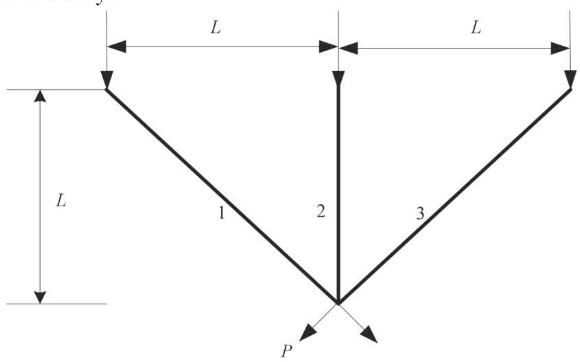


Fig. 6 Three-bar truss design problem model.

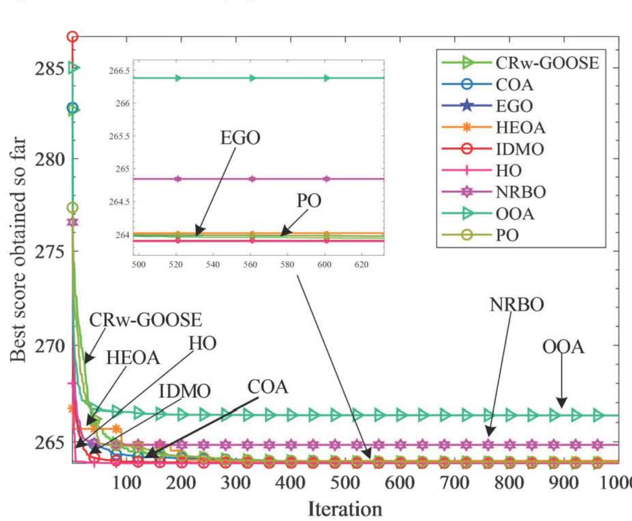


Fig. 7 The convergence diagram of three-bar truss design problem is optimized based on crossbar strategy and random walk improved goose optimization algorithm and other intelligent optimization algorithms.

The beam is made up of five hollow square sections with uniform thickness, where the height serves as the decision variable, and the thickness remains constant. The schematic representation of the cantilever beam design problem is depicted in Fig. 8. The objective function and constraints associated with this design challenge are delineated as follows:

Objective function:

$$f(X)=0.0624(X_1+X_2+X_3+X_4+X_5)$$

Constraints:

$$g(X)=\frac{61}{X_1^3}+\frac{37}{X_2^3}+\frac{19}{X_3^3}+\frac{7}{X_4^3}+\frac{1}{X_5^3}\le 0$$

Boundary conditions: $0.01\le X_i\le 100, i=1,2,3,4,5$

The convergence curve illustrating the optimization of the cantilever beam design problem using CRw-GOOSE, alongside seven other advanced intelligent optimization algorithms, is presented in Fig. 9. The optimal solutions for the cantilever beam design problem are detailed in Table Ⅴ. To facilitate a comparative analysis of the performance between the improved algorithm and other optimization methods, each algorithm was subjected to a maximum iteration limit of 1000 generations, with simulations conducted over 30 trials. The optimal value, average value, and variance were calculated and documented in Table Ⅴ; furthermore, the best experimental results have been highlighted in bold within the table. As can be seen from Table Ⅴ, the average value, optimal value and variance obtained by CRw-GOOSE optimization of the cantilever beam optimization issue are the best, which are 1.7038, 1.7330, and 0.010055 respectively. It can be observed from the data in the table that CRw-GOOSE has absolute advantages in solving cantilever beam design problems compared with other 7 intelligent optimization algorithms. Moreover, the convergence curve illustrated in Fig. 9 indicates that CRw-GOOSE converges at a relatively faster rate compared to other algorithms. As demonstrated in both Fig. 9 and Table Ⅴ, the overall performance of CRw-GOOSE in optimizing cantilever beam design is notably effective.

