mathematics

MDPI

Article

Mountaineering Team-Based Optimization: A Novel Human-Based Metaheuristic Algorithm

Iman Faridmehr [1], Moncef L. Nehdi [2,*], Iraj Faraji Davoudkhani [3] and Alireza Poolad [4]

[1] Department of Construction Production and Theory of Structures, Institute of Architecture and Construction, South Ural State University, 454080 Chelyabinsk, Russia
[2] Department of Civil Engineering, McMaster University, Hamilton, ON L8S 4M6, Canada
[3] Energy Management Research Center, University of Mohaghegh Ardabili, Ardabil 5619911367, Iran
[4] Department of Electrical Engineering, Islamic Azad University, Bushehr Branch, Bushehr 7519619555, Iran
* Correspondence: nehdim@mcmaster.ca

Abstract: This paper proposes a novel optimization method for solving real-world optimization problems. It is inspired by a cooperative human phenomenon named the mountaineering team-based optimization (MTBO) algorithm. Proposed for the first time, the MTBO algorithm is mathematically modeled to achieve a robust optimization algorithm based on the social behavior and human cooperation needed in considering the natural phenomena to reach a mountaintop, which represents the optimal global solution. To solve optimization problems, the proposed MTBO algorithm captures the phases of the regular and guided movement of climbers based on the leader's experience, obstacles against reaching the peak and getting stuck in local optimality, and the coordination and social cooperation of the group to save members from natural hazards. The performance of the MTBO algorithm was tested with 30 known CEC 2014 test functions, as well as on classical engineering design problems, and the results were compared with that of well-known methods. It is shown that the MTBO algorithm is very competitive in comparison with state-of-art metaheuristic methods. The superiority of the proposed MTBO algorithm is further confirmed by statistical validation, as well as the Wilcoxon signed-rank test with advanced optimization algorithms. Compared to the other algorithms, the MTBO algorithm is more robust, easier to implement, exhibits effective optimization performance for a wide range of real-world test functions, and attains faster convergence to optimal global solutions.

Keywords: optimization; mountaineering team-based optimization; human cooperation; benchmark function; heuristic algorithm

MSC: 5K10; 68Q25; 68T20

check for updates

Citation: Faridmehr, I.; Nehdi, M.L.; Davoudkhani, I.F.; Poolad, A. Mountaineering Team-Based Optimization: A Novel Human-Based Metaheuristic Algorithm. *Mathematics* **2023**, *11*, 1273. https://doi.org/10.3390/math11051273

Academic Editors: Wanquan Liu, Xianchao Xiu, Xuefang Li and David Barilla

Received: 7 January 2023
Revised: 1 March 2023
Accepted: 3 March 2023
Published: 6 March 2023

1. Introduction

Numerous metaheuristic optimization algorithms have been proposed in recent years, several of which have been deployed in solving engineering problems. The main performance features of such methods include a simple structure with easy implementation, not requiring gradient data, and not getting caught in premature convergence [1,2]. Metaheuristic algorithms inspired by nature solve optimization problems by imitating biological or physical phenomena. They are generally divided into four categories (Figure 1), including evolutionary algorithms, methods based on swarm intelligence, physics-based algorithms, and human-based algorithms.

Evolution-based methods are modeled on the laws of natural selection. In these methods, upon random generation of a population, the search is started and evolves in the next generations. The advantage of these methods is the combining of the fittest individuals to form the next generation for optimizing the population. The most well-known of these methods include the genetic algorithm (GA) [3], the evolution strategy [4], biogeography-based optimization (BBO) [5], and genetic programming (GP) [6]. The second group of metaheuristic methods includes swarm intelligence methods based on the social behavior of animals. The most popular of these methods include the particle swarm optimization (PSO) algorithm [7], the ant colony optimization algorithm (ACO) [8], the artificial bee algorithm (ABC) [9], the glowworm swarm optimization algorithm (GSO) [10], the grey wolf optimization (GWO) algorithm [11], the firefly algorithm (FA) [12], and the spotted hyena optimization (SHO) algorithm [13]. In Ref. [14], the optimization of the non-linear Hammerstein model is evaluated via the marine predator algorithm's (MPA) capabilities as a population-based optimization based on the predators' strategy for catching prey. In Ref. [15], an optimization method is presented based on the dwarf mongoose optimization algorithm (DMOA) to estimate the autoregressive exogenous (ARX) model parameter. In Ref. [16], a metaheuristic algorithm named the Aquila optimizer (AO) is used to determine the control autoregressive (CAR) model parameter. In Ref. [17], the parameter estimation of power system harmonics is investigated through the swarm intelligence-based optimization strength of the cuckoo search algorithm (CSA). In Ref. [18], a fractional hierarchical gradient descent (FHGD) algorithm is presented based on the standard hierarchical gradient descent generalization of the fractional order to solve the non-linear system problem. In Ref. [19], an optimization method named the flower pollination algorithm is applied to estimate the identification problems in non-linear active noise control systems.

