Article

An Improved Gray Wolf Optimization Algorithm to Solve Engineering Problems

Yu Li ¹ , Xiaoxiao Lin ² and Jingsen Liu ^{3,*}

¹ Institute of Management Science and Engineering, and School of Business, Henan University, Kaifeng 475004, China; leey@henu.edu.cn

² School of Business, Henan University, Kaifeng 475004, China; 104753190931@henu.edu.cn

³ Institute of Intelligent Network Systems, and Software School, Henan University, Kaifeng 475004, China

* Correspondence: ljs@henu.edu.cn

Abstract: With the rapid development of the economy, the disparity between supply and demand of resources is becoming increasingly prominent in engineering design. In this paper, an improved gray wolf optimization algorithm is proposed (IGWO) to optimize engineering design problems. First, a tent map is used to generate the initial location of the gray wolf population, which evenly distributes the gray wolf population and lays the foundation for a diversified global search process. Second, Gaussian mutation perturbation is used to perform various operations on the current optimal solution to avoid the algorithm falling into local optima. Finally, a cosine control factor is introduced to balance the global and local exploration capabilities of the algorithm and to improve the convergence speed. The IGWO algorithm is applied to four engineering optimization problems with different typical complexity, including a pressure vessel design, a tension spring design, a welding beam design and a three-truss design. The experimental results show that the IGWO algorithm is superior to other comparison algorithms in terms of optimal performance, solution stability, applicability and effectiveness; and can better solve the problem of resource waste in engineering design. The IGWO also optimizes 23 different types of function problems and uses Wilcoxon rank-sum test and Friedman test to verify the 23 test problems. The results show that the IGWO algorithm has higher convergence speed, convergence precision and robustness compared with other algorithms.

Keywords: engineering optimization design; gray wolf optimization; tent chaotic map; Gaussian mutation disturbance; cosine control factor

Citation: Li, Y.; Lin, X.; Liu, J. An Improved Gray Wolf Optimization Algorithm to Solve Engineering Problems. *Sustainability* **2021**, *13*, 3208. <https://doi.org/10.3390/su13063208>

Academic Editor: Yuri Borgianni

Received: 27 December 2020

Accepted: 11 March 2021

Published: 15 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many countries see the importance of resource optimization and sustainable development and consequently direct research to save resources and maintain sustainable development. As an important part of the entire life cycle of engineering construction projects, engineering design is essential to transform science and technology into actual production and is also at the key stage of determining and controlling engineering costs. Engineering design directly affects the construction of the project.

In the process of engineering construction, an unreasonable construction process will consume large amounts of raw materials and produce large amounts of construction waste, lead to natural resources shortage and environmental pollution [1]. Reasonable engineering structure design can save materials and costs, thus saves resources and alleviates the disparity between resource supply and economic and social development. It is also important for achieving sustainable economic and social development and protecting national resources, security and the environment in the long run. Therefore, the engineering optimization design problem has attracted the increasing attention of scholars.

As a new optimization algorithm, an intelligent optimization algorithm can better solve engineering design optimization problems without global information and has gradually become a new method to solve engineering optimization problems [2]. It is based on

the collective behavior of creatures living in swarms or colonies. The common intelligent optimization algorithms include particle swarm optimization [3] (PSO), moth–flame optimization algorithm [4] (MFO), and the novel ant lion optimization [5] (ALO), sine–cosine algorithm [6] (SCA), the bat algorithm [7] (BA), flower pollination algorithm [8] (FPA) and salp swarm algorithm [9].

In recent years, many scholars have applied intelligent optimization algorithms to resource conservation, low green carbon and other problems. Afsharis used constrained particle swarm optimization to solve the multi-reservoir cogeneration scheduling problem [10]. Kim studied the relationship between road transportation cost and carbon dioxide emissions, proposed the main factors affecting carbon dioxide emissions in the transportation process, optimized the cargo transportation mode, and reduced carbon dioxide emissions within a reasonable cost and time limit [11]. Lin et al. proposed an integrated model and teaching optimization (TLBO) algorithm for machining parameter optimization and flow shop scheduling to minimize the maximum completion time and carbon emissions [12]. He et al. proposed an energy-saving optimization method. They used machine tool selection to reduce machine processing energy consumption and adjusted operation sequence to reduce energy waste when the machine was idle [13]. Du et al. explored a new solution to the optimal allocation of water resources to alleviate the contradiction between supply and demand of water resources and improve the utilization rate of water resources. The objective function is constructed by integrating social, economic and ecological benefits, and the optimal allocation model based on simulated annealing particle swarm optimization is designed [14]. Huang et al. proposed a multipollutant cost-effective optimization system based on genetic algorithm (GA) in machine learning to provide a cost-effective air quality control strategy for large-scale applications [2].

Gray wolf optimization (GWO) was proposed by scholar Mirjalili in 2014, which was inspired by the leadership hierarchy and hunting mechanism of gray wolves in nature [15]. The optimization process mainly includes social hierarchy, encircling the prey and attacking the prey. GWO algorithm is a new type of swarm intelligence optimization algorithm and has many characteristics of swarm intelligence optimization algorithm. For example, gray wolf has no special requirements on the objective function, does not depend on the rigorous mathematical characteristics of the optimization problem. Additionally, their mechanisms are easy to implement. In terms of algorithm implementation, the gray wolf algorithm can design a reasonable algorithm implementation method according to specific problems, so the algorithm has strong versatility. At present, the algorithm has been applied to practical problems, such as the workshop scheduling problem [16], control system [17], thermal power system [18,19], image segmentation [20], mechanical design [21], neural network [22], and achieved good optimization results.

As a new optimization method, the gray wolf algorithm also has pros and cons. The algorithm has good optimization potentials, and at the same time, it suffers from issues, including low exploration and slow convergence rates. With the continuous research of this algorithm, many scholars have analyzed and studied these existing problems of the algorithm and put forward some improvement methods [23]. Jagdish Chand Bansal et al. proposed an exploration equation to explore a large area of search space in view of the insufficient exploration ability and ease of falling into local optimization of the gray wolf algorithm. In addition, opposition-based learning was added to enrich the initial population diversity and improve the learning, search and optimization capabilities of the GWO [23]. Wang et al. proposed an improved gray wolf optimization algorithm with an evolutionary elimination mechanism. By adding the principle of survival of the fittest (SOF), biological evolution and differential evolution algorithm, the algorithm avoids falling into local optima, further accelerates the convergence speed of the GWO and improves the convergence accuracy [24]. Hi Jun T et al. proposed a GWO algorithm combined with a PSO algorithm, which retains the optimal position information of individuals and avoids the algorithm falling into local optima. This algorithm is used to test 18 benchmark functions and has better results than other algorithms [25]. Zhang et al. added elite opposition-based

learning and a simple method to solve the problem of poor population diversity and slow convergence speed of the GWO, which increased the population diversity of gray wolf and improved the exploration ability of gray wolf [26].

In the basic gray wolf algorithm, the initialization and evolution process of the gray wolf population is random, which produces certain blindness for the position and velocity update of the gray wolf, such as the tendency to fall into local optima and premature convergence. To solve the above problems, an improved optimization algorithm of gray wolf (IGWO) is proposed. The tent chaos is used to increase the population diversity, the Gaussian perturbation is used to expand the search range, and the cosine control factor is used to balance the global exploration and local development ability so as to avoid the basic gray wolf algorithm falling into the local optima and improve the solution accuracy. IGWO is applied to four engineering problems: pressure vessel design, spring design, welding beam design and three truss design. The results show that the cost of engineering problems solved by the IGWO algorithm is effectively reduced compared with other algorithms. The IGWO is also tested on seven unimodal, six multimodal and ten various fixed dimension multimodal functions, and the results are verified by a comparative study with SCA, MFO, PSO, BA, FPA and SSA. The validation of the proposed modification is done on a set of 23 standard benchmark test problems using the Wilcoxon rank-sum and Friedman test. The results show that the IGWO algorithm has higher convergence speed, convergence precision and robustness compared with other algorithms.

2. Gray Wolf Optimization

GWO is inspired by the hierarchy and hunting behavior of gray wolf populations. The algorithm achieves optimization by mathematically simulating the tracking, surrounding, hunting and attacking the process of gray wolf populations. The gray wolf hunting process involves three steps: social hierarchy stratification, encircling the prey and attacking the prey.

2.1. Social Hierarchy

Gray wolves are social canids at the top of the food chain and follow a strict hierarchy of social dominance. The best solution is marked as the α ; the second-best solutions are marked as the β , the third-best solutions are marked as the δ , the rest of solutions are marked as ω . Its dominant social hierarchy, as shown in Figure 1:

Figure 1. Hierarchy of wolves.

2.2. Encircling the Prey

Gray wolves encircle prey during the hunt; in order to mathematically model encircling behavior, the following equations are used:

$$X(t+1) = X_p(t) - A \cdot |C \cdot X_p(t) - X(t)| \quad (1)$$

$$A = 2a \cdot r_1 - a \quad (2)$$

$$C = 2 \cdot r_2 \quad (3)$$

$$a = 2 - 2 \frac{t}{Max_iter} \quad (4)$$

where X indicates the position vector of the gray wolf, X_p indicates the position vectors of prey, t indicates the current iteration, A and C are coefficient vectors, r_1 and r_2 are random vectors in $[0, 1]$, a is distance control parameter, and its value decreases linearly from 2 to 0 over the course of iterations, and Max_iter is the maximum iterations.

