applied sciences

MDPI

Article

A Modified Gorilla Troops Optimizer for Global Optimization Problem

Tingyao Wu [1], Di Wu [1,*], Heming Jia [2], Nuohan Zhang [1], Khaled H. Almotairi [3], Qingxin Liu [4] and Laith Abualigah [5,6]

1 School of Education and Music, Sanming University, Sanming 365004, China
2 School of Information Engineering, Sanming University, Sanming 365004, China
3 Department of Computer Engineering, Computer and Information Systems College, Umm Al-Qura University, Makkah 21955, Saudi Arabia
4 School of Computer Science and Technology, Hainan University, Haikou 570228, China
5 Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan
6 Faculty of Information Technology, Middle East University, Amman 11831, Jordan
* Correspondence: wudi@fjsmu.edu.cn

Abstract: The Gorilla Troops Optimizer (GTO) is a novel Metaheuristic Algorithm that was proposed in 2021. Its design was inspired by the lifestyle characteristics of gorillas, including migration to a known position, migration to an undiscovered position, moving toward the other gorillas, following silverback gorillas and competing with silverback gorillas for females. However, like other Metaheuristic Algorithms, the GTO still suffers from local optimum, low diversity, imbalanced utilization, etc. In order to improve the performance of the GTO, this paper proposes a modified Gorilla Troops Optimizer (MGTO). The improvement strategies include three parts: Beetle-Antennae Search Based on Quadratic Interpolation (QIBAS), Teaching–Learning-Based Optimization (TLBO) and Quasi-Reflection-Based Learning (QRBL). Firstly, QIBAS is utilized to enhance the diversity of the position of the silverback. Secondly, the teacher phase of TLBO is introduced to the update the behavior of following the silverback with 50% probability. Finally, the quasi-reflection position of the silverback is generated by QRBL. The optimal solution can be updated by comparing these fitness values. The performance of the proposed MGTO is comprehensively evaluated by 23 classical benchmark functions, 30 CEC2014 benchmark functions, 10 CEC2020 benchmark functions and 7 engineering problems. The experimental results show that MGTO has competitive performance and promising prospects in real-world optimization tasks.

Keywords: gorilla troops optimizer; beetle-antennae search based on quadratic interpolation; teaching–learning-based optimization; quasi-reflection-based learning; function optimization; engineering design

MSC: 49K35

Citation: Wu, T.; Wu, D.; Jia, H.; Zhang, N.; Almotairi, K.H.; Liu, Q.; Abualigah, L. A Modified Gorilla Troops Optimizer for Global Optimization Problem. *Appl. Sci.* **2022**, *12*, 10144. https://doi.org/10.3390/app121910144

Academic Editor: Giancarlo Mauri

Received: 14 August 2022
Accepted: 3 October 2022
Published: 9 October 2022

1. Introduction

Optimization is a vibrant field with diverse applications in optimal control, disease treatment and engineering design [1–5]. In the past few years, modeling and implementing Metaheuristic Algorithms (MAs) have proved their worth [6–9]. Compared with other conventional optimization algorithms, MAs are widely used in engineering applications for the following reasons: first and foremost, the concepts of MAs are accessible and easy to implement; second, MAs are better than local search algorithms; and finally, they do not need derivative-function information. Nature-inspired Metaheuristic Algorithms deal with optimization problems by mimicking biological or physical phenomena. MAs can be generally divided into three categories: the physical-based, the evolution-based and

the swarm-based [10] algorithms, as shown in Figure 1. Although these MAs have some differences, they all benefit from the above advantages.

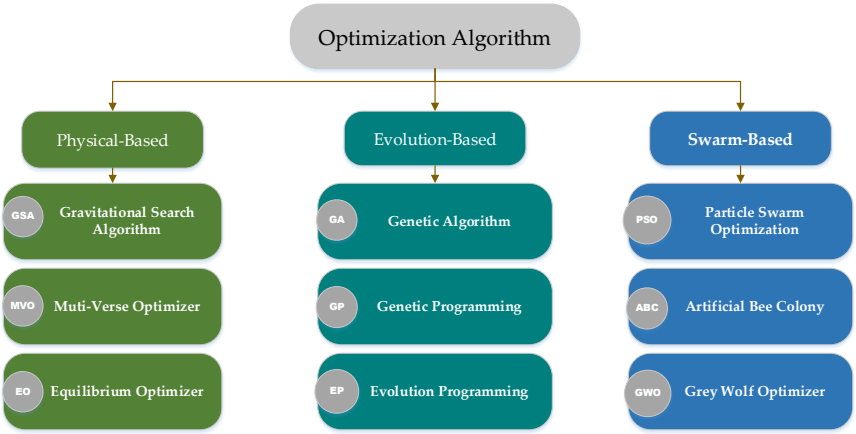

Figure 1. The category of Metaheuristic Algorithms.

A variety of MAs are introduced in this paper. To be specific, the physical-based algorithm originates from the physical, chemical phenomenon and the human intelligence. In the literature [11], the Gravitational Search Algorithm (GSA), which works on gravity and mass interactions, is discussed. The search agent is a set of interacting masses based on the Newtonian gravitation. Furthermore, GSA is wildly used in the field of machine learning. Other typical representatives are the Muti-Verse Optimizer (MVO) [12], Simulated Annealing algorithm (SA) [13], Equilibrium Optimizer (EO) [14], etc. In the following, one of the types which is affected by the biological evolution is introduced. In 1992, the genetic algorithm (GA) [15] was the first and the most popular algorithm to solve optimization problems, and it was established by Holland. The GA is derived from the laws of Darwinian evolution. This algorithm is regarded as one of the most effective algorithms, and it has been commonly utilized to solve substantial optimization problems with two recombination and mutation operators. This algorithm has been proposed with various modified and recombination versions [16]. Differential Evolution (DE) [17], Evolutionary Deduction (ED) [18] and Genetic Programming (GP) [19] are some well-known algorithms in this category. It is well-known that the swarm-based algorithm is a common approach, which derives from the survival habits of animal groups. Particle Swarm Optimization (PSO) [20] is an outstanding instance of the swarm-based algorithm, which was inspired by the swarm behavior of natural animals, such as birds and fish, in 1995. Since then, PSO has attracted humans' core attention and formed a desirable research subject called swarm intelligence. Another typical algorithm was the Artificial Bee Colony (ABC) [21], which was proposed by Karaboga in 2005; it originated from the collective behavior of bees. Like other Metaheuristic Algorithms, this algorithm also has some shortcomings. Therefore, a modified version was introduced later. Yang introduced a novel algorithm based on the luminosity of fireflies in 2010 [22]. The brightness or light of each firefly is compared to that of other fireflies. Of course, fireflies sometimes fly randomly, which improves the modified version of this algorithm. In addition, another swarm-based algorithm called Bat Algorithm (BA) [23] was proposed by Yang et al. in 2010; it was derived from the echolocation of bats. The Grey Wolf Optimizer (GWO) [6] is a well-established swarm intelligence algorithm that was developed by Mirjalili et al. in 2016. Moreover, the GWO was inspired by the social life and hunting activities of wolves. In 2019, Heidari et al. proposed the Harris Hawks Optimization (HHO) [24], which simulated the unique cooperative hunting of Harris hawks. Inspired by the biological behaviors of prey and predators, Faramarzi et al. established the swarm-based Marine Predators Algorithm (MPA) in 2020 [25].

In 2018, Lin et al. proposed a hybrid optimization method of the Beetle-Antenna Search Algorithm and Particle Swarm Optimization (PSO) [26]. The BAS has good opti-

mizing speed and accuracy in low-dimensional optimization problems, but it is easy to fall into local optimization in high-dimensional problems. Combining the PSO algorithm with the BAS can improve the BAS's optimization ability. Moreover, the BAS [27] can be applied to other Metaheuristic Algorithms to overcome the shortcomings of a single algorithm. For instance, Zhou et al. proposed a Flower-Pollination Algorithm (FPA) based on the Beetle-Antennae Search Algorithm to overcome the slow convergence problem of the original FPA algorithm [28]. The experimental results show that the improved optimization algorithm has a faster convergence rate. An improved Artificial-Bee-Colony Algorithm (ABC) [29] based on the Beetle-Antenna Search (BAS–ABC) was proposed by Cheng et al. in 2019 [30]. This algorithm makes use of the position-update ability of BAS to avoid the randomness of searching, so that the original algorithm can converge to the optimal solution more quickly. The Beetle-Antennae Search Based on Quadratic Interpolation (QIBAS) is an effective swarm intelligence optimization algorithm. QIBAS has been applied to the Inverse Kinematics Solution Algorithm of Electric Climbing Robot Based on the Improved Beetle-Antennae Search [31]. It is universally believed that this improved algorithm improves the convergence accuracy. The teaching–Learning-Based Optimization (TLBO) [32] is a very mature swarm intelligence optimization algorithm, which has been used in the improvement and hybrid of many MAs. Tuo et al. proposed a hybrid algorithm based on Harmonious Search (HS) and Teaching–Learning-Based Optimization for complex high-dimensional problems [33]. HS has a strong global search capability, but its convergence speed is slow. TLBO can make up for this deficiency to increase the convergence rate. Keesari et al. used the TLBO algorithm to solve the job-shop-scheduling problem [34] and compared with other optimization algorithms. The experimental results show that TLBO algorithm is more efficient in the job-shop-scheduling problem. In order to solve the problem that the TLBO algorithm is prone to local convergence in complex problems, Chen et al. designed the local learning and self-learning methods to improve the original TLBO algorithm [35]. It is proved that the improved TLBO has a better global search ability than other algorithms by testing on a few functions. Quasi-Reflection-Based Learning (QRBL) [36] is a variant of Opposition-Based Learning (OBL) [37], and it is an effective intelligent optimization technique. QRBL can be applied to Biogeography-Based Optimization (BBO) [38], Ion Motion Optimization (IMO) [39] and Symbiotic Organisms Search (SOS) [40]. The modified algorithm with QRBL has better convergence speed and the better ability to avoid the local optimal than the basic algorithm.

The Gorilla Troops Optimizer (GTO) [41] is a new swarm intelligence optimization algorithm that was established by Abdollahzadeh et al. in 2021. The inspiration of GTO is the migration, competition for adult females and following behavior of the gorilla colony. Currently, it has been applied to several subject-design and engineering-optimization problems. However, like other swarm intelligence optimization algorithms, GTO is difficult to obtain a balance between exploration and exploitation due to the randomness of the optimization process. Therefore, the algorithm still has some problems, such as low accuracy, slow convergence and ease of falling into local optimal. It is worth mentioning that the No Free Lunch (NFL) [42] theorem indicates that no algorithm can solve all optimization problems perfectly. Therefore, this theorem and the defects of GTO prompt us to improve and develop the modified swarm-based algorithm to deal with more engineering problems:

(1) A modified GTO, which is called MGTO, is proposed in this paper. The modified algorithm introduces three improvement strategies. Firstly, the Quadratic Interpolated Beetle-Antennae Search (QIBAS) [31] is embedded into the GTO that can get the diversity of the silverback's position. In addition, Teaching–Learning-Based Optimization (TLBO) [32] is hybridized with GTO to stabilize the performance between the silverback and other gorillas. Finally, the Quasi-Reflection-Based Learning (QRBL) [36] mechanism is used to enhance the quality of the optimal position.
(2) To verify the effectiveness of the MGTO, 23 classical benchmark functions, 30 CEC2014 benchmark functions and 10 CEC2020 benchmark functions are adopted to conduct a

simulation experiment. The performance of the MGTO is evaluated through a variety of comparisons with the basic GTO and eight state-of-the-art optimization algorithms.

(3) Furthermore, the MGTO is applied to solve the welded-beam-design problem, pressure-vessel-design problem, reducer problem, compression/tension-spring problem, three-bar-truss-design problem, crash-worthiness-design problem and string-design problem. The experimental results indicate that MGTO has a strong convergence ability and global search ability.

The rest of this paper is organized as follows: Section 2 introduces the basic GTO. In Section 3, three strategies and the modified GTO named MGTO are proposed. The experimental results and the discussion of this work are presented in Section 4. In Section 5, the MGTO is tested to solve seven kinds of real-world engineering problems. Finally, the conclusion and future work are given in Section 6.

2. Gorilla Troops Optimizer (GTO)

The Gorilla Troops Optimizer is a swarm-inspired algorithm that simulates the social life of gorillas. The gorilla is a social animal, and it is the largest primate on earth at present. Because of the white hair on its back, the adult male is also known as a silverback.

A gorilla group always consists of an adult male gorilla, several adult female gorillas and their offspring. Among them, the adult male gorilla is the leader, whose responsibilities are to defend the territory, make decisions, direct other gorillas to find abundant food and so on. The research shows that the male and female gorillas deviate from their birth with high probability. Generally, the male gorillas incline to abandon their quondam groups for appealing female gorillas, and then they will form a new group. Nevertheless, the male gorillas sometimes prefer to stay in the group in which they were born, with the hope that they will have the chance to dominate the whole group one day. The fierce competition for females between male gorillas is inevitable. Male gorillas can expand their territory by competition. The relationship between male and female gorillas is close and stable, while the relationship among female gorillas is cold relatively.

This algorithm includes two stages: Exploration and exploitation. Five different operators emulate the optimization operation for the behavior of gorillas in this algorithm. There are three operators in the exploration stage: moving to an undiscovered position, moving toward the other gorillas, moving to a known position. In the exploitation stage, two different operators of tracking the silverback and competing for adult females are adopted to improve the search performance.

2.1. Exploration

The operation procedures of the exploration stage are described in this subsection. It is commonly known that a gorilla group is governed by a silverback who has the capacity to conduct all actions. Sometimes gorillas will go to other places which they have visited before or that are new to them in nature. At each optimization operation stage, the optimal candidate solution is regarded as a silverback solution. Furthermore, three mechanisms at this stage are introduced.

Equation (1) is used to denote three mechanisms in the exploration stage. In the equation, p is a parameter ranging from 0 to 1 which is utilized to choose the mechanism of migration for an unknown position. For the sake of clarity, the mechanism for migration to an unknown position will be chosen when $rand < p$. Then, if $rand \geq 0.5$, the second mechanism, that of movement toward the other gorillas, will be selected. If $rand < 0.5$, the mechanism for migration to a known position will be selected.

$$GX(t+1) = \begin{cases} (UB - LB) \times r_1 + LB & rand < p \\ (r_2 - C) \times X_r(t) + L \times H & rand \geq 0.5 \\ X(i) - L \times (L \times (X(t) - GX_r(t)) + r_3 \times (X(t) - GX_r(t))) & rand < 0.5 \end{cases} \quad (1)$$

where $GX(t+1)$ indicates the candidate position vector of the gorilla at the next iteration, and $X(t)$ is the current position vector of the gorilla. Moreover, r_1, r_2, r_3 and *rand* are the random values between 0 and 1. *UB* and *LB* indicate the upper and lower bounds of the variables, respectively. X_r and GX_r are the candidate position vectors of gorillas that are selected randomly.

The equations that are used to calculate C, L and H are as follows:

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

where *It* indicates the current iteration value, and *MaxIt* is the maximum iteration value. At the initial stage, the variation values are generated in a large interval, and then the changed interval of variation values will decrease in the final optimization stage. F can be calculated by the following equation:

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

where r_4 is a random value that is in between $[-1,1]$.

L is a parameter for which the calculation equation is as follows:

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

where l is a random value from 0 to 1. Moreover, the Equation (4) is utilized to simulate the silverback leadership. Because of inexperience, silverback gorillas always hard to make the correct decisions to find food or manage the group. However, they can obtain adequate experience and extreme stability in the leadership process. Additionally, H in Equation (1) is calculated by Equation (5). Z in Equation (5) is calculated by Equation (6), where Z is a random value in the range of $[-C, C]$:

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

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

At the end of the exploration, a group operation is performed to calculate the cost of all GX solutions. If the cost is identified as $GX(t) < X(t)$, the $X(t)$ solution will be substituted by *GX(t)* solution. Therefore, the best solution at this stage is regarded as the silverback, as well.