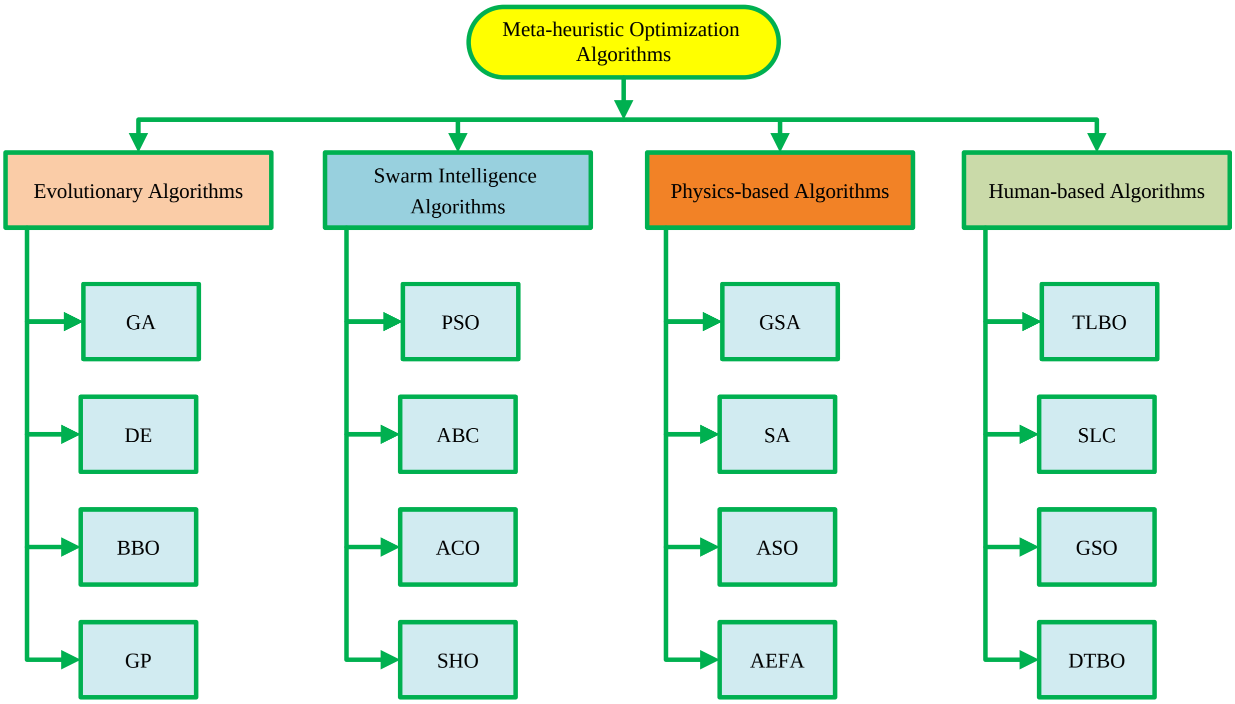


Figure 1. Categories of metaheuristic optimization algorithms.

Physics-based algorithms are inspired by nature's physical laws. The most popular of these methods include the gravitational search algorithm (GSA) [20], the simulated annealing algorithm (SA) [21], the atom search optimization (ASO) algorithm [22], the artificial electric field algorithm (AEFA) [23], the big bang–big crunch (BBBC) algorithm [24], the small world optimization algorithm (SWOA) [25], the galaxy-based search algorithm (GbSA) [26], the black hole (BH) algorithm [27], the vortex search algorithm (VSA) [28], and the electromagnetism-like mechanism (EM) algorithm [29]. The fourth category includes metaheuristic algorithms inspired by human behavior. The most popular of these methods include teaching–learning-based optimization (TLBO) [30], the harmony search (HS) [31], the tabu search (TS) [32], the group search optimizer (GSO) [33], the imperialist competitive algorithm (ICA) [34], the league championship algorithm (LCA) [35], the firework algorithm (FA) [36], the soccer league competition (SLC) [37], the seeker optimization algorithm (SOA) [38], the exchange market algorithm (EMA) [39], group counseling optimization (GCO) [40], and the driving training-based optimization (DTBO) algorithm [41].

Population-based metaheuristic algorithms share a common characteristic beyond their nature. These algorithms divide the search process into two phases: exploration and exploitation [42–46]. In the exploration phase, the algorithm must have operators to explore the search space to find the global optimum. In the exploitation phase, the algorithm can find the promising area of the search space. Therefore, the exploitation phase is related to the local search capability in the promising area of the search space discovered in the exploration phase. Creating a balance between these two phases, owing to the optimization randomness, is an essential challenge for developing the metaheuristic algorithm. A question that arises is, considering all these algorithms, is there a real need for new algorithms?

The significance and function of optimization in numerous disciplines of science have become more obvious with the development of science and technology. Hence, to meet the numerous optimization issues, useful tools are required. Faced with the diversity of challenging real-world problems in different areas of science and engineering [47–63], scientists must solve a wide range of complex problems with different objective functions, including linear or non-linear, single-objective or multi-objective functions, which are unlike each other. No single algorithm can solve all such optimization problems based on the no-free-lunch (NFL) theorem [64]. On the other hand, the inherent nature of metaheuristic algorithms is such that they may have the best possible performance in solving several functions, while on the other hand, the same algorithm may not perform well at solving other functions of a different type. Therefore, each algorithm can cover only a certain set of test functions well. Therefore, most scientific branches have widely recognized the need for a comprehensive and robust algorithm that is versatile to handle a comprehensive set of functions with various objectives.

In the present study, a new metaheuristic algorithm, the mountaineering team-based optimization (MTBO) algorithm, inspired by humans' social performance and cooperation, by considering natural phenomena, is presented. This algorithm is novel, and as far as the authors know, there is no previous study on this algorithm in previous optimization studies. The performance of the MTBO algorithm in solving real-world functions, basic and common standard test functions, CEC 2014 benchmark functions (unimodal, simple multimodal, hybrid, and composition test functions), as well as a wide range of common engineering design problems, is investigated herein. In each optimization stage, the MTBO algorithm's performance is compared with that of several modern and standard algorithms. The optimization results have shown that the MTBO algorithm is very competitive compared to common optimization methods. The advantages of the proposed MTBO algorithm are as follows:

i. A novel metaheuristic algorithm inspired by the social performance and cooperation of humans by considering natural phenomena;
ii. Has proper and effective MTBO optimization performance for a wide range of real-world functions compared to other well-known algorithms;
iii. Has a simple optimization process with superior robustness compared to other algorithms;
iv. Characteristic of fast and appropriate convergence to an acceptable and global optimal solution compared to modern algorithms;
v. A new algorithm based on the effective optimization of real and modern test functions.

The subsequent structure of this paper is as follows. In Section 2, the optimization through a mountaineering team-based algorithm is formulated. In Section 3, Appendix A, and Appendix B, the performance of the proposed algorithm is implemented on test functions, and the optimization results are presented. In Section 4, the performance of the proposed algorithm in solving real-world engineering problems is evaluated, and the results are analyzed. The findings obtained from the proposed algorithm and suggestions for future work are presented in Section 5.

2. Mountaineering Team-Based Optimization (MTBO)

2.1. Inspiration

Common optimization algorithms can be classified into local optimization and global optimization. Evolutionary methods are often used for global optimization. It is clear that intellectual and environmental evolution with the coordinated behavior of humans takes place much faster than physical and genetic evolution. Therefore, the cultural evolution and human perspective have not been ignored, and a group of algorithms, known as cultural algorithms, have been introduced. Cultural algorithms are actually not a completely new category of algorithms. Rather, the main idea is that by adding the possibility of cultural evolution (by capturing the possibility of exchanging information between members of the population) to the existing algorithms, they increase the speed of convergence, as expected.