C. *Pressure Vessel Design Problem*

Pressure vessel design is an important subject in the field of engineering optimization, which involves many disciplines such as structural engineering, material science, and optimization algorithms. As a key industrial equipment, the pressure vessel is widely used in chemical, oil, natural gas, nuclear energy, and other industries, and its design quality is directly related to production safety and economic benefits. The schematic representation of the pressure vessel problem is depicted in Fig. 10. The objective function and constraints associated with the optimal design problem of the pressure vessel are outlined as follows:

Objective function:

$$f(X)=0.6224X_1X_3X_4+1.7781X_2X_3^2+3.1661X_1^2X_4+19.84X_1^2X_3$$

Constraints: $g_1(X)=0.0193X_3-X_1\le 0$

$$g_2(X)=0.00954X_3-X_2\le 0$$

$$g_3(X)=1296000-\pi X_3^2X_4-\frac{4}{3}\pi X_3^3\le 0$$

$$g_4(X)=X_4-240\le 0$$

Boundary conditions: $0.0625\le X_1,X_2\le 6.1875$

where, X_1 and X_2 represent the cylinder head (*Th*) and cylinder wall thickness (*Ts*), X_3 represents the radius of the cylinder and cylinder head (*R*), and X_4 denotes the cylinder's length (*L*).

Fig. 11 illustrates the convergence plot of CRw-GOOSE alongside seven other advanced intelligent optimization methods employed for optimizing pressure vessel design problems. The experimental results pertaining to the optimal solutions for these design challenges are presented in Table VI. To facilitate a comparative analysis of the performance between the improved algorithm and other optimization techniques, each algorithm was subjected to an iteration limit set at 1000 generations, with simulations conducted over 30 trials. The optimal value, average value, and variance were computed and documented in Table VI; Notably, the best experimental data highlighted within this table is presented in bold. As evidenced by Table VI, both the average value and optimal value achieved by CRw-GOOSE in addressing the pressure vessel design problem are superior to those obtained through other methods, 5.7357E+03 and 6.2414e+03 respectively. The convergence diagram of Fig. 11 also shows that CRw-GOOSE optimization of pressure vessel design has obvious advantages compared with some intelligent optimization algorithms. Fig. 11 and Table VI show that CRw-GOOSE achieves an excellent overall performance in optimizing pressure vessel design.

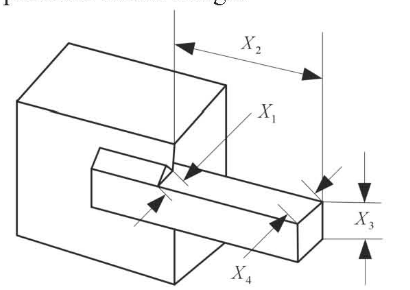


Fig. 8 Cantilever beam design problem model.

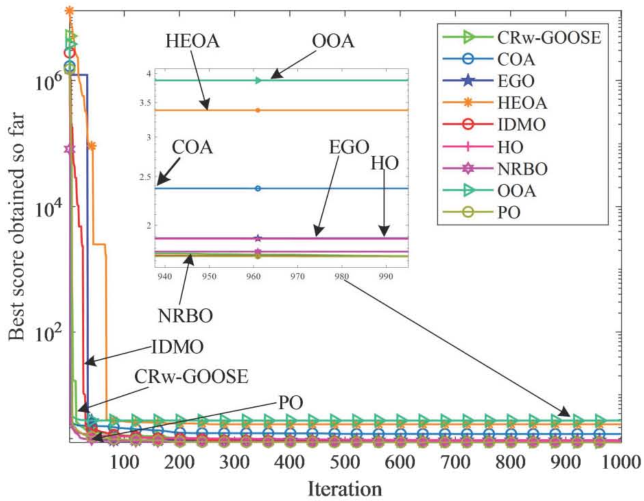


Fig. 9 The convergence graph of cantilever beam design problem is optimized based on crossbar strategy and random walk improved goose optimization algorithm and other intelligent optimization algorithms.