2.2. Exploitation

In the exploitation stage of the GTO algorithm, the two behaviors of following the silverback and competing for adult females are adopted. The silverback leads all the gorillas in the group, and it is responsible for various activities in the group. Competing for adult females is another behavior. The C value indicates that the adult males can choose to follow the silverback or compete with other males. W is a parameter which should be set before the optimization operation. If C satisfies different conditions, the above mechanism will be selected.

2.2.1. Following the Silverback

When the silverback and other gorillas are young, they can perform their duties well. For instance, male gorillas follow the silverback easily. Moreover, each member can influence other members. That is to say, if $C \geq W$, the strategy will be performed. By mimicking this behavior, Equation (7) is used to illustrate this mechanism, as follows:

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

where $X_{silverback}$ is the vector of the silverback, which presents the optimal solution. M can be expressed as follows:

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

where $GX_i(t)$ refers to the vector position of each candidate gorilla at iteration, t; N indicates the sum of gorillas; and g is estimated by Equation (9) as follows:

$$g = 2^L \quad (9)$$

2.2.2. Competition for Adult Females

Competing for females with other male gorillas is a main stage of puberty for young gorillas. This competition is always fierce, which will persist for days and affect other members. Equation (10) is used to emulate this behavior:

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

$$Q = 2 \times r_5 - 1 \quad (11)$$

$$A = \beta \times E \quad (12)$$

$$E = \begin{cases} N_1, & rand \geq 0.5 \\ N_2, & rand < 0.5 \end{cases} \quad (13)$$

where Q is adopted to simulate the impact, which is calculated by Equation (11). Moreover, r_5 is a random value between 0 and 1. Equation (12) is used to calculate the coefficient vector of the violence degree in conflict, where β is the parameter that needs to be given before the optimization operation. E is used to simulate the effect of violence on the solution's dimensions. If $rand \geq 0.5$, E will be equal to a random value in the normal distribution and the problem's dimensions. However, if $rand < 0.5$, E will be equal to a random value in the normal distribution; $rand$ is a random value between 0 and 1.

3. Modified Algorithm Implementation

3.1. Beetle-Antennae Search Based on Quadratic Interpolation (QIBAS)

The beetle hunts through two antennae. Inspired by this, the Beetle-Antennae Search (BAS) algorithm was proposed by Jiang et al. in 2017 [27]. Different odors in the space correspond to different function values. The beetle can detect odor values on both sides of itself and search for the position with the highest odor. The position with the highest odor is the specific position of food. The habits of the beetle are shown in Figure 2, where the black line represents the propagation of odor, and the blue line denotes the trajectory of the beetle.

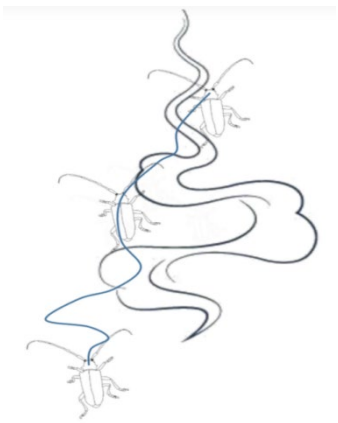

Figure 2. The habits of the beetle.

3.1.1. Searching Behavior of Beetles

To model the searching behavior, the random direction of the beetle's searching is represented as follows:

$$\vec{b} = \frac{rands(k,1)}{\|rands(k,1)\|} \tag{14}$$

where $\vec{b}$ is the unit vector, *rands* is a random value and k is the dimension of the position.

In addition, the searching behavior of the left and right sides of the beetle are given, respectively, as follows:

$$\begin{cases} x_l = x(t) + d \times b \\ x_r = x(t) - d \times b \end{cases} \tag{15}$$

where x_r indicates the right side of searching area, x_l denotes the left side of searching area and d represents the length of the antennae.

3.1.2. Detecting Behavior of Beetles

To simulate the detecting behavior, the iterative model corresponding to the odor detection is as shown below:

$$x_{t+1} = x_t - s \times b \times sign(f(x_l) - f(x_r)) \tag{16}$$

where x represents the position of the beetle, and $f(x)$ represents the strength of the odor at position x. The maximum value of $f(x)$ donates the source point of the odor, and s is the step length of the searching. The function, $sign(x)$; the antennae length, d; and the step length, s, are represented as follows:

$$sign(x) = \begin{cases} 1 & if \quad x > 0 \\ 0 & if \quad x = 0 \\ -1 & otherwise \end{cases} \tag{17}$$

$$d_{t+1} = 0.95 d_t + 0.01 \tag{18}$$

$$s_{t+1} = 0.95 s_t \tag{19}$$

3.1.3. Quadratic Interpolation Based on Beetle-Antennae Search (QIBAS)

The BAS algorithm has the advantages of accessible principle and high convergence speed. In order to further improve its ability to solve optimization problems, the quadratic interpolation operator is introduced, which can be presented as follows:

$$x_i = \frac{1}{2} \frac{(x_{lk}^2 - x_{bk}^2) f(x_r) + (x_{bk}^2 - x_{rk}^2) f(x_l) + (x_{rk}^2 - x_{lk}^2) f(x_b)}{(x_{lk} - x_{bk}) f(x_r) + (x_{bk} - x_{rk}) f(x_l) + (x_{rk} - x_{lk}) f(x_b)} \tag{20}$$

where k is the dimension of the position, and x_b denotes the global optimal solution.

The quadratic interpolation is adopted to obtain a new solution, x_i, which varies from the existing solution, x_t. The fitness values of the two positions are compared to determine whether x_t is preserved or replaced.

3.2. Teaching–Learning-Based Optimization

The Teaching–Learning-Based Optimization (TLBO) was proposed in 2012; it was inspired by the influence of the teacher on the output of learners [32]. The output is evaluated by results and grades. Generally, a teacher is considered to be a knowledgeable person who trains and shares knowledge with students. Learners earn better grades with the help of a well-qualified teacher. Moreover, learners can learn interactively, as well, to improve their own knowledge. The result of learners can be considered as the "fitness", and the teacher can be considered as the optimal solution currently.

3.2.1. Teacher Phase

As mentioned above, a knowledgeable teacher can increase the mean level of a class. However, it is difficult for the teacher to bring learners up to the same level as him/her. In fact, a teacher can only improve the mean of the class to a certain degree according to the capability of this class. It is a random process that relies on some factors. M_i represents the mean at iteration, i. T_i is the teacher who strives to bring M_i up to his/her level. Hence, the new mean can be designated as M_{new}. The updated solution based on the difference between the current mean and new mean is given as follows:

$$Difference\_Mean_i = r_i(M_{new} - T_F M_i) \tag{21}$$

where T_F is the teaching factor to change the mean, and r_i is a random value ranging from 0 to 1. The value of T_F can be represented as follows:

$$T_F = round[1 + rand(0,1)] \tag{22}$$

To reduce the difference, the existing solution is modified according to the following equation:

$$X_{new,i} = X_{old,i} + Difference\_Mean_i \tag{23}$$

3.2.2. Learner Phase

Learners increase their knowledge in two different patterns: one is the input of the teacher, and the other is the interaction among themselves. Learners interact randomly with other learners through group cooperation, presentations, debates, etc. Learners learn from others who have more knowledge. The modified learner can be expressed as follows:

$$X_{new,i} = X_{old,i} + r_i(X_i - X_j); if\ f(X_j) < f(X_i) \tag{24}$$

$$X_{new,i} = X_{old,i} + r_i(X_j - X_i); if\ f(X_i) < f(X_j) \tag{25}$$

3.3. Quasi-Reflection-Based Learning

A new Quasi-Reflection-Based Learning (QRBL) mechanism was established based on Opposition-Based Learning (OBL) and Quasi-Opposition-Based Learning (QOBL) by Ewees et al. in 2018. The quasi-reflection number, x^{qr}, of the solution, x, is obtained as follows:

$$x^{qr} = rand\left(\frac{lb + ub}{2},\ x\right) \tag{26}$$

where $rand((lb+ub/2),x)$ is a random number which distributes uniformly between $(lb+ub/2)$ and x. The quasi-reflection value can be extended into D-dimensional space, which is expressed as follows:

$$x_i^{qr} = rand\left(\frac{lb_i + ub_i}{2},\ x_i\right) \tag{27}$$

3.4. The Proposed MGTO

Like other swarm intelligence algorithms, GTO still falls into local optimum and suffers from slow convergence easily. To overcome these shortcomings and further enhance the performance of the GTO, a modified MGTO is proposed that introduces QIBAS, TLBO and QRBL into the GTO. First, the QIBAS algorithm is employed to enrich the initial position of the silverback. Then TLBO is hybridized with the stage involving following the silverback into the exploitation phase. In this stage, the silverback can be considered as a teacher, and other gorillas learn from the silverback. Through this operation, the search ability of gorillas is enhanced, and the differences between the silverback and gorillas are reduced. Thirdly, QRBL is adopted to update the position of the silverback at the end of the stage, thus facilitating the quality of the optimal position.

In the initialization phase, the MGTO generates the random population, X_i, and initializes the position of the silverback. Then the QIBAS algorithm is used to calculate the search positions on both sides of the silverback. The *NewPosition* can be calculated by Equation (28). The quadratic interpolation function is employed to generate *NewPosition*1. *NewPosition*1 can be expressed by Equation (29). To choose an optimal position, the fitness values of these two positions are compared, and then, if the fitness value of the "new position" is better than the previous one, it will replace the previous position:

$$Newposition = x_t - s \times b \times sign(f(x_l) - f(x_r)) \tag{28}$$

$$Newposition1 = \frac{1}{2}\frac{\left(x_{lk}^2 - Silverback^2\right)f(x_r) + \left(Silverback^2 - x_{rk}^2\right)f(x_l) + \left(x_{rk}^2 - x_{lk}^2\right)f(Silverback)}{(x_{lk} - Silverback)f(x_r) + (Silverback - x_{rk})f(x_l) + (x_{rk} - x_{lk})f(Silverback)} \tag{29}$$

In the exploitation phase, the TLBO algorithm is adopted to update the behavior of following the silverback with 50% probability. The first step is calculating the mean, M, of the population. Secondly, the teach factor, F, and the difference, *Difference*, between gorillas and the silverback are calculated. In addition, the third step is to update the position, as shown below:

$$GX(i) = X_{old,i} + Difference\_Mean_i \tag{30}$$

In the final phase, in order to get a new position of the silverback, QRBL is used to generate the quasi-reflection position $X_{Silverback}{}^{qr}$. By comparing the fitness values of $X_{Silverback}$ and $X_{Silverback}{}^{qr}$, the position is selected from two positions as the final optimal position:

$$Positions\_ROL = rand\left(\frac{lb_i + ub_i}{2},\ x_i\right) \tag{31}$$

Eventually, repeating the steps mentioned until the maximum value of iterations is reached. The pseudo-code is represented below. And the flowchart of MGTO is shown in Figure 3.

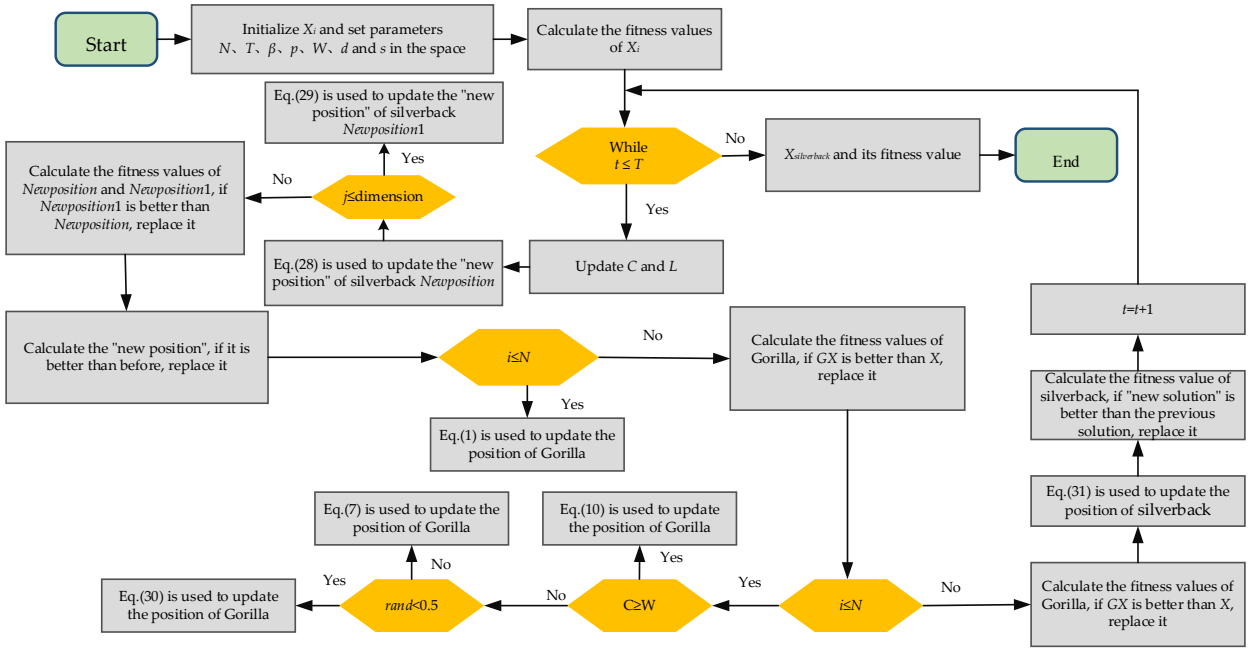

Figure 3. The flowchart of MGTO.

Algorithm 1. The pseudo-code of MGTO

% MGTO setting
Inputs: the size of population, N; the maximum number of iterations, T; and parameters β, p, W, d and s.
Outputs: $X_{silverback}$ and its fitness value
% Initialization
Initialize the random population Xi (I = 1,2, . . . ,N)
Calculate the fitness values of X_i
% Main Loop
while (stopping condition is not met) **do**
  Equation (2) is used to update C
  Equation (4) is used to update L
  Equation (28) is used to update the "new position" of silverback *Newposition*
  for ($j \leq$ variables_no) **do**
    Equation (29) is used to update the "new position" of silverback *Newposition*1
  end for
  Calculate the fitness values of *Newposition* and *Newposition*1
  if *Newposition*1 is better than *Newposition*, replace it
  if "new position" is better than the previous position, replace it
  % Exploration phase
  for (each Gorilla (X_i)) **do**
    Equation (1) is used to update the position of Gorilla
  end for
  % Establish group
  Calculate the fitness values of Gorilla
  if GX is better than X, replace it
  % Exploitation phase
  for (each Gorilla (X_i)) **do**
    if ($|C| \geq 1$) **then**
      if *rand*>0.5 **then**
        Update the position of Gorilla by using Equation (30)
      else
        Update the position of Gorilla by using Equation (7)
      end if
    else
    Update the position of Gorilla by using Equation (10)
    end if
  end for
  % Establish group
  Calculate the fitness values of Gorilla
  if GX is better than X, replace it
  Equation (31) is used to update the position of silverback
  Calculate the fitness value of silverback
  if "new solution" is better than the previous solution, replace it
end while
Return $X_{silverback}$ and its fitness value

4. The Results and Discussion of Experiment

In this section, 23 classical benchmark functions are used to evaluate the performance of the proposed algorithm. Furthermore, nine optimization algorithms are selected for comparison, namely the Gorilla Troops Optimization (GTO) [41], Arithmetic Optimization Algorithm (AOA) [43], Salp Swarm Algorithm (SSA) [2], Whale Optimization Algorithm (WOA) [44], Grey Wolf Optimizer (GWO) [6], Particle Swarm Optimization (PSO) [20], Random-Opposition-Based Learning Grey Wolf Optimizer (ROLGWO) [45], Dynamic Sine–Cosine Algorithm (DSCA) [46] and Hybridizing Sine–Cosine Algorithm with Harmony Search (HSCAHS) [47]. For the sake of fairness, the maximum iteration and population size of all algorithms are set to 500 and 30, respectively.