In this paper, a new optimization algorithm is introduced in the field of evolutionary computations, which is based on intellectual and environmental evolution with coordinated human behavior. A mountaineering team consists of a number of mountaineers with an experienced and professional leader whose goal is to conquer the mountaintop in the region, where the mountaintop is considered the final global solution to the optimization problem [64–67]. Like other evolutionary optimization methods, the developed algorithm starts with an initial population. In this algorithm, each population member is called a mountaineering team member or mountaineer. This algorithm's core is the mountaineers' regular and coordinated movement and the consideration of the natural phenomena. According to the regular and coordinated movement phase, the mountaineers are coordinated by their teammates and also the group leader, which in optimization science is equivalent to the best solution in the current iteration of the algorithm to reach their goal, which is to conquer the mountaintop, or in optimization, the science to reach the global optimum or the best solution. In presenting this algorithm, natural disasters such as avalanches are also considered, which can hinder the progress of the mountaineers and even endanger their lives. The main inspiration of the MTBO algorithm is the team's orderly and coordinated movement to conquer the mountaintop, considering the natural disasters, formulated below in rational steps.

2.2. Mathematical Model

2.2.1. First Phase: Coordinated Mountaineering

In a mountaineering team, the group's most experienced member is always chosen as the leader and front of the group, which in optimization science is equivalent to the best solution in the current iteration of the algorithm. Here, the best member of the population of the algorithm, or equivalently, the mountaineering group, assumes this role. This member leads the best or the whole group towards the destination or goal to conquer the mountaintop or, equivalently, to reach the optimal global solution. Therefore, the members move toward the group leader as follows:

$$X_i^{\text{new}} = X_i + rand \times (X_{Leader} - X_i) \tag{1}$$

It should be noted that in a mountaineering team, the movement is organized under the supervision of the group leader, and usually, the members are organized from the best to the worst, and each member, in addition to being guided by the group leader, is also guided, and directed by the member just in front. It can be assumed that the equivalent in the MTBO algorithm is that after each iteration, the population is ordered from the best to the worst, and each individual is guided through the group leader and the individual in front of him/her. Therefore, the equation of regular movement towards the mountaintop is modified in the following form:

$$X_i^{\text{new}} = X_i + rand \times (X_{Leader} - X_i) + rand \times (X_{ii} - X_i) \tag{2}$$

where X_{ii} is the position of the individual member directed by the member just in front.

On the other hand, in the optimization world, every action happens randomly, and the probability of this phase is assumed to be equal to Li, and hence the pseudo-code of this phase is as follows:

$$\begin{aligned}&if\ rand < Li\\&X_i^{\text{new}} = X_i + rand \times (X_{Leader} - X_i) + rand \times (X_{ii} - X_i)\\&\boldsymbol{end}\end{aligned}$$

2.2.2. Second Phase: Effect of Natural Disasters

Several natural disasters could threaten the lives of the mountaineers and prevent them from reaching the mountaintop, or in other words, trap the population in local optima that are likely to occur at any moment. Figure 2 shows the common threat that mountaineers may face when conquering the mountain peak. The most important thing in the MTBO algorithm is the occurrence of an avalanche and falling off the cliff. In the MTBO, the basis of the optimization process is mostly based on natural disasters, i.e., avalanches. Therefore, the probability of this phase or the occurrence of an avalanche is higher than in other conditions. Therefore, in the MTBO, the critical situation in the event of an avalanche that occurs randomly in nature is considered to be equivalent to the worst situation of the algorithm, X_{Worst} or its equivalent $X_{\text{Avalanche}}$, and that in the event of an avalanche and other calamities, the ith individual tries to get away from the calamity situation, X_{Worst} or its equivalent $X_{\text{Avalanche}}$, and save herself/himself through the below inspired equation. In other words, inspired by the science of optimization, the individual is saved from getting stuck in the optimal local solution and moves towards the global optimization of the best possible solution.

$$X_i^{\text{new}} = X_i - rand \times (X_{\text{Avalanche}} - X_i) \tag{3}$$

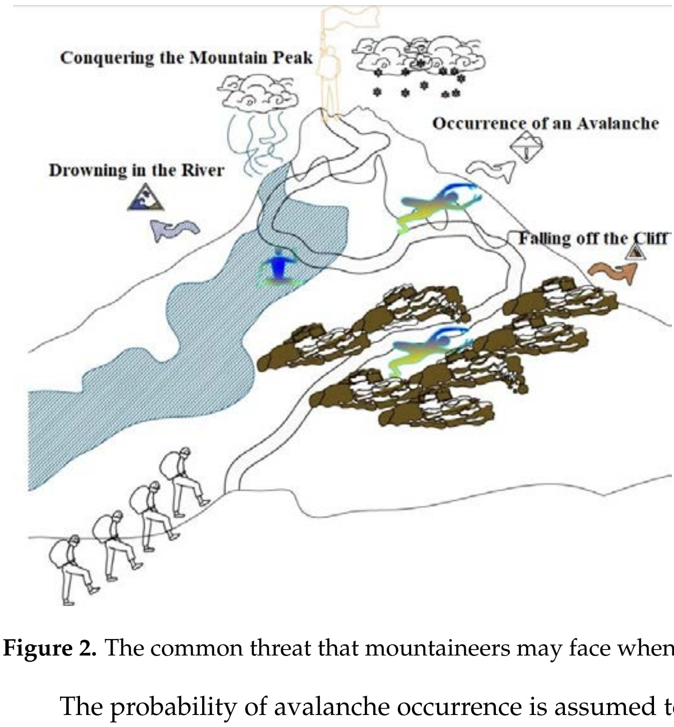


Figure 2. The common threat that mountaineers may face when conquering the mountain peak.

The probability of avalanche occurrence is assumed to be equal to Ai, and the pseudo-code of this phase is presented as follows:

$$\begin{aligned}&if\ rand < Ai\\&X_i^{\text{new}} = X_i - rand \times (X_{\text{Avalanche}} - X_i)\\&\boldsymbol{end}\end{aligned}$$

2.2.3. Third Phase: Coordinated and Group Effort against Disasters

The main difference between human groups and other phenomena and beings is that humans help and guide each other in an informed, organized, and highly effective manner. This social and cooperative behavior is a vital skill in a mountaineering team. Therefore, in a mountaineering team, when any calamity occurs, the entire team will try to save the trapped member in the case of possible disaster or getting stuck. Therefore, the MTBO is inspired by the concerted and social effort and cooperation of the group to save the trapped member, i.e., the position of all the members is considered equal to their average position, X_{mean} or X_{Team} that the ith individual is toward the position X_{mean} or X_{Team}; this behavior is modeled as follows:

$$X_i^{\text{new}} = X_i + rand \times (X_{\text{Team}} - X_i) \tag{4}$$

The probability of saving an individual trapped by an avalanche, or in other words, trapped in the optimal local solution, is assumed to be equal to Mi, and the pseudo-code of this phase is presented as follows:

$$\begin{aligned} &\textbf{\textit{if}}\ rand < Mi \\ &X_i^{\text{new}} = X_i + rand \times (X_{\text{Team}} - X_i) \\ &\textbf{\textit{end}} \end{aligned}$$

2.2.4. Fourth Phase: Possible Death of Members

Unfortunately, it has been observed that sometimes, due to an avalanche's intensity, a mountaineering team member is killed. Therefore, there is a possibility of the death of mountaineers in the disaster, and none of the above phases can save the mountaineer. This phase in the MTBO algorithm is considered in such a way that that member is removed from the group, and a new member is randomly replaced using the following equation:

$$X_i^{\text{new}} = X(X_{\max} - X_{min}())_{min} \tag{5}$$

Finally, the overall pseudo-code of the optimization process of the proposed MTBO algorithm is depicted in Algorithm 1. Additionally, the MTBO algorithm flowchart in the optimization process is illustrated in Figure 3.