2.3. Attacking the Prey

Gray wolves have the ability to recognize the location of potential prey, and the search process is mainly carried out by the guidance of α , β , δ wolves. In each iteration, the best three wolves (α , β , δ) in the current population are retained, and then the positions of other search agents are updated according to their position information. The following formulas are proposed in this regard:

$$X_1 = X_\alpha - A_1 \cdot |C_1 \cdot X_\alpha - X| \quad (5)$$

$$X_2 = X_\beta - A_2 \cdot |C_2 \cdot X_\beta - X| \quad (6)$$

$$X_3 = X_\delta - A_3 \cdot |C_3 \cdot X_\delta - X| \quad (7)$$

$$X(t+1) = \frac{X_1(t) + X_2(t) + X_3(t)}{3} \quad (8)$$

In the above equation, X_α , X_β and X_δ are the position vectors of α , β and δ wolves, respectively, the calculations of A_1 , A_2 and A_3 are similar to A , and the calculations of C_1 , C_2 and C_3 are similar to C . $D_\alpha = C_1 \cdot X_\alpha - X$, $D_\beta = C_2 \cdot X_\beta - X$, $D_\delta = C_3 \cdot X_\delta - X$ represents the distance between the current candidate wolves and the best three wolves. As can be seen from Figure 2, the candidate solution finally falls within the random circle defined by α , β and δ . Then, under the guidance of the current best three wolves, the other candidates randomly update their positions near the prey. They start to search the prey position information in a scattered way and then concentrate on attacking the prey.

Figure 2. Position updating in the gray wolf optimization (GWO).

The flowchart of the GWO is given in Figure 3.

Figure 3. GWO algorithm optimization process.

3. Improved Gray Wolf Optimization Algorithm (IGWO)

3.1. Tent Chaos Initialization

When solving practical problems, GWO usually uses the randomly generated data as the initial population information, which will make it difficult to retain the diversity of the population and lead to poor results. However, chaotic motion has the characteristics of randomness, ergodicity and regularity [27], and using these characteristics to optimize the search can maintain the diversity of the population and improve the global searchability. In general, the random motion state obtained by the deterministic equation is called chaos, and the variable representing the chaotic state is called chaotic variable. Chaos is a common phenomenon in nonlinear systems [28].

Many scholars have introduced chaotic maps and chaotic search into the GWO. The existing chaotic maps include logistic maps, tent maps, etc. However, different chaotic maps have a great impact on the chaos optimization process. At present, most of the existing chaotic maps used in the literature contain logistic maps, and the inhomogeneity of logistic traversals will affect the optimization speed, thereby reducing the efficiency of the algorithm.

Reference [29] pointed out that tent map has better ergodicity, regularity and faster speed than a logistic map. It is also proved that the tent map has a prerequisite for optimizing the chaotic sequence of algorithms by strict mathematical reasoning. Tent map is a kind of piecewise linear mapping mathematically, which is named because its function is like tent [28]. Moreover, it is a 2D chaotic map, which is widely used in chaotic encryption systems (such as image encryption), and the chaotic spread spectrum code generation, chaos encryption system construction and implementation of the chaotic optimization algorithm are also often used.

Figure 4 shows the distribution of logistic chaotic sequence and tent chaotic sequence. It can be found from Figure 4 that the probability of value in the interval of $[0, 0.05]$ and $[0.95, 1]$ is greater than that of other sections, and the probability of tent in each feasible region is relatively uniform. The uniformity of tent chaotic sequence generated by tent chaotic map is obviously better than that of logistic chaotic map.

Figure 4. Chaotic sequence distribution.

In this paper, the tent chaotic map is used to optimize the search by using randomness, ergodicity and regularity, which can effectively maintain the diversity of the population, suppress the algorithm to fall into the local optima, and improve the global searchability.

In recent years, tent chaotic map has been applied to various algorithms, and good results have been achieved. Li et al. proposed a new image encryption scheme based on

the tent chaotic map. They used known methods to analyze the performance and security of the proposed image encryption scheme and proved the effectiveness and security of the scheme through fault security analysis [30]. Indu et al. proposed an improved tent map particle swarm optimization algorithm (ITM-CPSO), which solved the nonlinear congestion management cost problem, realized the nonlinear congestion management cost problem of the algorithm, and further reduced the deviation of the timing pulse generator output predetermined level to reduce the overall unloading and cost [31]. Gokhale SS et al. proposed a tent chaos firefly algorithm (CFA) to optimize the time coordination of the relay, and tested it in a number of systems, obtained better results [32]. Petrovic et al. added chaotic map to the initialization of the fruit fly optimization algorithm and studied 10 different chaotic maps. The statistical results showed that FOA was improved in terms of convergence speed and overall performance [33].

The tent mapping model, which generates a chaotic sequence for population initialization, and its mathematical expression is:

$$y_{i+1} = \begin{cases} y_i/\alpha, & 0 \leq y_i < \alpha \\ (1 - y_i)/(1 - \alpha), & \alpha \leq y_i \leq 1 \end{cases} \quad (9)$$

Theoretical research shows that tent chaotic map is represented by Bernoulli shift transformation, as follows [29]:

$$y_{i+1} = (2y_i) \bmod 1 \quad (10)$$

where $y_i^j \in [0, 1]$ is a chaotic variable, $i = 1, 2 \dots n$ represents the ordinal number of chaotic variables, $j = 1, 2 \dots n$ represents the population size. In this paper, $\alpha = 0.7$.

Figures 5 and 6, respectively, show the initial population of the GWO and IGWO when solving the sphere function. Among them, $n = 30$, $d = 3$, and the search space range is $[-100, 100]$. The red circle represents the global optimal point, and the blue circle represents the gray wolf population. In the initialization phase of the algorithm, gray wolves are randomly dispersed in the search space, and the more uniform the population is, the stronger the global exploration ability will be. It can be seen from Figures 5 and 6 that the tent map makes the initial population distribution more uniform, enriches the diversity of the population, reduces the probability of the algorithm falling into local optima and improves the convergence accuracy.

Figure 5. Random initialization.

Figure 6. Tent chaotic initialization.

3.2. Gaussian Perturbation

According to Equation (1), the GWO's C also plays a decisive role, Equation (3) shows that C is a random vector in $[0, 2]$, the C provides random weights for prey, which can increase ($|C| > 1$) or reduce ($|C| < 1$) distance between the gray wolves and the prey. The size of C helps GWO to display random search behavior during the optimization process so as to avoid the algorithm falling into local optima [16]. The leading wolf position in a population plays an important role in guiding the group to move towards the best solution. If the leading wolf's position falls into local optima, it is easy to stop searching and lack diversity within the group. During the movement of wolves, their leading wolves were randomly generated and unpredictable. In order to avoid the premature convergence and balance the global and local exploration capabilities, the original random generation coefficient C was changed to Gaussian perturbation, which will cause some perturbation to the leader and maintain the diversity of the population.

Figure 7 is the random value generated by the original C and Gaussian distribution. It can be clearly seen from the figure that the random number generated by Gaussian distribution has a wider range, which can show the random walk behavior of gray wolves in the GWO algorithm more accurately.

$$X(t+1) = X_p(t) - A \cdot | \text{Gaussian}(\delta) \cdot X_p(t) - X(t) | \quad (11)$$

Figure 7. Values of Gaussian perturbations.

3.3. Cosine Control Factor

In the GWO, the coefficient vector A balances the global and local search capabilities. When $|A| > 1$, the gray wolves will expand the search scope, namely global searching. When

$|A| < 1$, the gray wolves will shrink the search scope to attack their prey, namely local search. In the early-stage of optimization, gray wolf individuals should be widely distributed in the entire search space. In the late-stage of optimization, gray wolf individuals should converge to the optimal global value by using the collected information. According to Equation (2), the attenuation factor a affects the coefficient vector A , which further affects the balance between GWO's exploration and development capacity. When $a > 1$ (the exploration stage), the gray wolves will engage in search and hunting activities, and when $a < 1$ (the development stage), the gray wolves will only engage in hunting activities.

In the GWO, the convergence factor A decreases linearly with the number of iterations from 2 to 0, but the individual gray wolf does not change linearly in the process of searching prey, so the linear decline of the convergence factor A cannot fully reflect the actual optimization search process. It can be seen from Figures 8 and 9, the improved a and A can better reflect the actual optimization search process.

Figure 8. Convergence curve of parameter a .

Figure 9. Value of control parameter A .

Therefore, inspired by reference [34], the control factor a of linear change is replaced by the control factor a' of cosine change, and its expression is as follows:

$$a' = 2 * \cos\left(\frac{\pi}{2} * \frac{t}{Max_{iter}}\right) \quad (12)$$

The inertia weight factor is a very important parameter [35]. When the inertia weight is large, the algorithm has a strong global searching ability, which can expand the search range. When the inertia weight is small, the local search ability of the algorithm is strong, and it can search around the optimal solution and accelerate the convergence speed.