The 23 classical benchmark functions can be divided into three categories: unimodal functions (UM), multimodal functions (MM) and composite functions (CM). The unimodal functions (F1~F7) have only one optimal solution that is fluently used to evaluate the exploration ability of the algorithm. The multimodal functions (F8~F13) are characterized by multiple optimal solutions. These functions can be utilized to evaluate the ability of jumping out from the local optimal solution in complex situations. The composite functions (F14~F23) are usually adopted to evaluate the stability of algorithms [48].

In addition, the benchmark functions (F1~F30) provided in CEC2014 and the benchmark functions (F1~F10) provided in CEC2020 are used in the other experiments [49]. These benchmarks are applied in many papers to evaluate the performance for the ability of solving problems. The standard to evaluate the performance of the optimization algorithm is whether it can keep the balance between exploration and development and avoid the local optimum.

4.1. The Experiments on Classical Benchmark

4.1.1. The Convergence Analysis

In order to evaluate the advantages of the MGTO algorithm on the benchmark functions, the MGTO is compared with traditional GTO, AOA, SSA, WOA, GWO, PSO, ROL-GWO, DSCA and HSCAHS algorithms. The results in Figure 4 show that the proposed MGTO can achieve more efficient and better results compared with other optimization algorithms. Furthermore, this paper selects "semi-logarithms" to draw the convergence curve with the purpose of making the difference of curve convergence obviously. Because "0" has no logarithms, the iterative curve is interrupted in the figure. As shown in F1~F4, F9 and F11, the modified algorithm does not display curves in the subsequent process, just because it converges to 0, which reflects the high convergence accuracy of this algorithm. In addition, the proposed algorithm is superior to other algorithms in the benchmark-functions (F1~F13) experiments. It is obvious that the MGTO has the excellent property of keeping balance between exploration and exploitation when solving complicated problems. In the benchmark functions (F14~F23), the MGTO can obtain great superiority, thus indicating that the proposed algorithm is competitive in solving composite functions. However, the results of the benchmark functions (F17~F18) and the DSCA and PSO algorithms demonstrate that these algorithms give an excellent performance to obtain high-quality results. In summary, the MGTO performs well in benchmark functions (F14~F23). What is more, the MGTO can find excellent solutions compared with other algorithms in most cases. Thus, in the comparative experiment, the MGTO has certain advantages over the other algorithms. The effectiveness of the MGTO can be proved by the experiment results.

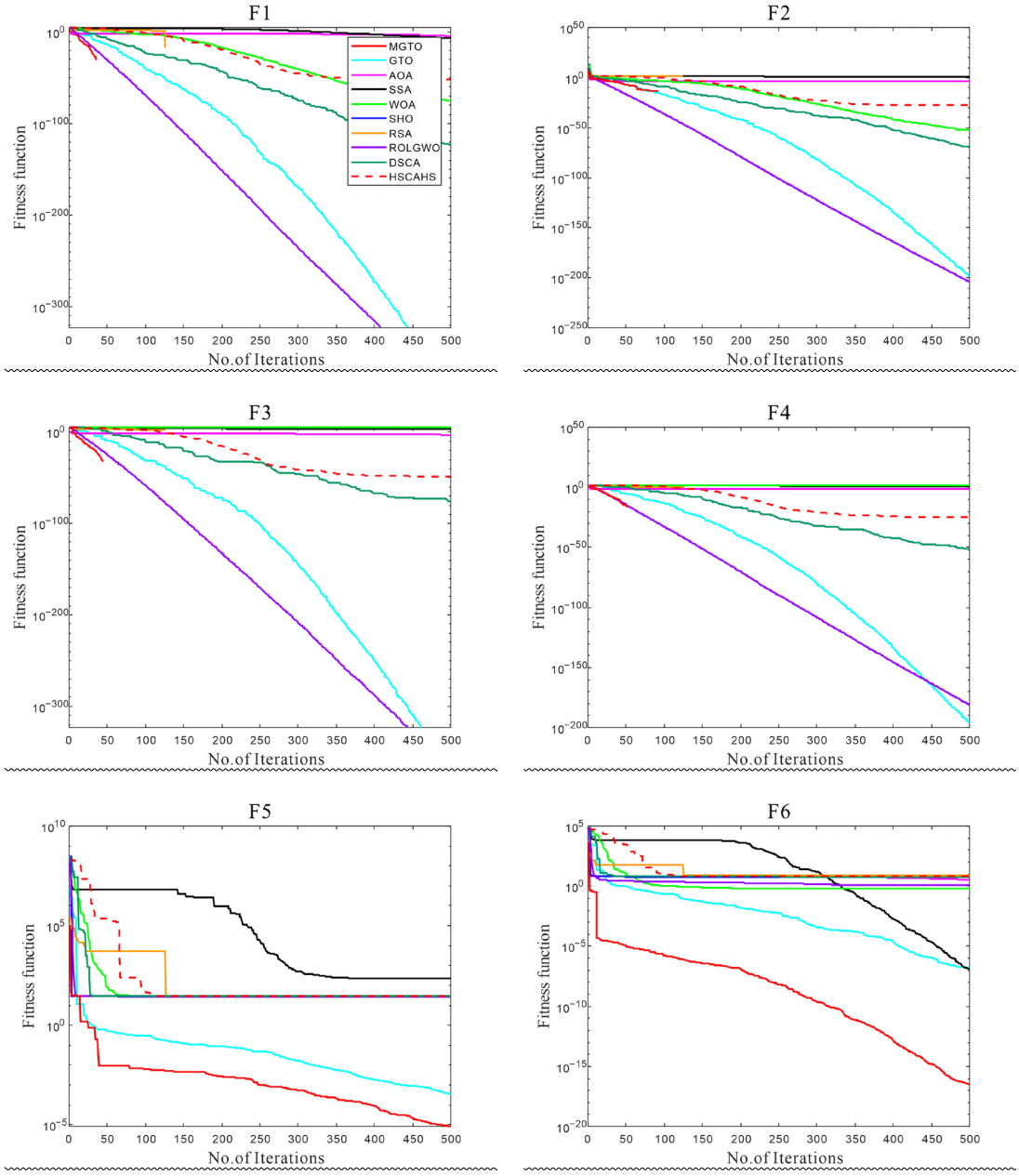

Figure 4. *Cont*.

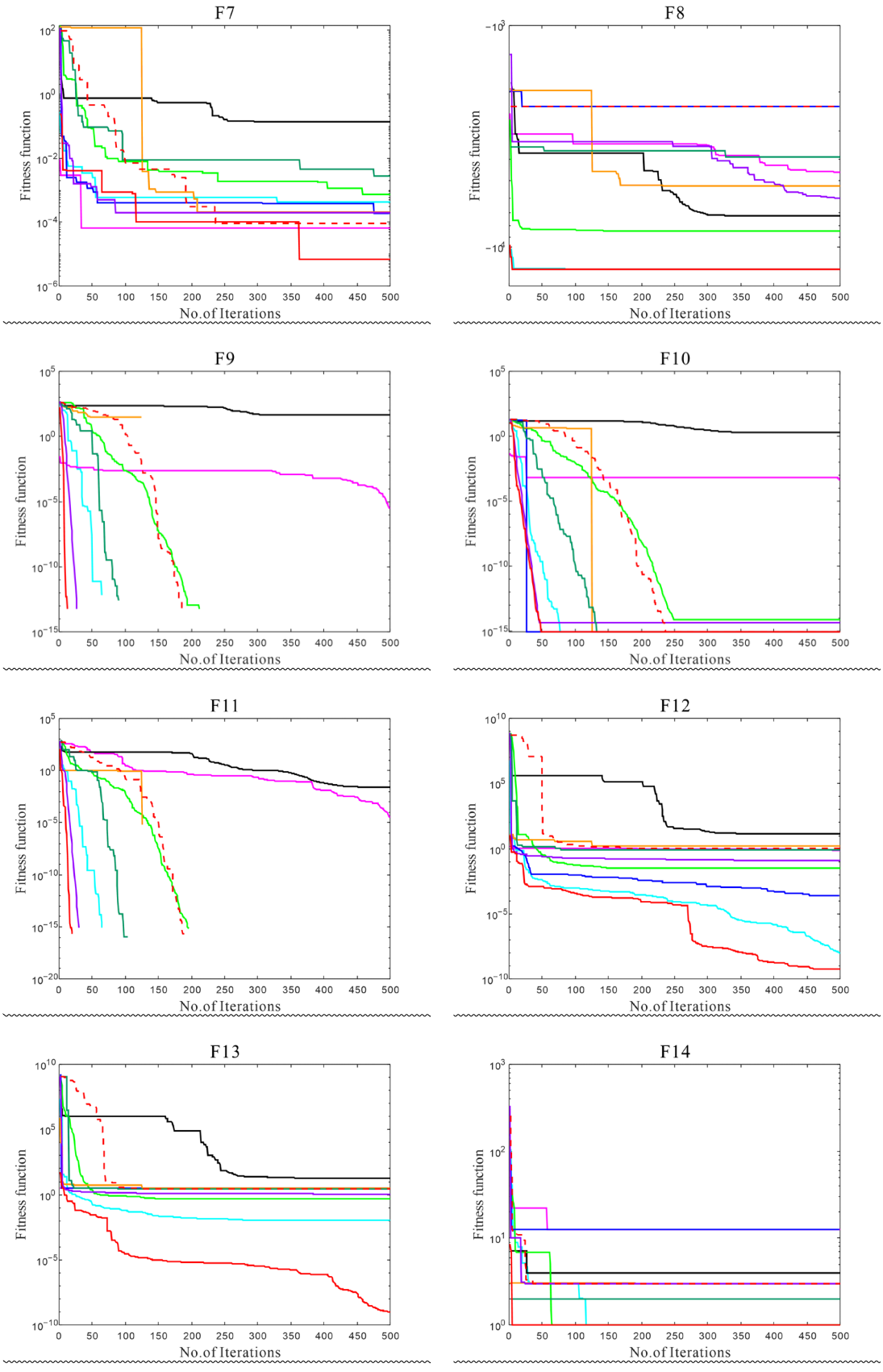

Figure 4. *Cont.*

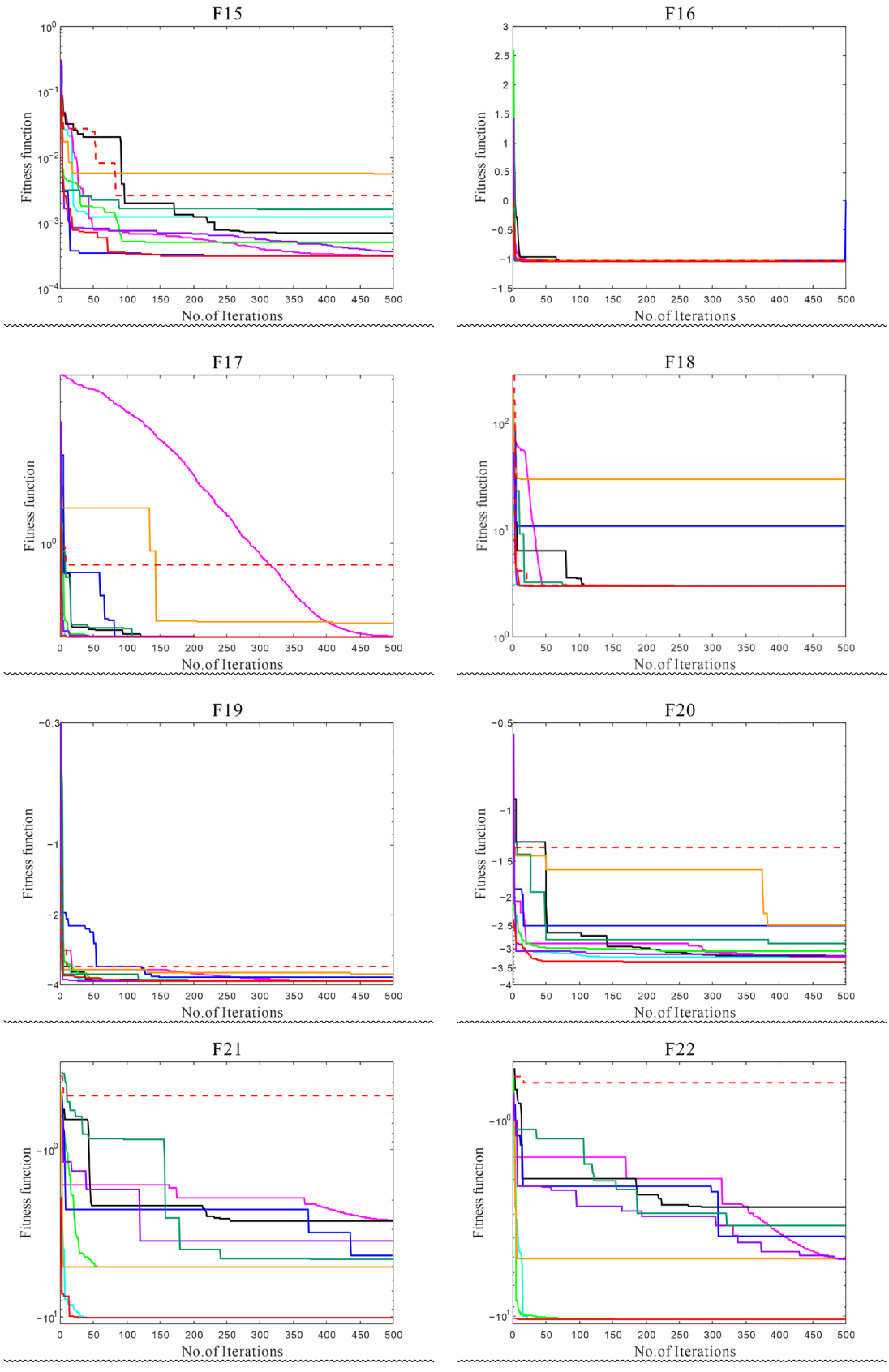

Figure 4. *Cont.*

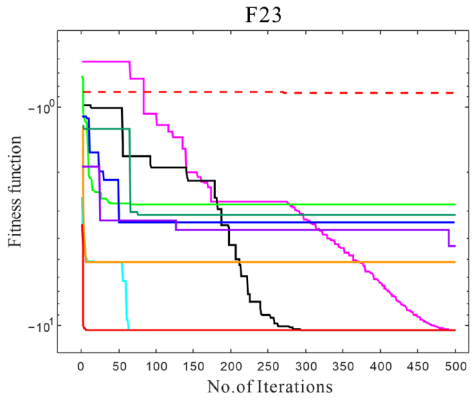

Figure 4. Diagram of the convergence curve.

4.1.2. The Results of the Classical Benchmark

The results of 23 classical functions are listed in Table 1. All results are expressed by average value (Avg) and standard deviation (Std): The average value represents better convergence performance, and the standard deviation indicates the better stability. According to Table 1, it is obvious that the MGTO performs better in major tests. In particular, the optimal value of most functions is precisely calculated by MGTO, whereas other algorithms cannot find the optimal solution. Compared with the GTO, the proposed mechanism can improve the performance of obtaining the best solution.

Table 1. The results of the classical test.