Algorithm 1: The MTBO Algorithm. https://github.com/Irajfaraji/MTBO (accessed on 6 January 2023)

to set values of the control parameters of MTBO algorithm: the scaling factors *Li*, *Ai*, and *Mi*, iterations maximum number $Iter_{max}$, and the population size *NP* and setting the iterations number *Iter* = 0 for individuals;
to generate the initial random population *NP* (*i* = 1, 2, . . . , *NP*);
$X_i = X(X_{\max} - X_{min}())_{min}$
to evaluate the fitness of each individual;
while The *i* **till** maximum no of iterations $Iter_{max}$ **do**
to set the iterations number *Iter* = *Iter* + 1;
for *i* = 1 to *NP* **do**
to choose the numbers X_{Leader}, X_{ii}, $X_{\text{Avalanche}}$;
if $rand < Li$
$X_i^{\text{new}} = X_i + rand \times (X_{Leader} - X_i) + rand \times (X_{ii} - X_i)$
else if $rand < Ai$
$X_i^{\text{new}} = X_i - rand \times (X_{\text{Avalanche}} - X_i)$
else if $rand < Mi$;
$X_i^{\text{new}} = X_i + rand \times (X_{\text{Team}} - X_i)$
else
$X_i^{\text{new}} = X(X_{\max} - X_{min}())_{min}$;
end if
if $f(X_i^{\text{new}}) < f(X_i)$
$X_i = X_i^{\text{new}}$ and $f(X_i) = f(X_i^{\text{new}})$;
end if
if $f(X_i) < f(X_{Leader})(or f(X_{Best}))$
$X_{Leader}(or X_{Best}) = X_i$ and $f(X_{Leader})(or f(X_{Best})) = f(X_i)$;
end if
end for
end while
Return the best solution has been achieved by MTBO algorithm: X_{Leader} or X_{Best}

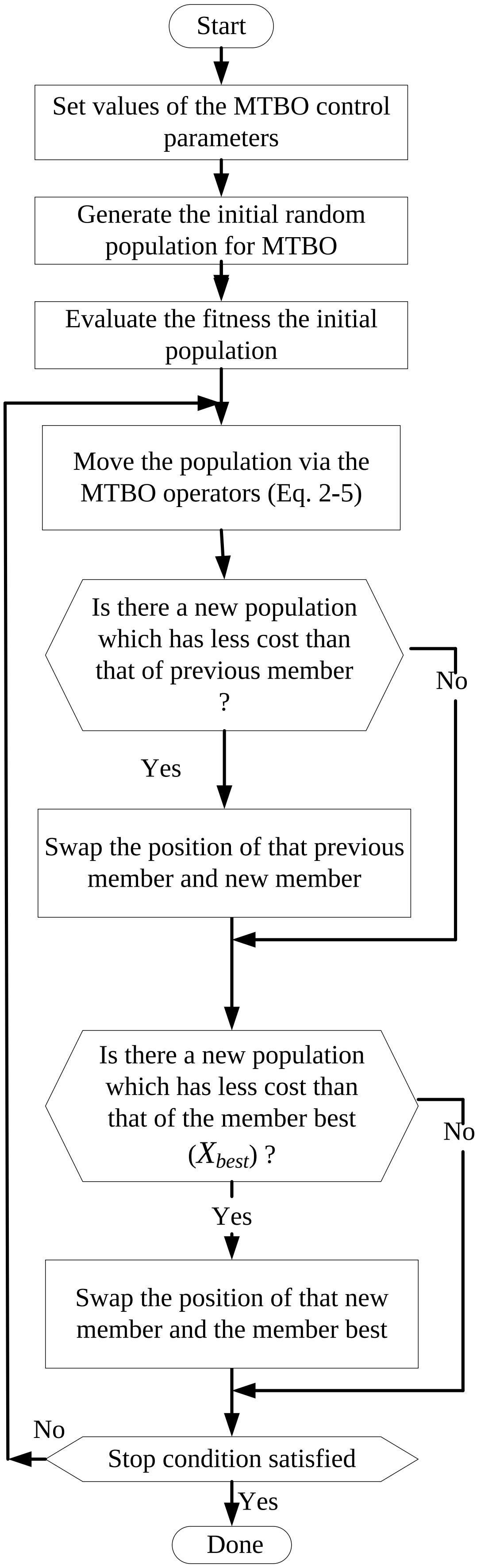


Figure 3. Flowchart of the proposed MTBO algorithm.

2.3. Computational Complexity of the MTBO

Note that the computational complexity of the MTBO mainly depends on three processes: initialization, fitness evaluation, and updating of the population. Note that with NP individuals, the computational complexity of the initialization process is O(NP). The computational complexity of the updating mechanism is O(Itermax × NP) + O(Itermax × NP × D), which is composed of searching for the best location and updating the location vector of all populations, where Itermax is the maximum number of iterations and D is the dimension of the specific problems. Therefore, the computational complexity of the MTBO is defined by:

$$\mathrm{O(MTBO)} = \mathrm{O}(\mathrm{NP} \times (\mathrm{Iter}_{\max} + \mathrm{Iter}_{\max} \times \mathrm{D} + 1)) \tag{6}$$

3. Results and Discussion

3.1. Understanding MTBO Performance

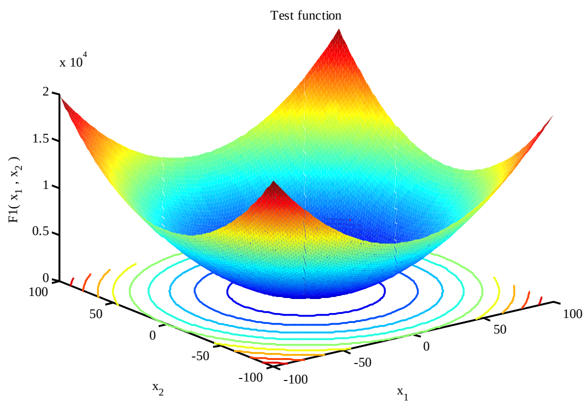
First, to describe the performance of the MTBO algorithm based on different populations and identify the best values of factors Li, Ai, and Mi for the MTBO optimization performance, three classic and diverse optimization functions [67] are considered, according to Table 1. In addition, a three-dimensional specification of these functions is provided in Figure 4.