Combined with the cosine change of the control parameter a' above, this paper introduces a weight cosine control factor $B(t)$ that changes synchronously with a' into the position update of the GWO to further enhance the global exploration capability. With the increase of iterations, the algorithm adjustment step length becomes smaller, the global search ability gradually weakens, and the local search ability gradually strengthens. When $B(t)$ is very small, the position of the individual population is finetuned instead of approaching the origin.

$$B(t) = \cos\left(\frac{\pi}{2} * \frac{t}{Max_{iter}}\right) \quad (13)$$

$$X_1 = X_\alpha - B(t) \cdot A_1 \cdot |C_1 \cdot X_\alpha - X| \quad (14)$$

$$X_2 = X_\beta - B(t) \cdot A_2 \cdot |C_2 \cdot X_\beta - X| \quad (15)$$

$$X_3 = X_\delta - B(t) \cdot A_3 \cdot |C_3 \cdot X_\gamma - X| \quad (16)$$

4. The Simulation Results

4.1. Optimization Function and Experimental Environment

In this section, to analyze the performance of the IGWO algorithm, 7 unimodal test functions, 6 multimodal optimization functions and 10 fixed dimensional multimodal optimization functions in reference [15] are selected. Tables 1–3 show the names, specific expressions, search space and optimal values of these functions. Among them, F1–F7 are the unimodal optimization functions, F8–F13 are the multimodal optimization functions, F14–F23 are the fixed dimensional multimodal functions. The unimodal test function is mainly used to estimate the solution accuracy and convergence speed of the IGWO, and the multimodal test function is mainly used to estimate the global surveying ability of the IGWO.

Table 1. Unimodal optimization function.

Number	Name	Benchmark	Dim	Range	f_{min}
F1	Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
F2	Schwefel'problem2.22	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
F3	Schwefel'problem1.2	$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
F4	Schwefel'problem2.21	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	$[-100, 100]$	0
F5	Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
F6	Step	$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	$[-100, 100]$	0
F7	Noise	$f_7(x) = \sum_{i=1}^n ix_i^4 + random[0, 1)$	30	$[-1.28, 1.28]$	0

Table 2. Multimodal optimization function.

Number	Name	Benchmark	Dim	Range	f_{min}
F8	Generalized Schwefel's problem	$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	-12,569.5
F9	Rastrigin	$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i + 10)]$	30	$[-5.12, 5.12]$	0
F10	Ackley	$f_{10}(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$	30	$[-32, 32]$	0
F11	Griewank	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
F12	Generalized penalized function 1	$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50, 50]$	0
F13	Generalized Penalized Function 2	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

Table 3. Fixed dimensional multimodal optimization function.

Number	Name	Benchmark	Dim	Range	f_{min}
F14	Shekel's foxholes function	$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^2} \right)^{-1}$	2	$[-65.536, 65.536]$	1
F15	Kowalik's Function	$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.0003
F16	Six-hump camelback	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	−1.0316
F17	Branin	$f_{17} = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	$[-5, 10] [10, 15]$	0.39788
F18	Goldstein-Price function	$f_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \times$ $\left[30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2^2 - 36x_1x_2 + 27x_2^2) \right]$	2	$[-2, 2]$	3
F19	Hartmann 1	$f_{19}(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$	3	$[0, 1]$	−3.86
F20	Hartmann 2	$f_{20}(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$	6	$[1, 6]$	−3.32
F21	Shekel 1	$f_{21}(x) = - \sum_{i=1}^5 \left[(X - a_i)(X - a_i)^T + C_i \right]^{-1}$	4	$[0, 10]$	−10.1532
F22	Shekel 2	$f_{22}(x) = - \sum_{i=1}^7 \left[(X - a_i)(X - a_i)^T + C_i \right]^{-1}$	4	$[0, 10]$	−10.4028
F23	Shekel 3	$f_{23}(x) = - \sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + C_i \right]^{-1}$	4	$[0, 10]$	−10.5363

In order to improve the accuracy of the experiment, the 8 selected algorithms all adopt the same experimental parameters, namely, the swarm size $n = 30$, dimension $d = 30$, the maximum iterations $\text{Max_iter} = 500$, each algorithm is run 30 times independently and the results recorded. In, which $C1 = C2 = 2$, $W = 0.7294$ in PSO, $A = 0.5$, $r = 0.5$ in BA and $P = 0.2$ in FPA. In the simulation experiment, the hardware configuration of the simulation environment is Windows 10 Professional operating system; the CPU is Intel® Celeron® CPU N3060 @1.60 GHz processor, 4 GB memory, and the software configuration of the simulation environment is MATLABR2016a.

4.2. Analysis of Different Strategies

Different improvement strategies have different effects on the optimization results, and their contributions to the whole improved algorithm are also different. In order to analyze the effectiveness of the improved strategy of the IGWO algorithm, three strategies of Tent map, Gaussian distribution and cosine control factor are combined with the basic GWO, respectively, to study the influence of each strategy on the effectiveness of the algorithm. GWO1 means the combination of the GWO algorithm and tent map, GWO2 means the combination of the GWO algorithm and Gaussian distribution, and GWO3 means the combination of the GWO algorithm and cosine control factor. The test results are shown in Table 4 below.

From Table 4, it can be seen that different strategies can improve the performance of the GWO algorithm, and GWO2 plays a greater role in improving the performance of the GWO algorithm. Overall, the combination of different strategies makes the performance of the IGWO algorithm greatly improved.

4.3. Analysis of Experimental Results

4.3.1. Compared with Other Algorithms

To further evaluate the performance of the IGWO, the IGWO algorithm was tested on 23 benchmark functions, and the results were compared with the GWO, SCA, MFO, PSO, BA, FPA and SSA 7 algorithms. The average values and standard deviation of each algorithm running 30 times are used as the evaluation criteria, and the solution with the highest accuracy is bolded in the table. The experimental test results are shown in Tables 5–7.

Table 4. Test the results of different strategies.

Function	GWO		GWO1		GWO2		GWO3		IGWO	
	ave	std	ave	std	ave	std	ave	std	ave	std
F1	1.55×10^{-27}	2.95×10^{-27}	1.18×10^{-30}	2.15×10^{-30}	1.72×10^{-38}	6.97×10^{-38}	3.25×10^{-28}	4.78×10^{-28}	8.34×10^{-40}	2.35×10^{-39}
F2	9.89×10^{-17}	9.40×10^{-17}	8.14×10^{-17}	4.84×10^{-18}	9.97×10^{-19}	1.92×10^{-18}	1.03×10^{-17}	6.89×10^{-18}	9.96×10^{-24}	1.23×10^{-23}
F3	2.47×10^{-5}	6.57×10^{-5}	3.03×10^{-5}	7.30×10^{-5}	3.19×10^{-10}	1.70×10^{-9}	3.74×10^{-5}	7.15×10^{-5}	5.95×10^{-8}	9.72×10^{-8}
F4	7.54×10^{-7}	1.16×10^{-6}	8.47×10^{-9}	9.94×10^{-9}	5.21×10^{-15}	1.04×10^{-14}	7.45×10^{-8}	5.55×10^{-7}	1.78×10^{-11}	3.00×10^{-11}
F5	2.73×10^1	7.53×10^{-1}	2.98×10^2	2.46×10^{-1}	2.98×10^2	2.39×10^{-1}	2.98×10^2	2.09×10^{-1}	2.72×10^1	7.30×10^{-1}
F6	7.37×10^{-1}	3.06×10^{-1}	7.26×10^{-1}	3.53×10^{-1}	1.72	5.35×10^{-1}	6.56×10^{-1}	3.95×10^{-1}	1.68	4.28×10^{-1}
F7	2.40×10^{-3}	1.40×10^{-3}	2.20×10^{-3}	9.42×10^{-4}	7.55×10^{-4}	4.52×10^{-4}	2.00×10^{-3}	1.60×10^{-3}	8.87×10^{-4}	5.52×10^{-4}
F8	1.84×10^{-14}	8.25×10^2	-6.09×10^3	6.96×10^2	-4.44×10^3	1.37×10^3	-5.98×10^3	9.15×10^2	-4.64×10^3	1.26×10^3
F9	2.41	3.30	1.99	3.53	0	0	3.51×10^{-2}	1.40×10^{-1}	0	0
F10	1.03×10^{-13}	1.84×10^{-14}	1.06×10^{-15}	2.26×10^{-14}	9.30×10^{-15}	2.38×10^{-15}	1.66×10^{-13}	5.28×10^{-14}	1.88×10^{-14}	4.01×10^{-15}
F11	3.20×10^{-3}	6.70×10^{-3}	1.20×10^{-4}	5.00×10^{-3}	0	0	3.50×10^{-6}	9.40×10^{-6}	0	0
F12	4.59×10^{-2}	2.13×10^{-2}	5.43×10^{-2}	3.12×10^{-2}	1.11×10^{-1}	4.30×10^{-2}	3.84×10^{-2}	2.28×10^{-2}	1.02×10^{-1}	3.95×10^{-2}
F13	6.41×10^{-1}	2.37×10^{-1}	6.48×10^{-1}	2.23×10^{-1}	1.10	2.32×10^{-1}	5.21×10^{-1}	1.92×10^{-1}	1.05	2.24×10^{-1}
F14	4.55	4.36	5.02	4.11	4.98	4.23	5.04	4.49	5.08	4.31
F15	6.50×10^{-3}	9.30×10^{-3}	4.40×10^{-3}	8.10×10^{-3}	5.19×10^{-4}	1.86×10^{-4}	5.10×10^{-3}	8.60×10^{-3}	4.94×10^{-4}	1.22×10^{-4}
F16	−1.03	4.65×10^{-8}	−1.03	1.69×10^{-8}	−1.03	1.04×10^{-5}	−1.03	1.86×10^{-11}	−1.03	4.84×10^{-8}
F17	3.98×10^{-1}	3.88×10^{-6}	3.98×10^{-1}	1.63×10^{-6}	3.98×10^{-1}	1.63×10^{-5}	3.98×10^{-1}	1.16×10^{-4}	3.98×10^{-1}	3.83×10^{-4}
F18	3.00	2.99×10^{-5}	5.70	1.48×10^1	3.00	3.17×10^{-5}	3.00	4.14×10^{-5}	3.00	3.15×10^{-5}
F19	4.56	2.90×10^{-3}	−3.86	2.40×10^{-3}	−3.86	2.90×10^{-3}	−3.86	2.70×10^{-3}	−3.86	2.90×10^{-3}
F20	−3.27	7.48×10^{-2}	−3.27	6.75×10^{-2}	−3.19	8.67×10^{-2}	−3.23	8.52×10^{-2}	−3.23	9.19×10^{-2}
F21	−9.39	2.00	−8.47	2.68	−8.28	2.49	−9.81	1.29	−7.87	2.69
F22	-1.04×10^1	9.22×10^{-4}	-1.02×10^1	9.70×10^{-1}	−9.78	1.90	-1.00×10^1	1.35	−9.69	1.83
F23	-1.04×10^1	9.87×10^{-1}	-1.05×10^1	7.63×10^{-4}	-1.03×10^1	1.07	-1.05×10^1	2.36×10^{-7}	-1.02×10^1	1.37