		MGTO	GTO	AOA	SSA	WOA	SHO	RSA	ROLGWO	DSCA	HSCAHS
F1	Avg	**0**	**0**	4.72×10^{-6}	1.49×10^{-7}	3.26×10^{-69}	**0**	**0**	**0**	2.89×10^{-111}	1.33×10^{-50}
	Std	**0**	**0**	1.97×10^{-6}	1.17×10^{-7}	1.78×10^{-68}	**0**	**0**	**0**	1.58×10^{-110}	5.89×10^{-50}
F2	Avg	**0**	3.69×10^{-189}	2.14×10^{-3}	1.93	8.20×10^{-51}	**0**	**0**	8.09×10^{-204}	6.39×10^{-59}	1.06×10^{-27}
	Std	**0**	**0**	2.06×10^{-3}	1.40	2.24×10^{-50}	**0**	**0**	**0**	3.49×10^{-58}	1.24×10^{-27}
F3	Avg	**0**	**0**	1.07×10^{-3}	1.35×10^{3}	4.45×10^{4}	**0**	**0**	**0**	7.60×10^{-62}	1.92×10^{-48}
	Std	**0**	**0**	7.62×10^{-4}	5.88×10^{2}	1.54×10^{4}	**0**	**0**	**0**	4.16×10^{-61}	7.43×10^{-48}
F4	Avg	**0**	8.19×10^{-192}	1.98×10^{-2}	1.31×10^{1}	5.01×10^{1}	**0**	**0**	3.78×10^{-183}	6.09×10^{-39}	1.26×10^{-25}
	Std	**0**	**0**	1.19×10^{-2}	4.83	2.94×10^{1}	**0**	**0**	**0**	3.34×10^{-38}	2.55×10^{-25}
F5	Avg	$\mathbf{2.54 \times 10^{-5}}$	4.81	2.80×10^{1}	5.26×10^{2}	2.80×10^{1}	2.88×10^{1}	1.55×10^{1}	2.74×10^{1}	2.86×10^{1}	2.88×10^{1}
	Std	$\mathbf{3.94 \times 10^{-5}}$	9.77	2.74×10^{-1}	1.15×10^{3}	4.71×10^{-1}	1.12×10^{-1}	1.47×10^{1}	8.20×10^{-1}	3.05×10^{-1}	5.19×10^{-2}
F6	Avg	$\mathbf{3.37 \times 10^{-14}}$	2.06×10^{-7}	3.11	1.73×10^{-7}	4.70×10^{-1}	3.55	7.24	9.06×10^{-1}	5.53	6.71
	Std	$\mathbf{3.76 \times 10^{-14}}$	2.68×10^{-7}	2.19×10^{-1}	2.69×10^{-7}	2.78×10^{-1}	2.47	4.48×10^{-1}	5.21×10^{-1}	2.78×10^{-1}	1.90×10^{-1}
F7	Avg	$\mathbf{7.67 \times 10^{-5}}$	8.68×10^{-5}	9.64×10^{-5}	1.77×10^{-1}	3.16×10^{-3}	1.13×10^{-4}	1.69×10^{-4}	8.21×10^{-5}	1.61×10^{-3}	9.55×10^{-5}
	Std	$\mathbf{4.50 \times 10^{-5}}$	5.90×10^{-5}	9.89×10^{-5}	7.77×10^{-2}	3.97×10^{-3}	1.81×10^{-4}	1.17×10^{-4}	6.62×10^{-5}	1.75×10^{-3}	1.70×10^{-4}
F8	Avg	$\mathbf{-1.26 \times 10^{4}}$	$\mathbf{-1.26 \times 10^{4}}$	-5.44×10^{3}	-7.32×10^{3}	-1.01×10^{4}	-2.45×10^{3}	-5.35×10^{3}	-5.17×10^{3}	-4.48×10^{3}	-2.53×10^{3}
	Std	$\mathbf{1.03 \times 10^{-6}}$	6.74×10^{-5}	3.23×10^{2}	6.74×10^{2}	1.77×10^{3}	4.12×10^{2}	4.01×10^{2}	1.50×10^{3}	3.50×10^{2}	2.93×10^{2}
F9	Avg	**0**	**0**	1.35×10^{-6}	6.17×10^{1}	1.89×10^{-15}	**0**	**0**	**0**	**0**	**0**
	Std	**0**	**0**	1.14×10^{-6}	1.96×10^{1}	1.04×10^{-14}	**0**	**0**	**0**	**0**	**0**
F10	Avg	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$	4.51×10^{-4}	2.49	5.03×10^{-15}	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$	2.07×10^{-15}	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$
	Std	**0**	**0**	1.81×10^{-4}	7.86×10^{-1}	3.11×10^{-15}	**0**	**0**	1.70×10^{-15}	**0**	**0**
F11	Avg	**0**	**0**	2.41×10^{-3}	1.87×10^{-2}	8.13×10^{-3}	**0**	**0**	**0**	**0**	**0**
	Std	**0**	**0**	6.04×10^{-3}	1.25×10^{-2}	4.45×10^{-2}	**0**	**0**	**0**	**0**	**0**
F12	Avg	$\mathbf{1.64 \times 10^{-8}}$	4.27×10^{-8}	7.42×10^{-1}	7.17	2.96×10^{-2}	2.09×10^{-4}	1.33	5.64×10^{-2}	7.75×10^{-1}	1.06
	Std	$\mathbf{3.12 \times 10^{-8}}$	9.20×10^{-8}	2.43×10^{-2}	4.31	3.44×10^{-2}	2.39×10^{-5}	4.55×10^{-1}	2.52×10^{-2}	8.18×10^{-2}	1.10×10^{-1}
F13	Avg	$\mathbf{1.00 \times 10^{-7}}$	1.77×10^{-3}	2.96	1.19×10^{1}	4.94×10^{-1}	2.94	1.19	1.09	2.84	2.84
	Std	$\mathbf{1.43 \times 10^{-7}}$	5.61×10^{-3}	2.90×10^{-2}	1.25×10^{1}	2.41×10^{-1}	2.41×10^{-2}	1.41	4.99×10^{-1}	4.52×10^{-2}	3.26×10^{-2}
F14	Avg	$\mathbf{9.98 \times 10^{-1}}$	$\mathbf{9.98 \times 10^{-1}}$	1.11×10^{1}	1.33	4.04	9.47	3.57	5.30	1.65	2.91
	Std	$\mathbf{4.12 \times 10^{-17}}$	7.14×10^{-17}	3.49	9.49×10^{-1}	3.73	4.18	2.23	4.44	8.33×10^{-1}	2.70×10^{-1}
F15	Avg	$\mathbf{3.07 \times 10^{-4}}$	3.99×10^{-4}	4.55×10^{-3}	1.59×10^{-3}	7.20×10^{-4}	3.19×10^{-4}	2.72×10^{-3}	3.62×10^{-4}	1.28×10^{-3}	2.70×10^{-3}
	Std	$\mathbf{3.29 \times 10^{-19}}$	2.79×10^{-4}	7.95×10^{-3}	3.56×10^{-3}	3.78×10^{-4}	7.46×10^{-6}	1.55×10^{-3}	7.11×10^{-5}	3.48×10^{-4}	1.69×10^{-3}
F16	Avg	**−1.03**	**−1.03**	**−1.03**	**−1.03**	**−1.03**	-9.27×10^{-1}	**−1.03**	**−1.03**	**−1.03**	−1.02
	Std	$\mathbf{5.45 \times 10^{-16}}$	6.58×10^{-16}	2.19×10^{-11}	2.99×10^{-14}	2.51×10^{-9}	2.01×10^{-1}	1.66×10^{-3}	6.88×10^{-5}	1.54×10^{-4}	7.55×10^{-3}
F17	Avg	$\mathbf{3.98 \times 10^{-1}}$	$\mathbf{3.98 \times 10^{-1}}$	3.99×10^{-1}	$\mathbf{3.98 \times 10^{-1}}$	$\mathbf{3.98 \times 10^{-1}}$	6.37×10^{-1}	4.23×10^{-1}	$\mathbf{3.98 \times 10^{-1}}$	4.02×10^{-1}	7.16×10^{-1}
	Std	**0**	**0**	3.58×10^{-3}	9.58×10^{-15}	1.50×10^{-5}	6.71×10^{-1}	2.04×10^{-2}	1.19×10^{-6}	3.22×10^{-3}	3.91×10^{-1}
F18	Avg	**3.00**	**3.00**	2.15×10^{1}	**3.00**	**3.00**	2.83×10^{1}	7.68	**3.00**	3.01	3.01
	Std	$\mathbf{1.28 \times 10^{-15}}$	1.29×10^{-15}	3.02×10^{1}	2.10×10^{-13}	2.62×10^{-4}	5.09×10^{1}	1.68×10^{1}	3.53×10^{-5}	5.91×10^{-3}	1.56×10^{-2}
F19	Avg	**−3.86**	**−3.86**	**−3.86**	**−3.86**	**−3.86**	−3.56	−3.79	**−3.86**	−3.83	−3.43
	Std	$\mathbf{2.60 \times 10^{-15}}$	2.67×10^{-15}	6.28×10^{-6}	1.31×10^{-12}	4.40×10^{-3}	3.82×10^{-1}	5.65×10^{-2}	1.29×10^{-4}	1.98×10^{-2}	2.62×10^{-1}
F20	Avg	**−3.30**	−3.27	−3.27	−3.25	−3.24	−2.57	−2.66	−3.26	−3.01	−1.64
	Std	$\mathbf{4.51 \times 10^{-2}}$	6.03×10^{-2}	5.93×10^{-2}	6.67×10^{-2}	1.25×10^{-1}	3.51×10^{-1}	2.86×10^{-1}	8.43×10^{-2}	9.72×10^{-2}	5.62×10^{-1}
F21	Avg	$\mathbf{-1.02 \times 10^{1}}$	$\mathbf{-1.02 \times 10^{1}}$	−7.38	−6.81	−8.01	−4.08	−5.06	−5.10	−4.19	-5.78×10^{-1}
	Std	$\mathbf{5.83 \times 10^{-15}}$	6.08×10^{-15}	2.91	3.50	2.66	1.11	3.09×10^{-7}	9.97×10^{-1}	1.29	1.62×10^{-1}
F22	Avg	$\mathbf{-1.04 \times 10^{1}}$	$\mathbf{-1.04 \times 10^{1}}$	−7.61	−8.48	−7.30	−3.65	−5.09	−5.78	−4.27	$\mathbf{-7.20 \times 10^{-1}}$
	Std	$\mathbf{6.60 \times 10^{-16}}$	9.33×10^{-16}	3.11	3.28	3.20	1.12	7.75×10^{-7}	2.28E	4.48×10^{-1}	1.79×10^{-1}
F23	Avg	$\mathbf{-1.05 \times 10^{1}}$	-1.05×10^{1}	−7.10	−8.56	−7.17	−4.09	−5.13	−6.83	−4.10	-8.79×10^{-1}
	Std	$\mathbf{1.23 \times 10^{-15}}$	1.51×10^{-15}	3.39	3.37	3.06	1.21	1.97×10^{-6}	2.81	5.89×10^{-1}	2.92×10^{-1}

Therefore, as far as numerous unimodal functions are concerned, the MGTO still remains the excellent searching performance, thus clearly proving the superiority of the exploitative ability in the MGTO. In contrast to unimodal functions, multimodal functions have multiple optimal solutions, which have many local optimal solutions. These multimodal functions are usually adopted to evaluate the convergence ability of algorithms. As Table 1 shows, the results can indicate that the proposed model has slight advantages in getting out of the local solutions. The primary cause is that the basic GTO has already provided competitive results. The functions F14~F23 can be used to test the stability and searching capability of algorithms. From Table 1, it can be seen that the performance of MGTO surpasses the traditional GTO, and the solution results of multiple functions are extremely close to the true value. Evidently, it can be concluded that the MGTO maintains high stability and exploitative ability consistently in the test.

4.2. The Experiments on CEC2014 and CEC2020

In this section, the CEC2014 tests are utilized to prove the performance of the MGTO. The results of the MGTO in complicated CEC2014 functions are compared with nine Metaheuristic Algorithms which are frequently quoted in the literature. The comparison results of CEC2014 benchmark functions are represented in Table 2. It can be seen from Table 2 that the MGTO provides the best results in 12 out of 30 cases for Avg and 15 out for 30 cases for Std. For other cases, it can be summarized as follows: SSA provides the best results in F24 and F25 for Avg; ROLGWO provides the best results in F6, F9 and F11 for Avg and F13 for Std; GTO provides the best results in F12 for Std; AOA provides the best results in F5 for Std; DSCA provides the best results in F6 for Std; and HSCAHS provides the best results in F9, F10, F11 and F16 for Std. In these cases, other algorithms are better than the MGTO, with a slightly different result. It can be proved that the MGTO has better efficiency and stability in solving global optimization problems. The MGTO can search the whole space consistently through subsequent iterations, and this can improve the convergence.

Table 2. The results of the CEC2014 test.