Table 1. Summary of the selected test functions with $f_{\min} = 0$.

Test Function	Search Range
$f_1 = \sum_{i-1}^{D} x_i^2$	[−100, 100]
$f_2 = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	[−2.048, 2.048]
$f_3 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[−5.12, 5.12]

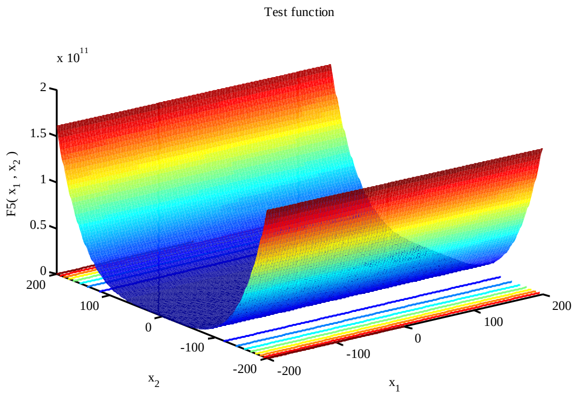
3.1.1. Investigation of MTBO Population Changes

This section uses different populations from 15 to 90 for the MTBO algorithm to solve three test functions with a dimension of 30 and several iterations of 1000. The mean value and standard deviation (Std.) for 30 independent executions for each test function are given in Table 2. It can be observed that the population between 60 and 75 is a suitable choice for this algorithm for dimension 30. Moreover, the convergence characteristics of the MTBO with different populations from 15 to 90 for the numerical results in Table 2 are shown in Figure 5. The symbol R indicates the rank of the obtained result in the total results.

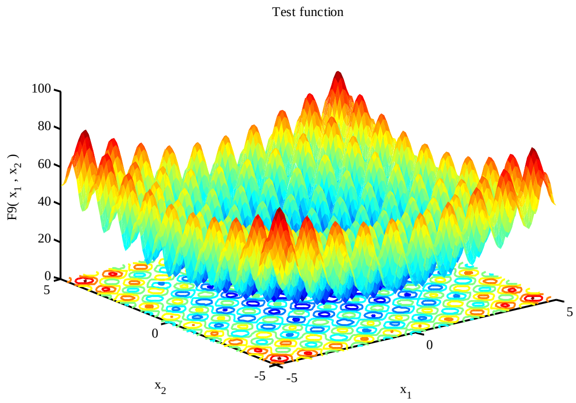


(a)

Figure 4. *Cont.*



(b)



(c)

Figure 4. Three-dimensional specification of (**a**) $F_{1,}$ (**b**) F_2, and (**c**) F_3.

Table 2. Mean value and standard deviation for 30 independent executions for each test function.

Population	F_1			F_2			F_3		
	Mean	**Std.**	R	**Mean**	**Std.**	R	**Mean**	**Std.**	R
15	0.5449	0.8326	6	32.9596	8.1957	6	49.9483	9.9646	6
30	9.81×10^{-7}	2.62×10^{-6}	5	23.8511	2.3849	5	33.6374	17.2550	5
45	1.67×10^{-11}	2.19×10^{-11}	4	22.8453	2.0032	4	25.9691	11.4754	4
60	1.33×10^{-15}	3.95×10^{-15}	3	21.9909	0.6714	3	19.2027	5.3488	3
75	4.76×10^{-20}	1.09×10^{-19}	2	21.6596	0.9834	1	16.9143	4.2987	1
90	1.83×10^{-23}	5.52×10^{-23}	1	21.8772	0.9459	2	18.0088	5.1476	2

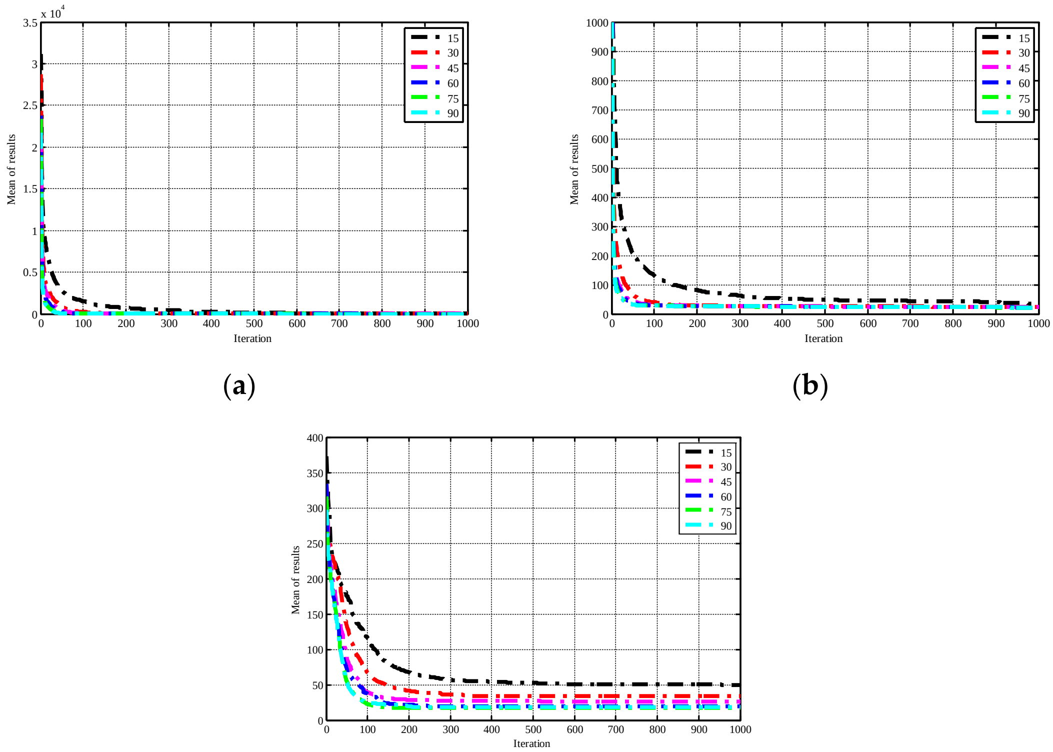
3.1.2. Determining Desirable Factors of MTBO

The performance of a metaheuristic algorithm depends on three factors: (1) the specific optimization problem at hand, (2) the values of the control parameters, and (3) the random variability inherent to stochastic algorithms. Therefore, the following aspects are taken into account.