TABLE IV. THE RESULTS OBTAINED FROM THE THREE-BAR TRUSS DESIGN PROBLEM

Algorithm	x	f(x)		
		Ave	Best	Std
CRw-GOOSE	[0.7893, 0.4066]	**2.6390E+02**	**2.6390E+02**	**3.4187E-03**
COA	[0.7965, 0.3865]	**2.6390E+02**	2.6394E+02	6.5029E-02
EGO	[0.7908, 0.3983]	2.6391E+02	2.6392E+02	2.2870E-02
HEOA	[0.7759, 0.4456]	**2.6390E+02**	2.6440E+02	5.4011E-01
IDMO	[0.7882, 0.4097]	**2.6390E+02**	**2.6390E+02**	6.8163E-03
HO	[0.7839, 0.4241]	**2.6390E+02**	2.6392E+02	1.6316E-02
NRBO	[0.7887, 0.4082]	**2.6390E+02**	**2.6390E+02**	3.0700E-02
OOA	[0.7945, 0.3921]	**2.6390E+02**	2.6637E+02	2.1991E+00
PO	[0.7821, 0.4270]	**2.6390E+02**	2.6395E+02	1.7070E-01

TABLE V. THE RESULTS OBTAINED FROM THE CANTILEVER BEAM DESIGN PROBLEM

Algorithm	x	f(x)		
		Ave	Best	Std
CRw-GOOSE	[0.1919, 4.7134, 9.0366, 0.2057]	**1.7038E+00**	**1.7330E+00**	**1.0055E-02**
COA	[0.2068, 9.0221, 9.0221, 0.2068]	1.7692E+00	2.3681E+00	2.4016E-01
EGO	[0.1899, 3.5656, 9.1559, 0.2053]	1.7078E+00	1.7409E+00	1.5541E-02
HEOA	[0.2787, 2.9130, 6.6869, 0.3900]	2.0476E+00	3.3781E+00	7.2155E-01
IDMO	[0.1895, 3.5717, 9.1356, 0.2054]	1.7066E+00	1.7334E+00	1.4455E-02
HO	[0.1968, 3.5219, 8.6880, 0.2226]	1.7845E+00	1.8788E+00	2.1099E-01
NRBO	[0.2062, 3.2665, 8.9673, 0.2090]	1.7089E+00	1.7341E+00	3.4437E-02
OOA	[0.2513, 6.0095, 5.0208, 0.6664]	2.2588E+00	3.8736E+00	9.4091E-01
PO	[0.1803, 3.7715, 9.0423, 0.2057]	1.7070E+00	1.7336E+00	2.8325E-02

D. *Tension Spring Design Problem*

The design of tensile spring is an important topic in the field of engineering optimization. It relates to the spring design in mechanical engineering, the purpose is to adjust the geometric parameters and material properties of the spring, so that the spring has the best performance and the minimum material cost while fulfilling the specified performance criteria. The constraints imposed on the design include minimum deflection, shear stress limits and vibration frequency requirements. The model for the stretch spring design problem is illustrated in Fig. 12. The objective function along with the constraints pertinent to the stretch spring design problem are detailed as follows:

Objective function: $f(X) = (X_3 + 2) X_2 X_1^2$

Constraints: $g_1(X) = 1 - \frac{X_2^3 X_3}{71785 X_1^4} \le 0$

$$g_2(X) = \frac{4X_2^2 - X_1 X_2}{12566\left(X_2 X_1^3 - X_1^4\right)} + \frac{1}{5108 X_1^2} \le 0$$

$$g_3(X) = 1 - \frac{140.45 X_1}{X_2^2 X_3} \le 0$$

$$g_4(X) = \frac{X_1 + X_2}{1.5} - 1 \le 0$$

Boundary conditions: $0.05 \le X_1 \le 2.00$, $0.25 \le X_2 \le 1.30$, $2.00 \le X_3 \le 15.0$.

Fig. 13 shows the convergence curve of CRw-GOOSE and seven other advanced intelligent optimization algorithms for optimizing the stretch spring design problem. The optimization results for the optimal solution of the stretch spring design problem are presented in Table Ⅶ. To enable a comparison of the improved algorithm's performance and other optimization algorithms, the iteration cap for each algorithm was established at 1000 generations, and the experiment was simulated 30 times. The optimal value, average value, and variance were obtained and recorded in Table Ⅶ, with the best experimental data highlighted in bold. As illustrated in Table Ⅶ, the average value, optimal value, and variance achieved by CRw-GOOSE in optimizing the design problem of stretch springs are superior when compared to those generated by the other seven intelligent optimization algorithms. In the index of average value, CRw-GOOSE and PO obtained the same minimum average value.

