

Article

Modified Artificial Gorilla Troop Optimization Algorithm for Solving Constrained Engineering Optimization Problems

Jinhua You ¹, Heming Jia ^{1,*} , Di Wu ^{2,*}, Honghua Rao ¹, Changsheng Wen ¹ , Qingxin Liu ³  and Laith Abualigah ^{4,5,6,7,8} ¹ School of Information Engineering, Sanming University, Sanming 365004, China² School of Education and Music, Sanming University, Sanming 365004, China³ School of Computer Science and Technology, Hainan University, Haikou 570228, China⁴ Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al al-Bayt University, Mafraq 25113, Jordan⁵ Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan⁶ Faculty of Information Technology, Middle East University, Amman 11831, Jordan⁷ Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan⁸ School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia

* Correspondence: jiaheming@fjismu.edu.cn (H.J.); wudi@fjismu.edu.cn (D.W.)

Abstract: The artificial Gorilla Troop Optimization (GTO) algorithm (GTO) is a metaheuristic optimization algorithm that simulates the social life of gorillas. This paper proposes three innovative strategies considering the GTO algorithm's insufficient convergence accuracy and low convergence speed. First, a shrinkage control factor fusion strategy is proposed to expand the search space and reduce search blindness by strengthening the communication between silverback gorillas and other gorillas to improve global optimization performance. Second, a sine cosine interaction fusion strategy based on closeness is proposed to stabilize the performance of silverback gorillas and other gorilla individuals and improve the convergence ability and speed of the algorithm. Finally, a gorilla individual difference identification strategy is proposed to reduce the difference between gorilla and silverback gorillas to improve the quality of the optimal solution. In order to verify the optimization effect of the modified artificial gorilla troop optimization (MGTO) algorithm, we used 23 classic benchmark functions, 30 CEC2014 benchmark functions, and 10 CEC2020 benchmark functions to test the performance of the proposed MGTO algorithm. In this study, we used a total of 63 functions for algorithm comparison. At the same time, we carried out the exploitation and exploration balance experiment of 30 CEC2014 and 10 CEC2020 functions for the MGTO algorithm. In addition, the MGTO algorithm was also applied to test seven practical engineering problems, and it achieved good results.

Keywords: artificial gorilla troop optimization algorithm; convergence strategy of contraction control factors; sine cosine interaction fusion strategy; identification strategies of individual differences in gorillas

MSC: 49K35

Citation: You, J.; Jia, H.; Wu, D.; Rao, H.; Wen, C.; Liu, Q.; Abualigah, L. Modified Artificial Gorilla Troop Optimization Algorithm for Solving Constrained Engineering Optimization Problems. *Mathematics* **2023**, *11*, 1256. <https://doi.org/10.3390/math11051256>

Academic Editor: Andrea Scozzari

Received: 2 February 2023

Revised: 20 February 2023

Accepted: 2 March 2023

Published: 5 March 2023



Copyright: © 2023 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

Metaheuristic Algorithms (MAs) have developed rapidly in recent decades, attracting the attention of scholars in many fields. They do not rely on gradient information but are inspired by nature, such as group behavior, social behavior, physics, etc. Its core idea is that several simple individuals form a group and show advanced and complex functions through cooperation, competition, interaction, and learning mechanisms. Therefore, in the process of solving optimization problems, it is not required to have continuity, derivative, and other conditions. Without local information and models, complex problems can still be

solved. With the progress of time, science and technology are also developing rapidly, but there are still many complex optimization problems that are difficult to describe and solve. In the face of these problems, traditional optimization methods require more time and cost. At the same time, with the change in application scenarios, the solutions are different. Therefore, it is necessary to improve the original algorithm to improve its performance in solving complex problems.

In recent decades, various metaheuristic algorithms were proposed. These metaheuristic algorithms are generally divided into three categories: the physical-based, the evolution-based, and the swarm-based algorithms. Although these metaheuristic algorithms are different, they all have advantages. The algorithms represented by physics-based algorithms include the Simulated Annealing Algorithm (SA) [1], the Equilibrium Optimizer Algorithm (EO) [2], the Multi-Verse Optimizer Algorithm (MVO) [3], the Arithmetic Optimization Algorithm (AOA) [4], the Ray Optimization Algorithm (RO) [5], the Sine Cosine Algorithm (SCA) [6], and the Thermal Exchange Optimization Algorithm (TEO) [7]. Some famous algorithms based on evolutionary algorithms include the Genetic Algorithm (SPGA) [8], the Evolutionary Strategy (ES) algorithm [9], the Genetic Programming (GP) algorithm [10], the Evolutionary Deduction (ED) algorithm [11], and the Differential Evolution (DE) algorithm [12]. As we all know, the swarm-based algorithm is common, and its inspiration comes from the living habits of animals. Representative algorithms include the Particle Swarm Optimization Algorithm (PSO) [13], the Gray Wolf Optimizer Algorithm (GWO) [14], the Artificial Bee Colony Optimization Algorithm (ABC) [15], the Remora Optimization Algorithm (ROA) [16], the Seagull Optimization Algorithm (SOA) [17], the Harris Hawk Optimization Algorithm (HHO) [18], the Whale Optimization Algorithm (WOA) [19], the Spotted Hyena Optimizer Algorithm (SHO) [20], etc.

The artificial Gorilla Troop Optimization Algorithm (GTO) [21] is a new metaheuristic algorithm proposed by Abdollahzadeh et al. in 2021. Its inspiration comes from gorilla social behavior, such as the gorilla's migration, competition, and following behaviors. The principle is to change other gorillas' positions with the silverback gorilla, a process divided into two stages: exploitation and exploration. So far, the algorithm has been used in many fields, such as shortest-path planning problems, engineering optimization problems, robot parking problems, etc. Jayashree Piri [22] and other scholars proposed an excellent gorilla initialization strategy based on mutual tag information. For the first time, a new discrete artificial gorilla force optimization technology was introduced to deal with FS tasks in the healthcare sector. Mahmoud A. El-Dabah [23] and other scholars proposed a method to optimize the stabilizer unit of a power system by combining GTO optimization algorithms. Hadel Alsolai [24] introduced a clustering protocol based on the Enhanced Artificial Gorilla Troops Optimizer-based Clustering Protocol for a UAV-Assisted Intelligent Vehicular Network (EAGTOC-UIVN) to achieve maximum life and energy efficiency from the vehicle network. Abdullah Shaheen [25] developed Gorilla Troops T technology for power system optimal power flow based on the gorilla force optimization algorithm. However, due to strong randomness in the optimization process, the GTO algorithm, like other swarm intelligence algorithms, has the problem of imbalance between exploration and exploitation, which leads to problems such as that the algorithm is easy to fall into the optimal local solution, and it has low convergence accuracy and slow convergence speed. In addition, the No Free Lunch (NFL) theorem [26] shows that no algorithm can perfectly solve all optimization problems. Therefore, we improved and modified the GTO algorithm according to the NFL theorem and the defects of the GTO algorithm to solve more practical engineering problems more effectively; our study is summarized as follows:

- (1) This paper proposes a modified artificial gorilla troop optimization algorithm called the MGTO algorithm. In this paper, three innovative strategies are proposed: a shrinkage control factor fusion strategy, a sine cosine interaction fusion strategy based on closeness, and a gorilla individual difference identification strategy. First, the shrinkage control factor fusion strategy is embedded into the GTO algorithm to expand the search

space of the algorithm and reduce the blindness of the search. In addition, the sine cosine interaction fusion strategy based on closeness is used to distinguish the distance between other gorilla individuals and silverback gorillas and stabilize their performance. Finally, the gorilla individual difference identification strategy is used to narrow the differences between the silverback gorilla and other gorillas to improve the quality of the optimal solution.

(2) To verify the MGTO algorithm's effectiveness, we used 23 classic benchmark functions, 30 CEC2014 benchmark functions, and 10 CEC2020 benchmark functions for testing. The performance of the MGTO algorithm was verified by comparing it with the traditional GTO algorithm and eight other popular optimization algorithms.

(3) At the same time, we also applied the MGTO algorithm to solve practical engineering problems. Seven engineering problems were selected for this paper: the pressure vessel design problem, the reducer design problem, the welding beam design problem, the tension/compression spring design problem, the cantilever beam design problem, a multi-plate clutch brake failure scenario, and the vehicle collision optimization problem. The final experimental results show that the MGTO algorithm has strong convergence ability and global search ability and achieves a good balance between exploitation and convergence.

The structure of this paper is as follows. The second section briefly describes the GTO algorithm, the third section introduces the three improved strategies and the framework of the improved algorithm, MGTO, the fourth section sets up and discusses the experiments and results, the fifth section applies the improved algorithm (MGTO) to engineering problems for testing, and the last section summarizes this paper.

2. Gorilla Troops Optimizer (GTO)

The artificial gorilla troop optimization algorithm is a metaheuristic algorithm based on gorilla social behavior. The gorilla is the largest primate and strongest primate in the world today. Gorillas are social animals. Each group is led by an adult male gorilla and has a strong sense of territory. Because the hair on the back of some male gorillas is white, these gorillas are also called silverback gorillas.

A gorilla colony usually consists of an adult male gorilla, several adult female gorillas, and their offspring. As leaders, adult male gorillas assume the responsibilities of defending territory, making decisions, and guiding other gorillas to find food. Male gorillas can expand their territory through competition, and competition between male gorillas and female gorillas is inevitable. Moreover, the relationship between male and female gorillas is close, while the relationship between female gorillas is cold.

2.1. Exploration Stage

In this section, the optimization process in the exploration phase is described. In the gorilla community, we know that there is a silverback gorilla who manages all decisions. Gorillas sometimes go places to look for food, which may be places they have been to or strange places. The silverback gorilla is considered the optimal candidate solution at each optimization exploration stage. This section also introduces three mechanisms that activate at this stage.

The three mechanisms in the exploration stage are expressed in Equation (1), where p is a parameter between 0 and 1 that controls the migration strategy of unknown positions. When $\text{rand} < p$, the current gorilla's position will move to an unknown position. This allows the GTO algorithm to better monitor the entire problem space, which makes the distribution of solutions more scattered and comprehensive. Conversely, if $\text{rand} \geq p$, two other mechanisms will be chosen. Then, if $\text{rand} \geq 0.5$, the gorilla will move in the direction of other gorillas. This mechanism makes the current solution closer to other solutions and improves the exploration performance of the GTO algorithm. When $\text{rand} < 0.5$, the gorillas migrate to the known position. This improved the ability of the GTO algorithm to escape from local optimal solutions. Equation (1) is as follows:

$$GX(t + 1) = \left\{ \begin{array}{l} (UB - LB) \times r_1 + LB, rand < p \\ (r_2 - C) \times X_r(t) + L \times H, rand \geq 0.5 \\ X(t) - L \times (L \times (X(t) - GX(t))) + r_3 \times (X(t) - GX_r(t)), rand < 0.5 \end{array} \right\}, rand \geq p \tag{1}$$

where $GX(t + 1)$ represents the position vector of the gorilla at the next iteration and $X(t)$ represents the current position of the gorilla. The position vector dimension here is defined by the dimension of the problem. r_1, r_2, r_3 , and $rand$ all represent random numbers between 0 and 1 produced with a uniform distribution. The variables' upper and lower bounds are expressed as UB and LB . X_r and GX_r are the position vectors of randomly selected gorillas. The equation for calculating C, L , and H in Equation (1) is as follows:

$$C = F \times \left(1 - \frac{t}{MaxIt} \right) \tag{2}$$

where t represents the current number of iterations and $MaxIt$ represents the maximum number of iterations. During the initial stage, the value of C has a large change in the early stage and a small change in the later stage. This makes the early stage have greater randomness to expand the exploration range, and in the later stage, it gets smaller to achieve the effect of rapid convergence. F is calculated as follows:

$$F = \cos(2 \times r_4) + 1 \tag{3}$$

where r_4 is a random number between 0 and 1 in a uniform distribution to ensure the randomness of the search, which is conducive to finding the global optimal solution.

L is a parameter that is used to simulate the silverback gorilla's leadership, which is calculated with the following equation:

$$L = C \times l \tag{4}$$

where l is a random number between -1 and 1 produced with uniform distribution. Silverback gorillas may make mistakes in finding food or managing groups due to a lack of experience. Therefore, they can obtain reliable experience and extreme stability under the guidance of leaders. At the same time, Equation (4) is used to simulate the silverback gorilla's leadership. Meanwhile, H in Equation (1) can be calculated with Equation (5). The Z in Equation (5) can be calculated with Equation (6), where Z is a random value in the problem dimensions and the range of $-C, C$.

$$H = Z \times X(t) \tag{5}$$

$$Z = [-C, C] \tag{6}$$

At the end of the exploration phase, the fitness values of GX and X are calculated. If the fitness value of $GX(t)$ is less than the fitness value of $X(t)$, then the position of $GX(t)$ will replace the position of $X(t)$.

2.2. Exploitation Stage

2.2.1. Following the Silverback

In the exploitation stage, two behaviors are taken: following the silverback gorilla and competing with adult female gorillas. In the gorilla group, the behavior of the silverback gorilla leading the other gorillas and competing with female gorillas are two different behaviors. Here, the value of C controls whether adult male gorillas follow the silverback gorilla or compete with other males. When C meets different conditions, the corresponding strategy will be selected. W is the parameter to be set before optimization.

Silverback gorillas and other gorillas are better able to perform their duties when young. At the same time, male gorillas prefer to follow silverback gorillas. Moreover, each gorilla can influence other gorillas. That is to say, the current individual position solution

of the gorilla will follow the optimal solution of the silverback gorilla. They will influence each other, and each solution will affect other solutions. That is, if $C \geq W$, then the policy will be executed. This behavior is simulated with Equation (7):

$$GX(t + 1) = L \times M \times (X(t) - X_{silverback}) + X(t) \tag{7}$$

where $X_{Silverback}$ represents the optimal solution, and M can be calculated with Equation (8):

$$M = \left(\left| \frac{1}{N} \sum_{i=1}^N GX_i(t) \right|^g \right)^{\frac{1}{g}} \tag{8}$$

where $GX_i(t)$ refers to the vector position of each candidate gorilla during the iteration. The position vector dimension here is defined by the dimension of the problem, N represents the total number of gorillas, and g can be calculated with Equation (9) as follows:

$$g = 2^L \tag{9}$$

2.2.2. Competition for Adult Females

One of adolescent gorillas' main behaviors is to compete with other male gorillas for the opposite sex. This kind of competition is characterized by being intense, lasting, and able to influence other members. The competition between them represents the mutual influence of solutions. The silverback gorilla's optimal solution moves towards the position of other solutions, thereby affecting the current solution to a certain extent and aiding the search for a better solution in the process. This behavior is simulated with Equation (10):

$$GX(t) = X_{silverback} - (X_{silverback} \times Q - X(t) \times Q) \times A \tag{10}$$

$$Q = 2 \times r_5 - 1 \tag{11}$$

$$A = \beta \times E \tag{12}$$

$$E = \begin{cases} N_1, rand \geq 0.5 \\ N_2, rand < 0.5 \end{cases} \tag{13}$$

where Q is the competition intensity of the simulated gorilla, r_5 is a random number between 0 and 1 in a uniform distribution, A is the coefficient vector used to simulate the degree of competition, β is the parameter set before the optimization operation, E is used to simulate the impact of violence on the solution dimension, and $Rand$ is a random number between 0 and 1. When $rand \geq 0.5$, E will be equal to the random value in the normal distribution and the problem dimension; otherwise, E will be equal to the random number in the normal distribution.

3. Modified Gorilla Troops Optimizer (MGTO)

3.1. Convergence Strategy of Contraction Control Factors

As we all know, communication between the silverback gorilla and its members is an important part of decision-making. Therefore, to enhance the algorithm's exploration ability, we propose a contraction control factor fusion strategy to simulate this link. Because the original algorithm fell into the problem of the local optimum, which is the problem of incomplete exploration of the entire space, we moved one solution to the position of another optimal solution so that the local space of the optimal solution can be fully explored and developed. In addition, we used randomness to move a solution to the exploration space that is not reached by the algorithm and made the algorithm have the ability to jump out of the local optimal solution so it can obtain a more reasonable and effective solution. Thus, we simulated the random movement process of the gorilla to better improve the quality of the solution. U simulates the degree of experience possessed by the gorillas at this stage. When gorillas are inexperienced, i.e., when $U > 1$, we introduce a contraction

factor CAN , which enables gorillas to explore more unknown spaces. There are two kinds of exploration of unknown spaces. If $|CAN| \geq 0.5$, then the behavior of gorillas to explore unknown positions is simulated completely according to their cognition. The other type of exploration is that of unknown positions after gorillas exchange experiences with one another, which is included to reduce the blindness of exploration. The parameter CAN controls how the gorillas choose between these two strategies, effectively expanding the gorilla's exploration of unknown fields and increasing the algorithm's search space. When experience is sufficient, gorillas still need to communicate with other gorillas to ensure a full, reliable, and stable pool of experience and reduce the blindness of their search. The specific update equation is as follows:

Shrinkage factor:

$$CAN = e^{1 - \frac{t}{MaxIt}} \times \cos\left(\frac{t}{2} + \frac{\pi}{4}\right) \tag{14}$$

Empirical parameters:

$$U = \frac{F_i - Silverback_Score}{Mean - Silverback_Score} \tag{15}$$

where, among them, F_i represents the fitness value of the i th gorilla, $Silverback_Score$ refers to the fitness value of the silverback gorilla, and $Mean$ refers to the average fitness value of all gorillas.

When $U > 1$, the update equation is as follows:

$$GX_i = [(UB - LB) \times (|CAN| - rand) \times rand(1, dim)] / 2 + LB, |CAN| \geq 0.5 \tag{16}$$

$$GX_i = (X_i - X_{r1}) \times D, |CAN| < 0.5 \tag{17}$$

where $rand$ refers to a random number between 0 and 1, $rand(1, dim)$ refers to a random vector with a problem dimension ranging from 0 to 1 with a uniform distribution, dim represents the dimension of the problem, X_{r1} represents a random gorilla individual, and D refers to a random vector with problem dimension generated in the interval $[-|CAN|, -|CAN|]$ with uniform distribution. The calculation equation is as follows:

$$D = unifrnd(-|CAN|, |CAN|, 1, dim) \tag{18}$$

Similarly, to increase the effectiveness of the exploration when $U \leq 1$, we fused current gorilla individuals with random gorilla individuals, multiplying empirical parameters and combining the influence of current gorilla individuals. We purposefully moved the position of the solution of the current individual to the random individual solution. The parameter U makes the range of movement large or small so the local space between the two solutions can be fully explored and a better solution can be found. This enhances the ability to explore effectively while largely avoiding blind searches. The following position updates are performed:

$$GX_i = X_{r2} + (X_i - X_{r2}) \times U + [X_i / (X_i \times rand(1, dim))] \times (1 - U) \tag{19}$$

3.2. Sine Cosine Interaction Fusion Strategy Based on Closeness

As we all know, due to the randomness of the algorithm, the solutions of the individual position of the gorilla are different. Therefore, in order to improve the exploitation ability of the original algorithm, we divided these individual solutions into good positions and bad positions, and we then improved the quality of the solution of the individual position to improve the exploitation performance of the algorithm. Thus, we divided the gorilla population according to the closeness of their relationships with the silverback gorilla, calculated the distance between the leader of the silverback gorilla and the gorilla in the gorilla population with the minimum average closeness, and normalized the calculated closeness value to the inverse cosine to obtain the Gamma. When the Gamma is greater

than 0, it means that the current relationship between gorillas and silverback gorillas is close and stable, which means that the current position of the individual gorilla is relatively good; otherwise, the relationship is cold. Gamma is calculated as follows:

$$x = 2 \times \text{sum}(\min(X_i, \text{Silverback})) \tag{20}$$

$$y = \text{sum}(X_i + \text{Silverback}) \tag{21}$$

$$C = x/y \tag{22}$$

$$\text{Gamma} = \text{atan}(C) \times 2/\pi \tag{23}$$

When the gorilla is closely related to the silverback gorilla, considering that the decision-making of the silverback gorilla is inevitably wrong sometimes, and that the silverback gorilla has a certain degree of humanization, the silverback gorilla should communicate with the group to make the decision conform to the wishes of the group and ensure the correctness of the decision. Here, the optimal individual solution is fused with the randomly generated individual position to ensure randomness and further improve exploration ability while taking into account the focus of the overall solution of the population, thereby avoiding blind search and exploitation. The specific calculation equation is shown in Equation (24). At the same time, it was also considered that individuals and silverback gorillas should fully communicate to increase their experience. This is to make the current individual solution more effective and move it to the position of the optimal individual solution so the local area of the current individual solution and the local area of the optimal individual solution can be explored to some extent, and thus, the quality of the current solution can be improved. Therefore, a sine cosine interaction equation is proposed here. Through the constant changes of sine cosine functions, individual gorillas and silverback gorillas can conduct in-depth communication in many aspects to obtain a more reliable decision-making scheme. Because the sine and cosine functions are periodic and are between $[-1, 1]$, this can make the distance between the individual solution and the optimal solution larger or smaller as the number of iterations increases; thus, the local space around them is fully developed, and the exploration of the global space is also taken into account so better location solutions can be found more comprehensively. The specific calculation equation is shown in Equation (25). The standard normal distribution random number h regulates the two aforementioned behaviors. The following dim represents the dimension of the problem. When $\text{Gamma} > 0$ at this time, the individual position is better; thus, we mainly let the area around it be comprehensively developed to find a better solution than the current one. The specific mathematical equation is as follows:

$$GX_i = (\text{Silverback} + ((UB - LB) \times \text{rand}(1, \text{dim})) + LB \times \text{rand}(1, \text{dim})) \times \text{rand}(1, \text{dim}) - P, h > 0.7 \tag{24}$$

$$GX_i = X_i + \text{mean}(\sin(3 \times t + \frac{\pi}{4}) \times X_i + \left| \sin(3 \times t + \frac{\pi}{3}) \right| \times \text{Silverback}) \times 2\text{rand}(1, \text{dim}), h \leq 0.7 \tag{25}$$

where P simulates the decision-making willingness of gorilla groups, and its calculation equation is as follows:

$$P = (\text{mean}(GX_{all}) \times \text{CAN2} - \text{mean}(X_{all})) \times (\text{rand} - 1) \tag{26}$$

where, among them, $\text{mean}(GX_{all})$ refers to the mean vector of all candidate gorilla positions, $\text{mean}(X_{all})$ refers to the mean vector of gorilla population positions, and rand is a random number between 0 and 1. The calculation equation of CAN2 is as follows:

$$\text{CAN2} = 3e^{-t/200} \times \left| \cos\left(\frac{1}{4}\left(t - \frac{3}{\pi}\right)\right) \right| \tag{27}$$