Table 5. Simulation results of unimodal functions.

Function	Index	IGWO	GWO	SCA	MFO	PSO	BA	FPA	SSA
F1	ave	8.34×10^{-40}	1.55×10^{-27}	2.70×10^{-12}	2.34×10^3	8.76×10^{-3}	4.64	4.22×10^1	2.67×10^{-7}
	std	2.35×10^{-39}	2.95×10^{-27}	7.91×10^{-12}	5.04×10^3	1.32×10^{-2}	1.58	1.61×10^1	3.31×10^{-7}
F2	ave	9.96×10^{-24}	9.89×10^{-17}	9.26×10^{-10}	2.92×10^1	1.12×10^{-1}	1.05×10^1	8.40	2.72
	std	1.23×10^{-23}	9.40×10^{-17}	2.51×10^{-9}	1.78×10^1	1.01×10^{-1}	2.10	1.65×10^1	1.94
F3	ave	5.95×10^{-8}	2.47×10^{-5}	5.49×10^{-1}	2.03×10^4	5.61×10^{-3}	7.05	6.02×10^1	1.38×10^3
	std	9.72×10^{-8}	6.57×10^{-5}	3.00	1.02×10^4	6.75×10^{-3}	2.42	3.33×10^1	6.63×10^2
F4	ave	1.78×10^{-11}	7.54×10^{-7}	1.00×10^{-3}	6.78×10^1	7.66×10^{-2}	1.23	2.61	1.14×10^1
	std	3.00×10^{-11}	1.16×10^{-6}	2.60×10^{-3}	1.02×10^1	8.77×10^{-2}	11.36×10^{-1}	5.22×10^{-1}	3.18
F5	ave	2.72×10^1	2.73×10^1	7.36	2.68×10^6	5.38×10^{-3}	4.98×10^2	2.48×10^4	3.11×10^2
	std	7.30×10^{-1}	7.53×10^{-1}	3.81×10^{-1}	1.46×10^7	7.82×10^{-3}	1.65×10^2	2.02×10^4	4.69×10^2
F6	ave	1.68	7.37×10^{-1}	4.53×10^{-1}	1.35×10^3	8.57×10^{-3}	6.07	5.04×10^1	2.11×10^{-7}
	std	4.28×10^{-1}	3.06×10^{-1}	1.55×10^{-1}	4.37×10^3	1.44×10^{-2}	1.72	1.54×10^1	3.85×10^{-7}
F7	ave	8.87×10^{-4}	2.40×10^{-3}	2.50×10^{-3}	2.83	1.12×10^{-1}	1.52×10^2	2.99×10^3	1.74×10^{-1}
	std	5.52×10^{-4}	1.40×10^{-3}	2.00×10^{-3}	4.45	1.03×10^{-1}	3.51×10^1	2.15×10^3	7.68×10^{-2}

Table 6. Simulation results of multimodal functions.

Function	Index	IGWO	GWO	SCA	MFO	PSO	BA	FPA	SSA
F8	ave	-4.64×10^3	1.84×10^{-14}	-2.16×10^3	-8.61×10^3	-5.57×10^2	-Inf	-4.73×10^1	-7.40×10^3
	std	1.26×10^3	8.25×10^2	1.77×10^2	8.71×10^2	4.79×10^{-3}	—	8.83	6.39×10^2
F9	ave	0	2.41	5.52×10^{-1}	1.64×10^2	5.70×10^{-1}	4.22×10^1	1.90×10^2	5.43×10^1
	std	0	3.30	2.92	2.98×10^1	6.01×10^{-1}	7.28	3.08×10^1	2.24×10^1
F10	ave	1.88×10^{-14}	1.03×10^{-13}	5.50×10^{-6}	1.59×10^1	3.96	3.44	5.60	2.46
	std	4.01×10^{-15}	1.84×10^{-14}	2.76×10^{-5}	6.56	7.34	2.51×10^{-1}	6.66×10^{-1}	8.54×10^{-1}
F11	ave	0	3.20×10^{-3}	8.05×10^{-2}	3.70×10^1	3.36×10^{-3}	2.32×10^{-1}	8.64×10^{-1}	1.95×10^{-2}
	std	0	6.70×10^{-3}	1.45×10^{-1}	5.06×10^1	3.83×10^{-3}	8.29×10^{-2}	1.12×10^{-1}	1.67×10^{-2}
F12	ave	1.02×10^{-1}	4.59×10^{-2}	9.27×10^{-2}	1.55×10^2	2.27×10^{-1}	2.33×10^{-1}	2.06	6.16
	std	3.95×10^{-2}	2.13×10^{-2}	4.55×10^{-2}	4.58×10^2	5.16×10^{-1}	9.85×10^{-2}	7.12×10^{-1}	2.26
F13	ave	1.05	6.41×10^{-1}	3.00×10^{-1}	1.37×10^7	1.38×10^{-2}	2.93	8.20	1.20×10^1
	std	2.24×10^{-1}	2.37×10^{-1}	1.08×10^{-1}	7.49×10^7	1.19×10^{-2}	7.01×10^{-1}	2.79	1.54×10^1

Table 7. Simulation results of the fixed dimensional multimodal function.

Function	Index	IGWO	GWO	SCA	MFO	PSO	BA	FPA	SSA
F14	ave	5.08	4.55	1.94	3.13	9.98×10^{-1}	1.26×10^1	1.27×10^1	1.26
	std	4.310	4.36	9.97×10^{-1}	2.51	8.07×10^{-6}	3.48×10^{-1}	1.88×10^{-14}	6.35×10^{-1}
F15	ave	4.94×10^{-4}	6.50×10^{-3}	1.00×10^{-3}	1.06×10^{-3}	5.30×10^{-3}	3.70×10^{-3}	6.31×10^{-3}	2.20×10^{-3}
	std	1.22×10^{-4}	9.30×10^{-3}	3.66×10^{-4}	4.24×10^{-4}	5.47×10^{-3}	6.10×10^{-3}	1.01×10^{-2}	4.90×10^{-3}
F16	ave	-1.03	-1.03	-1.03	-1.03	1.22×10^1	-1.03	-1.03	-1.03
	std	4.84×10^{-8}	4.65×10^{-8}	4.28×10^{-5}	6.78×10^{-16}	3.29×10^1	2.11×10^{-9}	4.99×10^{-7}	1.68×10^{-14}
F17	ave	3.98×10^{-1}	3.98×10^{-1}	4.01×10^{-1}	3.98×10^{-1}	3.20	3.99×10^{-1}	3.98×10^{-1}	3.81
	std	3.83×10^{-4}	3.88×10^{-6}	3.00×10^{-3}	0	3.30	2.90×10^{-3}	3.54×10^{-10}	3.57×10^{-2}
F18	ave	3.00	3.00	3.00	3.00	8.14×10^3	1.83×10^1	6.60	3.00
	std	3.15×10^{-5}	2.99×10^{-5}	1.54×10^{-4}	2.23×10^{-15}	1.88×10^4	2.53×10^1	1.54×10^1	2.45×10^{-13}
F19	ave	-3.86	4.56	-3.85	-3.86	-1.08	-3.42	-3.86	-3.86
	std	2.90×10^{-3}	2.90×10^{-3}	3.10×10^{-3}	2.71×10^{-15}	8.23×10^{-1}	9.85×10^{-1}	2.86×10^{-7}	5.68×10^{-9}
F20	ave	-3.23	-3.27	-2.93	-3.24	-5.03×10^{-1}	-3.16	-2.93	-3.22
	std	9.19×10^{-2}	7.48×10^{-2}	2.17×10^{-1}	7.08×10^{-2}	5.34×10^{-1}	3.74×10^{-1}	2.17×10^{-1}	6.12×10^{-2}
F21	ave	-7.87	-9.39	-1.93	-7.72	-2.49×10^{-1}	-5.23	-1.02×10^1	-7.15
	std	2.69	2.00	1.57	3.12	3.82×10^{-1}	9.31×10^{-1}	3.92×10^{-4}	3.56
F22	ave	-9.69	-1.04×10^1	-4.11	-7.46	-1.52×10^{-1}	-5.09	-1.04×10^1	-8.92
	std	1.83	9.22×10^{-4}	1.66	3.49	1.20×10^{-1}	5.23×10^{-7}	5.50×10^{-3}	2.78
F23	ave	-1.02×10^1	-1.04×10^1	-3.82	-7.58	-2.45×10^{-1}	-5.30	-1.05×10^1	-8.34
	std	1.37	9.87×10^{-1}	1.59	3.74	2.23×10^{-1}	9.38×10^{-1}	3.26×10^{-3}	3.46