		MGTO	GTO	AOA	SSA	WOA	SHO	RSA	ROLGWO	DSCA	HSCAHS
F1	Avg	$\mathbf{6.14 \times 10^{3}}$	1.89×10^{4}	2.45×10^{8}	2.37×10^{6}	1.52×10^{7}	2.07×10^{9}	1.10×10^{9}	8.34×10^{6}	3.34×10^{7}	1.04×10^{8}
	Std	$\mathbf{5.43 \times 10^{3}}$	2.30×10^{4}	2.19×10^{8}	2.25×10^{6}	1.04×10^{7}	2.44×10^{8}	2.67×10^{8}	4.68×10^{6}	1.45×10^{7}	4.35×10^{7}
F2	Avg	$\mathbf{2.00 \times 10^{2}}$	1.96×10^{3}	1.01×10^{10}	3.81×10^{3}	4.09×10^{7}	8.61×10^{10}	7.38×10^{10}	1.02×10^{8}	1.54×10^{9}	7.22×10^{9}
	Std	$\mathbf{1.91 \times 10^{-1}}$	2.38×10^{3}	2.64×10^{9}	3.73×10^{3}	3.33×10^{7}	7.49×10^{9}	4.16×10^{9}	3.24×10^{8}	4.83×10^{8}	9.48×10^{8}
F3	Avg	$\mathbf{3.07 \times 10^{2}}$	3.93×10^{2}	1.90×10^{4}	1.52×10^{4}	6.58×10^{4}	9.36×10^{5}	8.10×10^{4}	6.69×10^{3}	1.82×10^{4}	1.66×10^{4}
	Std	$\mathbf{1.65 \times 10^{1}}$	1.54×10^{2}	4.76×10^{3}	8.05×10^{3}	3.62×10^{4}	1.53×10^{6}	1.01×10^{4}	3.98×10^{3}	5.62×10^{3}	2.41×10^{3}
F4	Avg	$\mathbf{4.16 \times 10^{2}}$	4.24×10^{2}	2.67×10^{3}	4.32×10^{2}	4.55×10^{2}	1.94×10^{4}	1.00×10^{4}	4.37×10^{2}	5.51×10^{2}	1.55×10^{3}
	Std	$\mathbf{1.68 \times 10^{1}}$	1.79×10^{1}	1.43×10^{3}	1.73×10^{1}	3.67×10^{1}	3.33×10^{3}	2.91×10^{3}	2.48×10^{1}	4.59×10^{1}	4.77×10^{2}
F5	Avg	$\mathbf{5.20 \times 10^{2}}$	$\mathbf{5.20 \times 10^{2}}$	$\mathbf{5.20 \times 10^{2}}$	$\mathbf{5.20 \times 10^{2}}$	$\mathbf{5.20 \times 10^{2}}$	5.21×10^{2}	5.21×10^{2}	$\mathbf{5.20 \times 10^{2}}$	$\mathbf{5.20 \times 10^{2}}$	5.21×10^{2}
	Std	5.29×10^{2}	6.87×10^{2}	$\mathbf{3.95 \times 10^{-3}}$	1.05×10^{-1}	1.44×10^{-1}	5.63×10^{-2}	6.15×10^{-2}	1.86×10^{-1}	9.02×10^{-2}	1.11×10^{-1}
F6	Avg	6.04×10^{2}	6.06×10^{2}	6.11×10^{2}	6.05×10^{2}	6.09×10^{2}	6.47×10^{2}	6.40×10^{2}	$\mathbf{6.03 \times 10^{2}}$	6.10×10^{2}	6.10×10^{2}
	Std	1.67	1.80	9.21×10^{-1}	2.06	1.69	2.10	2.20	1.84	$\mathbf{4.79 \times 10^{-1}}$	7.12×10^{-1}
F7	Avg	$\mathbf{7.00 \times 10^{2}}$	$\mathbf{7.00 \times 10^{2}}$	9.03×10^{2}	$\mathbf{7.00 \times 10^{2}}$	7.02×10^{2}	1.57×10^{3}	1.35×10^{3}	7.02×10^{2}	7.27×10^{2}	8.47×10^{2}
	Std	$\mathbf{7.14 \times 10^{-2}}$	2.84×10^{-1}	6.65×10^{1}	1.19×10^{-1}	7.09×10^{-1}	9.37×10^{1}	1.10×10^{2}	2.39	7.94	3.10×10^{1}
F8	Avg	$\mathbf{8.07 \times 10^{2}}$	8.24×10^{2}	8.65×10^{2}	8.28×10^{2}	8.40×10^{2}	1.22×10^{3}	1.16×10^{3}	8.15×10^{2}	8.59×10^{2}	8.96×10^{2}
	Std	**5.31**	1.01×10^{1}	1.45×10^{1}	1.06×10^{1}	1.30×10^{1}	3.38×10^{1}	1.96×10^{1}	6.40	7.34	8.65
F9	Avg	9.29×10^{2}	9.31×10^{2}	9.53×10^{2}	9.34×10^{2}	9.52×10^{2}	1.31×10^{3}	1.24×10^{3}	$\mathbf{9.28 \times 10^{2}}$	9.61×10^{2}	9.70×10^{2}
	Std	1.01×10^{1}	1.03×10^{1}	5.49	1.85×10^{1}	2.00×10^{1}	2.21×10^{1}	2.15×10^{1}	1.40×10^{1}	7.87	**5.19**
F10	Avg	$\mathbf{1.25 \times 10^{3}}$	1.49×10^{3}	1.75×10^{3}	1.72×10^{3}	1.69×10^{3}	9.94×10^{3}	8.00×10^{3}	1.55×10^{3}	2.39×10^{3}	2.44×10^{3}
	Std	1.50×10^{2}	2.60×10^{2}	1.95×10^{2}	3.27×10^{2}	2.81×10^{2}	6.23×10^{2}	4.68×10^{2}	2.20×10^{2}	1.54×10^{2}	$\mathbf{1.48 \times 10^{2}}$
F11	Avg	1.88×10^{3}	1.93×10^{3}	2.25×10^{3}	2.07×10^{3}	2.31×10^{3}	3.48×10^{3}	2.69×10^{3}	$\mathbf{1.86 \times 10^{3}}$	2.81×10^{3}	2.88×10^{3}
	Std	3.06×10^{2}	2.61×10^{2}	2.40×10^{2}	3.48×10^{2}	3.51×10^{2}	2.70×10^{2}	2.12×10^{2}	3.39×10^{2}	1.86×10^{2}	$\mathbf{1.38 \times 10^{2}}$
F12	Avg	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$	$\mathbf{1.20 \times 10^{3}}$
	Std	2.19×10^{-1}	$\mathbf{2.12 \times 10^{-1}}$	5.82×10^{-1}	2.31×10^{-1}	3.83×10^{-1}	9.75×10^{-1}	3.40×10^{-1}	6.99×10^{-1}	3.63×10^{-1}	4.35×10^{-1}
F13	Avg	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$	$\mathbf{1.30 \times 10^{3}}$
	Std	9.91×10^{-2}	1.26×10^{-1}	7.31×10^{-1}	1.75×10^{-1}	1.62×10^{-1}	7.51×10^{-1}	7.51×10^{-1}	6.86×10^{-2}	2.14×10^{-1}	5.47×10^{-1}
F14	Avg	$\mathbf{1.40 \times 10^{3}}$	$\mathbf{1.40 \times 10^{3}}$	1.44×10^{3}	$\mathbf{1.40 \times 10^{3}}$	$\mathbf{1.40 \times 10^{3}}$	1.45×10^{3}	1.42×10^{3}	1.40×10^{3}	$\mathbf{1.40 \times 10^{3}}$	1.42×10^{3}
	Std	$\mathbf{7.90 \times 10^{-2}}$	1.85×10^{-1}	1.20×10^{1}	2.68×10^{-1}	2.00×10^{-1}	7.93	8.11	2.09×10^{-1}	1.40	4.30
F15	Avg	$\mathbf{1.50 \times 10^{3}}$	$\mathbf{1.50 \times 10^{3}}$	2.16×10^{4}	$\mathbf{1.50 \times 10^{3}}$	1.50×10^{3}	1.86×10^{4}	5.83×10^{3}	$\mathbf{1.50 \times 10^{3}}$	1.54×10^{3}	2.61×10^{3}
	Std	$\mathbf{4.31 \times 10^{-1}}$	2.30	1.14×10^{4}	1.01	4.80	1.98×10^{4}	4.32×10^{3}	7.88×10^{-1}	3.84×10^{1}	6.20×10^{2}
F16	Avg	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$	$\mathbf{1.60 \times 10^{3}}$
	Std	3.92×10^{-1}	3.28×10^{-1}	2.32×10^{-1}	3.26×10^{-1}	2.93×10^{-1}	2.41×10^{-1}	1.32×10^{-1}	3.24×10^{-1}	1.82×10^{-1}	$\mathbf{9.54 \times 10^{-2}}$
F17	Avg	$\mathbf{2.23 \times 10^{3}}$	2.68×10^{3}	4.73×10^{5}	2.78×10^{4}	4.02×10^{5}	4.92×10^{6}	4.59×10^{5}	3.11×10^{4}	1.70×10^{5}	4.89×10^{5}
	Std	$\mathbf{2.40 \times 10^{2}}$	1.18×10^{3}	9.71×10^{4}	3.72×10^{4}	7.54×10^{5}	4.15×10^{6}	1.27×10^{5}	9.63×10^{4}	1.19×10^{5}	9.53×10^{4}
F18	Avg	$\mathbf{1.87 \times 10^{3}}$	1.92×10^{3}	1.09×10^{4}	1.01×10^{4}	1.41×10^{4}	3.18×10^{7}	3.04×10^{5}	1.17×10^{4}	4.11×10^{4}	6.66×10^{4}
	Std	$\mathbf{4.04 \times 10^{1}}$	8.19×10^{1}	3.89×10^{3}	8.38×10^{3}	1.20×10^{4}	3.29×10^{7}	7.20×10^{5}	3.36×10^{3}	3.35×10^{4}	2.38×10^{4}
F19	Avg	$\mathbf{1.90 \times 10^{3}}$	$\mathbf{1.90 \times 10^{3}}$	1.95×10^{3}	$\mathbf{1.90 \times 10^{3}}$	1.91×10^{3}	1.97×10^{3}	1.93×10^{3}	$\mathbf{1.90 \times 10^{3}}$	1.91×10^{3}	1.92×10^{3}
	Std	$\mathbf{9.81 \times 10^{-1}}$	1.34	3.06×10^{1}	1.24	1.86	4.13×10^{1}	1.37×10^{1}	1.15	1.02	8.40
F20	Avg	$\mathbf{2.05 \times 10^{3}}$	2.08×10^{3}	1.26×10^{4}	7.78×10^{3}	1.08×10^{4}	5.54×10^{6}	2.19×10^{4}	7.86×10^{3}	3.39×10^{4}	2.06×10^{4}
	Std	$\mathbf{3.90 \times 10^{1}}$	6.52×10^{1}	3.80×10^{3}	6.77×10^{3}	7.51×10^{3}	9.13×10^{6}	3.30×10^{4}	3.72×10^{3}	2.41×10^{4}	8.97×10^{3}

Table 2. *Cont.*

		MGTO	GTO	AOA	SSA	WOA	SHO	RSA	ROLGWO	DSCA	HSCAHS
F21	Avg	**2.38×10^{3}**	2.48×10^{3}	1.95×10^{6}	7.03×10^{3}	6.03×10^{5}	3.97×10^{6}	9.29×10^{5}	1.11×10^{4}	5.15×10^{4}	2.00×10^{5}
	Std	**2.38×10^{2}**	3.13×10^{2}	2.72×10^{6}	6.91×10^{3}	1.99×10^{6}	5.03×10^{6}	1.66×10^{6}	6.76×10^{3}	3.60×10^{4}	9.92×10^{4}
F22	Avg	**2.22×10^{3}**	2.24×10^{3}	2.43×10^{3}	2.30×10^{3}	2.32×10^{3}	2.76×10^{3}	2.42×10^{3}	2.30×10^{3}	2.31×10^{3}	2.48×10^{3}
	Std	**6.65**	3.90×10^{1}	1.29×10^{2}	7.42×10^{1}	8.59×10^{1}	1.87×10^{2}	7.55×10^{1}	6.25×10^{1}	4.01×10^{1}	5.16×10^{1}
F23	Avg	**2.50×10^{3}**	**2.50×10^{3}**	**2.50×10^{3}**	2.63×10^{3}	2.64×10^{3}	**2.50×10^{3}**	**2.50×10^{3}**	2.58×10^{3}	2.52×10^{3}	**2.50×10^{3}**
	Std	**0.00**	**0.00**	3.81×10^{4}	8.83	2.83×10^{1}	**0.00**	**0.00**	6.56×10^{1}	6.02×10^{1}	**0.00**
F24	Avg	2.59×10^{3}	2.57×10^{3}	2.60×10^{3}	**2.54×10^{3}**	2.58×10^{3}	2.60×10^{3}	2.60×10^{3}	2.57×10^{3}	2.57×10^{3}	2.60×10^{3}
	Std	2.53×10^{1}	3.34×10^{1}	8.14	2.35×10^{1}	2.57×10^{1}	**0.00**	6.37×10^{-1}	3.80×10^{1}	9.06	6.25
F25	Avg	2.69×10^{3}	2.70×10^{3}	2.70×10^{3}	**2.68×10^{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}
	Std	1.03×10^{1}	1.37×10^{1}	**0.00**	2.60×10^{1}	3.74	**0.00**	8.21×10^{-2}	1.19×10^{-1}	4.70	1.48×10^{-1}
F26	Avg	**2.70×10^{3}**	**2.70×10^{3}**	2.71×10^{3}	**2.70×10^{3}**	**2.70×10^{3}**	2.71×10^{3}	2.71×10^{3}	**2.70×10^{3}**	**2.70×10^{3}**	**2.70×10^{3}**
	Std	**6.83×10^{-2}**	9.84×10^{-2}	2.43×10^{1}	1.24×10^{-1}	1.82×10^{1}	2.43×10^{1}	2.44×10^{1}	1.01×10^{-1}	1.83×10^{-1}	5.51×10^{-1}
F27	Avg	**2.83×10^{3}**	2.84×10^{3}	3.18×10^{3}	3.03×10^{3}	3.16×10^{3}	2.90×10^{3}	2.90×10^{3}	3.04×10^{3}	3.07×10^{3}	2.90×10^{3}
	Std	9.37×10^{1}	9.08×10^{1}	1.90×10^{2}	1.51×10^{2}	1.13×10^{2}	**0.00**	**0.00**	1.16×10^{2}	1.07×10^{2}	3.08
F28	Avg	**3.00×10^{3}**	**3.00×10^{3}**	3.37×10^{3}	3.21×10^{3}	3.42×10^{3}	**3.00×10^{3}**	**3.00×10^{3}**	3.24×10^{3}	3.24×10^{3}	**3.00×10^{3}**
	Std	**0.00**	**0.00**	5.04×10^{2}	6.65×10^{1}	1.26×10^{2}	**0.00**	**0.00**	6.45×10^{1}	1.10×10^{1}	**0.00**
F29	Avg	3.22×10^{3}	2.01×10^{5}	2.39×10^{7}	1.93×10^{5}	3.06×10^{5}	**3.10×10^{3}**	**3.10×10^{3}**	1.72×10^{5}	1.85×10^{4}	2.45×10^{6}
	Std	1.02×10^{2}	7.74×10^{5}	1.95×10^{7}	5.76×10^{5}	9.66×10^{5}	**0.00**	**0.00**	5.40×10^{5}	1.25×10^{4}	2.55×10^{6}
F30	Avg	3.93×10^{3}	3.96×10^{3}	1.52×10^{5}	4.68×10^{3}	6.29×10^{3}	**3.20×10^{3}**	**3.20×10^{3}**	4.39×10^{3}	6.54×10^{3}	4.03×10^{4}
	Std	3.99×10^{2}	3.55×10^{2}	3.73×10^{5}	7.67×10^{2}	1.86×10^{3}	**0.00**	**0.00**	6.23×10^{2}	1.48×10^{3}	3.56×10^{4}

For CEC2020 benchmark functions, the functions (F1~F4) are as follows: translational rotation function, translational rotation schwefel function, translational rotation lunacek bi-rastrigin function and expansion rosenbrock's plus griewangk function. F5~F7 functions are mixed functions. F8~F10 functions are composite functions. The results of CEC2020 calculated by MGTO, GTO, AOA, SSA, WOA, GWO, PSO, ROLGWO, DSCA and HACAHA are displayed in Table 3. According to Table 3, MGTO provides the best results in 6 out of 10 cases for Avg and 7 out for 10 cases for Std. For other cases, it can be summarized as follows: ROLGWO provides the best results in F3 for Avg, RSA provides the best results in F3 for Std and SSA provides the best results in F9 for Std. In function F4, all algorithms achieve the same Avg of 1.90E+03. Therefore, it can be concluded that the proposed algorithm can enhance the performance of GTO and solve CEC2020 functions better. Generally speaking, the MGTO has a better performance in solving optimization problems.

Table 3. The results of the CEC2020 test.