When the relationship between the two is cold, the silverback gorilla, as the leader, wants to conquer the individual gorilla. That is, considering that the current position of the

silverback gorilla may fall into the local optimal position, in order to make it jump out of the local defect, the sine and cosine function is used to migrate the position of the silverback gorilla to the individual position of other gorillas to find a better global solution. At the same time, considering that they also need to exchange experience, they also communicate through the constant changes of the sine and cosine functions to improve the quality of solutions. CAN3 regulates these two behaviors. The specific mathematical equation is as follows:

$$CAN3 = 5 \times \left(\frac{498}{500} + \left(1 - \frac{t}{MaxIt}\right) \times \cos\left(\frac{1}{3}\left(t + \frac{\pi}{4}\right)\right) - 1\right) \tag{28}$$

When $\Gamma \leq 0$,

$$GX_i = \text{mean}(X_i + Silverback) + CAN3 \times (X_i/2 - Silverback), |CAN3| > 0.8 \tag{29}$$

$$GX_i = X_i + \text{mean}\left(\cos\left(3 \times t + \frac{\pi}{4}\right) \times X_i - \left|\cos\left(3 \times t + \frac{\pi}{4}\right)\right| \times Silverback\right) \times 2rand(1, dim), |CAN3| \leq 0.8 \tag{30}$$

3.3. Identification Strategies of Individual Differences in Gorillas

There are some differences between gorillas and silverback gorillas. The solution to narrow the difference between the current solution and the optimal solution was inspired by the habits of the gorilla population. Gorillas will continue to learn from silverback gorillas and improve themselves to narrow the differences between them. Therefore, we took advantage of the differences between the current gorilla solitary solution and the optimal silverback gorilla optimal solution by having their positions update according to the magnitude of the difference. For example, when $X_{ij}/Silverback \geq 1$, it indicates that they are more different; thus, the maximum difference factor $D1$ and the influence degree b_j of the silverback gorilla’s optimal solution are used to adjust the differences between the current individual solution and individual optimal solutions. If the difference is small, that is, when $0 < X_{ij}/Silverback < 1$, the smallest difference factor $D2$ and the influence degree b_j of the silverback gorilla’s solution are used to adjust the difference between the gorilla and the silverback gorilla. These two mechanisms can further fully develop the optimal individual position and improve the convergence speed of the algorithm. At the same time, we considered that it may fall into the situation of the local optimum; thus, for other cases, we used $D1$ and $D2$ to further differentiate the current individual gorilla and the silverback gorilla, respectively, to let them get out of the area of local solution in an attempt to deviate the algorithm from the problem of the local optimum. The specific updates are as follows:

$$GX_{i,j} = (Silverback_j \times r_5 - X_{i,j} \times b_j \times D1) - \frac{D1}{3} \times d_1, X_{i,j}/Silverback_j \geq 1 \tag{31}$$

$$GX_{i,j} = (Silverback_j \times r_6 + |X_{i,j}| \times b_j \times D2) - D2 \times d_2, 1 > X_{i,j}/Silverback_j > 0 \tag{32}$$

$$GX_{i,j} = (X_{i,j} \times b_j \times D1 \times randi(-2,2) + Silverback_j) - D2 \times d_3, others \tag{33}$$

where r_5 and r_6 are random integers of -1 or 1 , d_1 , d_2 , and d_3 are uniformly distributed random values generated in the interval $[-2, 2]$, $D1$ and $D2$ are the maximum and minimum difference factors of the individual gorillas, and b is the vector of influence degree of the silverback gorilla on individual gorillas, which the following equation can calculate:

$$b = \frac{Silverback}{sum(Silverback^2)} \tag{34}$$

$$D1 = \max(|X_i| - Silverback) \tag{35}$$

$$D2 = \min(|X_i| - Silverback) \tag{36}$$

where, among them, $sum(Silverback^2)$ refers to the sum of the square of the vector positions of the silverback gorilla.

After the individual difference identification of gorillas, the fitness values of candidate gorilla GX and gorilla individual X are calculated and compared. Suppose the fitness value of GX is less than the fitness value of X . In that case, the individual position of GX will replace the individual position of X , and the position of the silverback gorilla will be updated. Then, the communication between the two can be improved through the change of sine and cosine functions to improve the corresponding exploitation convergence capability. The specific equation is as follows:

$$GX_i = \text{mean}(2 \times \cos(3 \times t + \frac{\pi}{4}) \times X_i - \cos(3 \times t + \frac{\pi}{2}) \times \text{Silverback}) \times \text{rand} \times \text{ones}(1, \text{dim}) \quad (37)$$

3.4. Implementation of the MGTO Algorithm

In the initialization phase, the MGTO algorithm randomly generates the population X_i and initializes the position of the silverback gorilla. In the exploration phase, the MGTO algorithm realizes the three mechanisms of the original algorithm, compares and updates the individuals' positions, and introduces the shrinkage control factor fusion strategy to expand the algorithm's search space and reduce its search blindness. After following the silverback gorilla and competing for adult female gorillas in the exploitation stage, the MGTO algorithm introduces the sine cosine interaction fusion strategy based on closeness and the gorilla individual difference identification strategy. After each strategy updates its position, they then update the position of the silverback gorilla in a timely manner to achieve an efficient search. Finally, the above steps are repeated until the maximum number of iterations is reached. The pseudocode of the MGTO algorithm is shown in Algorithm 1.

Algorithm 1: Pseudocode of MGTO algorithm

Initialize the population and set corresponding parameters β, w , and p , set the population size N and the maximum number of iterations T .

Calculate the fitness value of the initialized gorilla.

%Main Loop

While ($t \leq$ maximum iteration)

 Update C, L using Equations (2) and (4).

%Exploration phase

For (each gorilla(X_i)) **do**

 Use Equation (1) for position updates.

End for

 Calculate the fitness values of gorilla; if GX is better than X , replace them.

 Set $X_{\text{Silverback}}$ as the position of silverback(best position).

 Use Equations (14) and (15) to update U and CAN .

For (each gorilla(X_i)) **do**

 Use Equations (16)–(19) for position updates.

End for

 Calculate the fitness values of gorilla; if GX is better than X , replace them.

%Exploitation phase

For (each gorilla(X_i)) **do**

If ($|C| \geq 1$) **then**

 Update the position gorilla using Equation (7).

Else

 Update the position gorilla using Equation (10).

End if

End for

 Calculate the fitness values of the gorilla and replace them after comparison.

For (each gorilla(X_i)) **do**

If ($\text{Gamma} > 0$) **then**

 Update using Equations (24) and (25).

Else

Algorithm 1: Pseudocode of MGTO algorithm

```

Update using Equations (29) and (30).
End if
End for
Calculate the fitness values of the gorilla and replace them after comparison.
For (each gorilla( $X_i$ )) do
  For (1 to J dimension, J is a random integer between 1 and the total dimension) do
    Use Equations (31)–(33) for position updates.
  End for
End for
Calculate the fitness values of the gorilla and replace them after comparison.
For (each gorilla( $X_i$ )) do
  Use Equation (37) for position updates.
End for
Calculate the fitness values of the gorilla and replace them after comparison.
End while
Return  $X_{silverback}$  and its fitness value.
    
```

The corresponding flow chart is shown in Figure 1.

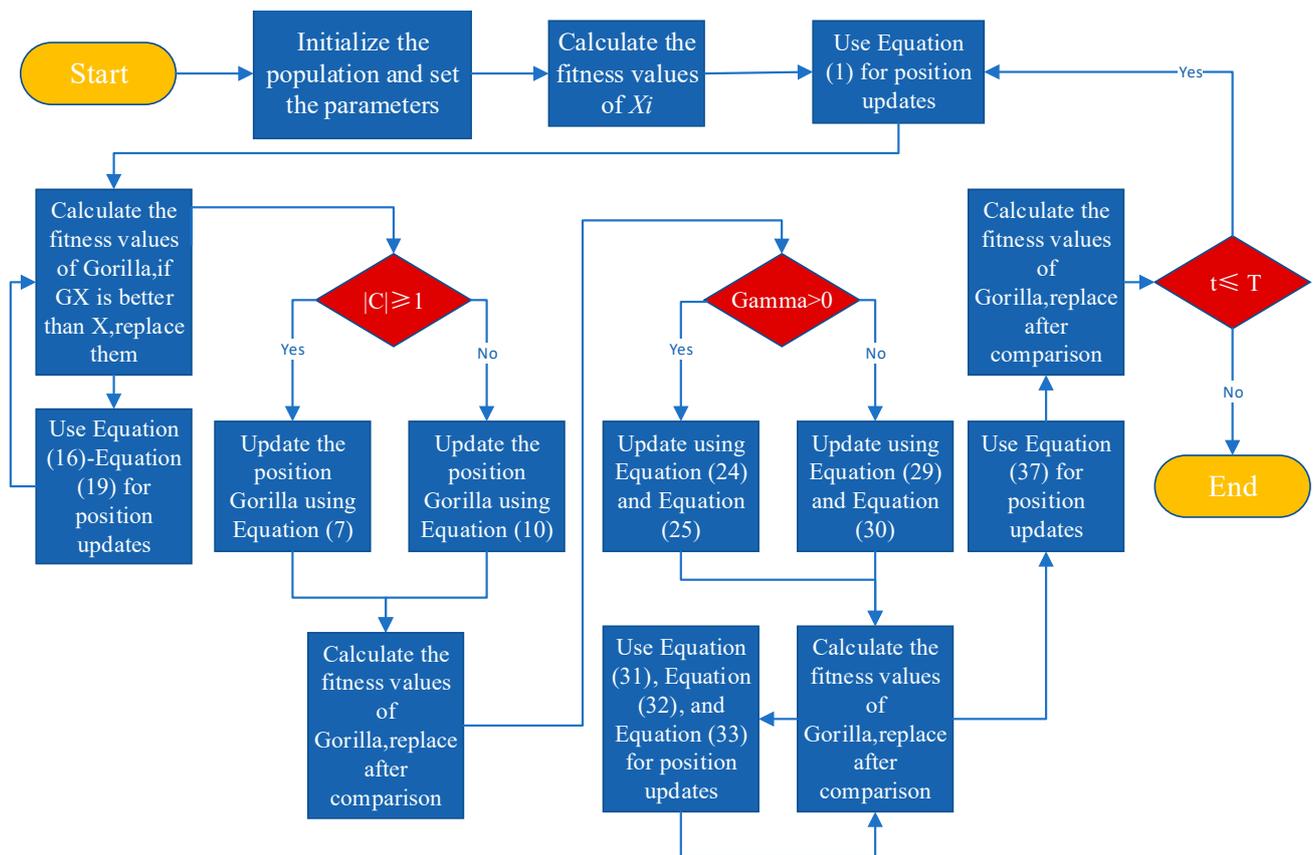


Figure 1. Flowchart for proposed MGTO algorithm.

3.5. Complexity Analysis

The time complexity depends on the size of the gorilla population (N), the given problem dimension (dim), the number of iterations of the algorithm (T), and the evaluation

cost of the solution function (C). Therefore, the time complexity of the MGTO algorithm is shown in Equation (38).

$$O(\text{MGTO}) = O(\text{define parameters}) + O(\text{population initialization}) + O(\text{dunction evaluation } \cos t) + O(\text{location update}) \tag{38}$$

The specific definition of each complexity of Equation (38) is as follows:

- (1) The initialization time of the problem definition is $O(1)$.
- (2) The time required to initialize the population position is $O(N \times \text{dim})$.
- (3) The update time of the gorilla position is $O(2 \times T \times N \times \text{dim})$.
- (4) The time required for the contraction control factor fusion strategy is $O(T \times N)$.
- (5) The time required for the sine cosine interactive fusion strategy based on closeness is $O(T \times N)$.

(6) The time required for the gorilla individual difference identification strategy is $O(2 \times T \times N)$.

(7) The time cost of calculating the function includes the calculation time cost of the algorithm itself, the calculation time of the fusion strategy of contraction control factors, the calculation time of the sine cosine interaction fusion strategy based on closeness, and the calculation time of the gorilla individual difference identification strategy. The calculation time cost of the algorithm itself is $O(2 \times T \times N \times C)$. The calculation time cost of the three strategies is $O(T \times N \times C)$. Therefore, the total time cost is $O(5 \times T \times N \times C)$.

Therefore, the time complexity of the MGTO algorithm is:

$$O(\text{MGTO}) = O(1 + 4 \times T \times N + N \times \text{dim} + 5 \times T \times N \times C + 2 \times T \times N \times \text{dim}) \tag{39}$$

Because $1 \ll T \times N \times C, 1 \ll T \times N \times \text{dim}, N \times \text{dim} \ll T \times N \times C$, and $N \times \text{dim} \ll T \times N \times \text{dim}$, Equation (39) can be replaced by Equation (40):

$$O(\text{MGTO}) \cong O(6 \times T \times N + 5 \times T \times N \times C + 2 \times T \times N \times \text{dim}) \tag{40}$$

The computational complexity of the GTO algorithm initialization process is $O(N)$. There are two stages of exploitation and exploration in the update process, and its computational complexity is equal to $O(T \times N) + O(T \times N \times D)$. Therefore, the computational complexity of GTO is $O(N \times (1 + T + TD) \times 2)$.

It can be seen that although the MGTO algorithm has increased in time complexity, it is far superior to the original algorithm.

4. Experimental Results and Discussion

In this section, we explain how 23 classic benchmark functions were used to evaluate the performance of the improved algorithm. In order to better demonstrate the performance of the improved algorithm, nine algorithms were selected for comparison. These nine algorithms are the Artificial Gorilla Troop Optimization Algorithm (GTO) [21], the Sine Cosine Algorithm (SCA) [6], the Remora Optimal Algorithm (ROA) [16], the Whale Optimization Algorithm (WOA) [19], the Reptile Search Algorithm (RSA) [27], the Spotted Hyena Optimizer Algorithm (SHO) [20], the Seagull Optimization Algorithm (SOA) [17], the Arithmetic Optimization Algorithm (AOA) [4], and the Honey Badger Algorithm (HBA) [28]. At the same time, to verify the optimization effect of improving the GTO algorithm, the maximum number of iterations and population size of the algorithms in all experiments were set to 500 and 30, respectively. In addition, to further prove the performance of the improved algorithm, 30 benchmark functions in CEC2014 and 10 in CEC2020 were used for further experiments. Many articles use these benchmark functions to evaluate the algorithm's performance. The criteria for evaluating the performance of the optimization algorithm depend on whether the algorithm can keep the balance between exploration and exploitation while jumping out of the local optimal solution.

All of the experiments reported in this paper were completed on a computer with an 11th generation Intel (R) Core (TM) i7-11700 processor. The main frequency of the processor was 2.50 GHz, the memory was 16 GB, the operating system was 64-bit Windows 11, and Matlab 2020b was used. The parameter settings of all of the above algorithms are shown in Table 1.

Table 1. Parameter settings for the comparative algorithms.

Algorithm	Parameters	Value
MGTO/GTO	p	0.03
	β	3
	w	0.8
SCA	α	2
ROA	C	0.1
WOA	Coefficient vectors \vec{A}	1
	Coefficient vectors \vec{C}	[-1, 1]
	Helical parameter b	0.75
	Helical parameter l	[-1, 1]
RSA	α	0.1
	β	0.005
SHO	\	\
SOA	b	1
AOA	MOP_Max	1
	MOP_Min	0.2
	A	5
	Mu	0.499
HBA	β	6
	C	2

4.1. Experiments on 23 Standard Benchmark Functions

These 23 classical benchmark functions can be divided into three categories: single-mode functions (UM), multimodal functions (MM), and composite functions (CM). Single mode functions (F1–F7) have only one optimal solution and are often used to evaluate the exploration ability of algorithms. Multiple optimal solutions characterize multimodal functions (F8–F13). These functions can be used to evaluate the ability to jump out of the optimal solution in complex situations. Composite functions (F14–F23) are usually used to evaluate the stability of algorithms. See Table 2 for details on the 23 benchmark functions, where F is the corresponding mathematical function, dim is the dimension, the range is the limit of the search space, and F_{min} is the optimal value that the corresponding function can achieve. The MGTO optimization algorithm and the other nine comparison algorithms were independently run 30 times to obtain the corresponding algorithm’s optimal value, average value, and standard deviation, as given in Table 3. The image F1–F13 of thirteen classical benchmark functions is shown in Figure 2.

Table 2. Details of 23 benchmark functions.

Type	F	dim	Range	F_{min}
Unimodal benchmark functions	$F_1(x) = \sum_{i=1}^n x_i^2$	30/500	[−100, 100]	0
	$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30/500	[−10, 10]	0
	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30/500	[−100, 100]	0
	$F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	30/500	[−100, 100]	0
	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30/500	[−30, 30]	0
	$F_6(x) = \sum_{i=1}^n (x_i + 5)^2$	30/500	[−100, 100]	0
	$F_7(x) = \sum_{i=1}^n i \times x_i^4 + random[0, 1)$	30/500	[−1.28, 1.28]	0
Multimodal benchmark functions	$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30/500	[−500, 500]	$-418.9829 \times dim$
	$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30/500	[−5.12, 5.12]	0
	$F_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i))) + 20 + e)$	30/500	[−32, 32]	0
	$F_{11}(x) = \frac{1}{400} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30/500	[−600, 600]	0
	$F_{12}(x) = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4)$, where $y_i = 1 + \frac{x_i + 1}{4}$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30/500	[−50, 50]	0
	$F_{13}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]) + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30/500	[−50, 50]	0
	$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	2	[−65, 65]	1
Fixed-dimension multimodal benchmark functions	$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[−5, 5]	0.00030
	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + x_2^4$	2	[−5, 5]	−1.0316
	$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	[−5, 5]	0.398

Table 2. Cont.

Type	F	dim	Range	F_{min}
	$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_2 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	5	[-2, 2]	3
	$F_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	[-1, 2]	-3.86
	$F_{20}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	-3.32
	$F_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
	$F_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
	$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

Table 3. Statistical results of 23 standard reference functions.

F	dim	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA
F1	30	min	0	0	3.02×10^{-3}	0	2.36×10^{-84}	0	0	1.76×10^{-14}	1.32×10^{-169}	9.41×10^{-144}
		mean	0	0	1.90×10^1	0	5.85×10^{-72}	0	0	6.04×10^{-12}	4.81×10^{-26}	1.61×10^{-135}
		std	0	0	3.51×10^1	0	3.20×10^{-71}	0	0	1.12×10^{-11}	1.56×10^{-25}	4.90×10^{-135}
	500	min	0	0	1.34×10^5	0	1.48×10^{-81}	0	0	1.34×10^{-2}	5.92×10^{-1}	1.40×10^{-119}
		mean	0	0	2.19×10^5	0	9.05×10^{-70}	0	0	1.20×10^{-1}	6.51×10^{-1}	3.42×10^{-112}
		std	0	0	4.83×10^4	0	4.94×10^{-69}	0	0	1.03×10^{-1}	4.14×10^{-2}	1.46×10^{-111}
F2	30	min	0	5.75×10^{-208}	3.05×10^{-4}	2.97×10^{-192}	1.50×10^{-57}	0	0	1.45×10^{-9}	0	2.52×10^{-76}
		mean	0	1.00×10^{-192}	1.75×10^{-2}	1.37×10^{-164}	2.03×10^{-51}	0	0	1.23×10^{-8}	0	7.34×10^{-71}
		std	0	0	2.42×10^{-2}	0	6.84×10^{-51}	0	0	1.10×10^{-8}	0	3.83×10^{-70}
	500	min	0	3.06×10^{-199}	5.04×10^1	3.00×10^{-187}	7.43×10^{-56}	0	0	2.62×10^{-3}	5.63×10^{-13}	2.68×10^{-63}
		mean	0	1.24×10^{-189}	1.22×10^2	2.35×10^{-159}	4.11×10^{-48}	0	0	6.53×10^{-3}	1.51×10^{-3}	4.29×10^{-61}
		std	0	0	5.60×10^1	1.29×10^{-158}	1.43×10^{-47}	0	0	2.81×10^{-3}	1.74×10^{-3}	8.32×10^{-61}

Table 3. Cont.