The average values and standard deviation in Tables 5–7 can reflect the convergence accuracy and optimization capability of the IGWO. For the 7 unimodal functions, the IGWO performs better in accuracy and standard deviation when solving F1, F2, F3, F4 and F6 functions, although the optimization accuracy does not reach the theoretical optimal value of 0. For the 7 multimodal functions, the optimization accuracy reaches the theoretical optimal value 0, and the standard deviation is also 0 when solving the F9 and F11 functions, which fully demonstrates the algorithm's solution accuracy and strong robustness. Meanwhile, compared with other optimization algorithms, the IGWO also achieves a better value when solving the F10 function. For the 10 fixed dimensional multimodal functions, F15 gets a better value than other algorithms, and the results of the remaining functions

are less different from those of the contrast algorithm. It can be seen that the IGWO has advantages in solving unimodal, multimodal and fixed dimensional multimodal functions.

4.3.2. Convergence Analysis

The convergence rate of the algorithm is an important index to measure the performance of the algorithm, and the algorithm convergence curve can directly compare the convergence performance of the algorithm. Figure 10a–w shows the convergence curves of fitness values of 8 algorithms, the IGWO, GWO, SCA, MFO, PSO, BA, FPA and SSA when solving 23 different types of test functions. The convergence curves are plotted by extracting the values generated by random running once when dimension $d = 30$.

Figure 10. Cont.

Figure 10. Evolution curve of the function ($d = 30$).

According to (a–w) in Figure 10, when solving the unimodal test function, the convergence curve of the IGWO algorithm decreases to the lower right corner as the iterations increase, and the convergence accuracy is higher than other algorithms. Among the multimodal test functions, the IGWO tests other functions except F8 have better performance. In particular, F9 and F11 can quickly find the optimal solution with fast convergence speed and high precision. For the fixed dimensional multimodal functions, most of the function curve drops in a stepped way. This is because the algorithm keeps exploring in the iterative process, jumps out of the optimal local solution and looks for the optimal global solution, and its convergence accuracy is relatively good.

In conclusion, the IGWO algorithm combining tent maps, Gaussian distribution and control factor can fully expand the search range in the early iteration, avoid the algorithm falling into local optima, and better balance the global and local search capabilities.

4.3.3. Numerical Result Test

In this section, Wilcoxon rank-sum test [36] will be carried out on the algorithm results of 30 independent runs, and the Friedman test will be carried out on the average values

in Tables 5–7. The following results were analyzed in IBM SPSS Statistics 21, and the significance level was set at 0.05. Table 8 shows the Wilcoxon rank-sum test results based on the IGWO algorithm. The bold data indicates that there is no significant difference between the comparison algorithm and the calculation results of the IGWO algorithm. Table 9 shows the rank generated by the Friedman test according to the mean values in Tables 5–7.

Table 8. Results of the p value Wilcoxon rank-sum test on the functions.

Function	IGWO vs. GWO p -Value Win	IGWO vs. SCA p -Value Win	IGWO vs. MFO p -Value Win	IGWO vs. PSO p -Value Win	IGWO vs. BA p -Value Win	IGWO vs. FPA p -Value Win	IGWO vs. SSA p -Value Win
F1	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F2	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F3	1.78×10^{-10}	3.029×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F4	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.01×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F5	4.13×10^{-2}	2.58×10^{-11}	2.58×10^{-11}	2.58×10^{-11}	8.26×10^{-6}	2.57×10^{-11}	1.26×10^{-10}
F6	2.97×10^{-11}	2.50×10^{-2}	2.70×10^{-2}	2.97×10^{-11}	2.97×10^{-11}	4.89×10^{-11}	2.97×10^{-11}
F7	7.27×10^{-6}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F8	3.01×10^{-11}	1.81×10^{-1}	3.01×10^{-11}	2.99×10^{-11}	3.01×10^{-11}	3.02×10^{-11}	1.21×10^{-10}
F9	1.19×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.20×10^{-12}	1.21×10^{-12}
F10	1.58×10^{-11}	1.68×10^{-11}	1.68×10^{-11}	5.65×10^{-13}	1.63×10^{-11}	1.68×10^{-11}	1.68×10^{-11}
F11	1.10×10^{-2}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
F12	1.06×10^{-7}	3.01×10^{-11}	3.01×10^{-11}	3.01×10^{-11}	9.12×10^{-1}	1.61×10^{-10}	3.01×10^{-11}
F13	4.50×10^{-11}	3.01×10^{-11}	3.01×10^{-11}	3.01×10^{-11}	8.14×10^{-5}	4.07×10^{-11}	1.86×10^{-6}
F14	3.26×10^{-1}	8.26×10^{-1}	6.58×10^{-1}	4.76×10^{-4}	5.12×10^{-11}	2.30×10^{-12}	1.79×10^{-4}
F15	1.84×10^{-2}	7.08×10^{-8}	7.64×10^{-8}	5.07×10^{-10}	1.10×10^{-4}	9.50×10^{-3}	8.84×10^{-7}
F16	1.06×10^{-11}	2.36×10^{-12}	1.61×10^{-1}	2.36×10^{-12}	6.50×10^{-14}	2.49×10^{-12}	1.61×10^{-1}
F17	8.44×10^{-7}	2.04×10^{-9}	8.59×10^{-7}	2.42×10^{-11}	1.20×10^{-7}	8.59×10^{-7}	3.71×10^{-7}
F18	3.79×10^{-1}	1.76×10^{-1}	1.21×10^{-12}	3.01×10^{-11}	1.21×10^{-12}	5.96×10^{-11}	1.21×10^{-12}
F19	8.00×10^{-3}	1.20×10^{-8}	1.20×10^{-12}	1.20×10^{-12}	3.71×10^{-7}	7.35×10^{-11}	1.20×10^{-12}
F20	2.23×10^{-1}	1.01×10^{-8}	1.50×10^{-3}	1.21×10^{-12}	6.25×10^{-4}	5.53×10^{-8}	1.40×10^{-2}
F21	5.11×10^{-1}	1.95×10^{-10}	2.48×10^{-1}	3.01×10^{-11}	2.55×10^{-1}	3.79×10^{-1}	1.02×10^{-1}
F22	5.07×10^{-10}	3.68×10^{-11}	5.73×10^{-2}	3.01×10^{-11}	7.73×10^{-11}	5.57×10^{-10}	6.35×10^{-2}
F23	3.01×10^{-11}	4.50×10^{-11}	1.70×10^{-1}	3.01×10^{-11}	8.82×10^{-10}	3.01×10^{-11}	1.23×10^{-9}

Table 9. The numerical results of 8 algorithms.

Function	IGWO	GWO	SCA	MFO	PSO	BA	FPA	SSA
F1	1.5	1.5	3	8	5	6	7	4
F2	1.5	1.5	3	8	4	7	6	5
F3	1	2	4	8	3	5	6	7
F4	1	2	3	8	4	5	6	7
F5	3	4	2	8	1	6	7	5
F6	5	4	3	8	2	6	7	1
F7	1	2	3	6	4	7	8	5
F8	3	7	4	1	5	—	6	2
F9	1	4	2	7	3	5	8	6
F10	1	2	3	8	6	5	7	4
F11	1	2	5	8	3	6	7	4
F12	3	1	2	8	4	5	6	7
F13	4	3	2	8	1	5	6	7
F14	6	5	3	4	1	7	8	2
F15	1	8	2	3	6	5	7	4
F16	1.5	5	5	5	8	5	5	1.5
F17	2.5	2.5	6	2.5	7	5	2.5	8
F18	3	3	3	3	8	7	6	3
F19	2	8	5	3.5	7	6	3.5	1
F20	3	1	6.5	2	8	5	6.5	4
F21	3	2	7	4	8	6	1	5
F22	3	1.5	7	5	8	6	1.5	4
F23	3	2	7	5	8	6	1	4

From Table 8, it can be concluded that the proposed IGWO significantly outperforms algorithms GWO, SCA, MFO, PSO, BA, FPA and SSA.