In addition, the variance obtained by CRw-GOOSE in optimizing the design problem of stretch spring is about 0, which is far superior to other intelligent optimization algorithms, and proves the stability of CRw-GOOSE in addressing the stretch spring design problem. As can be seen from Fig. 13 and Table Ⅶ, the comprehensive effect of CRw-GOOSE in optimizing the design of tensile spring is very good.

Fig. 13 shows the convergence curve of CRw-GOOSE and seven other advanced intelligent optimization algorithms for optimizing the stretch spring design problem. The optimization results for the optimal solution of the stretch spring design problem are presented in Table Ⅶ. To enable a comparison of the improved algorithm's performance and other optimization algorithms, the iteration cap for each algorithm was established at 1000 generations, and the experiment was simulated 30 times. The optimal value, average value, and variance were obtained and recorded in Table Ⅶ, with the best experimental data highlighted in bold.

As illustrated in Table Ⅶ, the average value, optimal value, and variance achieved by CRw-GOOSE in optimizing the design problem of stretch springs are superior when compared to those generated by the other seven intelligent optimization algorithms. In the index of average value, CRw-GOOSE and PO obtained the same minimum average value. In addition, the variance obtained by CRw-GOOSE in optimizing the design problem of stretch spring is about 0, which is far superior to other intelligent optimization algorithms, and proves the stability of CRw-GOOSE in addressing the stretch spring design problem. As can be seen from Fig. 13 and Table Ⅶ, the comprehensive effect of CRw-GOOSE in optimizing the design of tensile spring is very good.

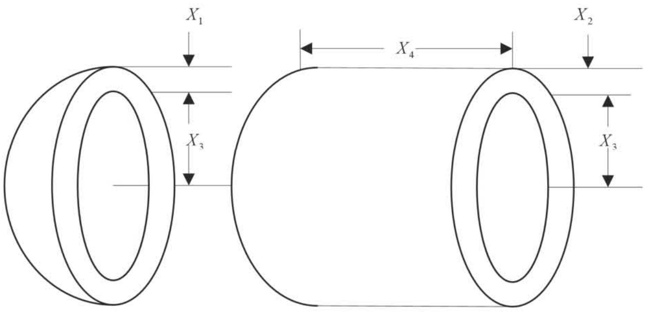


Fig. 10 Pressure vessel design problem model.

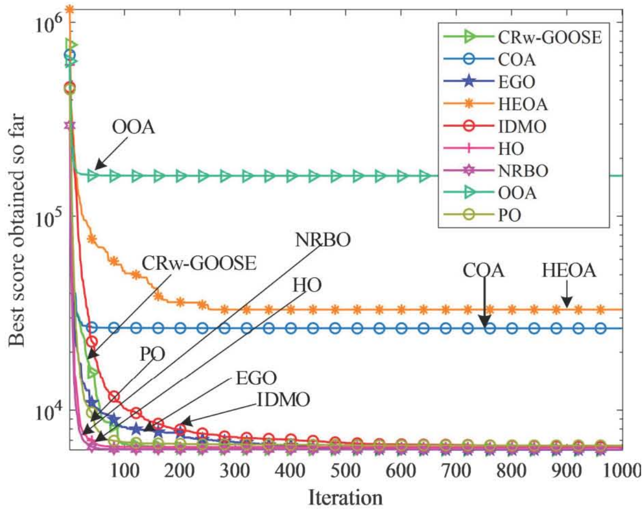


Fig. 11 The convergence graph of pressure vessel design problem is optimized based on crossbar strategy and random walk improved goose optimization algorithm and other intelligent optimization algorithms.

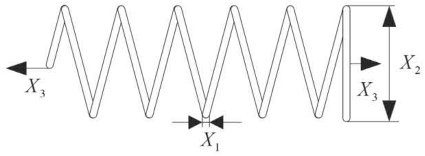


Fig. 12 Tension spring design problem model.