F	dim	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA	
F3	30	min	0	0	1.73×10^2	0	1.17×10^4	0	0	2.50×10^{-7}	3.81×10^{-152}	2.75×10^{-108}	
		mean	0	0	7.70×10^3	1.58×10^{-312}	4.67×10^4	0	0	6.28×10^{-4}	7.40×10^{-3}	8.30×10^{-97}	
		std	0	0	5.35×10^3	0	1.54×10^4	0	0	2.58×10^{-3}	1.50×10^{-2}	3.25×10^{-96}	
	500	min	0	0	4.64×10^6	1.57×10^{-313}	1.58×10^7	0	0	5.93×10^3	1.36×10^1	3.03×10^{-75}	
		mean	0	0	7.02×10^6	1.25×10^{-261}	3.13×10^7	0	0	1.43×10^5	3.12×10^1	1.41×10^{-64}	
		std	0	0	1.39×10^6	0	1.18×10^7	0	0	1.07×10^5	1.53×10^1	6.28×10^{-64}	
F4	30	min	0	3.03×10^{-208}	1.46×10^1	1.85×10^{-189}	2.99	0	0	1.20×10^{-4}	3.00×10^{-61}	1.53×10^{-61}	
		mean	0	2.36×10^{-192}	3.47×10^1	3.13×10^{-170}	5.09×10^1	0	0	3.97×10^{-3}	2.27×10^{-2}	2.89×10^{-57}	
		std	0	0	9.51	0	2.91×10^1	0	0	8.09×10^{-3}	2.09×10^{-2}	1.22×10^{-56}	
	500	min	0	2.43×10^{-203}	9.83×10^1	1.03×10^{-189}	3.54×10^1	0	0	9.71×10^1	1.54×10^{-1}	1.39×10^{-30}	
		mean	0	3.89×10^{-188}	9.90×10^1	1.86×10^{-159}	8.51×10^1	0	0	9.88×10^1	1.80×10^{-1}	3.38×10^{-28}	
		std	0	0	2.77×10^{-1}	1.02×10^{-158}	1.89×10^1	0	0	5.54×10^{-1}	1.58×10^{-2}	5.74×10^{-28}	
F5	30	min	1.30×10^{-10}	4.41×10^{-6}	5.03×10^1	2.87×10^1	2.72×10^1	9.87×10^{-26}	2.87×10^1	2.70×10^1	2.78×10^1	2.29×10^1	
		mean	2.55×10^{-7}	1.60	2.19×10^4	2.87×10^1	2.79×10^1	2.51×10^1	2.88×10^1	2.83×10^1	2.85×10^1	2.42×10^1	
		std	4.72×10^{-7}	6.10	3.92×10^4	1.76×10^{-2}	5.15×10^{-1}	1.00×10^1	1.10×10^{-1}	6.06×10^{-1}	3.21×10^{-1}	1.11	
	500	min	3.30×10^{-8}	7.75×10^{-5}	9.07×10^8	4.94×10^2	4.96×10^2	4.99×10^2	4.98×10^2	4.98×10^2	5.48×10^2	4.99×10^2	4.95×10^2
		mean	2.56×10^{-4}	1.48	2.00×10^9	4.94×10^2	4.96×10^2	4.99×10^2	4.99×10^2	4.99×10^2	9.52×10^2	4.99×10^2	4.98×10^2
		std	4.00×10^{-4}	2.89	5.16×10^8	2.32×10^{-1}	4.35×10^{-1}	0.00 × 100	1.33×10^{-1}	4.06×10^2	7.94×10^{-2}	7.46×10^{-1}	
F6	30	min	6.04×10^{-16}	6.47×10^{-9}	4.58	1.02×10^{-1}	5.67×10^{-2}	6.36	3.14×10^{-2}	2.31	2.47	2.53×10^{-6}	
		mean	2.71×10^{-11}	2.86×10^{-7}	3.69×10^1	6.11×10^{-1}	3.65×10^{-1}	7.26	3.32	3.25	3.12	2.52×10^{-2}	
		std	6.58×10^{-11}	5.21×10^{-7}	6.69×10^1	3.16×10^{-1}	2.35×10^{-1}	3.30×10^{-1}	2.49	5.10×10^{-1}	2.54×10^{-1}	7.63×10^{-2}	
	500	min	1.99×10^{-9}	1.06×10^{-3}	1.06×10^5	5.66×10^{-1}	1.39×10^1	1.25×10^2	1.16×10^2	1.14×10^2	1.13×10^2	1.13×10^2	9.44×10^1
		mean	9.03×10^{-7}	4.33×10^{-1}	2.03×10^5	8.85	3.27×10^1	1.25×10^2	1.23×10^2	1.16×10^2	1.16×10^2	1.16×10^2	9.78×10^1
		std	1.15×10^{-6}	3.85×10^{-1}	7.01×10^4	4.51	9.46	0	2.35	9.19×10^{-1}	1.35	2.15	
F7	30	min	3.12×10^{-7}	5.48×10^{-6}	1.06×10^{-2}	5.41×10^{-6}	1.51×10^{-4}	1.35×10^{-5}	8.03×10^{-6}	3.44×10^{-4}	9.78×10^{-7}	5.16×10^{-5}	
		mean	2.02×10^{-5}	1.12×10^{-4}	8.69×10^{-2}	1.54×10^{-4}	3.27×10^{-3}	9.68×10^{-5}	1.27×10^{-4}	2.89×10^{-3}	9.06×10^{-5}	3.78×10^{-4}	
		std	1.79×10^{-5}	1.01×10^{-4}	6.18×10^{-2}	1.52×10^{-4}	3.47×10^{-3}	9.60×10^{-5}	1.90×10^{-4}	2.40×10^{-3}	7.89×10^{-5}	2.91×10^{-4}	
	500	min	2.37×10^{-6}	5.82×10^{-6}	7.89×10^3	3.17×10^{-6}	1.73×10^{-4}	3.83×10^{-6}	2.47×10^{-6}	2.51×10^{-2}	2.05×10^{-6}	3.72×10^{-5}	
		mean	2.60×10^{-5}	8.81×10^{-5}	1.44×10^4	1.11×10^{-4}	4.26×10^{-3}	1.72×10^{-4}	8.06×10^{-5}	8.87×10^{-2}	8.52×10^{-5}	4.28×10^{-4}	
		std	2.03×10^{-5}	6.24×10^{-5}	3.00×10^3	8.38×10^{-5}	4.84×10^{-3}	1.61×10^{-4}	7.64×10^{-5}	4.69×10^{-2}	7.23×10^{-5}	2.90×10^{-4}	

Table 3. Cont.

F	dim	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA	
F8	30	min	-1.26×10^4	-1.26×10^4	-5.08×10^3	-1.26×10^4	-1.26×10^4	-5.66×10^3	-4.01×10^3	-6.87×10^3	-6.73×10^3	-1.02×10^4	
		mean	-1.26×10^4	-1.26×10^4	-3.82×10^3	-1.25×10^4	-1.01×10^4	-5.46×10^3	-2.66×10^3	-5.05×10^3	-5.39×10^3	-8.59×10^3	
		std	3.29×10^{-6}	4.27×10^{-5}	3.30×10^2	1.91×10^2	1.89×10^3	2.16×10^2	7.06×10^2	7.06×10^2	4.46×10^2	9.82×10^2	
	500	min	-2.09×10^5	-2.09×10^5	-1.72×10^4	-2.09×10^5	-2.09×10^5	-7.61×10^4	-2.05×10^4	-3.81×10^4	-2.64×10^4	-1.01×10^5	
		mean	-2.09×10^5	-2.09×10^5	-1.52×10^4	-2.07×10^5	-1.73×10^5	-6.42×10^4	-1.32×10^4	-2.30×10^4	-2.27×10^4	-7.32×10^4	
		std	2.66×10^{-2}	3.12×10^1	9.19×10^2	7.21×10^3	2.89×10^4	5.62×10^3	4.83×10^3	4.21×10^3	1.46×10^3	1.35×10^4	
F9	30	min	0	0	1.96×10^{-2}	0	0	0	0	8.53×10^{-13}	0	0	
		mean	0	0	3.74×10^1	0	0	0	0	3.19	0	0	
		std	0	0	4.58×10^1	0	0	0	0	5.87	0	0	
	500	min	0	0	4.93×10^2	0	0	0	0	4.11×10^{-5}	0	0	
		mean	0	0	1.21×10^3	0	6.06×10^{-14}	0	0	7.01	7.21×10^{-6}	0	
		std	0	0	4.62×10^2	0	3.32×10^{-13}	0	0	9.00	7.25×10^{-6}	0	
F10	30	min	8.88×10^{-16}	8.88×10^{-16}	3.07×10^{-2}	8.88×10^{-16}	2.00×10^1	8.88×10^{-16}	8.88×10^{-16}				
		mean	8.88×10^{-16}	8.88×10^{-16}	1.24×10^1	8.88×10^{-16}	4.80×10^{-15}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	2.00×10^1	8.88×10^{-16}	8.88×10^{-16}
		std	0	0	9.42	0	2.53×10^{-15}	0	0	1.62×10^{-3}	0	0	
	500	min	8.88×10^{-16}	8.88×10^{-16}	7.92	8.88×10^{-16}	2.00×10^1	7.22×10^{-3}	8.88×10^{-16}				
		mean	8.88×10^{-16}	8.88×10^{-16}	1.93×10^1	8.88×10^{-16}	4.68×10^{-15}	8.88×10^{-16}	8.88×10^{-16}	3.54	2.00×10^1	7.91×10^{-3}	4.64
		std	0	0	3.42	0	2.63×10^{-15}	0	0	4.27	7.17×10^{-5}	3.77×10^{-4}	8.56
F11	30	min	0	0	1.40×10^{-4}	0	0	0	0	7.85×10^{-13}	2.31×10^{-2}	0	
		mean	0	0	1.09	0	4.49×10^{-3}	0	0	2.00×10^{-2}	2.33×10^{-1}	0	
		std	0	0	8.47×10^{-1}	0	2.46×10^{-2}	0	0	3.85×10^{-2}	1.63×10^{-1}	0	
	500	min	0	0	2.64×10^2	0	0	0	0	8.27×10^{-4}	5.83×10^3	0	
		mean	0	0	1.74×10^3	0	0	0	0	5.30×10^{-2}	9.15×10^3	0	
		std	0	0	7.17×10^2	0	0	0	0	7.76×10^{-2}	2.35×10^3	0	

Table 3. Cont.

F	dim	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA
F12	30	min	2.50×10^{-15}	7.46×10^{-11}	9.42×10^{-1}	6.17×10^{-3}	3.37×10^{-3}	4.69×10^{-1}	1.42×10^{-4}	1.49×10^{-1}	4.13×10^{-1}	7.54×10^{-7}
		mean	3.67×10^{-12}	2.23×10^{-8}	4.99×10^4	3.28×10^{-2}	1.21×10^{-1}	1.38	2.06×10^{-4}	3.45×10^{-1}	5.14×10^{-1}	2.00×10^{-4}
		std	4.44×10^{-12}	3.29×10^{-8}	2.08×10^5	2.06×10^{-2}	4.07×10^{-1}	4.24×10^{-1}	2.56×10^{-5}	1.62×10^{-1}	4.73×10^{-2}	9.72×10^{-4}
	500	min	3.07×10^{-15}	7.19×10^{-8}	3.77×10^9	1.65×10^{-4}	2.03×10^{-2}	1.21	3.69×10^{-4}	1.10	1.06	6.93×10^{-1}
		mean	1.77×10^{-9}	2.47×10^{-4}	6.19×10^9	2.26×10^{-2}	8.65×10^{-2}	1.21	9.65×10^{-1}	1.94	1.08	7.48×10^{-1}
		std	4.59×10^{-9}	2.85×10^{-4}	1.30×10^9	1.40×10^{-2}	3.67×10^{-2}	4.52×10^{-16}	3.72×10^{-1}	9.48×10^{-1}	1.34×10^{-2}	2.73×10^{-2}
F13	30	min	2.10×10^{-14}	6.32×10^{-10}	2.99	1.42×10^{-2}	1.49×10^{-1}	2.20×10^{-30}	2.92	1.60	2.59	1.90×10^{-3}
		mean	1.89×10^{-10}	2.10×10^{-3}	2.64×10^5	3.49×10^{-1}	5.91×10^{-1}	5.73×10^{-1}	2.96	1.99	2.83	3.22×10^{-1}
		std	6.57×10^{-10}	6.90×10^{-3}	1.12×10^6	2.00×10^{-1}	2.23×10^{-1}	1.17	1.91×10^{-2}	2.03×10^{-1}	1.03×10^{-1}	2.82×10^{-1}
	500	min	1.55×10^{-10}	1.58×10^{-6}	4.36×10^9	1.37×10^{-1}	7.04	5.00×10^1	4.99×10^1	5.42×10^1	5.01×10^1	4.88×10^1
		mean	1.85×10^{-7}	6.74×10^{-2}	9.40×10^9	5.30	1.84×10^1	5.00×10^1	5.00×10^1	7.32×10^1	5.02×10^1	4.93×10^1
		std	3.95×10^{-7}	1.15×10^{-1}	2.32×10^9	2.88×100	6.99	0	1.76×10^{-2}	1.51×10^1	3.81×10^{-2}	2.79×10^{-1}
F14	2	min	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	1.06	1.06	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}
		mean	9.98×10^{-1}	9.98×10^{-1}	1.53	4.78	2.73	4.72	9.52	2.48	9.71	1.72
		std	0	0	8.92×10^{-1}	4.56	2.98	3.55	3.97	2.46	3.65	1.89
F15	4	min	3.07×10^{-4}	3.07×10^{-4}	5.14×10^{-4}	3.40×10^{-4}	3.09×10^{-4}	7.36×10^{-4}	3.10×10^{-4}	1.22×10^{-3}	3.72×10^{-4}	3.07×10^{-4}
		mean	3.07×10^{-4}	4.30×10^{-4}	9.37×10^{-4}	9.01×10^{-4}	6.56×10^{-4}	2.79×10^{-3}	3.17×10^{-4}	1.25×10^{-3}	1.08×10^{-2}	7.21×10^{-3}
		std	1.84×10^{-18}	3.17×10^{-4}	3.24×10^{-4}	5.74×10^{-4}	4.45×10^{-4}	1.97×10^{-3}	5.01×10^{-6}	5.07×10^{-5}	1.30×10^{-2}	1.00×10^{-2}
F16	2	min	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03
		mean	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-9.48×10^{-1}	-1.03	-1.03
		std	6.32×10^{-16}	6.39×10^{-16}	5.45×10^{-5}	4.96×10^{-8}	1.43×10^{-9}	5.90×10^{-3}	1.85×10^{-1}	2.56×10^{-6}	1.34×10^{-7}	6.05×10^{-16}
F17	2	min	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.99×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
		mean	3.98×10^{-1}	3.98×10^{-1}	4.00×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	4.26×10^{-1}	5.37×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
		std	0	0	1.60×10^{-3}	9.33×10^{-6}	1.54×10^{-5}	3.04×10^{-2}	4.25×10^{-1}	8.10×10^{-5}	1.07×10^{-7}	0
F18	5	min	3.00	3.00	3.00	3.00	3.00	3.00	3.40	3.00	3.00	3.00
		mean	3.00	3.00	3.00	3.00	3.00	3.00	5.78	2.26×10^1	3.00	5.70
		std	1.71×10^{-15}	1.21×10^{-15}	4.66×10^{-5}	9.32×10^{-4}	1.10×10^{-4}	8.48	3.39×10^1	4.74×10^{-4}	8.24	2.09×10^1

Table 3. Cont.

F	dim	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA
F19	3	min	−3.86	−3.86	−3.86	−3.86	−3.86	−3.86	−3.85	−3.86	−3.86	−3.86
		mean	−3.86	−3.86	−3.85	−3.76	−3.86	−3.79	−3.46	−3.85	−3.85	−3.86
		std	2.54×10^{-15}	2.65×10^{-15}	4.12×10^{-3}	1.63×10^{-1}	5.23×10^{-3}	5.49×10^{-2}	4.63×10^{-1}	7.04×10^{-3}	3.62×10^{-3}	2.60×10^{-15}
F20	6	min	−3.32	−3.32	−3.16	−3.28	−3.32	−2.98	−3.11	−3.19	−3.22	−3.32
		mean	−3.31	−3.28	−2.92	−3.04	−3.24	−2.47	−2.68	−3.01	−3.08	−3.26
		std	4.11×10^{-2}	5.70×10^{-2}	3.30×10^{-1}	1.61×10^{-1}	9.60×10^{-2}	4.39×10^{-1}	2.38×10^{-1}	1.68×10^{-1}	9.54×10^{-2}	7.68×10^{-2}
F21	4	min	−1.02 × 10¹	−1.02 × 10¹	−5.44	−1.02 × 10 ¹	−1.02 × 10 ¹	−5.06	−8.06	−1.01 × 10 ¹	−8.70	−1.02 × 10¹
		mean	−1.02 × 10¹	−1.02 × 10¹	−2.29	−9.85	−8.68	−5.01	−4.04	−4.17	−4.08	−1.02 × 10 ¹
		std	5.67×10^{-15}	6.04×10^{-15}	1.67	1.46	2.45	2.24×10^{-1}	1.43	4.03	1.50	8.14×10^{-6}
F22	4	min	−1.04 × 10¹	−1.04 × 10¹	−7.31	−1.04 × 10 ¹	−1.04 × 10 ¹	−5.09	−5.96	−1.04 × 10 ¹	−8.54	−1.04 × 10¹
		mean	−1.04 × 10¹	−1.04 × 10¹	−2.91	−1.04 × 10 ¹	−8.12	−5.09	−4.40	−5.64	−4.08	−9.07
		std	8.08×10^{-16}	8.08×10^{-16}	1.89	7.55×10^{-2}	3.08	8.73×10^{-7}	1.05	4.39	1.80	3.04
F23	4	min	−1.05 × 10¹	−1.05 × 10 ¹	−8.94	−1.05 × 10 ¹	−1.05 × 10 ¹	−5.13	−5.85	−1.05 × 10 ¹	−9.27	−1.05 × 10 ¹
		mean	−1.05 × 10¹	−1.05 × 10¹	−4.25	−1.03 × 10 ¹	−7.13	−5.13	−4.07	−6.74	−3.96	−8.32
		std	1.98×10^{-15}	2.84×10^{-15}	1.83	1.02	3.10	1.86×10^{-6}	1.28	4.19	1.96	3.47

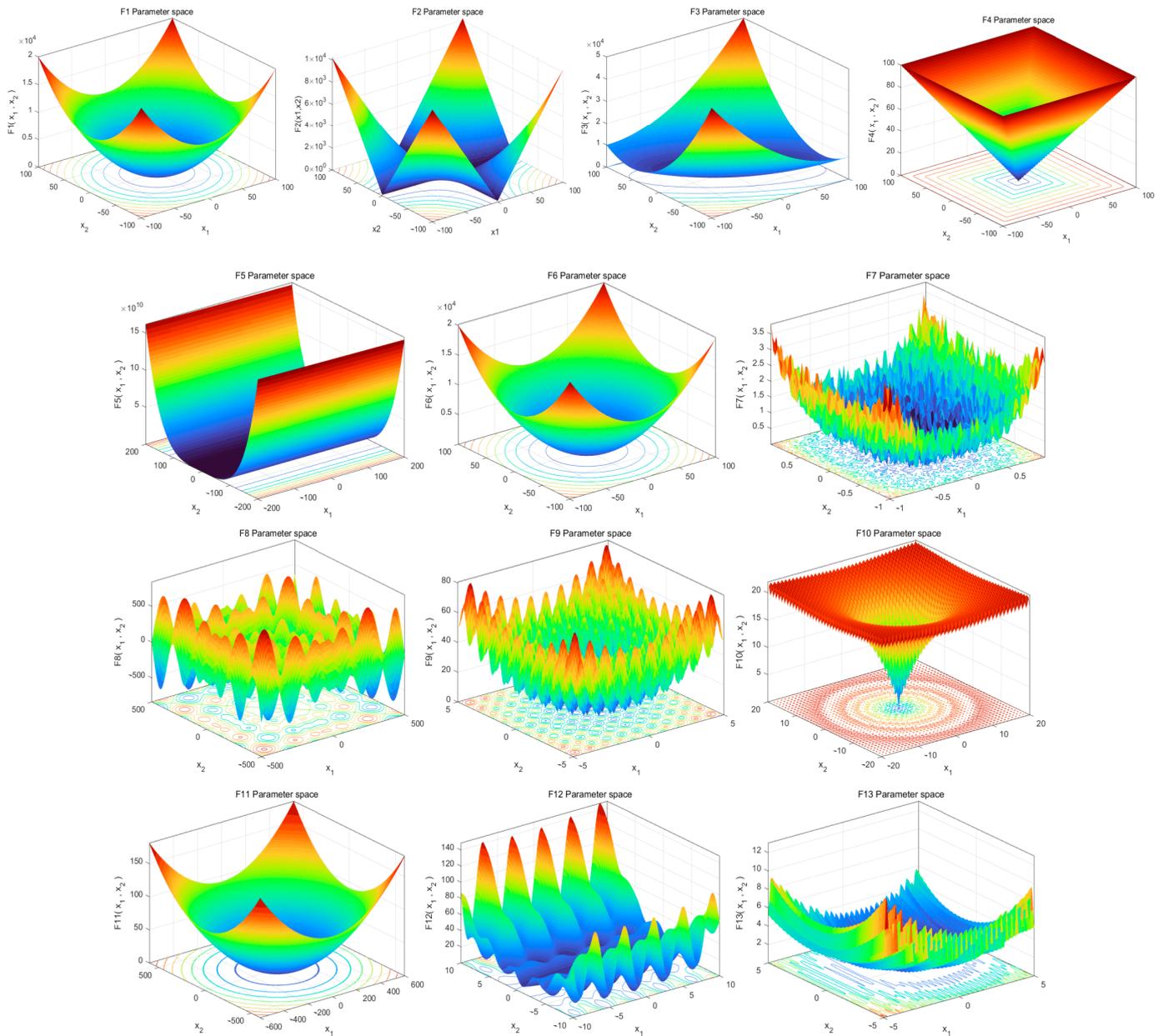


Figure 2. Schematic diagram of 23 standard reference functions.