According to the results of Table 9, the rank means of the 8 algorithms are 2.39 (IGWO), 3.22 (GWO), 3.93 (SCA), 5.70 (MFO), 4.96 (PSO), 5.73 (BA), 5.61 (FPA), and 4.37 (SSA). The priority order is IGWO > GWO > SCA > SSA > PSO > FPA > MFO > BA. IGWO compared with SCA $p = 0.011 < 0.05$ indicates that there is a statistically significant difference between

IGWO and SCA. IGWO compared with MFO $p = 0.001 < 0.05$ indicates that there is a statistically significant difference between IGWO and MFO. IGWO compared with PSO $p = 0.002 < 0.05$ indicated that there was a statistically significant difference between IGWO and PSO. IGWO compared with BA $p = 0 < 0.05$ indicates that there is a statistically significant difference between IGWO and BA. IGWO compared with FPA was $p = 0.01 < 0.05$ indicated that there was a statistically significant difference between IGWO and FPA. IGWO compared with SSA $p = 0.005 < 0.05$ indicates that there is a statistically significant difference between IGWO and SSA.

5. Application to Solve Engineering Optimization Problem

In this section, four structural design problems with constraints in mechanical engineering are optimized to reduce the cost of engineering design and save resources. For these constraints, the penalty function method is used to construct the fitness function. For the objective problem $f(x)$, the constraint $g_i(x) \leq 0, h = 0$ is treated, as shown in Equation (17), $\tilde{f}(x)$ is a newly constructed objective function. Both β_1 and β_2 represent a positive integer large enough (such as 10×10^{12}). m and l are the numbers of inequality constraints and equality constraints. Parameters α_1 and α_2 take 2 and 1, respectively. When dealing with the objective problem, this method excludes all the solutions that do not meet the constraints from the candidate solutions. If the search solution satisfies the constraints, then the objective function is $\tilde{f}(x) = f(x)$. When solving these engineering optimization problems, the population size $n = 30$, the maximum iteration number $\text{Max_iter} = 500$, and the results in the table are the best values obtained by running 30 times independently. The bold values indicate the best one among all methods.

$$\tilde{f}(x) = f(x) + \sum_i^m \beta_1 \cdot \max(0, g_i(x))^{\alpha_1} + \sum_j^l \beta_2 \cdot |h_j(x)|^{\alpha_2} \quad (17)$$

5.1. Pressure Vessel Design Problem

Figure 11 shows that the objective of the pressure vessel design optimization problem is to minimize the sum of material cost, synthesis cost and welding cost while satisfying the decision variables of pressure tube thickness $T_s(x_1)$, head cover thickness $T_h(x_2)$, inner radius $R(x_3)$ and cylindrical length. Equation (18) gives its mathematical model.

$$\text{Min } f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1x_3$$

Subjected to:

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(x) = -x_2 + 0.0954x_3 \leq 0$$

$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}x_3^2 + 1,296,000 \leq 0$$

$$g_4(x) = x_4 - 240 \leq 0$$

$$x_1, x_2 \in [0.0625, 99 * 0.0625]$$

$$x_3, x_4 \in [10, 200] \quad (18)$$

Figure 11. Pressure vessel design problem.

In order to objectively demonstrate the performance of the IGWO, the MVO in [37], GSA algorithm in [38], PSO algorithm in [39], MSCA algorithm in [40], GA (Coello) algorithm in [41], GA (Coello and Montes) algorithm in [42], GA (Deb et al.) algorithm in [43], ES algorithm in [44], DE (Huang et al.) algorithm in [45], ACO (Kaveh et al.) algorithm in [46], IHS algorithm in [47] and WOA algorithm in [48] are selected. Table 10 suggests that the IGWO finds a design with the minimum cost for this problem.

Table 10. Comparison results of pressure vessel design problem.

Algorithm	Optimum Variables				Optimum Cost
	Ts	Th	R	L	
MVO [37]	0.8125	0.4375	42.0907382	176.738690	6060.8066
GSA [38]	1.125000	0.6250	55.988659	84.4542025	8538.8359
PSO [39]	0.812500	0.437500	42.091266	176.746500	6061.0777
MSCA [40]	0.776256	0.399600	40.325450	199.9213	5935.7161
GA (Coello) [41]	0.812500	0.4345	40.323900	200.0000	6288.7445
GA (Coello and Montes) [42]	0.812500	0.4375	42.097397	176.654050	6059.9463
GA (Deb et al.) [43]	0.937500	0.50000	48.329000	112.679000	6410.3811
ES [44]	0.812500	0.437500	42.098087	176.640518	6059.745605
DE (Huang et al.) [45]	0.8125	0.4375	42.098411	176.637690	6059.7340
ACO (Kaveh et al.) [46]	0.8125	0.4375	42.103624	176.572656	6059.0888
HIS [47]	1.125000	0.625000	58.29015	43.69268	7197.7300
MFO	0.8125	0.4375	42.098445	176.636596	6059.7143
WOA [48]	0.812500	0.437500	42.098209	176.638998	6059.7410
IGWO [21]	0.8125	0.4375	42.0984456	176.636596	6059.7143
GWO	0.7852686	0.3891504	40.67564	195.6436	5913.8838
IGWO	0.7784458	0.3854034	40.33393	199.8019	5888.6000

According to the results of Table 10, the IGWO, obtaining the optimum cost of 5888.6000, found the best feasible optimal design among all these algorithms. Therefore, it can be concluded that the IGWO is excellent in solving the pressure vessel design problem.

5.2. Spring Design Problem

This subsection is an optimized test of the spring design problem. When the spring is applied in engineering, it is necessary to minimize the gravity of the spring and reduce the waste of materials under the setting conditions of factors, such as tortuosity, shear pressure, impact frequency and outer diameter. A schematic diagram of the spring design structure is shown in Figure 12; the effect variable is the diameter of the coil d (x_1), the wire diameter D (x_2), the number of effective windings P (x_3). For this problem, the mathematical model of the objective function and constraint function is shown in Equation (19).

$$\text{Min } f(x) = x_1^2 x_2 (x_3 + 2)$$

subjected to:

$$g_1(x) = 1 - \frac{x_2^3}{71,785x_1^4} \leq 0$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12,566x_2x_1^3} + \frac{1}{5108s_1^2} - 1 \leq 0$$

$$g_3(x) = 1 - \frac{140x_1}{x_2^2x_3} \leq 0$$

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

$$x_1 \in [0.05, 2]$$

$$x_2 \in [0.25, 1.3]$$

$$x_3 \in [2, 15]$$

(19)

Figure 12. Spring design problem.

In order to objectively demonstrate the performance of the IGWO, the mathematical optimization method (Belegundu) algorithm in [49], GSA algorithm in [38], GSA algorithm in [46], SCA algorithm in [6], and MVO algorithm in [37] are selected. The comparison results are shown in Table 11.

Table 11. Comparison results of results for the spring design problem.

Algorithm	Optimum Variables			Optimum Cost
	d	N	D	
Mathematical optimization method (Belegundu) [49]	0.053396	0.3177	14.0260	0.0127303
GSA (Kaveh) [46]	0.050000	0.317312	14.22867	0.0128739
GSA (Rashedi) [38]	0.050276	0.323680	13.525410	0.0127022
SCA [6]	0.050780	0.334779	12.72269	0.127097
MVO [37]	0.05000	0.315956	14.22623	0.128169
GWO	0.05000	0.31739	14.0351	0.012699
IGWO	0.05159	0.354337	11.4301	0.012700

As can be seen from Table 11, the IGWO outperforms other comparison algorithms except for the GWO. However, the difference between IGWO and GWO is small.

5.3. Welded Beam Design Problem

In practical engineering application, the stressed part of the engineering can be described as a welded beam design problem. That is, one end of the beam wall does not produce axial direction, does not rotate and has no vertical displacement, and the other end is a free end. As shown in Figure 13, this problem is very widespread in engineering design. The mathematical model is shown in Equation (20), where the function represents the shear stress of the joint, the bending stress of the member, the buckling stress of the member and the deflection function of the end of the member. The optimization goal of the welded beam design problem is to minimize the total manufacturing cost when the constraints are met, that is, to reduce the waste of resources.

$$\text{Min } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

Subjected to:

$$g_1(x) = \tau(x) - \tau_{\max} \leq 0$$

$$g_2(x) = \sigma(x) - \sigma_{\max} \leq 0$$

$$g_3(x) = x_1 - x_4 \leq 0$$

$$g_4(x) = 0.125 - x_1 \leq 0$$

$$g_5(x) = \sigma(x) - 0.25 \leq 0$$

$$g_6(x) = P - Pc(x) \leq 0$$

$$g_7(x) = 0.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0$$

$$0.1 \leq x_1, x_4 \leq 2; 0.1 \leq x_2, x_3 \leq 10$$

$$\tau(x) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2};$$

$$\begin{aligned}
\tau_1 &= \frac{P}{\sqrt{2}x_1x_2} \\
\tau_2 &= \frac{MR}{J} \\
M &= P\left(L + \frac{x_2}{2}\right) \\
J(x) &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\} \\
R &= \sqrt{\frac{X_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}; \\
\sigma(x) &= \frac{6PL}{x_4x_3^2} \\
\delta(x) &= \frac{4PL^3}{Ex_3^3x_4} \\
P_c(x) &= \frac{4.013\sqrt{EGx_3^2x_4^6}}{L^2}\left(1 - \frac{X_3}{2L}\sqrt{\frac{E}{4G}}\right) \\
G &= \times 10^6, E = 30 \times 10^6, P = 6000, L = 14
\end{aligned} \tag{20}$$

Figure 13. Welding beam problem.