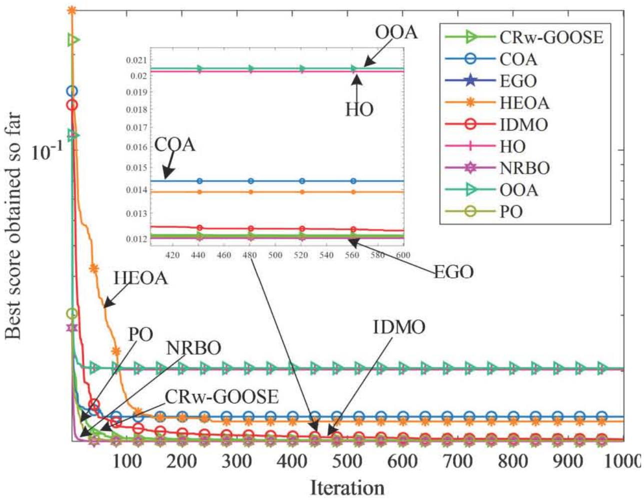


Fig. 13 The convergence graph of tension spring design problem is optimized based on crossbar strategy and random walk improved goose optimization algorithm and other intelligent optimization algorithms.

TABLE VI. THE RESULTS OBTAINED FROM THE PRESSURE VESSEL DESIGN PROBLEM

Algorithm	x	f(x)		
		Ave	Best	Std
CRw-GOOSE	[1.1278, 0.5369, 59.5367, 37.0000]	**5.7357E+03**	**6.2414E+03**	4.7247E+02
COA	[2.5411, 15.4987, 59.9535 , 34.8612]	6.8713E+03	2.6531E+04	2.8906E+04
EGO	[1.2131, 0.4229, 43.7087, 158.4888]	5.8499E+03	6.2571E+03	9.1285E+02
HEOA	[1.0772, 0.4875, 52.6928, 126.8476]	7.2549E+03	3.3095E+04	4.5349E+04
IDMO	[1.1725, 0.5367, 61.7582, 25.9534]	5.8129E+03	6.4274E+03	**4.4261E+02**
HO	[0.8065, 0.3970, 43.4379, 160.7174]	5.8200E+03	6.4424E+03	4.5082E+02
NRBO	[1.1789, 0.5718, 62.3243, 23.1048]	5.7560E+03	6.2694E+03	5.3448E+02
OOA	[15.7067, 12.7774, 52.6461, 78.6511]	9.8250E+03	1.6132E+05	1.3227E+05
PO	[1.2294, 0.5848, 64.8047, 11.8273]	5.7558E+03	6.5760E+03	5.2460E+02

TABLE VII. THE RESULTS OBTAINED FROM THE TENSION SPRING DESIGN PROBLEM

Algorithm	x	f(x)		
		Ave	Best	Std
CRw-GOOSE	[0.0555, 0.2917, 8.1624]	**1.2020E-02**	**1.2020E-02**	**3.7755E-18**
COA	[0.0527, 0.2500, 10.1432]	1.2044E-02	1.4374E-02	2.6311E-03
EGO	[0.0556, 0.2302, 10.1640]	1.2054E-02	1.2084E-02	3.0709E-05
HEOA	[0.0676, 0.5943, 2.0000]	1.2145E-02	1.3889E-02	1.1679E-03
IDMO	[0.0621, 0.4400, 3.6128]	1.2034E-02	1.2195E-02	2.3030E-04
HO	[0.0540, 0.2634, 4.1632]	1.2230E-02	2.0434E-02	9.8921E-03
NRBO	[0.0542, 0.2661, 9.7853]	1.2022E-02	1.2073E-02	2.9661E-05
OOA	[0.0544, 0.2677, 4.1928]	1.2252E-02	2.0443E-02	9.9205E-03
PO	[0.0535 0.2543, 10.4985]	**1.2020E-02**	1.2055E-02	2.5931E-05

VIII. CONCLUSION

The goose optimization algorithm is often susceptible to becoming trapped in local optima, exhibiting relatively low convergence accuracy and speed. This paper introduces an enhanced goose optimization algorithm that incorporates a crossbar strategy and random walk improvements. These enhancements significantly bolster the algorithm's exploration and exploitation capabilities, thereby mitigating the risk of local optimization and improving convergence accuracy.