4.1.1. Result Statistics and Convergence Curve Analysis of 23 Standard Data Functions

Table 3 shows the statistical results of 10 algorithms across 23 benchmark functions. It can be seen from the table that the MGTO algorithm obtained the theoretical optimal value in both the 30 and 500 dimensions of F1–F4 and had a very stable effect. The RSA and SHO algorithms had the same performance. The GTO algorithm obtained the theoretical optimal value in F1 and F3, and the ROA algorithm obtained the optimal value in F1. In F5–F8, the MGTO algorithm had almost all of the best fitness values. It was very stable and far superior to other algorithms, showing that the MGTO algorithm has better optimization than the GTO algorithm. For F9–F11, the MGTO algorithm, GTO algorithm, ROA, and RSA all obtained the theoretical optimal value. The SHO algorithm could obtain the optimal value in the 30 dimensions of F10 but could only approach the optimal value in 500 dimensions. Therefore, the SHO algorithm is not strong in multi-dimensional exploration. In the 30 dimensions of F12 and in 500 dimensions, the MGTO algorithm obtained the best fitness value and was stable. In the 30 dimensions of F13, the MGTO algorithm obtained the best fitness value only compared to the RSA algorithm, but it obtained a better fitness value,

one that was far better than those of the other algorithms. In 500 dimensions, the MGTO algorithm obtained the best fitness value and had better stability. The F14–F23 functions are relatively simple, and it was easy to find a better fitness value. The MGTO algorithm obtained the optimal fitness value in the optimal fitness test of combined functions, proving the superiority of the MGTO algorithm’s performance.

It is difficult to fully explain the advantages of the MGTO algorithm in the 23 standard reference functions only through Table 3; the optimization effect can be better understood by analyzing the convergence curve image. The convergence curves of each algorithm are shown in Figures 3–5. It can be seen from these images that the MGTO algorithm had strong convergence and exploitation ability in F1–F4 and quickly found the optimal values. For F5–F6, it can be seen from the figure that the MGTO algorithm found the optimal value faster and was more stable than other algorithms, which means its performance was far better than other algorithms. In F7, the MGTO algorithm quickly found the fitness value and, at the same time, continuously jumped out of the optimal local solution in the later stages, which shows that the exploration ability of the MGTO algorithm was enhanced over its predecessor. Good results were achieved in the later stage for the F8 function when testing the MGTO algorithm, GTO algorithm, and ROA algorithm. Still, the early exploitation performance of the MGTO algorithm was better, which made it converge rapidly. In F9–F11, the MGTO algorithm found the optimal fitness value faster than other algorithms. In F12 and F13, MGTO’s exploration and exploitation performance were superior to other algorithms. In F14–F23, these algorithms found better fitness values, which shows that these algorithms have good optimization effects, but the MGTO algorithm also found very good fitness values. It can be clearly seen from the results regarding F14, F15, and F19–F23 that the MGTO algorithm is better. The comprehensive analysis of these tables and images shows that the MGTO algorithm is relatively more stable and can find better values.

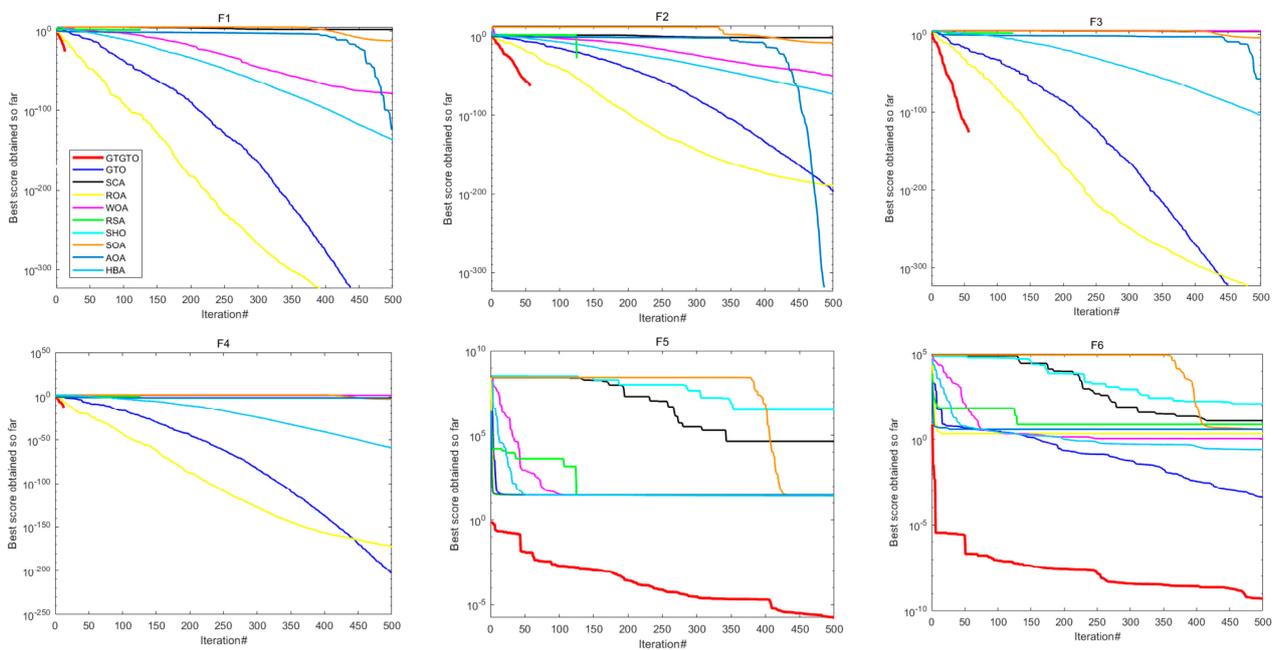


Figure 3. Cont.

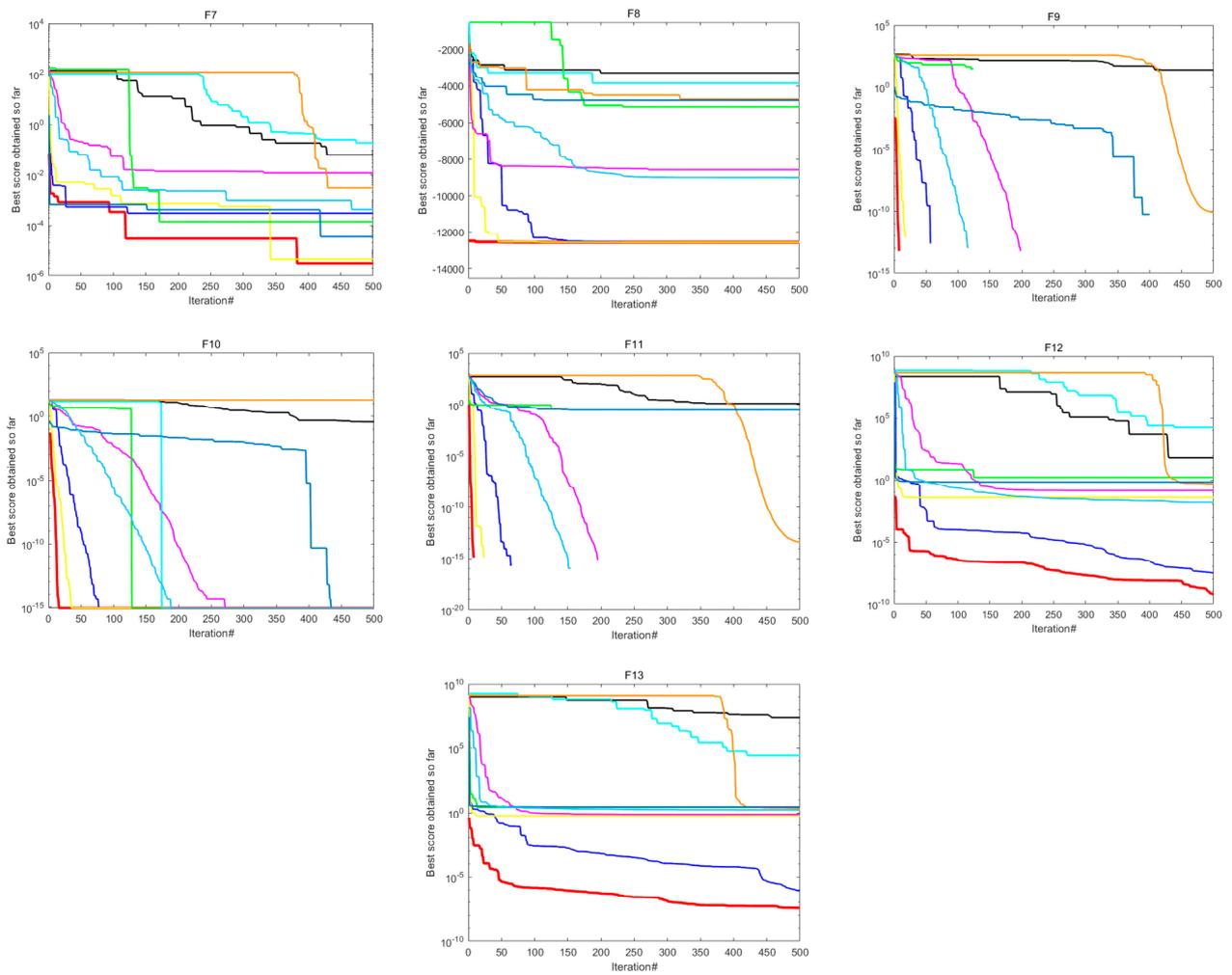


Figure 3. Convergence curves for the optimization algorithms for standard benchmark functions (F1–F13) with $dim = 30$.

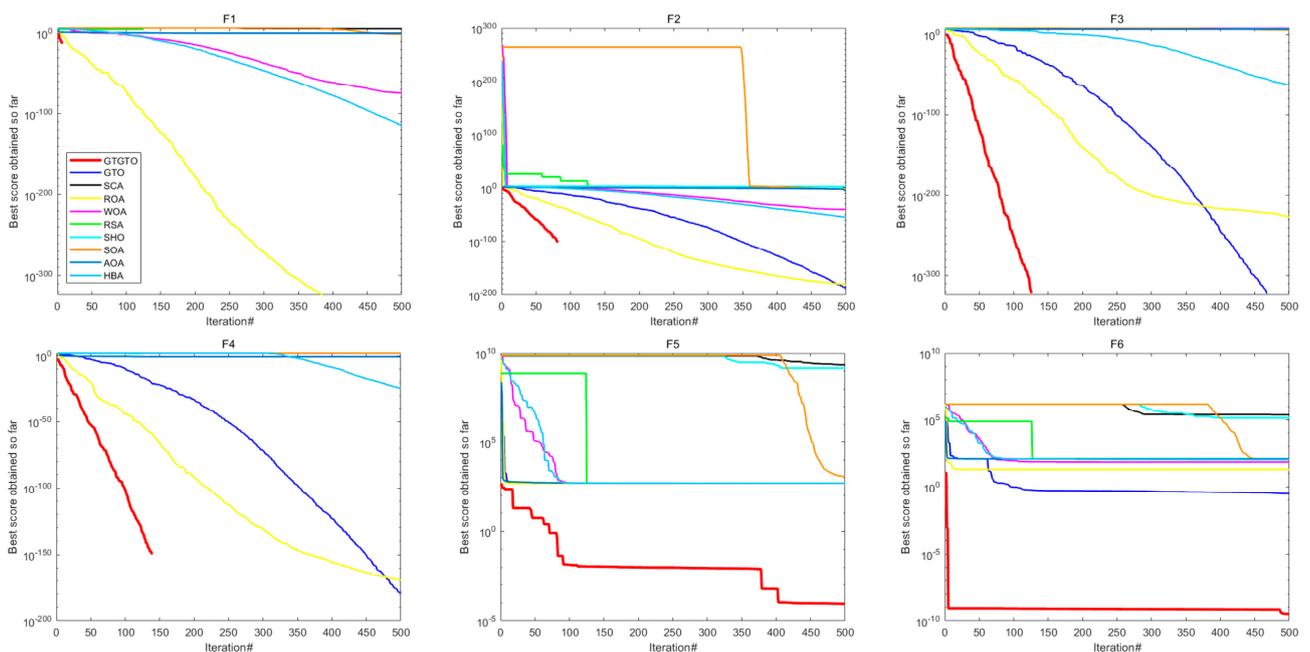


Figure 4. Cont.

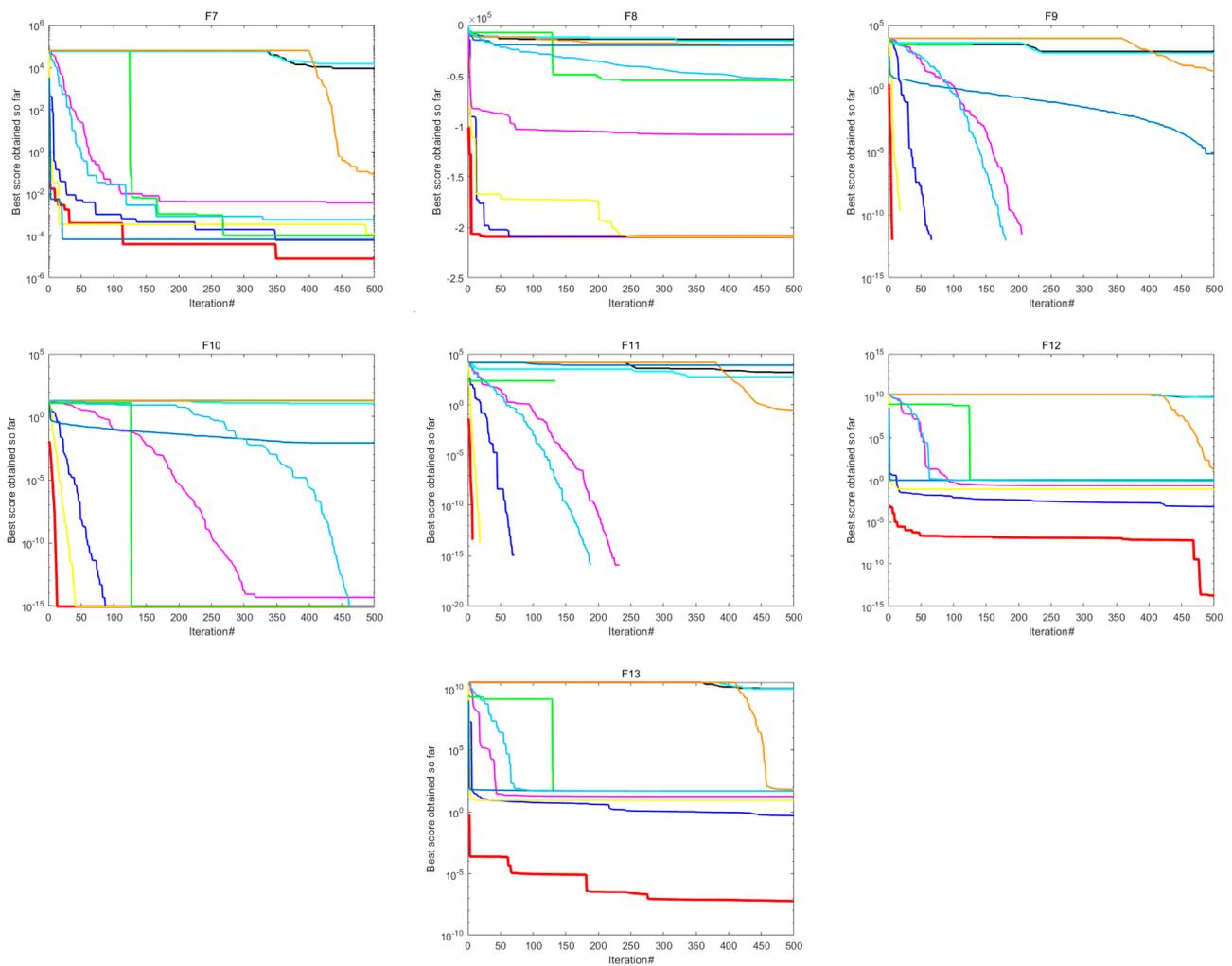


Figure 4. Convergence curves for the optimization algorithms for standard benchmark functions (F1–F13) with $dim = 500$.

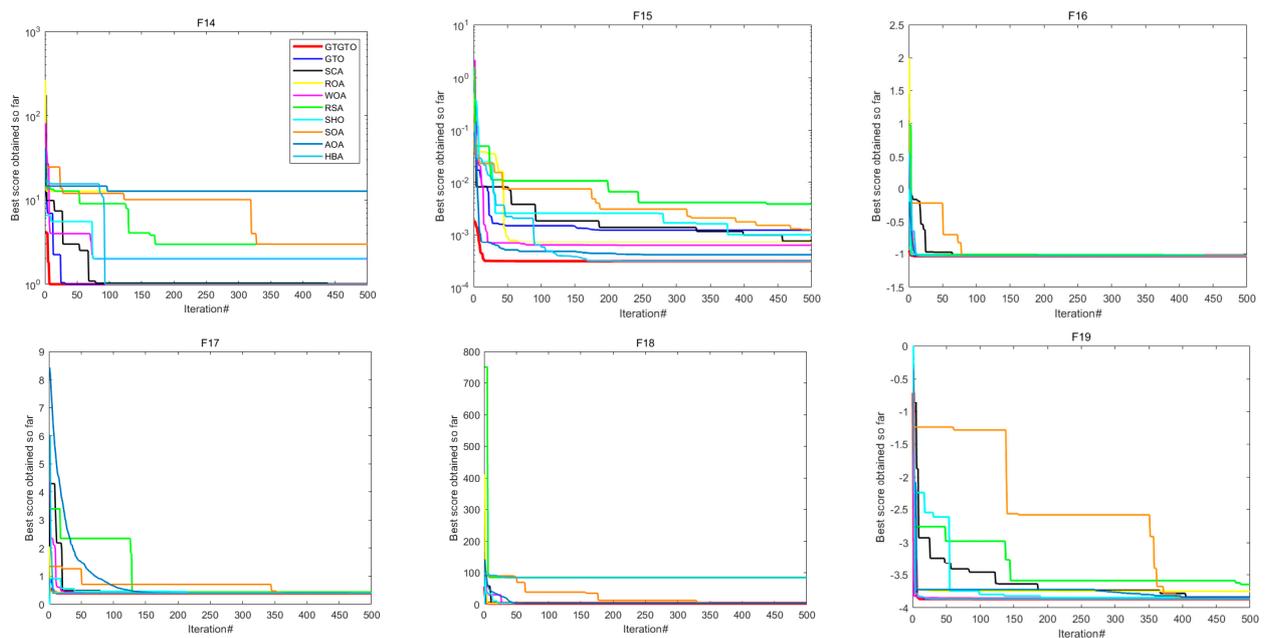


Figure 5. Cont.

Table 4. Cont.