(i) The regular and coordinated natural movement of the climbing team.

This algorithm chooses the group's most experienced member as the leader. The basis of the optimization process is the avalanche. In other words, each member is guided by the group leader and the member in front. In Table 3, various possibilities for continuing the regular movement of the population have been examined. The number of repetitions is 1000, the population of the algorithm is 60, and the dimension of the problem is 30.

According to the optimization results obtained in Table 3, it can be concluded that the suitable and desirable value for Li is equal to (0.25 + 0.25 × rand).



(**a**) (**b**) (**c**)

Figure 5. Convergence characteristics of the MTBO with different populations from 15 to 90 (**a**) F_1, (**b**) F_2, and (**c**) F_3.

Table 3. Various possibilities to continue the regular movement of the population.

Li	F_1			F_2			F_3		
	Mean	**Std.**	***R***	**Mean**	**Std.**	***R***	**Mean**	**Std.**	***R***
(0.25 + 0.25 × rand)	$\mathbf{1.33 \times 10^{-15}}$	$\mathbf{3.95 \times 10^{-15}}$	**1**	**21.9909**	**0.6714**	**1**	**19.2027**	**5.3488**	**3**
(0.5 + 0.5 × rand)	8.6845	9.4542	6	31.0271	4.4540	6	41.7866	11.7440	6
(0.25 + 0.5 × rand)	8.41×10^{-15}	1.51×10^{-14}	2	22.3998	1.2361	3	29.0528	12.9879	4
0.5 × rand	3.42×10^{-12}	8.85×10^{-12}	4	22.3536	1.1654	2	16.5163	8.3666	2
rand	1.41×10^{-14}	4.21×10^{-14}	3	23.1929	1.7868	4	29.6497	6.9378	5
0.1	1.09×10^{-7}	1.70×10^{-7}	5	23.4122	1.0828	5	14.6635	7.4747	1
0.9	389.8292	365.6368	7	67.3013	33.3700	7	57.9233	17.7395	7

(ii) Avalanche occurrence probability as a model of natural disasters.

In this algorithm, the optimization process is based on the avalanche. Therefore, the possibility of avalanche occurrence is more than in other conditions, which are analyzed in Table 4. The number of repetitions is 1000, the population of the algorithm is 60, and the dimension of the problem is 30. According to the optimization results obtained in Table 4, it can be concluded that the appropriate and desirable value for Ai is equal to (0.75 + 0.25 × rand).

Table 4. Various possibilities for the occurrence of an avalanche.

Ai	F_1			F_2			F_3		
	Mean	**Std.**	R	**Mean**	**Std.**	R	**Mean**	**Std.**	R
(0.5 + 0.5 × rand)	1.33×10^{-15}	3.95×10^{-15}	4	21.9909	0.6714	3	19.2027	5.3488	1
(0.5 + 0.25 × rand)	4.85×10^{-12}	7.99×10^{-12}	5	22.6942	0.9995	4	23.6800	12.5486	2
(0.75 + 0.25 × rand)	$\mathbf{2.28 \times 10^{-18}}$	$\mathbf{6.09 \times 10^{-18}}$	**3**	**21.7711**	**1.3563**	**2**	**26.7644**	**5.1901**	**3**
(0.9 + 0.1 × rand)	4.70×10^{-21}	1.13×10^{-20}	1	22.7972	2.6426	5	51.6382	22.0165	6
0.1	284.0440	242.0450	6	41.3510	6.7514	6	34.0024	12.2198	5
0.9	2.38×10^{-20}	6.09×10^{-20}	2	21.2388	1.1805	1	32.3361	11.3176	4

(iii) The possibility of rescuing an individual by the mountaineering team.

This algorithm establishes the probability of saving a person after the occurrence of an avalanche. Thus, the possibility of saving a person cannot be very high compared to the previous two processes, and the conditions of its occurrence are when the previous two processes do not happen to the person in question. In Table 5, different possibilities for rescue of an individual have been examined first. The number of repetitions is 1000, the population of the algorithm is 60, and the dimension of the problem is 30. According to the results obtained in Table 5, the appropriate and desirable value for Mi is equal to (0.75 + 0.25 × rand).

Table 5. Various possibilities for the rescue of an individual.

Mi	F_1			F_2			F_3		
	Mean	**Std.**	R	**Mean**	**Std.**	R	**Mean**	**Std.**	R
(0.5 + 0.5 × rand)	1.33×10^{-15}	3.95×10^{-15}	4	21.9909	0.6714	3	19.2027	5.3488	1
(0.5 + 0.25 × rand)	6.64×10^{-15}	2.10×10^{-14}	6	22.0837	0.9729	4	28.5554	9.8614	5
(0.75 + 0.25 × rand)	$\mathbf{8.66 \times 10^{-17}}$	$\mathbf{1.43 \times 10^{-16}}$	**2**	**21.9980**	**1.2873**	**2**	**26.7644**	**16.5657**	**3**
(0.9 + 0.1 × rand)	2.84×10^{-15}	6.05×10^{-15}	5	21.9107	1.1549	1	27.5603	15.6338	4
0.1	1.51×10^{-18}	3.27×10^{-18}	1	22.5299	0.5712	6	40.5943	10.7754	6
0.9	1.29×10^{-15}	3.96×10^{-15}	3	22.2802	0.8322	5	24.3765	6.1735	2

Appendix A examines the performance comparison of the MTBO based on basic test functions, while Appendix B discusses the performance comparison of the MTBO based on the CEC 2014 test functions.

4. MTBO for Real Engineering Problems

In this section, the performance of the MTBO algorithm has been evaluated with three constrained engineering design problems based on equality and inequality constraints [68], including the tension/compression spring design [69], the three-bar truss design [70,71], and pressure vessel optimization [72,73]. Due to the different constraints of these problems, the constraint management method should be used. Previous optimization studies have used different types of penalty functions, such as the static, dynamic, adaptive, co-evolutionary, and death penalty, as well as the special operator and goal separation methods. The death penalty is the simplest method with easy implementation and low computational cost, which assigns a large numerical value in the minimization of the problem, causing the algorithm to move away from impractical solutions in the optimization process. In this study, the MTBO algorithm is equipped with a death penalty to satisfy the constraints.