In this study, two types of simulation experiments are conducted to evaluate the effectiveness and superiority of the proposed CRw-GOOSE algorithm. The simulation results obtained by CRw-GOOSE on 12 different types of reference functions in CEC-BC-2022 all prove that the enhanced algorithm performs well. In the first simulation experiment, GOOSE and CRw-GOOSE, which introduced three strategies separately, were first simulated, and the results showed that each strategy had a better effect than the original GOOSE, among which the optimized test function of CRw-GOOSE obtained the best result. In the second simulation experiment, CRw-GOOSE is compared with other 7 advanced intelligent optimization algorithms. The results show that CRw-GOOSE can achieve very good results in most simulation experiments.

The results of two simulation experiments prove the superiority and advanced nature of the designed CRw-GOOSE. In addition, four engineering design issues were optimized. The test results indicate that all four engineering design challenges addressed by CRw-GOOSE yielded favorable outcomes. Notably, the average value, optimal value, and variance achieved through CRw-GOOSE are superior in three specific engineering problems: the three-bar truss optimization issue, the cantilever beam design challenge, and the tensile spring design problem. In summary, all simulation results demonstrate that the proposed CRw-GOOSE effectively tackles both function optimization and engineering optimization challenges.

REFERENCES

[1] F. S. Gharehchopogh, I. Maleki, and Z. A. Dizaji, "Chaotic vortex search algorithm: metaheuristic algorithm for feature selection," *Evolutionary Intelligence*, vol. 15, no. 3, pp. 1777-1808, 2022.

[2] Y. L. Chen, S. Y. Huang, Y. C. Chang, and H. C. Chao, "Resource allocation based on genetic algorithm for cloud computing," *2021 30th Wireless and Optical Communications Conference (WOCC)*, pp. 211-212, 2021.

[3] Y. L. Jin, Z. H. Jiang, and W. R. Hou, "Multi-objective integrated optimization research on preventive maintenance planning and production scheduling for a single machine," *International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9-10, pp. 954-964, 2008.

[4] F. S. Lobato, and V. S. Jr, "Fish swarm optimization algorithm applied to engineering system design," *Latin American Journal of Solids & Structures*, vol. 11, no. 1, pp. 143-156, 2014.

[5] S. C. Evans, T. Shah, H. Huang, and S. P. Ekanayake, "The entropy economy and the kolmogorov learning cycle: Leveraging the intersection of machine learning and algorithmic information theory to jointly optimize energy and learning," *Physica D: Nonlinear Phenomena*, vol. 461, pp. 134051-134051, 2024.

[6] S. Klinmalee, T. Naenna, and C. Woarawichai, "Application of a genetic algorithm for multi-item inventory lot-sizing with supplier selection under quantity discount and lead time," *International Journal of Operational Research*, vol. 38, no. 3, pp. 403-421, 2020.

[7] C. Chen, J. Zhou, L. Zheng, Y. Wang, X. Zheng, B. Wu, C. Chen, L. Wang, and J. Yin, "Toward scalable and privacy-preserving deep neural network via algorithmic-cryptographic co-design," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 4, pp. 1-21, 2022.

[8] V. H. Pacheco-Valencia, N. Vakhania, F. Á. Hernández-Mira, and J. A. Hernández-Aguilar, "A multi-phase method for euclidean traveling salesman problems," *Axioms*, vol. 11, no. 9, pp. 439-439, 2022.

[9] A. L. Bolaji, F. Z. Okwonu, P. B. Shola, B. S. Balogun, and O. D. Adubisi, "A modified binary pigeon-inspired algorithm for solving the multi-dimensional knapsack problem," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 1-14, 2021.

[10] Z. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60-68, 2001.

[11] S. Segura, R. Romero, and M. J. Rider, "Efficient heuristic algorithm used for optimal capacitor placement in distribution systems," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 1, pp. 71-78, 2010.

[12] M. S. Puga, and J. S. Tancrez, "A heuristic algorithm for solving large location–inventory problems with demand uncertainty," *European Journal of Operational Research*, vol. 259, no. 2, pp. 413-423, 2017.

[13] G. Tasoglu, and M. A. Ilgin, "A simulation-based genetic algorithm approach for the simultaneous consideration of reverse logistics network design and disassembly line balancing with sequencing," *Computers & Industrial Engineering*, vol. 187, pp. 109794-109794, 2024.