		MGTO	GTO	AOA	SSA	WOA	SHO	RSA	ROLGWO	DSCA	HSCAHS
F1	Avg	**1.51×10^{2}**	2.39×10^{3}	1.62×10^{10}	3.39×10^{3}	6.69×10^{7}	1.65×10^{10}	1.18×10^{10}	7.85×10^{7}	4.52×10^{9}	1.08×10^{10}
	Std	**1.76×10^{2}**	2.62×10^{3}	5.21×10^{9}	3.42×10^{3}	8.73×10^{7}	4.17×10^{9}	3.77×10^{9}	1.61×10^{8}	1.68×10^{9}	2.05×10^{9}
F2	Avg	**1.80×10^{3}**	1.92×10^{3}	2.32×10^{3}	2.01×10^{3}	2.24×10^{3}	3.55×10^{3}	2.82×10^{3}	1.98×10^{3}	2.59×10^{3}	2.98×10^{3}
	Std	**1.82×10^{2}**	2.81×10^{2}	2.40×10^{2}	3.22×10^{2}	3.68×10^{2}	2.70×10^{2}	1.98×10^{2}	2.98×10^{2}	2.16×10^{2}	1.90×10^{2}
F3	Avg	7.46×10^{2}	7.53×10^{2}	8.01×10^{2}	7.47×10^{2}	7.86×10^{2}	8.70×10^{2}	8.10×10^{2}	**7.44×10^{2}**	8.20×10^{2}	8.38×10^{2}
	Std	1.26×10^{1}	1.54×10^{1}	1.24×10^{1}	1.54×10^{1}	2.45×10^{1}	2.47×10^{1}	**1.18×10^{1}**	1.35×10^{1}	1.20×10^{1}	1.20×10^{1}
F4	Avg	**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	**0.00**	**0.00**	**0.00**	1.06	4.23×10^{-1}	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
F5	Avg	**2.15×10^{3}**	2.57×10^{3}	4.81×10^{5}	2.80×10^{4}	3.80×10^{5}	3.21×10^{6}	4.43×10^{5}	4.32×10^{4}	3.82×10^{5}	4.91×10^{5}
	Std	**2.15×10^{2}**	7.72×10^{2}	1.34×10^{5}	4.05×10^{4}	6.12×10^{5}	3.02×10^{6}	1.48×10^{5}	1.42×10^{5}	1.64×10^{5}	8.70×10^{4}
F6	Avg	**1.68×10^{3}**	1.77×10^{3}	2.22×10^{3}	1.77×10^{3}	1.90×10^{3}	2.65×10^{3}	2.22×10^{3}	1.76×10^{3}	1.90×10^{3}	2.26×10^{3}
	Std	**6.65×10^{1}**	1.25×10^{2}	1.76×10^{2}	1.22×10^{2}	1.25×10^{2}	2.82×10^{2}	1.68×10^{2}	8.14×10^{1}	1.15×10^{2}	1.46×10^{2}
F7	Avg	**2.40×10^{3}**	2.50×10^{3}	3.11×10^{6}	9.57×10^{3}	9.69×10^{5}	4.70×10^{6}	1.47×10^{6}	9.38×10^{3}	5.47×10^{4}	1.67×10^{5}
	Std	**1.83×10^{2}**	2.40×10^{2}	4.14×10^{6}	9.09×10^{3}	1.58×10^{6}	5.60×10^{6}	2.63×10^{6}	5.34×10^{3}	5.64×10^{4}	7.36×10^{4}
F8	Avg	**2.30×10^{3}**	2.30×10^{3}	3.48×10^{3}	2.38×10^{3}	2.48×10^{3}	3.70×10^{3}	3.37×10^{3}	2.32×10^{3}	2.64×10^{3}	3.03×10^{3}
	Std	**1.16**	1.60×10^{1}	3.78×10^{2}	2.79×10^{2}	4.36×10^{2}	5.55×10^{2}	3.61×10^{2}	2.31×10^{1}	1.14×10^{2}	1.32×10^{2}
F9	Avg	**2.66×10^{3}**	2.71×10^{3}	2.96×10^{3}	2.74×10^{3}	2.79×10^{3}	2.96×10^{3}	2.91×10^{3}	2.73×10^{3}	2.76×10^{3}	2.86×10^{3}
	Std	9.75×10^{1}	1.06×10^{2}	1.23×10^{2}	**4.72×10^{1}**	4.91×10^{1}	8.53×10^{1}	6.60×10^{1}	7.69×10^{1}	1.18×10^{2}	6.37×10^{1}
F10	Avg	**2.93×10^{3}**	2.94×10^{3}	3.86×10^{3}	**2.93×10^{3}**	2.97×10^{3}	3.90×10^{3}	3.46×10^{3}	2.94×10^{3}	3.18×10^{3}	3.53×10^{3}
	Std	**2.15×10^{1}**	2.79×10^{1}	3.67×10^{2}	2.37×10^{1}	6.41×10^{1}	2.85×10^{2}	2.01×10^{2}	2.46×10^{1}	7.61×10^{1}	5.98×10^{1}

4.3. The Non-Parametric Statistic Test

Although the superiority of the MGTO was confirmed by the above benchmark functions, in order to further demonstrate the advantages of the MGTO, the Wilcoxon's rank-sum test [50,51] with 5% accuracy is used to evaluate and research the difference between the proposed algorithm and other algorithms. The *p*-values of Wilcoxon's rank-sum test are shown in Tables 4–6. The *p*-value less than 0.05 decides the significant differences between two algorithms. According to Tables 4–6, the superiority of the MGTO is statistically significant in most of the benchmark functions since most of the *p*-values are less than 0.05. Thus, the MGTO is considered to have significant advantages compared with other algorithms.

Table 4. Wilcoxon's rank-sum test of classical function.

	GTO	AOA	SSA	WOA	SHO	RSA	ROLGWO	DSCA	HSCAHS
F1	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.000000	1.000000	1.000000	6.10×10^{-5}	6.10×10^{-5}
F2	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.000000	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F3	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.000000	1.000000	1.000000	6.10×10^{-5}	6.10×10^{-5}
F4	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.000000	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F5	0.041260	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.002625	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F6	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F7	0.006714	0.010254	6.10×10^{-5}	0.000122	0.015076	0.002014	0.041260	6.10×10^{-5}	0.018066
F8	0.018066	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F9	1.000000	6.10×10^{-5}	6.10×10^{-5}	0.250000	1.000000	1.000000	1.000000	1.000000	1.000000
F10	1.000000	6.10×10^{-5}	6.10×10^{-5}	0.000977	0.500000	1.000000	0.015625	1.000000	1.000000
F11	1.000000	6.10×10^{-5}	6.10×10^{-5}	0.125000	1.000000	1.000000	1.000000	1.000000	1.000000
F12	0.002014	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F13	0.047913	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F14	0.500000	6.10×10^{-5}	0.015625	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F15	0.005615	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F16	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F17	1.000000	6.10×10^{-5}	0.000977	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F18	0.001953	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F19	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F20	0.031250	6.10×10^{-5}	0.000610	0.005371	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F21	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F22	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F23	1.000000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}

Table 5. Wilcoxon's rank-sum test of CEC2014.

	GTO	AOA	SSA	WOA	SHO	RSA	ROLGWO	DSCA	HSCAHS
F1	0.006714	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F2	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F3	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F4	0.013428	6.10×10^{-5}	0.015076	0.002014	6.10×10^{-5}	6.10×10^{-5}	0.002625	6.10×10^{-5}	6.10×10^{-5}
F5	0.000854	0.002014	0.012451	0.002625	0.000122	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F6	0.000610	6.10×10^{-5}	0.030151	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.004272	6.10×10^{-5}	6.10×10^{-5}
F7	0.004272	6.10×10^{-5}	0.047913	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F8	0.000183	6.10×10^{-5}	0.000183	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.021545	6.10×10^{-5}	6.10×10^{-5}
F9	0.041260	6.10×10^{-5}	0.041260	0.000305	6.10×10^{-5}	6.10×10^{-5}	0.012451	6.10×10^{-5}	6.10×10^{-5}
F10	0.010254	0.000305	0.003357	0.000427	6.10×10^{-5}	6.10×10^{-5}	0.001526	6.10×10^{-5}	6.10×10^{-5}
F11	0.000854	0.000122	0.018066	0.005371	6.10×10^{-5}	6.10×10^{-5}	0.002014	6.10×10^{-5}	6.10×10^{-5}
F12	0.025574	0.012451	0.021545	0.000610	6.10×10^{-5}	6.10×10^{-5}	0.030151	6.10×10^{-5}	6.10×10^{-5}
F13	0.041260	6.10×10^{-5}	0.002014	0.000305	6.10×10^{-5}	6.10×10^{-5}	0.030151	6.10×10^{-5}	6.10×10^{-5}
F14	0.021545	6.10×10^{-5}	0.021545	0.021545	6.10×10^{-5}	6.10×10^{-5}	0.006714	6.10×10^{-5}	6.10×10^{-5}
F15	6.10×10^{-5}	6.10×10^{-5}	0.002014	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.047913	6.10×10^{-5}	6.10×10^{-5}
F16	0.047913	6.10×10^{-5}	0.008362	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.012451	6.10×10^{-5}	6.10×10^{-5}
F17	0.047913	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F18	0.021545	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F19	0.004272	6.10×10^{-5}	0.018066	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.001160	6.10×10^{-5}	6.10×10^{-5}
F20	0.018066	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F21	0.041260	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F22	0.030151	6.10×10^{-5}	6.10×10^{-5}	0.000427	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F23	1.000000	1.000000	6.10×10^{-5}	0.000122	1.000000	1.000000	0.000977	0.003906	1.000000
F24	0.035278	0.031250	0.000854	0.041260	0.031250	0.031250	0.016113	0.004272	0.015625
F25	0.046875	0.031250	0.035339	0.021484	0.031250	0.031250	0.031250	0.031250	0.031250
F26	0.047913	6.10×10^{-5}	0.035339	0.005371	6.10×10^{-5}	6.10×10^{-5}	0.000122	6.10×10^{-5}	6.10×10^{-5}
F27	0.037109	6.10×10^{-5}	0.000610	0.000122	0.031250	0.031250	0.003357	0.015076	0.031250
F28	1.000000	0.000977	6.10×10^{-5}	6.10×10^{-5}	1.000000	1.000000	6.10×10^{-5}	6.10×10^{-5}	1.000000
F29	0.012451	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.000488	0.000488	6.10×10^{-5}	6.10×10^{-5}	0.000305
F30	0.047913	6.10×10^{-5}	0.006714	0.000610	0.000122	0.000122	0.021545	6.10×10^{-5}	6.10×10^{-5}

Table 6. Wilcoxon's rank-sum test of CEC2020.

	GTO	AOA	SSA	WOA	SHO	RSA	ROLGWO	DSCA	HSCAHS
F1	0.000122	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F2	0.010254	0.000427	0.012451	0.003357	6.10×10^{-5}	6.10×10^{-5}	0.041260	6.10×10^{-5}	6.10×10^{-5}
F3	0.021545	6.10×10^{-5}	0.025574	0.000305	6.10×10^{-5}	6.10×10^{-5}	0.021545	6.10×10^{-5}	6.10×10^{-5}
F4	1.000000	1.000000	6.10×10^{-5}	0.031250	1.000000	1.000000	1.000000	1.000000	1.000000
F5	0.025574	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F6	0.047913	6.10×10^{-5}	0.041260	0.001160	6.10×10^{-5}	6.10×10^{-5}	0.000610	6.10×10^{-5}	6.10×10^{-5}
F7	0.035339	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F8	0.002625	6.10×10^{-5}	0.015076	0.002625	6.10×10^{-5}	6.10×10^{-5}	0.041260	6.10×10^{-5}	6.10×10^{-5}
F9	0.006714	0.000122	0.003357	0.000305	6.10×10^{-5}	6.10×10^{-5}	0.041260	0.000183	6.10×10^{-5}
F10	0.021545	6.10×10^{-5}	0.041260	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	0.041260	6.10×10^{-5}	6.10×10^{-5}

5. MGTO for Solving Engineering-Optimization Problems

In this section, seven engineering problems are used to evaluate the superiority of the MGTO, namely the welded-beam-design problem, pressure-vessel-design problem, speed-reducer-design problem, compression/tension-spring-design problem, three-bar-truss-design problem, car-crashworthiness-design problem and the tubular-column-design problem. The MGTO runs independently 30 times for each issue, with the maximum iterations setting at 500 and the population size at 30. In addition, the MGTO is compared

to other algorithms in regard to dealing with engineering problems, as represented in the following tables; the corresponding conclusions are also discussed.

5.1. Welded-Beam Design

The welded-beam-design problem is a common engineering-optimization problem that was developed by Rao et al. [52]. Figure 5 offers a schematic diagram of a welded beam. In this problem, the constraints may be divided into four categories: shear stress, bending stress, buckling load and deflection of beam. The cost of welding materials is minimized as much as possible under the four constraints. Furthermore, there are four variables in the welded-beam-design problem: The thickness of the weld (h), the length of the weld (l), the height of the welded beam (t) and the thickness of the bar (b).

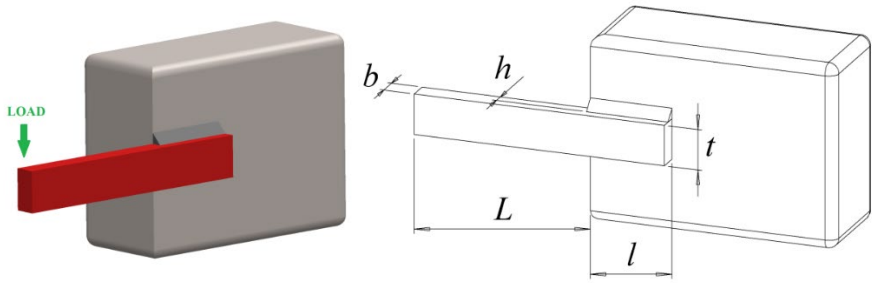

Figure 5. Welded-beam-design problem: three-dimensional model diagram (**left**) and the structural parament (**right**).

The mathematical model of the above problem is presented as follows:
Consider:

$$\vec{x} = [x_1,\ x_2,\ x_3,\ x_4] = [h,\ l,\ t,\ b] \tag{32}$$

Minimize:

$$f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{33}$$

Constraints:

$$\begin{cases} g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \\ g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \\ g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \\ g_4(\vec{x}) = x_1 - x_4 \le 0 \\ g_5(\vec{x}) = P - P_C(\vec{x}) \le 0 \\ g_6(\vec{x}) = 0.125 - x_1 \le 0 \\ g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5 \le 0 \end{cases} \tag{34}$$

where we have the following:

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \tag{35}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},\ M = P\left(L + \frac{x_2}{2}\right) \tag{36}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{37}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \tag{38}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_3^2 x_4}, \ \delta(\vec{x}) = \frac{6PL^3}{Ex_3^3 x_4} \tag{39}$$

$$P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{40}$$

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \tag{41}$$

Variable range:

$$P = 6000 \text{ lb}, L = 14 \text{ in}, \ E = 30 \times 10^6 \text{ psi}, G = 12 \times 10^6 \text{ psi} \tag{42}$$

$$\tau_{max} = 13600 \text{ psi}, \sigma_{max} = 30000 \text{ psi}, \delta_{max} = 0.25 \text{ in} \tag{43}$$

The results of the MGTO and other comparative optimization algorithms are listed in Table 7. According to Table 7, it can be concluded that the MGTO was able to obtain the effective solution for welded-beam-design problems compared with other algorithms.

Table 7. The comparative optimization results for the welded-beam design.

Algorithm	Optimal Values for Variables				Optimal Cost
	h	*L*	*t*	*B*	
MGTO	0.2057	3.2531	9.0366	0.2057	**1.6952**
GTO	0.2059	3.2510	9.0331	0.2059	1.6958
HS	0.2442	6.2231	8.2915	0.2400	2.3807
WOA	0.205395	3.528467	9.004233	0.207241	1.735344
GSA	0.182129	3.856979	10.000	0.202376	1.87995
MVO	0.205463	3.473193	9.044502	0.205695	1.72645
OBSCA	0.230824	3.069152	8.988479	0.208795	1.722315
PHSSA	0.202369	3.544214	9.04821	0.205723	1.72802

5.2. The Pressure-Vessel Problem

The pressure-vessel problem proposed by Kannan and Kramer [53] aims to reduce the cost of the pressure vessel under the pressure requirements. Figure 6 is a schematic diagram of the pressure vessel. The parameters include the following details: the thickness of the shell (*Ts*), the thickness of the head (*Th*), the inner radius of the vessel (*R*) and the length of the cylindrical shape (*L*).

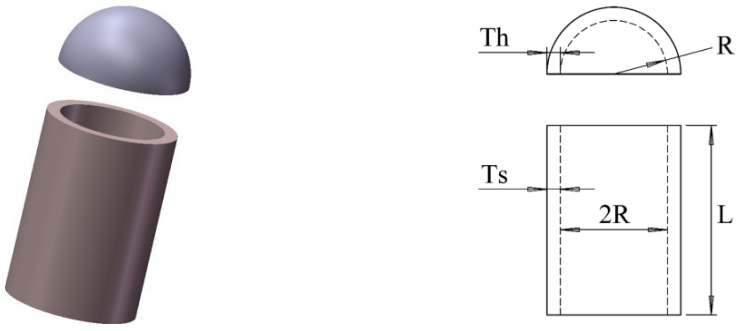

Figure 6. Pressure-vessel-design problem.