IGWO was used to solve the welding beam problem. The comparison algorithm includes NGS-WOA [50], WOA algorithm [48], RO algorithm [51], MVO algorithm [37], CPSO algorithm [52], CPSO algorithm [53], HS algorithm [54], GSA algorithm [38], GA algorithm [55], GA algorithm [56], Coello algorithm [40], and Coello and Monters algorithm [41]. The comparison results are shown in Table 12.

Table 12. Comparison results of results for welding beam problem.

Algorithm	Optimum Variables				Optimum Cost
	h	l	t	b	
NGS-WOA [50]	0.202369	3.544214	9.04821	0.205723	1.72802
WOA [48]	0.205396	33.484293	9.037426	0.206276	1.730499
RO [51]	0.203687	3.528467	9.004233	0.207241	1.735344
MVO [37]	0.20722744	3.393969312	9.018874001	0.207225774	1.7250
CPSO [52]	0.205463	3.473193	9.044502	0.205695	1.72645
CPSO [53]	0.202369	3.544214	9.048210	0.205723	1.73148
HS [54]	0.2442	6.2231	8.2915	0.2433	2.3807
GSA [38]	0.182129	3.856979	10.0000	0.202376	1.87995
GA [55]	0.1829	4.0483	9.3666	0.2059	1.82420
GA [56]	0.2489	6.1730	8.1789	0.2533	2.43312
Coello [40]	0.208800	3.420500	8.997500	0.2100	1.74831
Coello and Monters [41]	0.205986	3.471328	9.020224	0.206480	1.72822
GWO	0.20527	3.4819	9.0389	0.20583	1.7269
IGWO	0.20496	3.4872	9.0366	0.20573	1.7254

According to Table 12 statistics, the data results of each algorithm in solving the welding beam design problem; compared with the basic GWO algorithm, the solution result of the IGWO algorithm is better than that of the GWO algorithm. Judging from the results of other algorithms, the IGWO algorithm is superior to other algorithms except for MVO.

5.4. Three Truss Design Problem

The three-truss design problem is a nonlinear optimization problem with three nonlinear inequality constraints and two continuous decision variables to minimize the cost. The goal is to optimize the cross-sectional areas of rod 1 and 3, x_1 , and the cross-sectional areas of rod 2, x_2 . The mathematical model is shown in Equation (21). The schematic diagram is shown in Figure 14.

$$\begin{aligned} \text{Min } f(x) &= \sqrt{2}\sqrt{2}x_1 + x_2) \times l \\ \text{Subjected to:} \\ g_1(x) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}p - \delta \leq 0 \\ g_2(x) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}p - \delta \leq 0 \\ g_3(x) &= \frac{1}{\sqrt{2}x_2 + x_1}p - \delta \leq 0 \\ 0 \leq x_1, x_2 \leq 1, l &= 100 \text{ cm}, p = 2 \text{ kN/cm}, \delta = 2 \text{ kN/cm}^2 \end{aligned} \quad (21)$$

Figure 14. The three truss design problem.

In order to objectively demonstrate the performance of the IGWO, the PSO-DE in [39], MBA algorithm in [57], DEDS algorithm in [58] CS algorithm in [59], Ray and Sain algorithm in [60], and Tsa algorithm in [61] are selected. Table 13 shows the comparison of solution results between the IGWO algorithm and other algorithms.

Table 13. Comparison results of results for the three truss design problem.

Algorithm	Optimum Variables		Optimum Cost
	X1	X2	
PSO-DE [39]	0.7886751	0.4082482	263.8958433
MBA [57]	0.7885650	0.4085597	263.8958522
DEDS [58]	0.78867513	0.40824828	263.8958434
CS [59]	0.78867	0.40902	263.9716
Ray and Sain [60]	0.795	0.395	264.3
Tsa [61]	0.788	0.408	263.68
WOA [48]	0.789050544	0.407187512	263.8959474
MFO	0.788244770931922	0.409466905784741	263.895979682
GWO	0.78769	0.41108	263.9011
IGWO	0.78846	0.40884	263.8959

It can be seen from Table 13 that the solution result of the IGWO algorithm is slightly inferior to the Tsa algorithm. However, compared with other algorithms, the IGWO algorithm still has some advantages. Therefore, the IGWO algorithm has certain practicability in the engineering design of three truss structures.

6. Conclusions and Future Work

To save materials and cost in the process of engineering construction, an improved gray wolf optimization algorithm termed IGWO is proposed. It combines tent chaos, Gaussian perturbation and cosine control factor to achieve a better promote the conservation and rational use of resources, alleviate the contradiction between resource supply and socioeconomic development. Twenty-three functions with different characteristics of uni-modal, multimodal and fixed dimensional multimodal with different characteristics are tested, and the results are compared with seven algorithms. The experimental results show that the IGWO optimization performance and stability of the algorithm are superior to the other seven different types of comparable algorithms. Then four challenging engineering constraint optimization problems with different objective functions, constraint conditions and properties, including pressure vessel design, spring design, welded beam design and truss design, are solved. Meanwhile, the Wilcoxon rank-sum test and Friedman were used to evaluate the results of the IGWO algorithm. The experimental results show that the IGWO is more competitive than other comparison algorithms and can be an effective tool to solve engineering design problems and save resources.

For the study of this paper, the results show that the IGWO was successfully implemented to solve the practical constraints of engineering design problems. Additionally, the IGWO is able to provide very competitive results in terms of minimizing total cost. It was observed that the IGWO has the ability to converge to a better quality near-optimal solution and possesses better convergence characteristics than other prevailing techniques reported in the literature. It is also clear from the results obtained by different functions that the IGWO shows a good balance between exploration and exploitation that results in high local optima avoidance. It can be concluded that the IGWO is an effective and efficient algorithm.

However, the IGWO algorithm still has some shortcomings in some aspects, for example, it cannot be effectively applied to all problems, and the convergence accuracy can be further improved. Therefore, research on the IGWO still has much room for improvement. Our future work is divided into two parts: (1) to apply the proposed method to more practical cases to exert its full effective merits, especially in the areas of engineering optimization control; and (2) to reconstruct the proposed method as a fusion procedure with other metaheuristics to further investigate its performance thoroughly.

Author Contributions: J.L.: methodology. X.L.: resources, writing—original draft. Y.L.: writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by the National Natural Science Foundation of China (No. 71601071), the Science & Technology Program of Henan Province, China (No. 182102310886 and 162102110109), and an MOE Youth Foundation Project of Hu-manities and Social Sciences (No. 15YJC630079).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Acknowledgments: This study is supported by the National Natural Science Foundation of China (No. 71601071), the Science and Technology Program of Henan Province, China (No. 182102310886 and 162102110109), and a MOE Youth Foundation Project of Humanities and Social Sciences (No. 15YJC630079).

Conflicts of Interest: There are no conflicts to declare.

References

1. Begum, R.A.; Siwar, C.; Pereira, J.J.; Jaafar, A.H. A benefit–cost analysis on the economic feasibility of construction waste minimisation: The case of Malaysia. *Resour. Conserv. Recycl.* **2006**, *48*, 86–98. [[CrossRef](#)]
2. Huang, J.; Zhu, Y.; Kelly, J.T.; Jang, C.; Wang, S.; Xing, J.; Yu, L. Large-scale optimization of multi-pollutant control strategies in the Pearl River Delta region of China using a genetic algorithm in machine learning. *Sci. Total Environ.* **2020**, *722*, 137701. [[CrossRef](#)] [[PubMed](#)]
3. Guo-Chu, C.; Jin-Shou, Y.U. Particle swarm optimization algorithm. *Inf. Control* **2005**, *186*, 454–458.

4. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
5. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
6. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [CrossRef]
7. Gandomi, A.H.; Yang, X.-S. Chaotic bat algorithm. *J. Comput. Sci.* **2014**, *5*, 224–232. [CrossRef]
8. Yang, X.-S. Flower Pollination Algorithm for Global Optimization. In *International Conference on Unconventional Computing and Natural Computation*; Springer: Berlin, Heidelberg, 2012; Volume 7445, pp. 240–249.
9. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
10. Afshar, M.H. Extension of the constrained particle swarm optimization algorithm to optimal operation of mul-ti-reservoirs system. *Int. J. Electr. Power Energy Syst.* **2013**, *51*, 71–81. [CrossRef]
11. Kim, N.S.; Janic, M.; Van Wee, B. Trade-off between carbon dioxide emissions and logistics costs based on multi-objective optimization. *Transp. Res. Rec.* **2009**, *2139*, 107–116. [CrossRef]
12. Lin, W.; Yu, D.Y.; Zhang, C.; Liu, X.; Zhang, S.; Tian, Y.; Xie, Z. A multi-objective teaching–learning-based opti-mization algorithm to scheduling in turning processes for minimizing makespan and carbon footprint. *J. Clean. Prod.* **2015**, *101*, 337–347. [CrossRef]
13. He, Y.; Li, Y.; Wu, T.; Sutherland, J.W. An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops. *J. Clean. Prod.* **2015**, *87*, 245–254. [CrossRef]
14. Du, B.; Zhang, J.F.; Gao, Z.H.; Li, T.; Huang, Z.Q.; Zhang, N. Based on simulated annealing particle swarm algorithm of optimal allocation of water resources research. *J. Drain. Irrig. Mach. Eng.* **2020**, 1–10. Available online: <http://kns.cnki.net/kcms/detail/32.1814.th.20200927.0952.002.html> (accessed on 1 February 2021).
15. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
16. Maharana, D.; Kotecha, P. Optimization of Job Shop Scheduling Problem with Grey Wolf Optimizer and JAYA Al-gorithm. In *Smart Innovations in Communication and Computational Sciences*; Springer: Singapore, Singapore, 2019; Volume 669, pp. 47–58.
17. Precup, R.-E.; David, R.-C.; Petriu, E.M. Grey Wolf Optimizer Algorithm-Based Tuning of Fuzzy Control Systems With Reduced Parametric Sensitivity. *IEEE Trans. Ind. Electron.* **2016**, *64*, 527–534. [CrossRef]
18. Sharma, Y.; Saikia, L.C. Automatic generation control of a multi-area ST—Thermal power system using Grey Wolf Optimizer algorithm based classical controllers. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 853–862. [CrossRef]
19. Shakarami, M.; Davoudkhani, I.F. Wide-area power system stabilizer design based on Grey Wolf Optimization algorithm considering the time delay. *Electr. Power Syst. Res.* **2016**, *133*, 149–159. [CrossRef]
20. Yao, X.; Li, Z.; Liu, L.; Cheng, X. Multi-Threshold Image Segmentation Based on Improved Grey Wolf Optimization Algorithm. *IOP Conf. Series: Earth Environ. Sci.* **2019**, *252*, 042105. [CrossRef]
21. Yang, J.C.; Long, W. Improved Grey Wolf Optimization Algorithm for Constrained Mechanical Design Problems. *Appl. Mech. Mater.* **2016**, *851*, 553–558. [CrossRef]
22. Chandar, S.K. Grey Wolf optimization-Elman neural network model for stock price prediction. *Soft Comput.* **2021**, *25*, 649–658. [CrossRef]
23. Bansal, J.C.; Singh, S. A better exploration strategy in Grey Wolf Optimizer. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 1099–1118. [CrossRef]
24. Wang, J.-S.; Li, S.-X. An Improved Grey Wolf Optimizer Based on Differential Evolution and Elimination Mechanism. *Sci. Rep.* **2019**, *9*, 1–21. [CrossRef] [PubMed]
25. Teng, Z.-J.; Lv, J.-L.; Guo, L.-W. An improved hybrid grey wolf optimization algorithm. *Soft Comput.* **2019**, *23*, 6617–6631. [CrossRef]
26. Zhang, S.; Luo, Q.; Zhou, Y. Hybrid Grey Wolf Optimizer Using Elite Opposition-Based Learning Strategy and Simplex Method. *Int. J. Comput. Intell. Appl.* **2017**, *16*, 1750012. [CrossRef]
27. Tian, D.P. Particle Swarm Optimization Based on Tent Chaotic Sequences. *Comput. Eng.* **2010**, *4*, 180–182.
28. Shan, L.; Qiang, H.; Li, J.; Wang, Z. Chaos optimization algorithm based on Tent mapping. *Control Decis.* **2005**, *2*, 179–182.
29. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, Perth, WA, Australia, 27 November–1 December 1995.
30. Li, C.; Luo, G.; Qin, K.; Li, C. An image encryption scheme based on chaotic tent map. *Nonlinear Dyn.* **2017**, *87*, 127–133. [CrossRef]
31. Batra, I.; Ghosh, S. An Improved Tent Map-Adaptive Chaotic Particle Swarm Optimization (ITM-CPSO)-Based Novel Approach Toward Security Constraint Optimal Congestion Management. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2018**, *42*, 261–289. [CrossRef]
32. Gokhale, S.; Kale, V. An application of a tent map initiated Chaotic Firefly algorithm for optimal overcurrent relay coordination. *Int. J. Electr. Power Energy Syst.* **2016**, *78*, 336–342. [CrossRef]
33. Mitić, M.; Vuković, N.; Petrović, M.; Miljković, Z. Chaotic fruit fly optimization algorithm. *Knowl. Based Syst.* **2015**, *89*, 446–458. [CrossRef]
34. Huang, Q.; Li, J.; Song, C.; Xu, C.; Lin, X. A whale optimization algorithm based on cosine control factor and polynomial variation. *Control Decis.* **2020**, *35*, 50–59.
35. Chatterjee, A.; Siarry, P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Comput. Oper. Res.* **2006**, *33*, 859–871. [CrossRef]

36. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [\[CrossRef\]](#)
37. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [\[CrossRef\]](#)
38. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [\[CrossRef\]](#)
39. Chen, H.; Wang, M.; Zhao, X. A multi-strategy enhanced sine cosine algorithm for global optimization and con-strained practical engineering problems. *Appl. Math. Comput.* **2020**, *369*, 124872.
40. Coello, C.A.C.; Montes, E.M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv. Eng. Inform.* **2002**, *16*, 193–203. [\[CrossRef\]](#)
41. Coello, C.A.C. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput. Ind.* **2000**, *41*, 113–127. [\[CrossRef\]](#)
42. Deb, K. GeneAS: A Robust Optimal Design Technique for Mechanical Component Design. In *Evolutionary Algorithms in Engineering Applications*; Springer: Berlin, Heidelberg, 1997; pp. 497–514.
43. Mezura-Montes, E.; Coello, C.A.C. An empirical study about the usefulness of evolution strategies to solve con-strained optimization problems. *Int. J. Gen. Syst.* **2008**, *37*, 443–473. [\[CrossRef\]](#)
44. Huang, F.-Z.; Wang, L.; He, Q. An effective co-evolutionary differential evolution for constrained optimization. *Appl. Math. Comput.* **2007**, *186*, 340–356. [\[CrossRef\]](#)
45. Kaveh, A.; Talatahari, S. An improved ant colony optimization for constrained engineering design problems. *Eng. Comput.* **2010**, *27*, 155–182. [\[CrossRef\]](#)
46. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [\[CrossRef\]](#)
47. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
48. Belegundu, A.D.; Arora, J.S. A study of mathematical programming methods for structural optimization. Part I: Theory. *Int. J. Numer. Methods Eng.* **1985**, *21*, 1583–1599. [\[CrossRef\]](#)
49. Zhang, J.; Wang, J.S. Improved Whale Optimization Algorithm Based on Nonlinear Adaptive Weight and Golden Sine Operator. *IEEE Access* **2020**, *8*, 77013–77048. [\[CrossRef\]](#)
50. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray Optimization. *Comput. Struct.* **2012**, *112*–113, 283–294. [\[CrossRef\]](#)
51. Krohling, R.A.; Coelho, L.D.S. Coevolutionary Particle Swarm Optimization Using Gaussian Distribution for Solving Constrained Optimization Problems. *IEEE Trans. Syst. Man, Cybern. Part B* **2006**, *36*, 1407–1416. [\[CrossRef\]](#)
52. He, Q.; Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intell.* **2007**, *20*, 89–99. [\[CrossRef\]](#)
53. Sun, W.-Z.; Wang, J.-S.; Wei, X. An Improved Whale Optimization Algorithm Based on Different Searching Paths and Perceptual Disturbance. *Symmetry* **2018**, *10*, 210. [\[CrossRef\]](#)
54. Coello, C.A.C. Constraint-handling using an evolutionary multiobjective optimization technique. *Civ. Eng. Environ. Syst.* **2000**, *17*, 319–346. [\[CrossRef\]](#)
55. Deb, K. An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **2000**, *186*, 311–338. [\[CrossRef\]](#)
56. Coello, C.A.C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Comput. Methods Appl. Mech. Eng.* **2002**, *191*, 1245–1287. [\[CrossRef\]](#)
57. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [\[CrossRef\]](#)
58. Zhang, M.; Luo, W.; Wang, X. Differential evolution with dynamic stochastic selection for constrained optimization. *Inf. Sci.* **2008**, *178*, 3043–3074. [\[CrossRef\]](#)
59. Gandomi, A.H.; Yang, X.-S.; Alavi, A.H. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **2013**, *29*, 17–35. [\[CrossRef\]](#)
60. Ray, T.; Saini, P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng. Optim.* **2001**, *33*, 735–748. [\[CrossRef\]](#)
61. Tsai, J.-F. Global optimization of nonlinear fractional programming problems in engineering design. *Eng. Optim.* **2005**, *37*, 399–409. [\[CrossRef\]](#)