F	dim	MGTO Vs GTO	MGTO Vs SCA	MGTO Vs ROA	MGTO Vs WOA	MGTO Vs RSA	MGTO Vs SHO	MGTO Vs SOA	MGTO Vs AOA	MGTO Vs HBA
F6	30	1.73×10^{-6}								
	50	1.73×10^{-6}								
F7	30	6.98×10^{-6}	1.73×10^{-6}	1.25×10^{-4}	1.73×10^{-6}	6.89×10^{-5}	1.73×10^{-6}	1.73×10^{-6}	7.51×10^{-5}	1.73×10^{-6}
	50	3.06×10^{-4}	1.73×10^{-6}	6.32×10^{-5}	1.92×10^{-6}	1.48×10^{-4}	1.73×10^{-6}	1.73×10^{-6}	2.26×10^{-3}	6.98×10^{-6}
F8	30	3.61×10^{-3}	1.73×10^{-6}							
	50	4.73×10^{-6}	1.73×10^{-6}	9.32×10^{-6}	1.73×10^{-6}					
F9	30	1	1.73×10^{-6}	1	5.00×10^{-1}	1	1.73×10^{-6}	1.73×10^{-6}	1	1
	50	1	1.73×10^{-6}	1	1	1	1.73×10^{-6}	1.73×10^{-6}	8.86×10^{-5}	1
F10	30	1	1.73×10^{-6}	1	2.41×10^{-6}	1	1.73×10^{-6}	1.73×10^{-6}	1	5.00×10^{-1}
	50	1	1.73×10^{-6}	1	3.22×10^{-5}	1	1.73×10^{-6}	1.72×10^{-6}	1.73×10^{-6}	9.77×10^{-4}
F11	30	1	1.73×10^{-6}	1	5.00×10^{-1}	1	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1
	50	1	1.73×10^{-6}	1	1	1	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1
F12	30	1.73×10^{-6}								
	50	1.73×10^{-6}								
F13	30	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.48×10^{-2}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}
	50	1.73×10^{-6}								
F14	2	2.50×10^{-2}	1.73×10^{-6}	1.72×10^{-1}						
F15	4	4.49×10^{-2}	1.73×10^{-6}	1.64×10^{-5}	2.16×10^{-5}	1.73×10^{-6}	1.73×10^{-6}	2.13×10^{-6}	6.34×10^{-6}	8.32×10^{-6}
F16	2	1	1.73×10^{-6}	1						
F17	2	1	1.73×10^{-6}	1						
F18	5	4.81×10^{-1}	1.73×10^{-6}	5.93×10^{-2}						
F19	6	1	1.73×10^{-6}	1						
F20	3	1.25×10^{-1}	1.73×10^{-6}	1.73×10^{-6}	8.47×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	7.57×10^{-4}
F21	4	1	1.73×10^{-6}	1.25×10^{-1}						
F22	4	1	1.73×10^{-6}	7.81×10^{-3}						
F23	4	1	1.73×10^{-6}	3.13×10^{-2}						

Therefore, the experimental results show that the MGTO algorithm has a good optimization effect in 23 benchmark functions, performs better than the original algorithm, and has significant differences compared to other algorithms.

4.2. Experiments on 30 CEC2014 and 10 CEC2020 Benchmark Function

The above 23 benchmark functions are simple test functions, which are insufficient to prove the MGTO algorithm superior performance fully. In order to effectively prove the effectiveness of the MGTO algorithm, the CEC2014 and CEC2020 test functions were used for further testing. Within the set of CEC2014 test functions, CEC1–CEC3 are unimodal functions, CEC4–CEC16 are simple multiple functions, CEC17–CEC22 are hybrid functions, and CEC23–30 are composition functions. For CEC2020 test functions, functions CEC1–CEC4 are, in ascending order, the translation rotation function, the translation rotation Schwefel function, the translation rotation Lunacek double grating function, and the extended Rosenbrock plus Griewangk function. The CEC5–CEC7 functions in this set are mixed functions, and the CEC8–CEC10 functions are compound functions. All algorithms were independently run 30 times to obtain each algorithm’s best value, average value, and standard deviation.

See Table 5 for the test results of all algorithms with the CEC2014 function. It can be seen from the table that the MGTO algorithm provided 25 best results for the average value, 28 best results for the optimal value, and 21 best results for the standard deviation of 30 benchmark functions. HBA provided the optimal value for CEC6, CEC9, and CEC25 in terms of average value, and it provided the best value for CEC6 and CEC9 in terms of optimal value. For the CEC6 and CEC9 test functions, the mean values and optimal values of the MGTO algorithm were all inferior to HBA and superior to other algorithms. Regarding STD, the improved algorithm was more stable than the original algorithm and superior to most algorithms. In the CEC8 and CEC25 test functions, the MGTO algorithm obtained the optimal values; its mean value was only inferior to HBA, and its stability was also stronger than those of most other algorithms. For the CEC24 test function, the MGTO

algorithm also obtained the optimal value, with its average value second only to those of SOA, SCA, and HBA. This shows that the MGTO algorithm is a stronger optimizer.

Table 5. CEC2014 Algorithm Results of Benchmark Functions.

CEC2014	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA
CEC1	MIN	1.04×10^6	1.63×10^6	2.82×10^8	3.52×10^8	5.42×10^7	6.49×10^8	1.53×10^9	5.83×10^7	4.26×10^8	2.22×10^6
	MEAN	3.91×10^6	7.97×10^6	4.96×10^8	7.72×10^8	2.05×10^8	1.11×10^9	1.99×10^9	1.79×10^8	1.19×10^9	1.12×10^7
	STD	2.28×10^6	4.64×10^6	1.68×10^8	2.59×10^8	9.96×10^7	2.42×10^8	2.53×10^8	9.38×10^7	4.08×10^8	5.76×10^6
CEC2	MIN	4.11×10^3	1.36×10^4	1.76×10^{10}	3.19×10^{10}	4.30×10^9	6.54×10^{10}	7.74×10^{10}	6.73×10^9	4.77×10^{10}	9.76×10^4
	MEAN	2.72×10^4	2.83×10^5	2.90×10^{10}	5.94×10^{10}	7.41×10^9	7.37×10^{10}	8.67×10^{10}	1.55×10^{10}	7.06×10^{10}	1.28×10^7
	STD	2.79×10^4	3.27×10^5	5.85×10^9	1.38×10^{10}	2.42×10^9	4.98×10^9	4.91×10^9	4.83×10^9	1.00×10^{10}	5.40×10^7
CEC3	MIN	1.77×10^3	2.62×10^3	4.44×10^4	5.80×10^4	5.48×10^4	6.52×10^4	9.39×10^4	4.83×10^4	6.66×10^4	5.22×10^3
	MEAN	5.44×10^3	6.90×10^3	7.65×10^4	8.84×10^4	1.27×10^5	7.84×10^4	9.21×10^5	6.29×10^4	8.39×10^4	1.06×10^4
	STD	1.87×10^3	3.69×10^3	1.78×10^4	7.90×10^3	6.42×10^4	7.39×10^3	1.33×10^6	9.32×10^3	8.35×10^3	4.16×10^3
CEC4	MIN	4.19×10^2	4.88×10^2	1.72×10^3	4.23×10^3	8.31×10^2	6.30×10^3	1.18×10^4	8.52×10^2	5.07×10^3	4.90×10^2
	MEAN	5.19×10^2	5.43×10^2	2.83×10^3	9.27×10^3	1.39×10^3	1.10×10^4	1.79×10^4	1.47×10^3	5.70×10^2	5.70×10^2
	STD	3.79×10^1	4.79×10^1	7.54×10^2	3.22×10^3	4.24×10^2	3.11×10^3	3.23×10^3	5.36×10^2	4.53×10^3	5.05×10^1
CEC5	MIN	5.20×10^2	5.20×10^2	5.21×10^2							
	MEAN	5.20×10^2	5.20×10^2	5.21×10^2							
	STD	2.38×10^{-1}	3.06×10^{-1}	5.20×10^{-2}	8.38×10^{-2}	1.08×10^{-1}	6.51×10^{-2}	7.36×10^{-2}	9.35×10^{-2}	7.12×10^{-2}	1.75×10^{-1}
CEC6	MIN	6.18×10^2	6.24×10^2	6.35×10^2	6.35×10^2	6.35×10^2	6.38×10^2	6.43×10^2	6.28×10^2	6.36×10^2	6.17×10^2
	MEAN	6.25×10^2	6.30×10^2	6.39×10^2	6.40×10^2	6.39×10^2	6.40×10^2	6.46×10^2	6.34×10^2	6.39×10^2	6.23×10^2
	STD	2.93×100	3.72×100	2.29×100	2.95×100	3.27×100	1.89×100	1.91×100	3.57×100	2.82×100	4.72×100
CEC7	MIN	7.00×10^2	7.00×10^2	8.76×10^2	9.58×10^2	7.26×10^2	1.12×10^3	1.34×10^3	7.39×10^2	1.14×10^3	7.00×10^2
	MEAN	7.00×10^2	7.00×10^2	9.62×10^2	1.13×10^3	7.47×10^2	1.33×10^3	1.57×10^3	8.37×10^2	1.35×10^3	7.01×10^2
	STD	4.82×10^{-2}	2.23×10^{-1}	4.99×10^1	9.85×10^1	1.40×10^1	1.03×10^2	8.82×10^1	5.37×10^1	9.97×10^1	2.84×10^{-1}
CEC8	MIN	8.56×10^2	9.10×10^2	1.05×10^3	1.06×10^3	9.92×10^2	1.13×10^3	1.18×10^3	9.77×10^2	1.09×10^3	8.61×10^2
	MEAN	9.02×10^2	9.33×10^2	1.08×10^3	1.12×10^3	1.03×10^3	1.16×10^3	1.22×10^3	1.03×10^3	1.15×10^3	8.92×10^2
	STD	2.92×10^1	1.97×10^1	2.46×10^1	2.94×10^1	4.83×10^1	1.93×10^1	4.28×10^1	2.84×10^1	3.53×10^1	2.38×10^1
CEC9	MIN	1.01×10^3	1.03×10^3	1.18×10^3	1.20×10^3	1.14×10^3	1.21×10^3	1.29×10^3	1.08×10^3	1.20×10^3	9.84×10^2
	MEAN	1.06×10^3	1.08×10^3	1.23×10^3	1.25×10^3	1.24×10^3	1.25×10^3	1.32×10^3	1.14×10^3	1.24×10^3	1.03×10^3
	STD	2.35×10^1	2.50×10^1	2.83×10^1	2.85×10^1	6.58×10^1	2.10×10^1	2.56×10^1	3.20×10^1	2.15×10^1	2.99×10^1
CEC10	MIN	1.69×10^3	3.28×10^3	7.32×10^3	6.98×10^3	5.74×10^3	7.44×10^3	8.82×10^3	5.51×10^3	6.55×10^3	2.82×10^3
	MEAN	2.78×10^3	4.74×10^3	8.00×10^3	7.69×10^3	6.45×10^3	7.90×10^3	9.91×10^3	7.02×10^3	7.39×10^3	4.09×10^3
	STD	4.72×10^2	1.10×10^3	4.09×10^2	7.01×10^2	7.26×10^2	5.58×10^2	7.55×10^2	8.43×10^2	6.20×10^2	1.25×10^3
CEC11	MIN	1.23×10^3	1.57×10^3	2.26×10^3	1.85×10^3	1.59×10^3	2.26×10^3	2.99×10^3	1.70×10^3	1.71×10^3	1.27×10^3
	MEAN	1.63×10^3	2.01×10^3	2.56×10^3	2.56×10^3	2.22×10^3	2.61×10^3	3.48×10^3	2.25×10^3	2.11×10^3	1.99×10^3
	STD	1.75×10^2	3.45×10^2	2.54×10^2	2.59×10^2	3.67×10^2	2.28×10^2	2.77×10^2	3.19×10^2	2.84×10^2	4.29×10^2
CEC12	MIN	1.20×10^3									
	MEAN	1.20×10^3									
	STD	1.37×10^{-1}	2.15×10^{-1}	2.91×10^{-1}	3.49×10^{-1}	5.20×10^{-1}	4.10×10^{-1}	7.98×10^{-1}	3.41×10^{-1}	2.77×10^{-1}	3.64×10^{-1}
CEC13	MIN	1.30×10^3									
	MEAN	1.30×10^3									
	STD	6.59×10^{-2}	1.02×10^{-1}	1.58×10^{-1}	1.08	1.89×10^{-1}	6.43×10^{-1}	1.13	1.64×10^{-1}	1.19	9.69×10^{-2}
CEC14	MIN	1.40×10^3	1.41×10^3	1.43×10^3	1.40×10^3	1.41×10^3	1.40×10^3				
	MEAN	1.40×10^3	1.40×10^3	1.41×10^3	1.41×10^3	1.40×10^3	1.41×10^3	1.44×10^3	1.40×10^3	1.43×10^3	1.40×10^3
	STD	6.10×10^{-2}	2.36×10^{-1}	6.06×10^{-1}	8.30	2.40×10^{-1}	5.84	1.06×10^1	1.03	9.46	2.04×10^{-1}
CEC15	MIN	1.50×10^3	1.50×10^3	1.51×10^3	1.51×10^3	1.50×10^3	1.76×10^3	3.23×10^3	1.50×10^3	1.58×10^3	1.50×10^3
	MEAN	1.50×10^3	1.50×10^3	1.59×10^3	2.20×10^3	1.51×10^3	5.83×10^3	2.19×10^4	1.66×10^3	5.27×10^3	1.50×10^3
	STD	4.57×10^{-1}	1.49	4.45×10^2	1.51×10^3	5.63	3.46×10^3	2.19×10^4	8.45×10^2	5.14×10^3	5.89×10^{-1}
CEC16	MIN	1.60×10^3									
	MEAN	1.60×10^3									
	STD	3.06×10^{-1}	3.86×10^{-1}	2.39×10^{-1}	3.20×10^{-1}	3.69×10^{-1}	2.15×10^{-1}	2.86×10^{-1}	3.03×10^{-1}	3.05×10^{-1}	3.51×10^{-1}
CEC17	MIN	1.77×10^3	1.96×10^3	8.48×10^3	3.77×10^3	7.57×10^3	2.59×10^5	6.07×10^5	5.17×10^3	2.89×10^4	2.24×10^3
	MEAN	2.39×10^3	2.61×10^3	6.86×10^4	4.35×10^5	2.41×10^5	4.84×10^5	6.05×10^6	1.06×10^5	3.95×10^5	9.16×10^3
	STD	3.66×10^2	6.66×10^2	7.75×10^4	3.62×10^5	3.35×10^5	1.17×10^5	8.34×10^6	1.82×10^5	1.76×10^5	1.04×10^4
CEC18	MIN	1.81×10^3	1.82×10^3	5.30×10^3	2.51×10^3	2.02×10^3	8.98×10^3	2.19×10^5	2.36×10^3	1.95×10^3	1.97×10^3
	MEAN	1.85×10^3	1.90×10^3	3.55×10^4	1.20×10^4	9.68×10^3	1.47×10^5	2.12×10^7	1.48×10^4	1.09×10^4	9.75×10^3

Table 5. Cont.

CEC2014	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA
CEC24	MIN	2.51×10^3	2.53×10^3	2.55×10^3	2.60×10^3	2.54×10^3	2.59×10^3	2.60×10^3	2.53×10^3	2.56×10^3	2.52×10^3
	MEAN	2.57×10^3	2.58×10^3	2.56×10^3	2.60×10^3	2.58×10^3	2.60×10^3	2.60×10^3	2.55×10^3	2.59×10^3	2.56×10^3
	STD	2.50×10^1	2.93×10^1	7.07	0	2.99×10^1	4.07	0	2.46×10^1	1.81×10^1	3.64×10^1
CEC25	MIN	2.63×10^3	2.70×10^3	2.67×10^3	2.70×10^3	2.67×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.67×10^3
	MEAN	2.69×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.69×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.68×10^3
	STD	1.23×10^1	1.70×10^1	1.44×10^1	0	1.53×10^1	0	0	1.55	3.43	2.70×10^1
CEC26	MIN	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3
	MEAN	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.70×10^3	2.71×10^3	2.70×10^3	2.70×10^3	2.70×10^3
	STD	7.71×10^{-2}	8.27×10^{-2}	2.06×10^{-1}	1.80×10^1	1.82×10^{-1}	7.36×10^{-1}	1.75×10^1	1.36×10^{-1}	2.45×10^1	1.17×10^{-1}
CEC27	MIN	2.70×10^3	2.70×10^3	2.72×10^3	2.90×10^3	2.71×10^3	2.90×10^3	2.90×10^3	3.10×10^3	2.90×10^3	2.90×10^3
	MEAN	2.83×10^3	2.84×10^3	3.05×10^3	2.89×10^3	3.18×10^3	2.90×10^3	2.90×10^3	3.14×10^3	2.92×10^3	3.00×10^3
	STD	4.96×10^1	9.30×10^1	1.43×10^2	3.22×10^1	1.19×10^2	0	0	5.98×10^1	1.02×10^2	9.06×10^1
CEC28	MIN	3.00×10^3	3.00×10^3	3.24×10^3	3.00×10^3	3.00×10^3	3.00×10^3	3.00×10^3	3.17×10^3	3.00×10^3	3.03×10^3
	MEAN	3.00×10^3	3.00×10^3	3.30×10^3	3.00×10^3	3.46×10^3	3.00×10^3	3.00×10^3	3.19×10^3	3.00×10^3	3.22×10^3
	STD	0	0	5.49×10^1	0	2.05×10^2	0	0	9.96×100	0	1.44×10^2
CEC29	MIN	3.10×10^3	3.11×10^3	5.34×10^3	3.10×10^3	3.22×10^3	3.10×10^3	3.10×10^3	3.48×10^3	3.10×10^3	3.33×10^3
	MEAN	3.10×10^3	6.88×10^4	2.26×10^4	2.05×10^5	2.99×10^5	3.10×10^3	3.10×10^3	5.62×10^3	7.72×10^3	8.44×10^5
	STD	0	3.59×10^5	1.89×10^4	7.73×10^5	8.25×10^5	0	0	2.70×10^3	2.33×10^6	1.28×10^6
CEC30	MIN	3.20×10^3	3.49×10^3	4.13×10^3	3.73×10^3	3.79×10^3	3.20×10^3	3.20×10^3	3.79×10^3	4.89×10^3	3.46×10^3
	MEAN	3.20×10^3	3.91×10^3	5.17×10^3	9.73×10^3	6.36×10^3	3.20×10^3	3.20×10^3	4.14×10^3	4.88×10^4	1.31×10^4
	STD	0	4.51×10^2	1.07×10^3	1.11×10^4	1.99×10^3	0	0	2.74×10^2	1.06×10^5	3.24×10^4

Table 6 shows the results of all algorithms for all CEC2020 test functions. It can be seen from the table that the MGTO algorithm provided the optimal results in 10 benchmark functions, no matter the optimal value or the average value, and the MGTO algorithm provided seven optimal results. This shows that the MGTO algorithm has a very significant optimization effect. TD. For CEC3, CEC8, and CEC9, the MGTO algorithm was not optimal for STD in numerical value, but its stability was also stronger than those of most other algorithms.

Table 6. CEC2020 Algorithm Results of Benchmark Functions.