Consider:

$$\vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ \mathrm{R} \ L] \tag{44}$$

Minimize:

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{45}$$

Constraints:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0 \tag{46}$$

$$g_2(\vec{x}) = -x_3 + 0.00954x_3 \leq 0 \tag{47}$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \tag{48}$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0 \tag{49}$$

Variable range:

$$\begin{cases} 0 \leq x_1 \leq 99 \\ 0 \leq x_2 \leq 99 \\ 10 \leq x_3 \leq 200 \\ 10 \leq x_4 \leq 200 \end{cases} \tag{50}$$

From Table 8, we can see that the optimal solution of the function obtained by the MGTO is 5734.9131, while x is equal to 0.7424, 0.3702, 40.3196 and 200. The lowest cost calculated by the MGTO is superior to other optimization algorithms, including the GTO [41], HHO [24], SMA [54], WOA [44], GWO [6], MVO [12] and GA [15], indicating that the proposed model has the merits in solving the pressure-vessel problem.

Table 8. The comparative optimization results for pressure-vessel-design problem.

Algorithm	Optimum Variables				Optimum Cost
	Ts	Th	R	L	
MGTO	0.7424	0.3702	40.3196	200	**5734.9131**
GTO	1.2404	0.5844	65.2252	10	7141.3611
HHO	0.8175838	0.4072927	42.09174576	176.7196	6000.46259
SMA	0.7931	0.3932	40.6711	196.2178	5994.1857
WOA	0.8125	0.4375	42.0982699	176.6389	6059.7410
GWO	0.8125	0.4345	42.0892	176.7587	6051.5639
MVO	0.8125	0.4375	42.090738	176.7386	6060.8066
GA	0.8125	0.4375	42.097398	176.6540	6059.94634

5.3. Speed-Reducer Design

The main intention of this problem is to minimize the weights of the speed reducer, which can be limited by seven variables, namely the face width (x_1), the teeth module (x_2), the discrete design variables representing the teeth in the pinion (x_3), the length of the first shaft between the bearings (x_4), the length of the second shaft between the bearings (x_5), the diameters of the first shaft (x_6) and diameters of the second shaft (x_7) [55]. In this case, there are four constraints that should be satisfied: covered stress, bending stress of the gear teeth, stresses in shafts and transverse deflection of the shafts, which are shown in Figure 7.

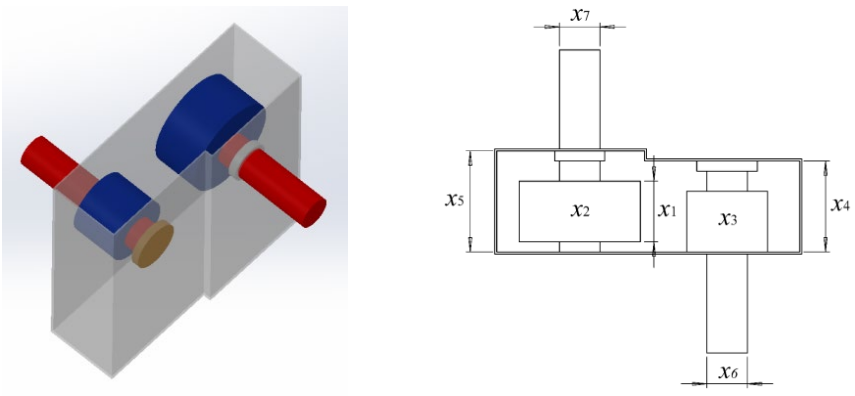

Figure 7. Speed-reducer-design problem.

In addition, the formulas of this problem are expressed as follows:

Minimize:

$$f(\vec{x}) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \quad (51)$$

Constraints:

$$g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0 \quad (52)$$

$$g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0 \quad (53)$$

$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0 \quad (54)$$

$$g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0 \quad (55)$$

$$g_5(\vec{x}) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6}}{110.0x_6^3} - 1 \le 0 \quad (56)$$

$$g_6(\vec{x}) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5 \times 10^6}}{85.0x_6^3} - 1 \le 0 \quad (57)$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \le 0 \quad (58)$$

$$g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \le 0 \quad (59)$$

$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \le 0 \quad (60)$$

$$g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0 \quad (61)$$

$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0 \quad (62)$$

Variable range:

$$\begin{cases} 2.6 \le x_1 \le 3.6 \\ 0.7 \le x_2 \le 0.8 \\ 17 \le x_3 \le 28 \\ 7.3 \le x_4 \le 8.3 \\ 7.8 \le x_5 \le 8.3 \\ 2.9 \le x_6 \le 3.9 \\ 5.0 \le x_7 \le 5.5 \end{cases} \quad (63)$$

The comparative optimization results of the speed-reducer-design problem are shown in Table 9.

Table 9. The comparative optimization results of the speed-reducer-design problem.

Algorithm	Optimum Variables							Optimum Weight
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
MGTO	3.4975	0.7	17	7.3000	7.8000	3.3500	5.2855	**2995.4373**
AO	3.5021	0.7	17	7.3099	7.7476	3.3641	5.2994	3007.7328
PSO	3.5001	0.7	17.0002	7.5177	7.7832	3.3508	5.2867	3145.922
AOA	3.50384	0.7	17	7.3	7.72933	3.35649	5.2867	2997.9157
MFO	3.49745	0.7	17	7.82775	7.71245	3.35178	5.28635	2998.9408
GA	3.51025	0.7	17	8.35	7.8	3.36220	5.28772	3067.561
SCA	3.50875	0.7	17	7.3	7.8	3.46102	5.28921	3030.563
MDA	3.5	0.7	17	7.3	7.67039	3.54242	5.24581	3019.5833

Compared with the AO [56], PSO [20], AOA [43], MFO [57], GA [15], SCA [58] and MDA [59], it can be observed that the MGTO achieves 2995.4373, which is the best optimum weight. Thus, it can be certified that the MGTO has an obvious advantage in solving the speed-reducer-design problem.

5.4. Compression/Tension-Spring Design

Compression/tension-spring-design problems minimize the weight of the spring (shown in Figure 8) [60]. The three constraints impacting frequency, shear stress and deflection should be satisfied in the optimization design. There are three variables displayed in Figure 8, namely the wire diameter (d), mean coil diameter (D) and the number of active coils (N).

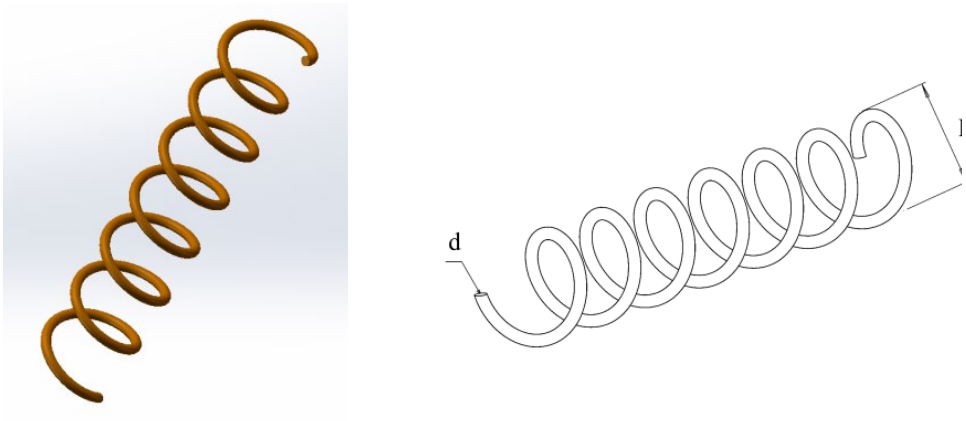

Figure 8. The compression/tension-spring-design problem.

The mathematical model is as follows:
Consider:

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [d\ \mathrm{D}\ \mathrm{N}] \tag{64}$$

Minimize:

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2 \tag{65}$$

Constraints:

$$g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \tag{66}$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} \leq 0 \tag{67}$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \tag{68}$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \tag{69}$$

Variable range:

$$\begin{cases} 0.05 \leq x_1 \leq 2.00 \\ 0.25 \leq x_2 \leq 1.30 \\ 2.00 \leq x_3 \leq 15.00 \end{cases} \tag{70}$$

As Table 10 shows, the optimal solution calculated by MGTO is 0.0099, when variable x is equal to 0.05000, 0.3744 and 8.5465. According to the results, it is clear that the MGTO is capable of obtaining the best solution in comparison with other optimization algorithms.

Table 10. The comparative optimization results of the compression/tension-spring problem.

Algorithm	Optimum Variables			Optimum Weight
	D	D	N	
MGTO	0.05000	0.3744	8.5465	**0.0099**
AO	0.0502439	0.35262	10.5425	0.011165
HHO	0.051796393	0.359305355	11.138859	0.012665443
SSA	0.051207	0.345215	12.004032	0.0126763
WOA	0.051207	0.345215	12.004032	0.0126763
GWO	0.05169	0.356737	11.28885	0.012666
PSO	0.051728	0.357644	11.244543	0.0126747
HS	0.051154	0.349871	12.076432	0.0126706

5.5. Three-Bar-Truss Design

The three-bar-truss-design problem [61] is a typical engineering problem which can be utilized to evaluate the performance of algorithms. The main intention is to minimize the weight, which is subject to deflection, stress and buckling constraints on each of the truss members, by adjusting the cross-sectional areas A_1, A_2 and A_3. It is illustrated in Figure 9.

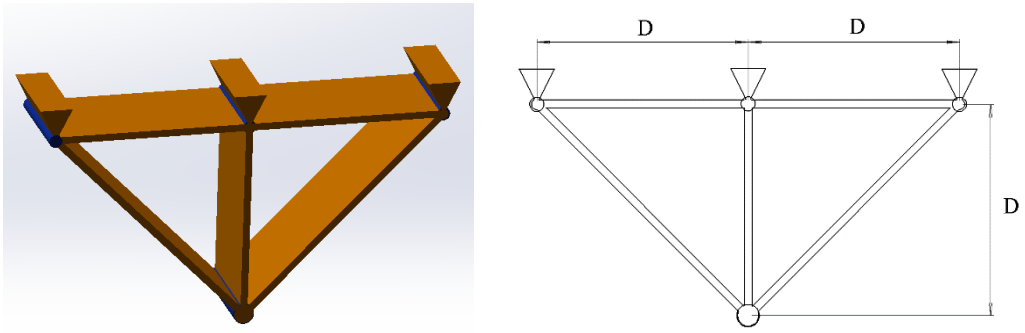

Figure 9. Three-bar-truss-design problem.

The problem considers a non-linear function with three constraints and two decision variables. The mathematical model is given as follows:

Consider:

$$\vec{x} = [x_1\ x_2] = [A_1\ A_2] \tag{71}$$

Minimize:

$$f(\vec{x}) = \left(2\sqrt{2}x_1 + x_2\right) * l \tag{72}$$

Constraints:

$$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \tag{73}$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \tag{74}$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0 \tag{75}$$

Variable range:

$$0 \leq x_1, x_2 \leq 1 \tag{76}$$

where we have the following:

$$l = 100\ \text{cm},\ P = 2KN/\text{cm}^2,\ \sigma = 2KN/\text{cm}^2 \tag{77}$$

The results of the three-bar-truss-design problem are shown in Table 11, and it can be seen that the MGTO has significant superiority compared with other optimization

algorithms in solving this problem. The proposed model can obtain the lowest optimum weight when x is equal to 0.7884 and 0.4081.

Table 11. The comparative optimization results of the three-bar-truss-design problem.

Algorithm	Optimum Variables		Optimum Weight
	x_1	x_2	
MGTO	0.7884	0.4081	**263.8523464**
AO	0.7926	0.3966	263.8684
HHO	0.788662816	0.408283133832900	263.8958434
SSA	0.78866541	0.408275784	263.89584
AOA	0.79369	0.39426	263.9154
MVO	0.78860276	0.408453070000000	263.8958499
MFO	0.788244771	0.409466905784741	263.8959797
GOA	0.788897555578973	0.407619570115153	263.895881496069

5.6. Car-Crashworthiness Design

The car-crashworthiness-design problem proposed by Gu et al. aims to improve the safety of the vehicle and reduce casualties. The details are shown in Figure 10 [62].

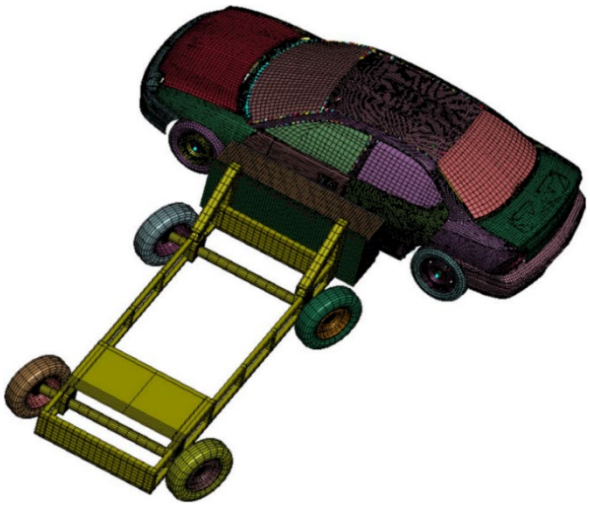

Figure 10. Three-dimensional model diagram of car's side crash.

In this case, eleven parameters are adopted to reduce the weight, namely the inside of the B-pillar, reinforcement of the B-pillar, the inside of the floor side, the crossbeam, the door beam, the door-strip-line reinforcement, the roof rail of the car (x_1–x_7), the materials of inside of B-pillar and reinforcement of B-pillar (x_8, x_9), the thickness of barrier height and the impact position (x_{10}, x_{11}). The mathematical formula is represented as follows:

Minimize:

$$f(\vec{x}) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7 \tag{78}$$

Constraints:

$$g_1(\vec{x}) = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10} \leq 1 \tag{79}$$

$$\begin{gathered} g_2(\vec{x}) = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 \\ +0.0144x_3x_5 + 0.0008757x_5x_{10} + 0.080405x_6x_9 \\ +0.00139x_8x_{11} + 0.00001575x_{10}x_{11} \leq 0.32 \end{gathered} \tag{80}$$

$$g_3(\vec{x}) = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 \\ +0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 \\ +0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} \\ -0.0005354x_6x_{10} + 0.00121x_8x_{11} \leq 0.32 \tag{81}$$

$$g_4(\vec{x}) = 0.074 - 0.061x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} \\ -0.166x_7x_9 + 0.227x_2^2 \leq 0.32 \tag{82}$$

$$g_5(\vec{x}) = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} \\ +6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} \leq 32 \tag{83}$$

$$g_6(\vec{x}) = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 \\ -11.0x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22.0x_8x_9 \leq 32 \tag{84}$$

$$g_7(\vec{x}) = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10} \leq 32 \tag{85}$$

$$g_8(\vec{x}) = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} \\ +0.009325x_6x_{10} + 0.000191x_{11}^2 \leq 4 \tag{86}$$

$$g_9(\vec{x}) = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} \\ -0.0198x_4x_{10} + 0.028x_6x_{10} \leq 9.9 \tag{87}$$

$$g_{10}(\vec{x}) = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} \\ -0.0556x_9x_{11} - 0.000786x_{11}^2 \leq 15.7 \tag{88}$$

Variable range:

$$\begin{cases} 0.5 \leq x_1 - x_7 \leq 1.5 \\ x_8,\ x_9 \in (0.192,\ 0.345) \\ -30 \leq x_{10},\ x_{11} \leq 30 \end{cases} \tag{89}$$

The results of the car-crashworthiness-design problem are listed in Table 12. From the contents, we can see that the optimal weight obtained by the MGTO is 23.1894. Furthermore, it is certain that the MGTO is an efficient algorithm in solving the car-crashworthiness-design problem.