4.1. Tension/Compression Spring Design Problem

The aim of the studied problem is the minimization of the tension/compression spring weight. The design problem is depicted in Figure 6. The appropriate design should satisfy the constraints of the shear stress, deflection, and surge frequency. In this design, the problem has three variables, including the diameter of the wire (d), the average diameter of the coil (D), and the number of active coils (N).

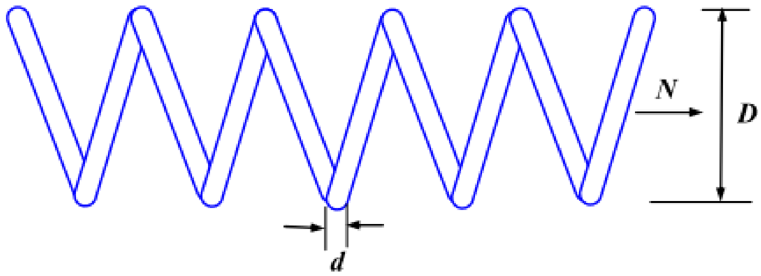


Figure 6. Problem of tension/compression spring design.

The optimization problem is presented as follows [69]:
Minimize:

$$F_1(X) = (x_3 + 2)x_2 x_1^2. \tag{7}$$

Subject to:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0,$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \le 0,$$

$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0,$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \le 0.$$

Variable range: $0.05 \le x_1 \le 2$, $0.25 \le x_2 \le 1.3$, $2 \le x_3 \le 15$.

This design problem is conducted using the MTBO algorithm, as well as the Rao-1, BA, PSO, and WOA algorithms for comparative purposes. The results, including the decision variables, constraints, and function values, are given in Tables 6 and 7, which indicate that the MTBO algorithm has obtained better results than that of the other algorithms. Also, the convergence process for the tension/compression spring problem using different algorithms is demonstrated in Figure 7, which shows that the MTBO algorithm achieved lower mean and best values of the objective function.

4.2. Three-Bar Truss Design Problem

This section presents the three-bar truss design problem to minimize its weight. The objective function is bounded, and structural design problems have many constraints. The constraints include stress, deflection, and buckling. Figure 8 shows the three-bar truss design problem.

This design problem is defined as follows [70,71]:
Minimize:

$$F_2(X) = 100 \times \left(2\sqrt{2}x_1 + x_2\right). \tag{8}$$

Subject to:

$$g_1(X) = P \times \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} - \sigma \le 0,$$

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

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

Table 6. Tension/compression spring optimal design problem.

Variable	MTBO
x_1	**0.05173**
x_2	**0.35771**
x_3	**11.2312**
g_1	$\mathbf{-1.02 \times 10^{-5}}$
g_2	$\mathbf{-3.49 \times 10^{-6}}$
g_3	$\mathbf{-4.06}$
g_4	$\mathbf{-0.73}$
Best	**0.012665**
Mean	**0.012684**
Worst	**0.012702**
Std.	$\mathbf{5.60 \times 10^{-05}}$

Table 7. Statistical results for the tension/compression spring problem by the studied algorithms.

Variable	Best	Mean	Worst	Std.	*p*-Values
MTBO	**0.012665**	**0.012684**	**0.012702**	$\mathbf{5.60 \times 10^{-5}}$	–
Rao-1	0.012666	0.012725	0.012875	7.94×10^{-5}	**0.020840**
BA	0.012666	0.013495	0.016673	9.18×10^{-3}	**0.009918**
PSO	0.012675	0.012728	0.012899	4.26×10^{-4}	**0.009235**
WOA	0.012672	0.012711	0.012946	1.84×10^{-3}	**0.008167**

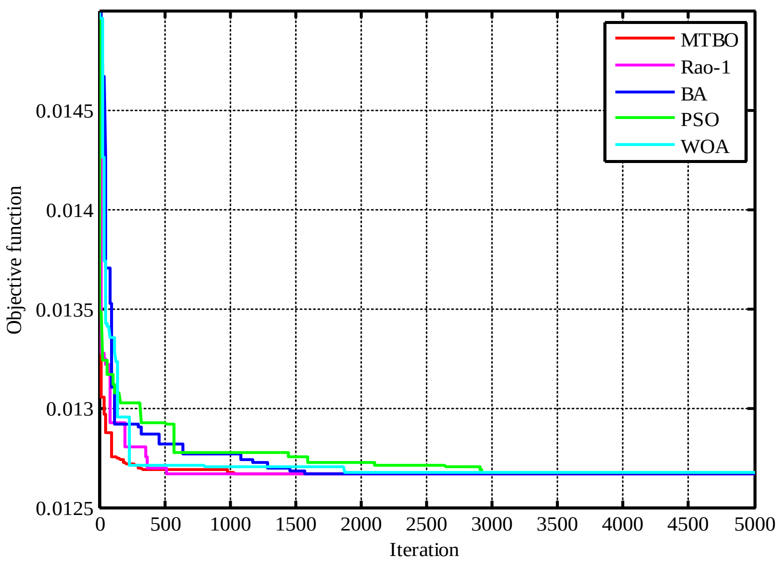


Figure 7. Convergence process for tension/compression spring problem using different algorithms.

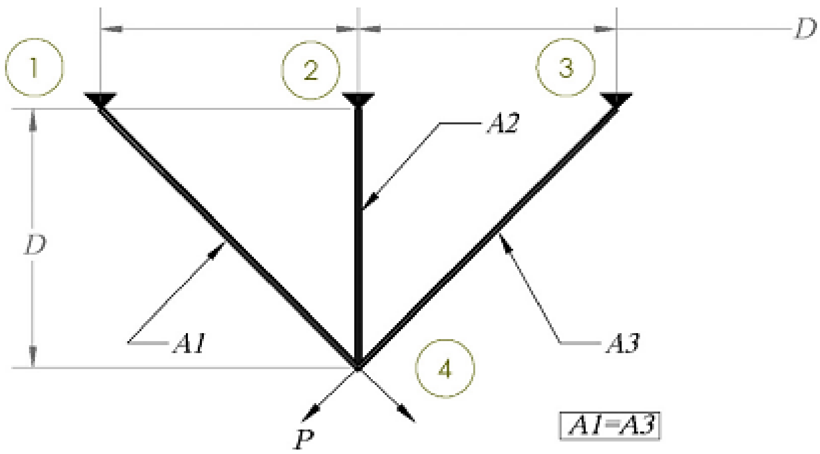


Figure 8. Problem of three-bar truss design. The numbers 1 to 3 are the number of elements of truss.