CEC2020	Metric	MGTO	GTO	SCA	ROA	WOA	RSA	SHO	SOA	AOA	HBA
CEC1	MIN	1.00×10^2	1.02×10^2	5.21×10^8	1.29×10^9	5.98×10^6	6.48×10^9	7.07×10^9	1.25×10^7	3.59×10^9	1.78×10^2
	MEAN	8.74×10^2	2.57×10^3	1.15×10^9	4.79×10^9	6.72×10^7	1.19×10^{10}	1.62×10^{10}	4.00×10^8	1.01×10^{10}	4.59×10^3
	STD	5.15×10^2	3.18×10^3	3.57×10^8	3.22×10^9	8.06×10^7	4.44×10^9	5.00×10^9	3.69×10^8	4.32×10^9	3.93×10^3
CEC2	MIN	1.15×10^3	1.50×10^3	2.26×10^3	2.09×10^3	1.72×10^3	2.33×10^3	3.04×10^3	1.66×10^3	1.90×10^3	1.47×10^3
	MEAN	1.66×10^3	2.02×10^3	2.55×10^3	2.59×10^3	2.33×10^3	2.78×10^3	3.47×10^3	2.11×10^3	2.29×10^3	1.89×10^3
	STD	2.15×10^2	3.02×10^2	2.51×10^2	2.80×10^2	3.13×10^2	2.46×10^2	2.80×10^2	2.23×10^2	2.70×10^2	4.87×10^2
CEC3	MIN	7.07×10^2	7.26×10^2	7.70×10^2	7.74×10^2	7.55×10^2	7.97×10^2	8.42×10^2	7.55×10^2	7.74×10^2	7.21×10^2
	MEAN	7.39×10^2	7.55×10^2	7.87×10^2	8.16×10^2	8.02×10^2	8.12×10^2	8.73×10^2	7.69×10^2	8.00×10^2	7.39×10^2
	STD	1.46×10^1	1.73×10^1	1.41×10^1	2.89×10^1	2.83×10^1	1.13×10^1	2.26×10^1	1.58×10^1	1.65×10^1	1.47×10^1
CEC4	MIN	1.90×10^3									
	MEAN	1.90×10^3	1.90×10^3	1.90×10^3	1.90×10^3	1.90×10^3	1.90×10^3	1.90×10^3	1.90×10^3	1.90×10^3	1.90×10^3
	STD	0	0	1.32	0	4.09×10^{-1}	0	0	4.46×10^{-1}	0	0
CEC5	MIN	1.78×10^3	1.94×10^3	1.25×10^4	5.46×10^3	3.68×10^3	3.26×10^5	4.12×10^5	5.66×10^3	9.75×10^3	1.87×10^3
	MEAN	2.35×10^3	2.54×10^3	1.02×10^5	3.38×10^5	2.13×10^5	4.84×10^5	3.90×10^6	1.18×10^5	3.63×10^5	6.28×10^3
	STD	2.73×10^2	5.79×10^2	1.16×10^5	2.98×10^5	3.73×10^5	7.88×10^4	5.11×10^6	1.85×10^5	1.72×10^5	8.09×10^3
CEC6	MIN	1.60×10^3	1.60×10^3	1.71×10^3	1.74×10^3	1.71×10^3	1.92×10^3	2.25×10^3	1.66×10^3	1.84×10^3	1.60×10^3
	MEAN	1.64×10^3	1.75×10^3	1.87×10^3	1.92×10^3	1.87×10^3	2.27×10^3	2.75×10^3	1.82×10^3	2.09×10^3	1.81×10^3
	STD	5.06×10^1	1.21×10^2	9.28×10^1	1.25×10^2	1.30×10^2	1.71×10^2	3.37×10^2	9.10×10^1	1.89×10^2	1.87×10^2
CEC7	MIN	2.10×10^3	2.11×10^3	4.96×10^3	2.93×10^3	8.95×10^3	2.35×10^4	3.66×10^4	3.11×10^3	5.39×10^3	2.18×10^3
	MEAN	2.36×10^3	2.64×10^3	1.52×10^4	2.19×10^5	1.06×10^6	3.59×10^6	3.59×10^6	1.80×10^4	7.18×10^5	3.05×10^3
	STD	2.07×10^2	6.00×10^2	1.07×10^4	7.49×10^5	2.93×10^6	2.02×10^6	3.57×10^6	3.67×10^4	1.14×10^6	7.36×10^2
CEC8	MIN	2.20×10^3	2.30×10^3	2.35×10^3	2.36×10^3	2.28×10^3	2.74×10^3	3.21×10^3	2.26×10^3	2.66×10^3	2.30×10^3
	MEAN	2.30×10^3	2.30×10^3	2.46×10^3	2.74×10^3	2.52×10^3	3.29×10^3	3.97×10^3	3.02×10^3	3.10×10^3	2.30×10^3
	STD	1.55×10^1	1.61×10^1	2.93×10^2	3.98×10^2	5.31×10^2	3.30×10^2	4.78×10^2	7.18×10^2	3.37×10^2	1.16×10^1
CEC9	MIN	2.50×10^3	2.50×10^3	2.77×10^3	2.72×10^3	2.60×10^3	2.81×10^3	2.86×10^3	2.74×10^3	2.77×10^3	2.74×10^3
	MEAN	2.65×10^3	2.70×10^3	2.79×10^3	2.80×10^3	2.79×10^3	2.90×10^3	2.96×10^3	2.76×10^3	2.86×10^3	2.74×10^3
	STD	1.22×10^2	1.11×10^2	3.66×10^1	7.18×10^1	5.20×10^1	6.80×10^1	7.01×10^1	4.58×10^1	8.25×10^1	6.91×10^1
CEC10	MIN	2.60×10^3	2.90×10^3	2.94×10^3	2.97×10^3	2.91×10^3	3.23×10^3	3.48×10^3	2.92×10^3	3.11×10^3	2.90×10^3
	MEAN	2.92×10^3	2.94×10^3	2.99×10^3	3.22×10^3	2.98×10^3	3.51×10^3	3.90×10^3	2.95×10^3	3.39×10^3	2.93×10^3
	STD	2.15×10^1	3.19×10^1	3.20×10^1	2.13×10^2	8.10×10^1	2.49×10^2	4.14×10^2	3.69×10^1	2.32×10^2	2.30×10^1

Analysis of Wilcoxon Rank-Sum Test Results

According to this analysis, the MGTO algorithm achieved good results in the CEC2014 and CEC2020 tests. Tables 7 and 8 show the Wilcoxon rank-sum test results for the CEC2014 and CEC2020 results, respectively. In most functions' results, the p-values are less than 0.05,

whereas a small part is greater than 0.05, indicating that the fitness values obtained by the two algorithms in these functions are similar, and there is no significant difference between them. In the results for the CEC2020 function, CEC4 function, CEC23 function, and CEC28 function, many values of 1 indicate that the MGTO algorithm obtained the same value as these comparison algorithms. In most cases, they are less than 0.05, which shows that the similarity between the MGTO algorithm and other algorithms is very low, and there are significant differences.

Table 7. CEC2014 Experimental Results of Wilcoxon Rank-Sum Test on Benchmark Functions.

CEC2014	MGTO VS GTO	MGTO VS SCA	MGTO VS ROA	MGTO VS WOA	MGTO VS RSA	MGTO VS SHO	MGTO VS SOA	MGTO VS AOA	MGTO VS HBA
CEC1	3.32×10^{-4}	1.73×10^{-6}	2.41×10^{-4}						
CEC2	2.60×10^{-6}	1.73×10^{-6}							
CEC3	3.59×10^{-4}	1.73×10^{-6}	2.88×10^{-6}						
CEC4	4.28×10^{-2}	1.73×10^{-6}	4.65×10^{-1}						
CEC5	2.85×10^{-2}	1.73×10^{-6}	1.73×10^{-6}	1.92×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.92×10^{-6}
CEC6	3.88×10^{-4}	1.73×10^{-6}	1.73×10^{-6}	1.92×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.64×10^{-5}	1.73×10^{-6}	9.63×10^{-4}
CEC7	9.32×10^{-6}	1.73×10^{-6}							
CEC8	5.75×10^{-6}	1.73×10^{-6}	8.22×10^{-2}						
CEC9	7.34×10^{-1}	1.73×10^{-6}	4.29×10^{-6}	1.73×10^{-6}	6.98×10^{-6}				
CEC10	2.35×10^{-6}	1.73×10^{-6}	5.22×10^{-6}						
CEC11	1.89×10^{-4}	1.73×10^{-6}	1.73×10^{-6}	2.35×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	2.35×10^{-6}	2.88×10^{-6}	2.16×10^{-5}
CEC12	3.82×10^{-1}	1.73×10^{-6}	1.92×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.97×10^{-5}	4.72×10^{-2}
CEC13	1.64×10^{-5}	1.73×10^{-6}	1.73×10^{-6}	1.92×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.02×10^{-5}
CEC14	2.22×10^{-4}	1.73×10^{-6}	2.88×10^{-6}	2.41×10^{-3}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	2.84×10^{-5}
CEC15	1.11×10^{-3}	1.73×10^{-6}	8.29×10^{-1}						
CEC16	2.58×10^{-3}	1.73×10^{-6}	2.88×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.92×10^{-6}	2.18×10^{-2}
CEC17	4.05×10^{-1}	1.73×10^{-6}	4.29×10^{-6}						
CEC18	1.85×10^{-2}	1.73×10^{-6}							
CEC19	4.20×10^{-4}	1.73×10^{-6}	8.47×10^{-6}	1.73×10^{-6}	2.05×10^{-4}				
CEC20	8.94×10^{-4}	1.73×10^{-6}							
CEC21	5.45×10^{-2}	1.73×10^{-6}	1.92×10^{-6}						
CEC22	3.39×10^{-1}	1.73×10^{-6}	5.75×10^{-6}	2.35×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	2.13×10^{-6}	3.32×10^{-4}
CEC23	1	1.73×10^{-6}	1	2.56×10^{-6}	1	1	1.73×10^{-6}	1	9.77×10^{-4}
CEC24	6.96×10^{-1}	1.92×10^{-6}	1	1.75×10^{-2}	1	1	2.60×10^{-6}	3.42×10^{-2}	1.15×10^{-4}
CEC25	1.74×10^{-2}	1.73×10^{-6}	2.44×10^{-4}	4.88×10^{-2}	2.44×10^{-4}	2.44×10^{-4}	2.13×10^{-6}	2.44×10^{-4}	2.47×10^{-1}
CEC26	5.04×10^{-1}	1.73×10^{-6}	1.92×10^{-6}	2.88×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	4.29×10^{-6}	1.73×10^{-6}	7.81×10^{-1}
CEC27	5.37×10^{-2}	1.24×10^{-5}	5.00×10^{-1}	1.92×10^{-6}	5.00×10^{-1}	5.00×10^{-1}	3.18×10^{-6}	5.00×10^{-1}	8.22×10^{-3}
CEC28	1	1.73×10^{-6}	1	1.73×10^{-6}	1	1	1.73×10^{-6}	5.00×10^{-1}	1.73×10^{-6}
CEC29	1.73×10^{-6}	1.73×10^{-6}	5.95×10^{-5}	1.73×10^{-6}	1	1	1.73×10^{-6}	1.22×10^{-4}	1.73×10^{-6}
CEC30	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1	1	1.73×10^{-6}	5.61×10^{-6}	1.73×10^{-6}

Table 8. CEC2020 Experimental Results of Wilcoxon Rank-Sum Test on Benchmark Functions.

CEC2020	MGTO VS GTO	MGTO VS SCA	MGTO VS ROA	MGTO VS WOA	MGTO VS RSA	MGTO VS SHO	MGTO VS SOA	MGTO VS AOA	MGTO VS HBA
CEC1	4.72×10^{-2}	1.73×10^{-6}	7.19×10^{-2}						
CEC2	1.48×10^{-4}	1.73×10^{-6}	1.73×10^{-6}	7.69×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	4.45×10^{-5}	4.29×10^{-6}	1.25×10^{-2}
CEC3	2.22×10^{-4}	1.73×10^{-6}	2.35×10^{-6}	2.35×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	2.60×10^{-5}	2.13×10^{-6}	4.07×10^{-2}
CEC4	1	1.22×10^{-4}	1	3.13×10^{-2}	1	1	2.50×10^{-1}	1	1
CEC5	4.72×10^{-2}	1.73×10^{-6}	1.64×10^{-5}						
CEC6	2.61×10^{-4}	1.73×10^{-6}	4.29×10^{-6}	3.18×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	8.92×10^{-5}	1.73×10^{-6}	1.36×10^{-4}
CEC7	4.11×10^{-3}	1.73×10^{-6}	1.92×10^{-6}	9.32×10^{-6}					
CEC8	2.70×10^{-2}	1.73×10^{-6}	6.87×10^{-2}						
CEC9	2.58×10^{-3}	1.24×10^{-5}	1.73×10^{-6}	2.88×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	3.72×10^{-5}	1.92×10^{-6}	1.11×10^{-3}
CEC10	2.80×10^{-1}	4.73×10^{-6}	1.73×10^{-6}	1.92×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	3.82×10^{-1}	1.73×10^{-6}	6.73×10^{-1}

4.3. Experimental Analysis between Exploration and Exploitation

In order to show that the exploration and exploitation of MGTO algorithm reach a good balance, we carried out an exploration and exploitation balance experiment on 30

CEC2014 functions and 10 CEC2020 functions for the MGTO algorithm. The following Figures 6 and 7 show the specific results. Figure 6 shows the balance between exploration and exploitation within MGTO algorithm when tested on the CEC2014 function. It can be seen from the figure that the optimal value of most functions was obtained by a search balance of nearly 90% exploitation and 10% exploration. For the best solution to the fifth function of CEC2014, the algorithm was based on exploration. In the test of 10 CEC2020 functions, the MGTO algorithm also obtained most results at the balance point of 90% exploitation and 10% exploration. Of course, only proper balance is not enough to achieve good results. In most of the functions of the CEC2014 and CEC2020 sets, the MGTO algorithm maintained an irregularly balanced response, which means that the shares of exploration and exploitation continued to fluctuate. The algorithm is not always limited to local exploitation, but it also pays attention to the comprehensive exploration of the optimal solution that jumps out of the local optimum, which reflects the improvement of the algorithm's performance. It is also necessary to produce promising solutions to make full use of the appropriate diversity of responses obtained within such a balance. Therefore, the MGTO algorithm has a good exploration and exploitation mechanism that makes the search scope comprehensive and extensive, which it then fully develops to obtain the global optimal value and achieve the effect of rapid convergence.

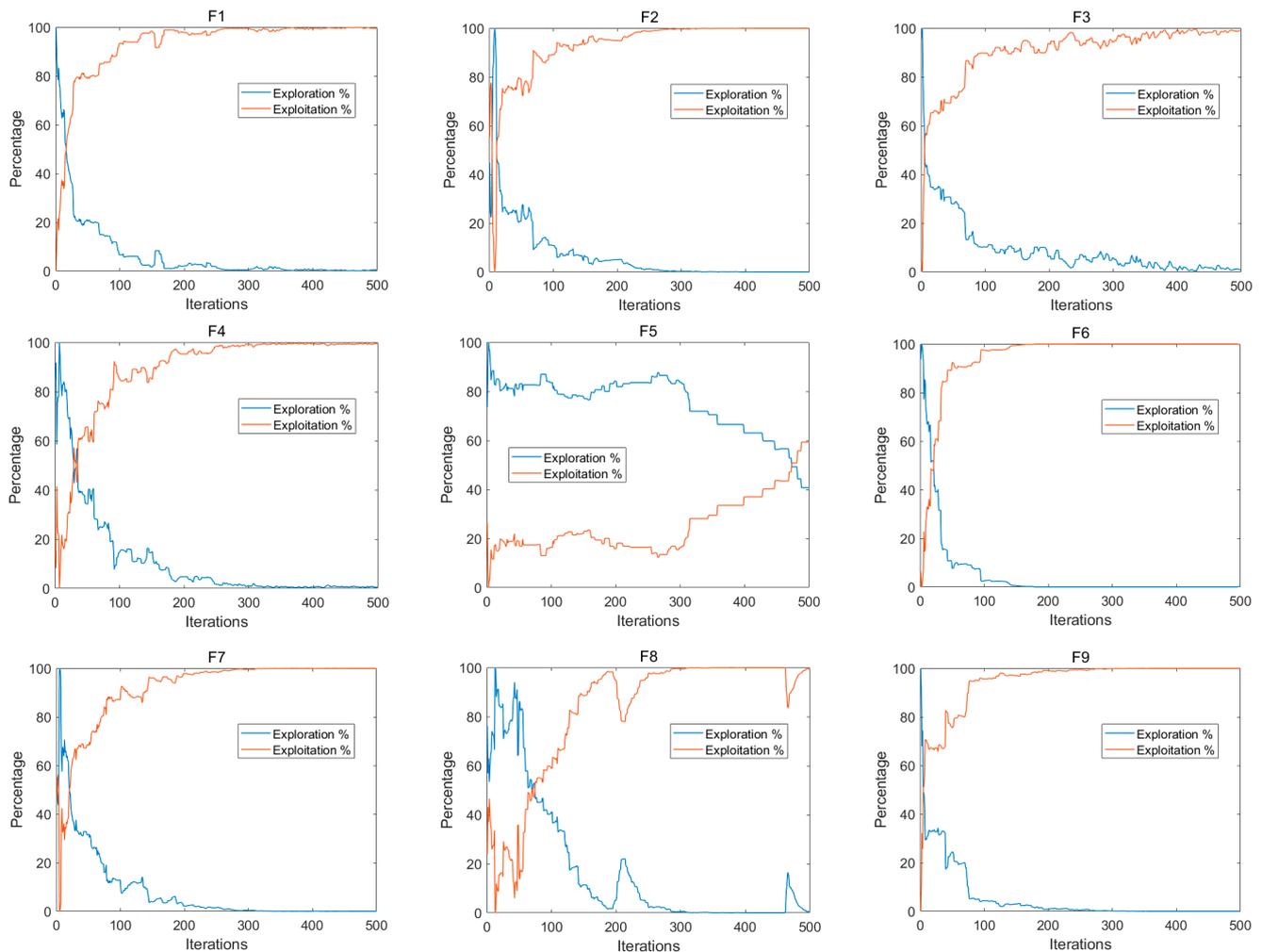


Figure 6. Cont.

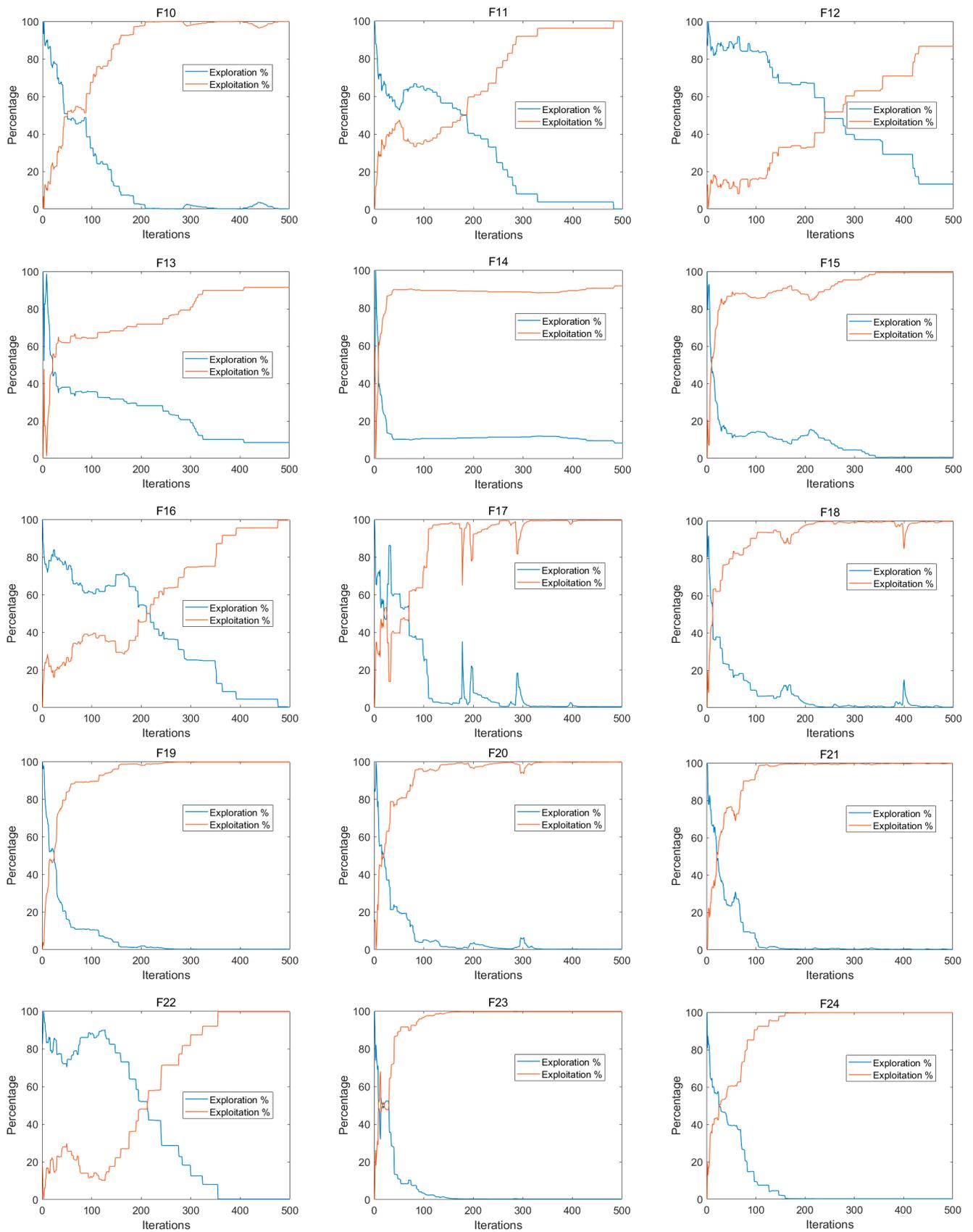


Figure 6. Cont.

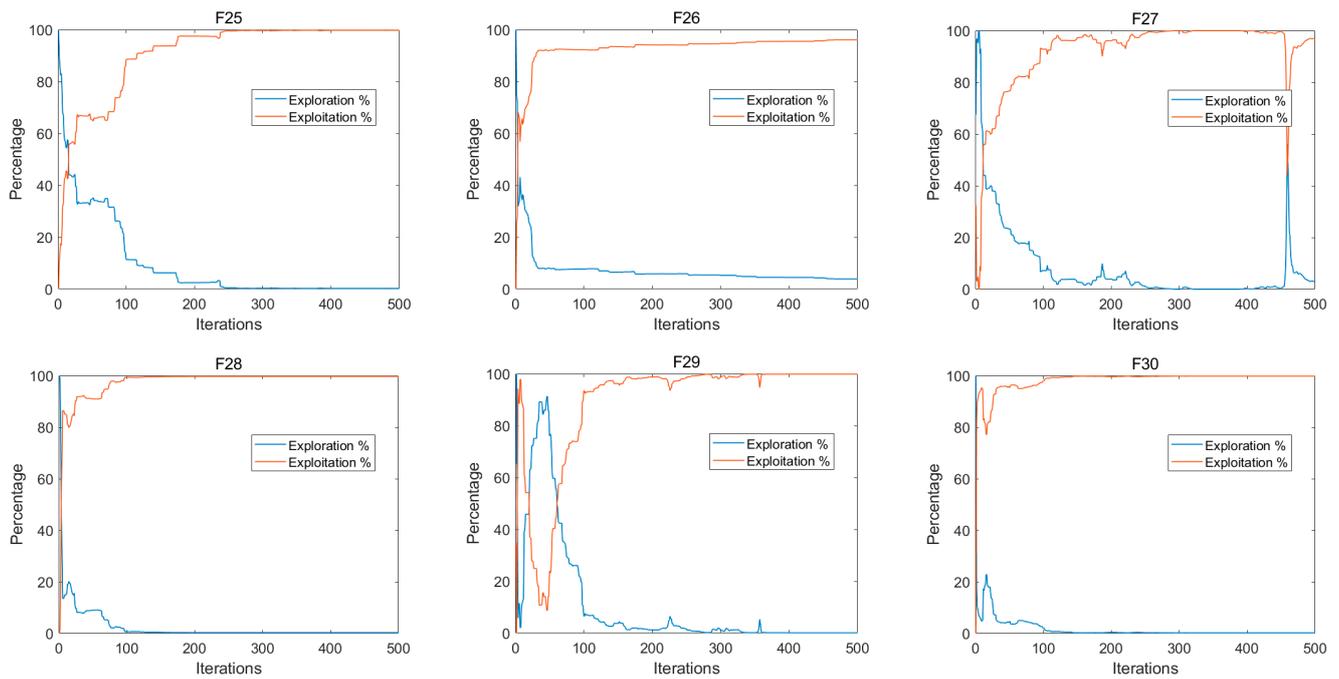


Figure 6. Exploration and exploitation phases for the MGTO algorithm using CEC2014.