Table 12. The comparative optimization results of the car-crashworthiness problem.

Algorithm	MGTO	GTO	AOA	SSA	WOA	ROLGWO	PSO
x1	0.5000	1.3249	0.50000	0.6770	1.1079	0.5008	0.5000
x2	1.2294	0.8819	1.2827	1.1695	1.1202	1.2439	1.2222
x3	0.5000	0.5000	0.6364	0.5000	0.6965	0.5000	1.5000
x4	1.2006	1.2322	1.3636	1.1805	1.1535	1.1921	0.7412
x5	0.5000	0.5000	0.50000	0.9549	0.6840	0.5037	0.5000
x6	1.0917	1.0985	1.50000	0.9809	0.8071	1.2917	1.5000
x7	0.5000	0.5000	0.50000	0.5000	0.9805	0.5012	0.5000
x8	0.3450	0.34385	0.3450	0.3413	0.2748	0.3449	0.34500
x9	0.3450	0.19200	0.3084	0.2229	0.2587	0.2536	0.34500
x10	0.6436	4.4782	0.5795	3.5908	8.4836	3.1707	−0.20845
x11	0.3162	2.6609	−9.62291	−1.22041	17.8255	3.3177	3.2759
Optimal weight	**23.1894**	23.2046	25.8678	23.3121	28.9580	23.2527	23.1902

5.7. Tubular-Column Design

According to Figure 11, a tubular column is given to decrease the cost of supporting the compressive load, p = 2500 kgf, and it consists of the yield stress, elastic modulus and density.

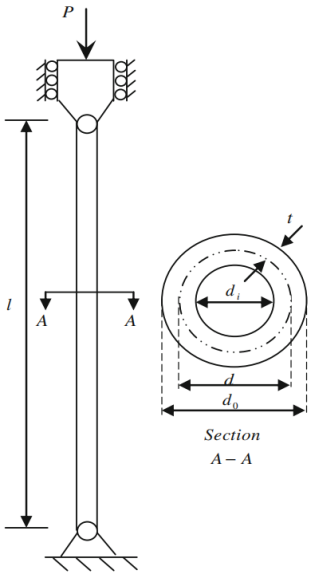

Figure 11. The schematic diagram of the tubular column.

The functions, including the material and manufacturing costs, are considered. The formulas can be expressed as follows:

Minimize:

$$f(d,\ t) = 9.8dt + 2d \tag{90}$$

Constraints:

$$g_1 = \frac{P}{\pi dt\sigma_y} - 1 \le 0 \tag{91}$$

$$g_2 = \frac{8PL^2}{\pi^3 Edt(d^2 + t^2)} - 1 \le 0 \tag{92}$$

$$g_3 = \frac{2.0}{d} - 1 \le 0 \tag{93}$$

$$g_4 = \frac{d}{14} - 1 \le 0 \tag{94}$$

$$g_5 = \frac{0.2}{t} - 1 \le 0 \tag{95}$$

Variable range:

$$0.01 \le d,\ T \le 100 \tag{96}$$

The comparative optimization results of the tubular-column-design problem are shown in Table 13.

Table 13. The comparative optimization results of the tubular-column-design problem.

Algorithm	Optimal Values for Variables		Optimal Cost
	d	*T*	
MGTO	**5.4511**	**0.2919**	**26.5313**
HSCAHS	5.4170	0.3128	27.4745
AOA	7.5313	0.2223	31.5088
WOA	5.7562	0.2764	27.1415
GWO	5.4513	0.2919	26.5333
ROLGWO	5.4510	0.2919	26.5326
DSCA	5.5179	0.2899	26.7494

The optimal solution obtained by the MGTO is 26.5313 when the variable x is 5.4511 and 0.2919. It is obvious that the proposed algorithm has advantages in solving the tubular-column-design problem.

In summary, the advantages of the proposed algorithm are shown in this section. Because the MGTO has great exploration and exploitation ability, it is superior to the traditional GTO and other existing algorithms in regard to performance. Therefore, the MGTO can be applied in practical engineering problems.

6. Conclusions and Future Work

The MGTO was proposed to modify the performance of GTO in this paper. Three strategies were used to modify the basic GTO. Firstly, the QIBAS algorithm was utilized to increase the diversity of the position of the silverback. Secondly, TLBO was introduced to the exploitation phase to reduce the difference of the silverback and gorillas. Thirdly, the QRBL generates the quasi-refraction position of the silverback to enhance the quality of the optimal position.

For the comprehensive evaluation, the proposed MGTO was compared to the basic GTO and eight other state-of-the-art optimization algorithms, namely AOA, SSA, WOA, GWO, PSO, ROLGWO, DSCA and HSCAHS, on 23 classical benchmark functions, 30 CEC2014 benchmark functions, and 10 CEC2020 benchmark functions. The statistical analysis results disclosed that MGTO is a very competitive algorithm and outperforms other algorithms in regard to exploitation, exploration, escaping local optimum, and convergence behavior. In order to further investigate the superior capability of the MGTO in solving real-word engineering problems, the proposed algorithm was compared with other algorithms, using seven constrained, complex and challenging problems. The obtained results confirmed the high competency of MGTO to optimize real-word problems with complicated and unknown search domains.

In future works, we hope that the MGTO will perform particularly well on more real-world problems, such as image segmentation, feature selection and so on. Moreover, the NFL theorem has also prompted researchers to improve more algorithms.

Author Contributions: Conceptualization, T.W. and D.W.; methodology, T.W. and D.W.; software, T.W. and H.J.; validation, T.W. and N.Z; formal analysis, T.W. and D.W.; investigation, T.W. and D.W; resources, D.W. and H.J.; data curation, T.W. and N.Z.; writing—original draft preparation, T.W. and D.W.; writing—review and editing, Q.L., K.H.A., L.A. and H.J.; visualization, T.W. and D.W.; supervision, D.W. and H.J.; funding acquisition, D.W., K.H.A. and H.J. 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 the Deanship of Scientific Research at Umm AI-Qura University for supporting this work thought grant (22UQU4320277DSR14). The authors would like to thank the anonymous reviewers and the editor for their careful reviews and constructive suggestions to help us improve the quality of this paper.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Khajehzadeh, M.; Iraji, A.; Majdi, A.; Keawsawasvong, S.; Nehdi, M.L. Adaptive Salp Swarm Algorithm for Optimization of Geotechnical Structures. *Appl. Sci.* **2022**, *12*, 6749. [CrossRef]
2. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]

3. Das, S.; Das, P.; Das, P. Chemical and Biological Control of Parasite-Borne Disease Schistosomiasis: An Impulsive Optimal Control Approach. *Nonlinear Dyn.* **2021**, *104*, 603–628. [CrossRef]
4. Das, P.; Upadhyay, R.K.; Misra, A.K.; Rihan, F.A.; Das, P.; Ghosh, D. Mathematical Model of COVID-19 with Comorbidity and Controlling Using Non-Pharmaceutical Interventions and Vaccination. *Nonlinear Dyn.* **2021**, *106*, 1213–1227. [CrossRef]
5. Das, P.; Das, S.; Upadhyay, R.K.; Das, P. Optimal Treatment Strategies for Delayed Cancer-Immune System with Multiple Therapeutic Approach. *Chaos Solitons Fractal.* **2020**, *136*, 109806. [CrossRef]
6. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
7. Liu, Q.; Li, N.; Jia, H.; Qi, Q.; Abualigah, L. Modified Remora Optimization Algorithm for Global Optimization and Multilevel Thresholding Image Segmentation. *Mathematics* **2022**, *10*, 1014. [CrossRef]
8. Das, P.; Upadhyay, R.K.; Das, P.; Ghosh, D. Exploring Dynamical Complexity in a Time-Delayed Tumor-Immune Model. *Chaos* **2020**, *30*, 123118. [CrossRef]
9. Das, P.; Das, S.; Das, P.; Rihan, F.A.; Uzuntarla, M.; Ghosh, D. Optimal Control Strategy for Cancer Remission Using Combinatorial Therapy: A Mathematical Model-Based Approach. *Chaos Solitons Fractals* **2021**, *145*, 110789. [CrossRef]
10. Wang, S.; Jia, H.; Laith, A.; Liu, Q.; Zheng, R. An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems. *Processes* **2021**, *9*, 1551. [CrossRef]
11. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
12. 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]
13. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
14. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium Optimizer: A Novel Optimization Algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [CrossRef]
15. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
16. Hussain, S.F.; Iqbal, S. Genetic ACCGA: Co-Similarity Based Co-Clustering Using Genetic Algorithm. *Appl. Soft Comput.* **2018**, *72*, 30–42. [CrossRef]
17. 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]
18. Yao, X.; Liu, Y.; Lin, G. Evolutionary Programming Made Faster. *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–102.
19. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1922.
20. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN95—International Conference on Neural Networks, Perth, WA, Australia, 27 November 1995–1 December 1995; pp. 1942–1948.
21. 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.
22. Yang, X.S. Firefly algorithm. In *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Bristol, UK, 2010; Volume 2, pp. 1–148.
23. Yang, X.S.; Gandomi, A.H. Bat Algorithm: A Novel Approach for Global Engineering Optimization. *Eng. Comput.* **2012**, *29*, 464–483. [CrossRef]
24. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris Hawks Optimization: Algorithm and Applications. *Future Gener. Comput. Systems.* **2019**, *97*, 849–872. [CrossRef]
25. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A Nature-Inspired Metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 11337. [CrossRef]
26. 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]
27. Jiang, X.; Li, S. BAS: Beetle Antennae Search Algorithm for Optimization Problems. *Int. J. Robot. Control* **2017**. [CrossRef]
28. Zhou, J.; Qian, Q.; Fu, Y.; Feng, Y. Flower pollination algorithm based on beetle antennae search method. In *Smart Innovation, Systems and Technologies*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 181–189.
29. Karaboga, D.; Akay, B. A Comparative Study of Artificial Bee Colony Algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132. [CrossRef]
30. Cheng, L.; Yu, M.; Yang, J.; Wang, Y. An improved artificial bee colony algorithm based on beetle antennae search. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; p. 23122316.
31. Li, Z.; Li, S.; Luo, X. A novel quadratic interpolated beetle antennae search for manipulator calibration. *arXiv* **2022**, arXiv:2204.06218.
32. Rao, R.V.; Savsani, V.J.; Vakharia, D. Teaching-Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems. *Comput. Aided Des.* **2011**, *43*, 303–315. [CrossRef]
33. Tuo, S.; Yong, L.; Deng, F.; Li, Y.; Lin, Y.; Lu, Q. HSTLBO: A hybrid algorithm based on Harmony Search and Teaching-Learning-Based Optimization for Complex High-Dimensional Optimization Problems. *PLoS ONE.* **2017**, *12*, e0175114. [CrossRef]
34. Keesari, H.S.; Rao, R.V. Optimization of Job Shop Scheduling Problems Using Teaching-Learning-Based Optimization Algorithm. *Opsearch* **2014**, *51*, 545–561. [CrossRef]
35. Chen, D.; Zou, F.; Li, Z.; Wang, J.; Li, S. An Improved Teaching-Learning-Based Optimization Algorithm for Solving Global Optimization Problem. *Inf. Sci.* **2015**, *297*, 171–190. [CrossRef]

36. Fan, Q.; Chen, Z.; Xia, Z. A Novel Quasi-Reflected Harris Hawks Optimization Algorithm for Global Optimization Problems. *Soft Comput.* **2020**, *24*, 14825–14843. [CrossRef]
37. Ahandani, M.A.; Alavi-Rad, H. Opposition-Based Learning in the Shuffled Bidirectional Differential Evolution Algorithm. *Soft Comput.* **2016**, *16*, 1303–1337. [CrossRef]
38. Simon, D. Biogeography-Based Optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
39. Wang, Y.J.; Ma, C.L. Opposition-Based Learning Differential Ion Motion Algorithm. *J. Inf. Hid. Multimed. Signal Process.* **2018**, *9*, 987–996.
40. Abedinia, O.; Naslian, M.D.; Bekravi, M. A New Stochastic Search Algorithm Bundled Honeybee Mating for Solving Optimization Problems. *Neural Comput. Appl.* **2014**, *25*, 1921–1939. [CrossRef]
41. Abdollahzadeh, B.; Soleimanian 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]
42. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
43. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The Arithmetic Optimization Algorithm. *Comput Meth. Appl. Mat.* **2021**, *376*, 113609. [CrossRef]
44. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
45. Long, W.; Jiao, J.; Liang, X.; Cai, S.; Xu, M. A Random Opposition-Based Learning Grey Wolf Optimizer. *IEEE Access* **2019**, *7*, 113810–113825. [CrossRef]
46. Li, Y.; Zhao, Y.; Liu, J. Dynamic Sine Cosine Algorithm for Large-Scale Global Optimization Problems. *Expert Syst. Appl.* **2021**, *177*, 114950. [CrossRef]
47. Singh, N.; Kaur, J. Hybridizing Sine-Cosine Algorithm with Harmony Search Strategy for Optimization Design Problems. *Soft Comput.* **2021**, *25*, 11053–11075. [CrossRef]
48. Sun, K.; Jia, H.; Li, Y.; Jiang, Z. Hybrid Improved Slime Mould Algorithm with Adaptive B Hill Climbing for Numerical Optimization. *J. Intell. Fuzzy Syst.* **2021**, *40*, 1667–1679. [CrossRef]
49. 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.
50. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced Nonparametric Tests for Multiple Comparisons in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power. *Inf. Sci.* **2010**, *180*, 2044–2064. [CrossRef]
51. Demsar, J. Statistical Comparisons of Classifiers Over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
52. Jia, H.; Peng, X.; Lang, C. Remora Optimization Algorithm. *Expert Syst. Appl.* **2021**, *185*, 115665. [CrossRef]
53. Kannan, B.; Kramer, S. An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design. *J. Mech. Des.* **1994**, *116*, 405–411. [CrossRef]
54. Li, S.; Che, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Gener. Comp. Sy.* **2020**, *111*, 300–323. [CrossRef]
55. Jia, H.; Sun, K.; Zhang, W.; Leng, X. An Enhanced Chimp Optimization Algorithm for Continuous Optimization Domains. *Complex Intell. Syst.* **2022**, *8*, 65–82. [CrossRef]
56. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-qaness, M.A.A.; Gandomi, A.H. Aquila Optimizer: A Novel Meta-Heuristic Optimization Algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
57. Mirjalili, S. Moth–Flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm. *Knowl.–Based Syst.* **2015**, *89*, 228–249. [CrossRef]
58. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl.–Based Syst.* **2016**, *96*, 120–133. [CrossRef]
59. Lu, S.; Kim, H.M. A Regularized Inexact Penalty Decomposition Algorithm for Multidisciplinary Design Optimization Problems with Complementarity Constraints. *J. Mech. Des.* **2010**, *132*, 041005. [CrossRef]
60. Liu, Q.; Li, N.; Jia, H.; Qi, Q.; Abualigah, L.; Liu, Y. A Hybrid Arithmetic Optimization and Golden Sine Algorithm for Solving Industrial Engineering Design Problems. *Mathematics* **2022**, *10*, 1567. [CrossRef]
61. Ray, T.; Saini, P. Engineering Design Optimization Using a Swarm with an Intelligent Information Sharing Among Individuals. *Eng. Optim.* **2001**, *33*, 735–748. [CrossRef]
62. Gu, L.; Yang, R.; Tho, C.H.; Makowskit, M.; Faruquet, O.; Li, Y.L. Optimisation and Robustness for Crashworthiness of Side Impact. *Int. J. Veh. Des.* **2001**, *26*, 348–360. [CrossRef]