[14] S. A. Ludwig, "Particle swarm optimization approach with parameter-wise hill-climbing heuristic for task allocation of workflow applications on the cloud," *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pp. 201-206, 2013.

[15] B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovský, F. S. Gharehchopogh, E. M. El-kenawy, L. Abualigah, and Seyedali Mirjalili, "Puma optimizer (PO): A novel metaheuristic optimization

algorithm and its application in machine learning," *Cluster Computing*, vol. 27, no, 4, pp. 5235-5283, 2024.

[16] Y. Fu, D. Liu, J. Chen, and L. He, "Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems," *Artificial Intelligence Review*, vol. 57, no. 5, 2024.

[17] H. B. Zhang, H. J. San, J. P. Chen, H. J. Sun, L. Ding, and X. M. Wu, "Black eagle optimizer: A metaheuristic optimization method for solving engineering optimization problems," *Cluster Computing*, vol. 27, pp. 12361-12393, 2024.

[18] M. Abdel-Basset, R. Mohamed, and M. Abouhawwash, "Crested porcupine optimizer: A new nature-inspired metaheuristic," *Knowledge-Based Systems*, vol. 284, pp. 111257-111257, 2024.

[19] Y. C. Liang, and A. E. Smith, "An ant colony optimization algorithm for the redundancy allocation problem (RAP)," *IEEE Transactions on Reliability*, vol. 53, no. 3, pp. 417-423, 2004.

[20] W. Z. Liao, X. Y. Xia, X. J. Jia, S. G. Shen, H. L. Zhuang, and X. C. Zhang, "A spider monkey optimization algorithm combining opposition-based learning and orthogonal experimental design," *Computers, Materials and Continua*, vol. 76, no. 3, pp. 3297-3323, 2023.

[21] M. Y. Li, Z. L. Liu, and H. X. Song, "An improved algorithm optimization algorithm based on RungeKutta and golden sine strategy," *Expert Systems with Applications*, vol. 247, pp. 123262-123262, 2024.

[22] X. Wang, H. Zhao, T. Han, Z. Wei, Y. Liang, and Y. Li, "A gaussian estimation of distribution algorithm with random walk strategies and its application in optimal missile guidance handover for multi-UCAV in over-the-horizon air combat," *IEEE Access*, vol. 7, pp. 43298-43317, 2019.

[23] X. Cai, H. Wang, Z. Cui, J. Cai, Y. Xue, and L. Wang, "Bat algorithm with triangle-flipping strategy for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 9, pp. 199-215, 2017.

[24] R. K. Hamad, and T. A. Rashid, "Goose algorithm: A powerful optimization tool for real-world engineering challenges and beyond," *Evolving Systems*, vol. 15, no. 4, pp. 1249-1274, 2024.

[25] A. Mohammadzadeh, and S. Mirjalili, "Eel and grouper optimizer: A nature-inspired optimization algorithm," *Cluster Computing*, vol. 27, pp. 12745-12786, 2024.

[26] J. Lian, and G. Hui, "Human evolutionary optimization algorithm," *Expert Systems With Applications*, vol. 241, pp. 122638-122638, 2024.

[27] A. I. Hammouri, M. A. Awadallah, M. S. Braik, M. A. Al-Betar, and M. Beseiso, "Improved dwarf mongoose optimization algorithm for feature selection: Application in software fault prediction datasets," *Journal of Bionic Engineering*, vol. 21, no. 4, pp. 2000-2033, 2024.

[28] M. H. Amiri, N. M. Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm," *Scientific Reports*, vol. 14, no. 1, pp. 5032-5032, 2024.

[29] R. Sowmya, M. Premkumar, and P. Jangir, "Newton raphson based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 128, pp. 107532-107532, 2024.

[30] M. Dehghani M and P. Trojovsk, "Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems," *Frontiers of Mechanical Engineering*, vol. 8, 2022.

[31] J. Lian, G. Hui, L. Ma, T. Zhu, X. Wu, A. A. Heidari Y. Chen, and H. Chen, "Parrot optimizer: Algorithm and applications to medical problems," *Computers in Biology and Medicine*, vol. 172, pp. 108064-108064, 2024.