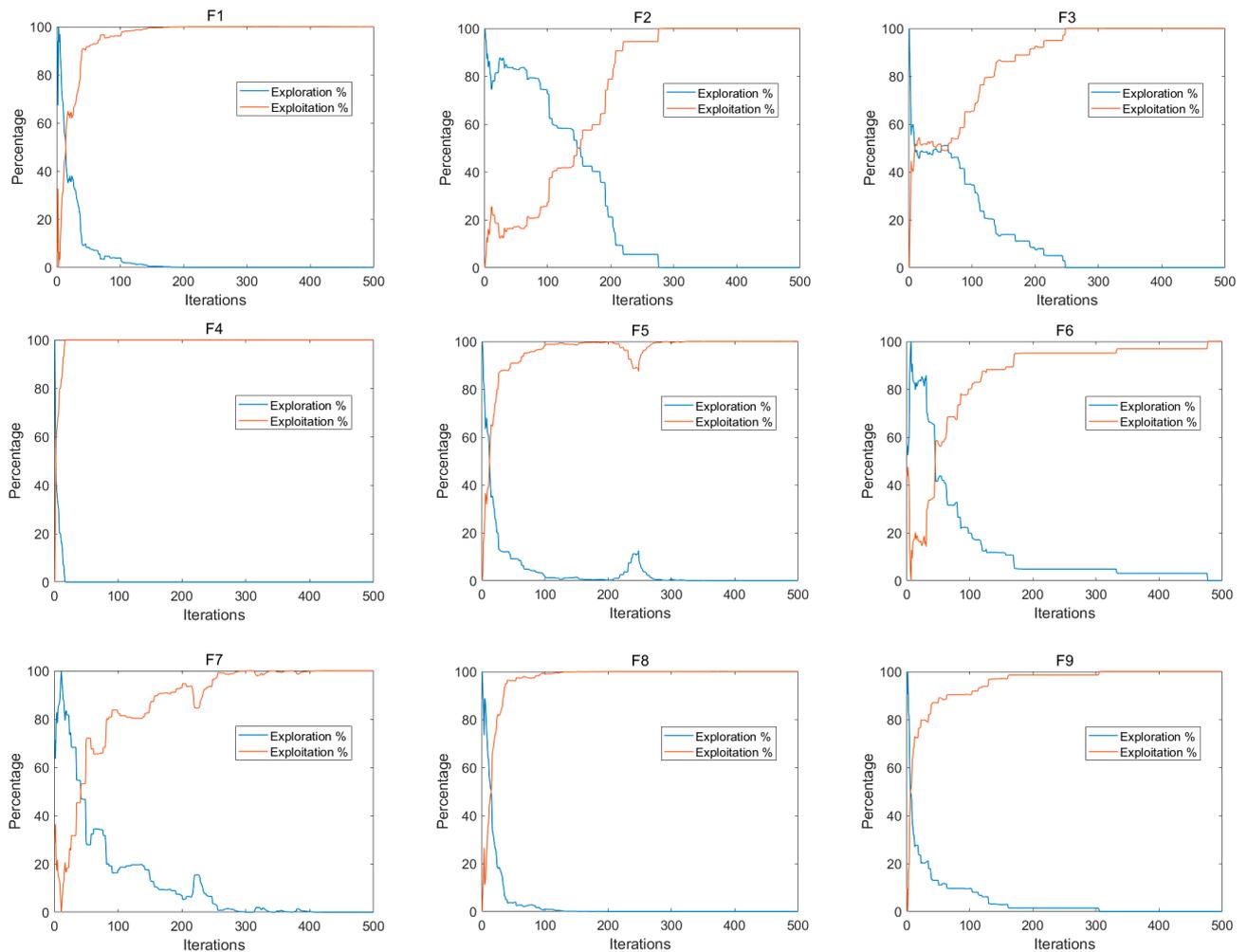


Figure 7. Cont.

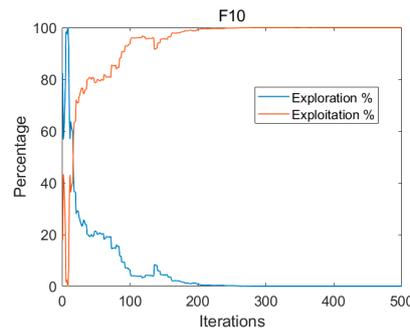


Figure 7. Exploration and exploitation phases for the MGTO algorithm using CEC2020.

5. Constrained Engineering Design Problems

In this section, seven engineering problems were selected to test the optimization performance of the MGTO algorithm. The selected engineering problems are those of pressure vessel design, reducer design, welding beam design, tension/compression spring design, cantilever beam design, multi plate clutch brake failure, and vehicle collision optimization. Each algorithm was run independently 30 times. The specific experimental results are as follows.

5.1. Pressure Vessel Design Problem

The pressure vessel design problem aims to minimize the total cost of cylindrical pressure vessels to meet pressure requirements. There are four variables to be optimized for this problem, namely: vessel wall thickness T_s , head wall thickness T_h , inner diameter R , and vessel body length L . See Figure 8 for a detailed schematic diagram. The specific mathematical model is as follows:

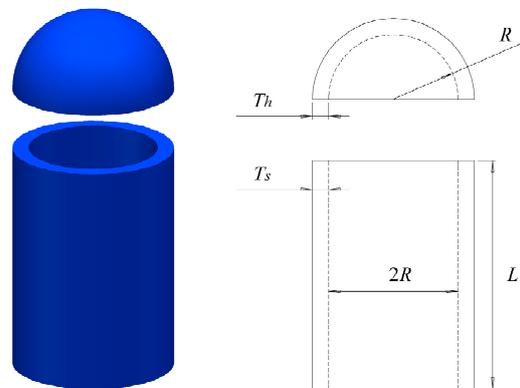


Figure 8. Model of pressure vessel design.

Consider:

$$\vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ R \ L] \tag{41}$$

Objective function:

$$f(\vec{x}) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{42}$$

Subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0 \tag{43}$$

$$g_2(\vec{x}) = -x_3 + 0.00954x_3 \leq 0 \tag{44}$$

$$g_3(\vec{x}) = -\pi x_3^2x_4 + \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \tag{45}$$

$$g_4(\vec{x}) = -x_4 - 240 \leq 0 \tag{46}$$

Variable Range:

$$0 \leq x_1 \leq 99, 0 \leq x_2 \leq 99, 10 \leq x_3 \leq 200, 10 \leq x_4 \leq 200 \tag{47}$$

The results of this problem are shown in Table 9. They show that the improved MGTO algorithm has advantages in solving pressure vessel design problems. It can be seen from the table that the MGTO algorithm obtained $T_s = 0.7424$, $T_h = 0.3702$, $R = 40.3196$, and $L = 200$; thus, the lowest cost of pressure vessels is 5734.9132. Among the algorithms used for comparison, eight algorithms generated cost values greater than 6000, and three algorithms generated cost values less than 6000.

Table 9. Experimental results of the pressure vessel design problem test.

Algorithm	T_s	T_h	R	L	Best Cost
MGTO	0.7424	0.3702	40.3196	200.0000	5734.9132
MGTOA [29]	0.754363	0.366375	40.42809	198.5652	5752.402458
EO [2]	1.0480	0.5114	55.3848	60.6465	6470.6125
HBA [28]	1.2404	0.5844	65.2252	10.0000	7141.3612
GWO [14]	0.8125	0.4375	42.0984	176.6366	6059.7143
HHO [18]	1.2492	0.5810	65.2139	10.0487	7149.8665
WOA [19]	0.8125	0.4375	42.09845	176.6366	6059.714335
MVO [3]	1.2269	0.5928	64.3294	14.0578	7106.0065
ACO [30]	0.8125	0.4375	42.10362	176.7387	6059.0888
NGO [31]	0.7427	0.3708	40.3199	200.0000	5735.0462
EROA [32]	0.84343	0.400762	44.786	145.9578	5935.7301
PSO [13]	0.8861	0.4306	46.9699	124.3784	6024.2816

5.2. Speed Reducer Design Problem

The speed reducer design problem’s goal is to minimize the weight of the speed reducer. There are seven variable restrictions and four design constraints. The constraints are the bending stress of the gear teeth, covering stress, lateral deflection of the shaft, and stress in the shaft. The seven decision variables are tooth face width x_1 , gear module x_2 , number of teeth on pinion x_3 , length of the first shaft between bearings x_4 , length of the second shaft between bearings x_5 , the diameter of the first shaft x_6 , and diameter of the second shaft x_7 . The variable diagram is shown in Figure 9.

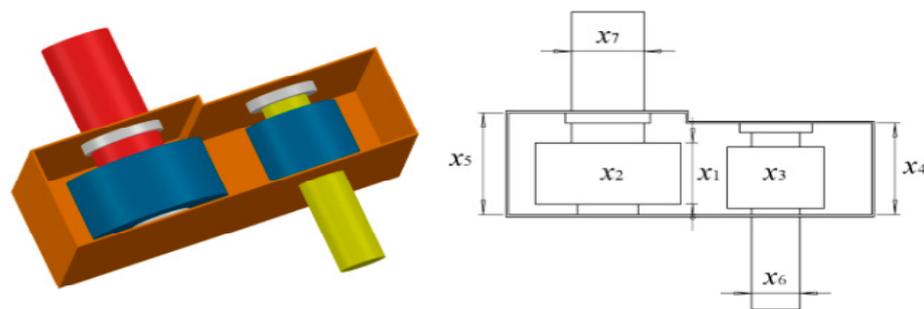


Figure 9. Model of speed reducer design.

The mathematical formulation of this problem is shown below:

Consider:

$$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7] \tag{48}$$

Objective function:

$$f(\vec{x}) = 07854 \times x_1 \times x_2^2 \times (3.3333 \times x_3^2 + 14.9334 \times x_3 - 43.0934) - 1.508 \times x_1 \times (x_6^2 + x_7^2) + 7.4777 \times x_6^3 + x_7^3 + 0.7854 \times x_4 \times x_6^2 + x_5 \times x_7^2 \tag{49}$$

Subject to:

$$g_1(\vec{x}) = \frac{27}{x_1 \times x_2^2 \times x_3} - 1 \leq 0 \tag{50}$$

$$g_2(\vec{x}) = \frac{397.5}{x_1 \times x_2^2 \times x_3^2} - 1 \leq 0 \tag{51}$$

$$g_3(\vec{x}) = \frac{1.93 \times x_4^3}{x_2 \times x_3 \times x_6^4} - 1 \leq 0 \tag{52}$$

$$g_4(\vec{x}) = \frac{1.93 \times x_5^3}{x_2 \times x_3 \times x_7^4} - 1 \leq 0 \tag{53}$$

$$g_5(\vec{x}) = \frac{1}{110 \times x_6^3} \times \sqrt{\left(\frac{745 \times x_4}{x_2 \times x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0 \tag{54}$$

$$g_6(\vec{x}) = \frac{1}{85 \times x_7^3} \times \sqrt{\left(\frac{745 \times x_5}{x_2 \times x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0 \tag{55}$$

$$g_7(\vec{x}) = \frac{x_2 \times x_3}{40} - 1 \leq 0 \tag{56}$$

$$g_8(\vec{x}) = \frac{5 \times x_2}{x_1} - 1 \leq 0 \tag{57}$$

$$g_9(\vec{x}) = \frac{x_1}{12 \times x_2} - 1 \leq 0 \tag{58}$$

$$g_{10}(\vec{x}) = \frac{1.5 \times x_6 + 1.9}{x_4} - 1 \leq 0 \tag{59}$$

$$g_{11}(\vec{x}) = \frac{1.1 \times x_7 + 1.9}{x_5} - 1 \leq 0 \tag{60}$$

Variable range:

$$2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5 \tag{61}$$

The MGTO algorithm obtained the optimal solution of 2988.2713, and the corresponding variable values for this solution are $x_1 = 3.47641$, $x_2 = 0.7$, $x_3 = 17$, $x_4 = 7.3$, $x_5 = 7.8$, $x_6 = 3.3486$, and $x_7 = 5.27678$. Therefore, the MGTO algorithm can effectively solve this problem. The results of the reducer design problem test are shown in Table 10.

Table 10. Experimental results of the speed reducer design problem test.

Algorithm	Optimal Values for Variables							Optimal Weight
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
MGTO	3.47641	0.7	17	7.3	7.8	3.3486	5.27678	2988.2713
MDA [33]	3.5	0.7	17	7.3	7.67039	3.54242	5.2481	3019.5833
MFO [34]	3.497455	0.7	17	7.82775	7.712457	3.351787	5.286352	2998.94083
CS [35]	3.5015	0.7	17	7.605	7.8181	3.352	5.2875	3000.981
RSA [27]	3.50279	0.7	17	7.30812	7.74715	3.35067	5.28675	2996.5157
HS [36]	3.520124	0.7	17	8.37	7.8	3.36697	5.288719	3029.002

5.3. Welded Beam Design Problem

As shown in Figure 10, the purpose of the welded beam design problem is to minimize the total cost of the design of welded beams. The problem is composed of four decision variables and seven constraints. The four decision variables are weld width h , connecting beam length l , beam height t , and connecting beam thickness b . The objective function, constraint conditions, and variable range of the problem are as follows:

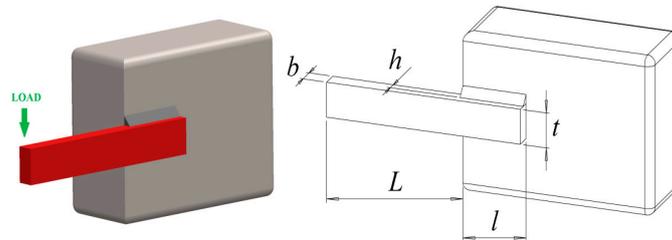


Figure 10. Model of welded beam design.

The mathematical formulation of this problem is shown below:

Consider:

$$x = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b] \tag{62}$$

Objective function:

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{63}$$

Subject to:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0 \tag{64}$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \leq 0 \tag{65}$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0 \tag{66}$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0 \tag{67}$$

$$g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0 \tag{68}$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0 \tag{69}$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 0.5 \leq 0 \tag{70}$$

where:

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2x_1x_2}}, \tau'' = \frac{MR}{J}, \tag{71}$$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \tag{72}$$

$$J = 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \delta(\vec{x}) = \frac{6PL^3}{Ex_4x_3^3}, \tag{73}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{x_3^2x_4^6}}{L^2}, \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \tag{74}$$

$$P = 6000lb, L = 14 \text{ in}, \delta_{\max} = 0.25\text{in}, E = 30 \times 10^6 \text{ psi}, \tag{75}$$

$$\tau_{\max} = 13600 \text{ psi}, \text{ and } \sigma_{\max} = 30000 \text{ psi} \tag{76}$$

Variable range:

$$0.1 \leq x_i \leq 2, i = 1, 4; 0.1 \leq x_i \leq 10, i = 2, 3 \tag{77}$$

Table 11 shows the operating results for the design of welded beams. It can be seen from the table that the MGTO algorithm obtains the minimum cost of welded beam design of 1.6952, and the corresponding decision variables are $h = 0.2057$, $l = 3.2531$, $t = 9.03662$, and $b = 0.2057$. Compared to the GTO, WOA, ROA, GWO, GA, MFO, MVO, GSA, and TSA algorithms, the MGTO algorithm has better practicability.

Table 11. Experimental results of the welded beam design problem test.

Algorithm	h	l	t	b	Best Weight
MGTO	0.2057	3.2531	9.03662	0.2057	1.6952
GTO [21]	0.2094	3.21	8.9565	0.2094	1.7087
WOA [19]	0.20536	3.48293	9.03746	0.206276	1.730499
ROA [16]	0.200077	3.365754	9.011182	0.206893	1.706447
GWO [14]	0.205676	3.478377	9.03681	0.205778	1.72624
GA [37]	0.1829	4.0483	9.3666	0.2059	1.8242
MFO [34]	0.2057	3.4703	9.0364	0.2057	1.72452
MVO [3]	0.205463	3.473193	9.044502	0.205695	1.72645
GSA [38]	0.182129	3.856979	10	0.202376	1.879952
TSA [39]	0.24415	6.223	8.2955	0.2444	2.3824
MROA [40]	0.2062185	3.254893	9.020003	0.206489	1.699058

5.4. Tension/Compression Spring Design Problem

The purpose of the tension/compression spring design problem is to minimize the weight of the spring, as shown in Figure 11. Three constraints affecting frequency, shear stress, and deflection in this optimization problem should be satisfied. At the same time, there are three decision variables: conductor diameter d , average coil diameter D , and effective coil number N .

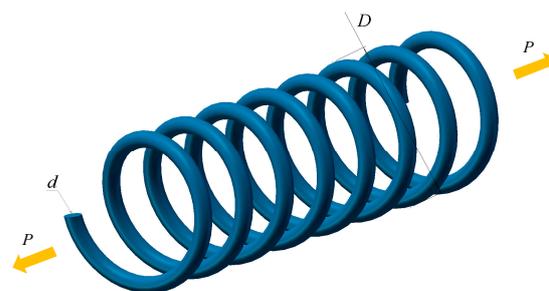


Figure 11. Model of tension/compression spring design.

The mathematical formulation of this problem is shown below:

Consider:

$$x = [x_1 \ x_2 \ x_3] = [d \ D \ N] \tag{78}$$

Objective function:

$$f(x) = (x_3 + 2) \times x_2 \times x_1^2 \tag{79}$$

Subject to:

$$g_1(x) = 1 - \frac{x_3 \times x_2^3}{71785 \times x_1^4} \leq 0 \tag{80}$$

$$g_2(x) = \frac{4 \times x_2^2 - x_1 \times x_2}{12566 \times x_1^4} + \frac{1}{5108 \times x_1^2} - 1 \leq 0 \tag{81}$$

$$g_3(x) = 1 - \frac{140.45 \times x_1}{x_2^2 \times x_3} \leq 0 \tag{82}$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \tag{83}$$

Variable range:

$$\begin{aligned} 0.05 \leq x_1 \leq 2.0; & 0.25 \leq x_2 \leq 1.3; \\ 2.0 \leq x_3 \leq 15.0 & \end{aligned} \tag{84}$$

The results show that the MGTO algorithm was effective at solving this problem. The optimization results of the MGTO algorithm and other algorithms are shown in Table 12. It can be seen from the table that the results obtained by the MGTO algorithm were far better than other comparison algorithms, and the minimum weight obtained was 0.009872, where $d = 0.05$, $D = 0.37443$, and $V = 8.546566$.

Table 12. Experimental results of the tension/compression spring design problem.

Algorithm	d	D	V	Best Weight
MGTO	0.05	0.37443	8.546566	0.009872
SSA [41]	0.051207	0.345215	12.00403	0.012676
ES [9]	0.051989	0.363965	10.89052	0.012681
PSO [13]	0.051728	0.357644	11.24454	0.012675
EROA [32]	0.053799	0.46951	5.811	0.010614
HHO [18]	0.051796	0.359305	11.13886	0.012665
HS [36]	0.051154	0.349871	12.07643	0.012671
AO [42]	0.0502439	0.35262	10.5425	0.011165
DE [12]	0.051609	0.354714	11.41083	0.01267

5.5. Cantilever Beam Design Problem

The goal of the cantilever beam design problem is to minimize the weight of the cantilever. The decision variable is the height or width of five hollow square blocks with a constant thickness. The cantilever beam design problem model is shown in Figure 12.

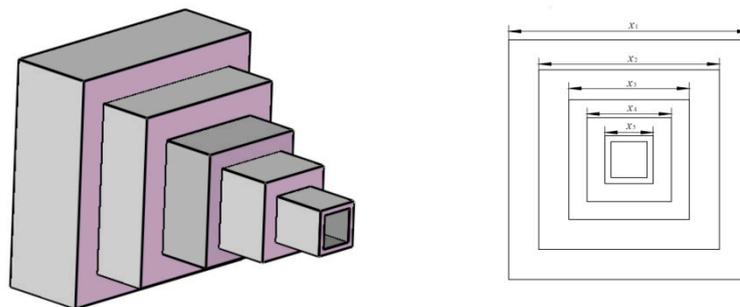


Figure 12. Model of cantilever beam design.

The mathematical formulation of this problem is shown below:
Consider:

$$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5] \tag{85}$$

Objective function:

$$f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \tag{86}$$

Subject to:

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{87}$$

Variable range:

$$0.01 \leq x_i \leq 100 (i = 1, 2, \dots, 5) \tag{88}$$

The test results of the cantilever beam design problem can be seen in Table 13. It can be seen from the table that compared to the WOA, BWO, PSO, GSA, ERHHO, and SCA algorithms, the weight obtained by the MGTO algorithm was the best weight at 1.33995647611238, which proves the feasibility and effectiveness of the MGTO algorithm in solving the cantilever beam design problem.

Table 13. Experimental results of the cantilever beam design problem test.

Algorithm	Optimal Values for Variables					Optimum Weight
	x_1	x_2	x_3	x_4	x_5	
MGTO	6.0142	5.3107	4.4942	3.5010	2.15338	1.33995647611238
WOA [19]	5.1261	5.6188	5.0952	3.9329	2.3219	1.37873150673956
BWO [43]	6.2094	6.2094	6.2094	6.2094	6.2094	1.93736251728534
PSO [13]	6.0040	5.2950	4.4915	3.5125	2.1710	1.33998298081255
GSA [38]	5.6052	4.9553	5.6619	3.1959	3.2026	1.41155753917296
ERHHO [44]	6.0509	5.2639	4.514	3.4605	2.1878	1.3402
SCA [6]	5.1096	5.9911	5.0150	3.7095	3.2744	1.44143866919587

5.6. Multiple Disc Clutch Brake Problem

The goal of the multiple disc clutch brake problem is to find five variables of the minimum-mass multi-plate brake. The problem has eight constraints. The five variables are the inner radius r_i , outer radius r_o , brake disc thickness t , driving force F , and surface friction number Z . Its model is shown in Figure 13.

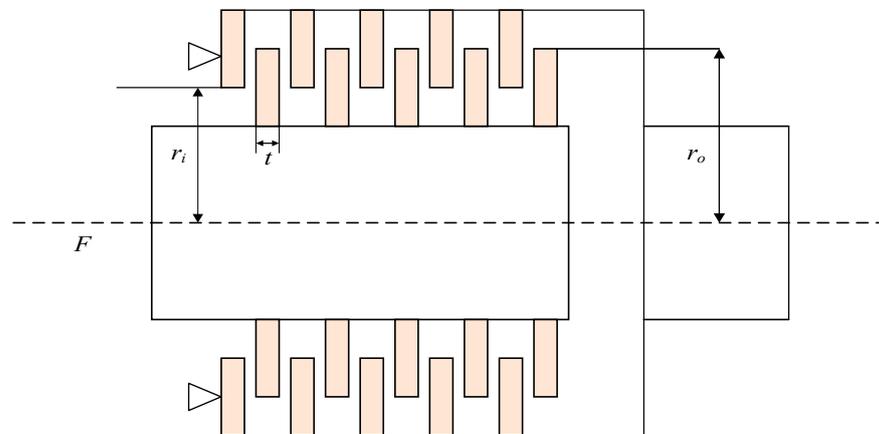


Figure 13. Model of the multiple disc clutch brake.

The mathematical formulation of this problem is shown below:

Consider:

$$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5] = [r_i \ r_o \ t \ F \ Z] \tag{89}$$

Objective function:

$$f(x) = \Pi (r_o^2 - r_i^2) t (Z + 1) \rho \ (\rho = 0.0000078) \tag{90}$$

Subject to:

$$g_1(x) = r_o - r_i - \Delta r \geq 0 \tag{91}$$

$$g_2(x) = l_{max} - (Z + 1)(t + \delta) \geq 0 \tag{92}$$

$$g_3(x) = P_{max} - P_{rz} \geq 0 \tag{93}$$

$$g_4(x) = P_{max}v_{sr \ max} - P_{rz}v_{sr} \geq 0 \tag{94}$$

$$g_5(x) = v_{sr \ max} - v_{sr} \geq 0 \tag{95}$$

$$g_6(x) = T_{max} - T \geq 0 \tag{96}$$

$$g_7(x) = M_h - sM_s \geq 0 \tag{97}$$

$$g_8(x) = T \geq 0 \tag{98}$$

Variable range:

$$60 \leq x_1 \leq 80, 90 \leq x_2 \leq 110, 1 \leq x_3 \leq 3, \tag{99}$$

$$600 \leq x_4 \leq 1000, 2 \leq x_5 \leq 9$$

Other parameters:

$$M_h = \frac{2}{3}\mu FZ \frac{r_o^3 - r_i^2}{r_o^2 - r_i^3}, P_{rz} = \frac{F}{\Pi(r_o^2 - r_i^2)}, \tag{100}$$

$$v_{rz} = \frac{2\Pi(r_o^3 - r_i^3)}{90(r_o^2 - r_i^2)}, T = \frac{I_z \Pi n}{30(M_h + M_f)} \tag{101}$$

$$\Delta r = 20 \text{ mm}, I_z = 55 \text{ kgmm}^2, P_{max} = 1 \text{ MPa}, F_{max} = 1000 \text{ N}, \tag{102}$$

$$T_{max} = 15 \text{ s}, \mu = 0.5, s = 1.5, M_s = 40 \text{ Nm}, M_f = 3 \text{ Nm}, \tag{103}$$

$$n = 250 \text{ rpm}, v_{sr \ max} = 10 \text{ m/s}, l_{max} = 30 \text{ mm} \tag{104}$$

In Table 14, we can observe that with the decision variables $x_1 = 70, x_2 = 90, x_3 = 1, x_4 = 600,$ and $x_5 = 2,$ the MGTO algorithm provided the best weight of 0.235242. Obviously, the MGTO algorithm is superior to other algorithms in solving this problem.

Table 14. Experimental results of the multiple disc clutch brake problem test.

Algorithm	Optimal Values for Variables					Optimum Weight
	x_1	x_2	x_3	x_4	x_5	
MGTO	70	90	1	600	2	0.235242
TLBO [45]	70	90	1	810	3	0.313656611
WCA [46]	70	90	1	910	3	0.313656
SCA [6]	69.516	90	1	1000	2	0.24019
CMVO [47]	70	90	1	910	3	0.313656
MFO [34]	70	90	1	910	3	0.313656

5.7. Car Crashworthiness Design

The car crashworthiness design problem’s goal is to balance a vehicle’s safety performance with the total mass to ensure the vehicle’s safety performance while remaining lightweight. Its model can be seen in Figure 14. This problem needs to be satisfied with ten constraints and, at the same time, needs to minimize an objective function with 11 decision variables. The decision variables are the internal thickness of the B-pillar, the thickness of the B-pillar reinforcement, the internal thickness of the floor, the thickness of the beam, the thickness of the door beam, the width of the door safety belt line reinforcement, the internal materials of the B-pillar, the internal materials of the floor, the height of the obstacle, and the impact position of the obstacle. The abdominal load, upper viscosity standard, middle viscosity standard, lower viscosity standard, upper rib deflection, middle rib deflection, lower rib deflection, pubic symphysis force, B-pillar midpoint velocity, and B-pillar front door velocity are constraints.



Figure 14. Three-dimensional model diagram of a car’s side crash.

The mathematical formulation of this problem is shown below:

Minimize:

$$f(\vec{x}) = \text{Weight}, \tag{105}$$

Subject to:

$$g_1(\vec{x}) = F_a(\text{load in abdomen}) \leq 1 \text{ kN}, \tag{106}$$

$$g_2(\vec{x}) = V \times Cu(\text{dummy upper chest}) \leq 0.32 \text{ m/s}, \tag{107}$$

$$g_3(\vec{x}) = V \times Cm(\text{dummy middle chest}) \leq 0.32 \text{ m/s}, \tag{108}$$

$$g_4(\vec{x}) = V \times Cl(\text{dummy lower chest}) \leq 0.32 \text{ m/s}, \tag{109}$$

$$g_5(\vec{x}) = \Delta_{ur}(\text{upper rib deflection}) \leq 32 \text{ mm}, \tag{110}$$

$$g_6(\vec{x}) = \Delta_{mr}(\text{middle rib deflection}) \leq 32 \text{ mm}, \tag{111}$$

$$g_7(\vec{x}) = \Delta_{lr}(\text{lower rib deflection}) \leq 32 \text{ mm}, \tag{112}$$

$$g_8(\vec{x}) = F(\text{Public force})_p \leq 4 \text{ kN}, \tag{113}$$

$$g_9(\vec{x}) = V_{MBP}(\text{Velocity of V-Pillar at middle point}) \leq 9.9 \text{ mm/ms}, \tag{114}$$

$$g_{10}(\vec{x}) = V_{FD}(\text{Velocity of front door at V-Pillar}) \leq 15.7 \text{ mm/ms}, \tag{115}$$

Variable Range:

$$0.5 \leq x_1 - x_7 \leq 1.5, x_8, x_9 \in (0.192, 0.345), -30 \leq x_{10}, x_{11} \leq 30, \tag{116}$$

Table 15 shows the test results of all algorithms for vehicle crashworthiness. It can be seen from the table that the MGTO algorithm obtained the best weight of 23.18916 among all other algorithms; thus, the MGTO algorithm is an effective algorithm to solve this problem.

Table 15. Experimental results of the car crashworthiness design problem test.

Algorithm	MGTO	GTO [21]	ROA [16]	MPA [48]	SOA [17]	HHOCM [49]	MSCSO [50]	MALO [51]
x_1	0.5	0.5	0.5	0.5	0.50063	0.500164	0.50011	0.5
x_2	1.2292	1.2607	1.22942	1.22823	1.25921	1.248612	1.22826	1.2281
x_3	0.5	0.5	0.5	0.5	0.5	0.659558	0.50001	0.5
x_4	1.20079	1.1495	1.21197	1.2049	1.26308	1.098515	1.20254	1.2126
x_5	0.5	0.6205	0.5	0.5	0.9377	0.757989	0.50019	0.5
x_6	1.07605	0.8860	1.37798	1.2393	1.11573	0.767268	1.05280	1.308
x_7	0.5	0.5	0.50005	0.5	0.5	0.500055	0.50002	0.5
x_8	0.345	0.34485	0.34489	0.34498	0.334889	0.343105	0.34499	0.3449
x_9	0.345	0.344608	0.19263	0.192	0.252275	0.192032	0.33595	0.2804
x_{10}	0.62110	6.202292	0.62239	0.44035	4.3435	2.898805	0.46117	0.4242
x_{11}	0.64810	7.3429	-	1.78504	16.2208	-	1.05012	4.6565
Best Weight	23.18916	23.4084	23.23544	23.19982	24.42114	24.48358	23.19085	23.2294

6. Conclusions

This paper proposes three strategies to improve the performance of the traditional GTO algorithm, and the improved algorithm is called MGTO. First, the shrinkage control factor fusion strategy improves the algorithm's search space and reduces search blindness. Second, the sine cosine interaction fusion strategy based on closeness is proposed to enhance and stabilize the performance between the silverback gorilla and other gorilla individuals. Finally, the gorilla individual difference identification strategy is used to reduce the difference between the silverback gorilla and gorilla individuals to improve the quality of the optimal solution.

In order to verify the effectiveness of the algorithm, a comprehensive evaluation was conducted on 23 classical benchmark functions, 30 CEC2014 benchmark functions, and 10 CEC2020 benchmark functions. The proposed MGTO algorithm was compared with the original GTO algorithm and eight other advanced optimization algorithms. The final results show that MGTO is a very excellent algorithm and is superior to other optimization algorithms in terms of exploitation, exploration, convergence speed, etc. At the same time, in order to further verify the superiority of the algorithm, seven complex, highly constrained, and challenging practical engineering problems were used for testing, and the MGTO algorithm's performance was compared with those of other algorithms. The results also prove the efficient ability of the MGTO algorithm to solve complex practical problems.

In future work, we hope that the MGTO algorithm can be used for its superior performance in more practical problems, such as image segmentation, feature selection, etc. Further evaluation of the algorithm's performance compared to other optimization algorithms should be performed on a wider range of engineering optimization problems, especially those with complex constraints. Modifying the algorithm to handle multi-objective optimization problems is becoming increasingly common in engineering. The sensitivity of the algorithm's performance to its hyperparameters and optimizing them for better performance should be investigated. It should also be applied to real-world engineering optimization problems to demonstrate its practical utility. Finally, the integration of machine learning techniques should be explored to enhance the algorithm's performance, such as incorporating neural networks to improve its global search capability.

Author Contributions: Conceptualization, J.Y. and H.J.; methodology, J.Y.; software, J.Y. and H.J.; validation, H.J. and D.W.; formal analysis, J.Y., H.J., and D.W.; investigation, H.R. and C.W.; resources, H.R., C.W., and Q.L.; data curation, J.Y. and L.A.; writing—original draft preparation, J.Y. and H.J.; writing—review and editing, D.W. and L.A.; visualization, J.Y., H.J., and D.W.; supervision, H.J. and D.W.; funding acquisition, D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by National Education Science Planning Key Topics of the Ministry of Education—“Research on the core quality of applied undergraduate teachers in the intelligent age” (DIA220374).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank their anonymous reviewers for helping them improve this paper’s quality.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#)
- Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium Optimizer: A Novel Optimization Algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [\[CrossRef\]](#)
- Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A Nature-Inspired Algorithm for Global Optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [\[CrossRef\]](#)
- Abualigah, L.; Diabat, A.; Mirjalili, S.; Elaziz, M.A.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [\[CrossRef\]](#)
- Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray optimization. *Comput. Struct.* **2012**, *112*, 283–294. [\[CrossRef\]](#)
- Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [\[CrossRef\]](#)
- Kaveh, A.; Dadras, A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Adv. Eng. Softw.* **2017**, *110*, 69–84. [\[CrossRef\]](#)
- Zitzler, E.; Laumanns, M.; Thiele, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*; Technical Report Gloriestrasse; Computer Engineering and Networks Laboratory: Zurich, Switzerland, 2001.
- Beyer, H.G.; Schwefel, H.P. Evolution strategies—A comprehensive introduction. *Nat. Comput.* **2002**, *1*, 3–52. [\[CrossRef\]](#)
- Banzhaf, W.; Koza, J.R. Genetic programming. *IEEE Intell. Syst.* **2000**, *15*, 74–84. [\[CrossRef\]](#)
- Yao, X.; Liu, Y.; Lin, G. Evolutionary Programming Made Faster. *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–102.
- Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
- Lin, M.; Li, Q. A Hybrid Optimization Method of Beetle Antennae Search Algorithm and Particle Swarm Optimization. *DEStech Trans. Eng. Technol. Res.* **2018**, *1*, 396–401. [\[CrossRef\]](#)
- Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
- Kiran, M.S.; Gunduz, M. A Novel Artificial Bee Colony-based Algorithm for Solving the Numerical Optimization Problems. *Int. J. Innov. Comput. I* **2012**, *8*, 6107–6121.
- Jia, H.; Peng, X.; Lang, C. Remora optimization algorithm. *Expert Syst. Appl.* **2021**, *185*, 115665. [\[CrossRef\]](#)
- Dhiman, G.; Kumar, V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowl.-Based Syst.* **2019**, *165*, 169–196. [\[CrossRef\]](#)
- Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris Hawks Optimization: Algorithm and Applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [\[CrossRef\]](#)
- Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
- Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [\[CrossRef\]](#)
- Abdollahzadeh, B.; Soleimani Gharehchopogh, F.; Mirjalili, S. Artificial Gorilla Troops Optimizer: A New Nature-Inspired Metaheuristic Algorithm for Global Optimization Problems. *Int. J. Intell. Syst.* **2021**, *36*, 5887–5958. [\[CrossRef\]](#)
- Piri, J.; Mohapatra, P.; Acharya, B.; Gharehchopogh, F.S.; Gerogiannis, V.C.; Kanavos, A.; Manika, S. Feature selection using artificial gorilla troop optimization for biomedical data: A case analysis with COVID-19 data. *Mathematics* **2022**, *10*, 2742. [\[CrossRef\]](#)
- El-Dabah, M.A.; Kamel, S.; Khamies, M.; Shahinzadeh, H.; Gharehpetian, G.B. Artificial gorilla troops optimizer for optimum tuning of TID based power system stabilizer. In Proceedings of the 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Mashhad, Iran, 2–4 March 2022; IEEE: New York, NY, USA, 2022; pp. 1–5.
- Alsolai, H.; Alzahrani, J.S.; Maray, M.; Alghamdi, M.; Qahmash, A.; Alnfai, M.M.; Mustafa Hilal, A. Enhanced Artificial Gorilla Troops Optimizer Based Clustering Protocol for UAV-Assisted Intelligent Vehicular Network. *Drones* **2022**, *6*, 358. [\[CrossRef\]](#)
- Shaheen, A.; Ginidi, A.; El-Sehiemy, R.; Elsayed, A.; Elattar, E.; Dorrah, H.T. Developed Gorilla troops technique for optimal power flow problem in electrical power systems. *Mathematics* **2022**, *10*, 1636. [\[CrossRef\]](#)
- Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [\[CrossRef\]](#)

27. Abualigah, L.; Elaziz, M.A.; Sumari, P.; Zong, W.G.; Gandomi, A.H. Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2021**, *191*, 116158. [[CrossRef](#)]
28. Hashim, F.A.; Houssein, E.H.; Hussain, K.; Mabrouk, M.S.; Al-Atabany, W. Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Math. Comput. Simul.* **2022**, *192*, 84–110. [[CrossRef](#)]
29. Rao, H.; Jia, H.; Wu, D.; Wen, C.; Liu, Q.; Abualigah, L. A Modified Group Teaching Optimization Algorithm for Solving Constrained Engineering Optimization Problems. *Mathematics* **2022**, *10*, 3765. [[CrossRef](#)]
30. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [[CrossRef](#)]
31. Dehghani, M.; Hubálovský, Š.; Trojovský, P. Northern goshawk optimization: A new swarm-based algorithm for solving optimization problems. *IEEE Access* **2021**, *9*, 162059–162080. [[CrossRef](#)]
32. Wang, S.; Hussien, A.G.; Jia, H.; Abualigah, L.; Zheng, R. Enhanced Remora Optimization Algorithm for Solving Constrained Engineering Optimization Problems. *Mathematics* **2022**, *10*, 1696. [[CrossRef](#)]
33. Lu, S.; Kim, H.M. A Regularized Inexact Penalty Decomposition Algorithm for Multidisciplinary Design Optimization Problem with Complementarity Constraints. *J. Mech. Des.* **2010**, *132*, 041005. [[CrossRef](#)]
34. Hussien, A.G.; Amin, M.; Abd El Aziz, M. A comprehensive review of moth-flame optimisation: Variants, hybrids, and applications. *J. Exp. Theor. Artif. Intell.* **2020**, *32*, 705–725. [[CrossRef](#)]
35. 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](#)]
36. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *2*, 60–68. [[CrossRef](#)]
37. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [[CrossRef](#)]
38. Rashedi, E.; Nezamabadi-Pour, H.S. GSA: A Gravitational Search Algorithm. *Inform. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
39. Babalik, A.; Cinar, A.C.; Kiran, M.S. A modification of tree-seed algorithm using Deb's rules for constrained optimization. *Appl. Soft. Comput.* **2018**, *63*, 289–305. [[CrossRef](#)]
40. Wen, C.; Jia, H.; Wu, D.; Rao, H.; Li, S.; Liu, Q.; Abualigah, L. Modified Remora Optimization Algorithm with Multistrategies for Global Optimization Problem. *Mathematics* **2022**, *10*, 3604. [[CrossRef](#)]
41. Hussien, A.G. An enhanced opposition-based salp swarm algorithm for global optimization and engineering problems. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 129–150. [[CrossRef](#)]
42. Laith, A.; Dalia, Y.; Mohamed, A.E.; Ahmed, A.E.; Mohammed, A.A.A.; Amir, H.G. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250.
43. Hayyolalam, V.; Kazem, A.A.P. Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103249. [[CrossRef](#)]
44. Song, M.; Jia, H.; Abualigah, L.; Liu, Q.; Lin, Z.; Wu, D.; Altalhi, M. Modified Harris Hawks Optimization Algorithm with Exploration Factor and Random Walk Strategy. *Comput. Intell. Neurosci.* **2022**, *2022*, 23. [[CrossRef](#)]
45. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching-Learning-Based Optimization: An optimization method for continuous nonlinear large scale problems. *Inform. Sci.* **2012**, *183*, 1–15. [[CrossRef](#)]
46. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110*, 151–166. [[CrossRef](#)]
47. Sayed, G.I.; Darwish, A.; Hassani, A.E. A new chaotic multi-verse optimization algorithm for solving engineering optimization problems. *J. Exp. Theor. Artif. Intell.* **2018**, *30*, 293–317. [[CrossRef](#)]
48. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine predators algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [[CrossRef](#)]
49. Houssein, E.H.; Neggaz, N.; Hosney, M.E.; Mohamed, W.M.; Hassaballah, M. Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities. *Neural Comput. Appl.* **2021**, *33*, 13601–13618. [[CrossRef](#)]
50. Wu, D.; Rao, H.; Wen, C.; Jia, H.; Liu, Q.; Abualigah, L. Modified Sand Cat Swarm Optimization Algorithm for Solving Constrained Engineering Optimization Problems. *Mathematics* **2022**, *10*, 4350. [[CrossRef](#)]
51. Wang, S.; Sun, K.; Zhang, W.; Jia, H. Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation. *Math. Biosci. Eng.* **2021**, *18*, 3092–3143. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.