Variable range: $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$. This problem is performed via the MTBO, as well as the Rao-1, BA, PSO, and WOA algorithms. The obtained optimal decision variables, constraints, and function values are presented in Tables 8 and 9, which clearly indicates the superiority of the MTBO algorithm in obtaining better results compared to that of the Rao-1, BA, PSO, and WOA algorithms. Also, the convergence curve of different algorithms in the design problem is depicted in Figure 9, which shows that the MTBO algorithm obtained lower mean and best values.

Table 8. Three-bar truss structure optimal design problem using MTBO.

Variable	MTBO
x_1	0.78868
x_2	0.40825
$g_1(X)$	−2.52
$g_2(X)$	−1.4639
$g_3(X)$	−0.5360
Best	263.8958434
Mean	263.895844
Worst	263.8958442
Std.	7.19×10^{-7}

Table 9. Statistical results for the three-bar truss structure optimal design problem by the studied algorithms.

Variable	Best	Mean	Worst	Std.	*p*-Values
MTBO	**263.8958434**	**263.895844**	**263.8958444**	$\mathbf{7.19 \times 10^{-7}}$	–
Rao-1	263.8958441	263.897012	263.897528	2.58×10^{-3}	0.00481
BA	263.8958449	263.910134	263.931824	8.29×10^{-3}	2.8772×10^{-4}
PSO	263.8958608	263.898057	263.941265	1.67×10^{-2}	9.3245×10^{-5}
WOA	263.8958505	263.897963	263.928429	6.43×10^{-3}	8.4932×10^{-5}

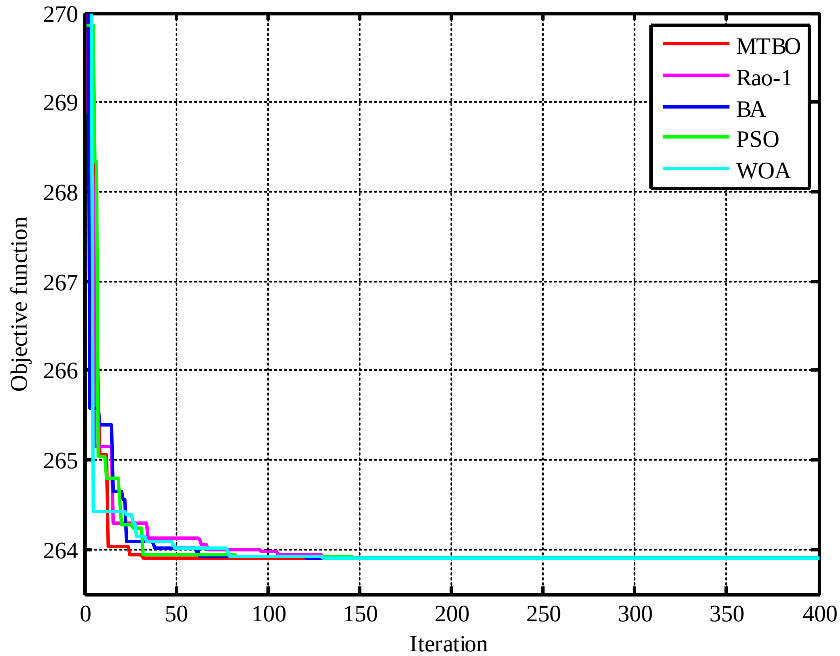


Figure 9. Convergence graphs for the three-bar truss design problem using different algorithms.

4.3. Pressure Vessel Optimization Problem

In pressure vessel optimization, the objective is to minimize the total cost, including the materials, shaping, and welding of the cylindrical pressure vessel, as shown in Figure 10. Decision variables include shell thickness (Ts), head thickness (Th), inner radius (R), and length of cylindrical section excluding head (L).

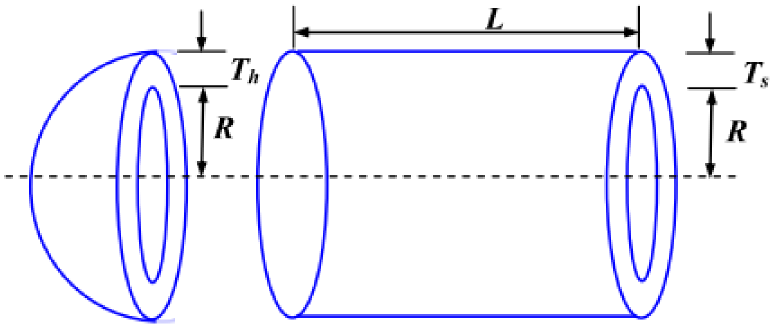


Figure 10. Problem of the pressure vessel optimization.

The pressure vessel optimization problem is presented as follows [72,73]:
Minimize:

$$F_3(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3. \tag{9}$$

Subject to:

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

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,$$

$$g_4(X) = x_4 - 240 \leq 0,$$

Variable range: $0 \leq x_1 \leq 100$, $0 \leq x_2 \leq 100$, $10 \leq x_3 \leq 200$, $10 \leq x_4 \leq 200$.

This problem of pressure vessel optimization is implemented via the MTBO, as well as the Rao-1, BA, PSO, and WOA algorithms. The decision variables, constraints, and function values are given in Tables 10 and 11, which prove the superiority of the MTBO in achieving superior results compared to that of the Rao-1, BA, PSO, and WOA algorithms. Moreover, the convergence process of the Rao-1, BA, PSO, and WOA algorithms in the design problem is demonstrated in Figure 11, which shows that the MTBO obtained lower mean and best values.

Table 10. Pressure vessel optimal design problem using MTBO.

Variable	MTBO
x_1	0.8125
x_2	0.4375
x_3	42.09845
x_4	1.76637
$g_1(X)$	0.0
$g_2(X)$	-0.036
$g_3(X)$	-3.5×10^{-10}
$g_4(X)$	-63.40
Best	6059.714335
Mean	6168.7825
Worst	6304.2583
Std.	95.37

Table 11. Statistical results for pressure vessel optimal design problem by the studied algorithms.

Variable	Best	Mean	Worst	Std.	*p*-Values
MTBO	6059.714335	6168.7825	6304.2583	95.37	–
Rao-1	6059.714335	6182.7054	6391.1278	242.93	0.020568
BA	6059.714335	6195.1006	6325.3192	308.64	0.0164401
PSO	6061.592462	7982.6379	9296.1815	693.51	0.0065754
WOA	6059.715963	6314.8562	7142.5356	500.78	0.0081667

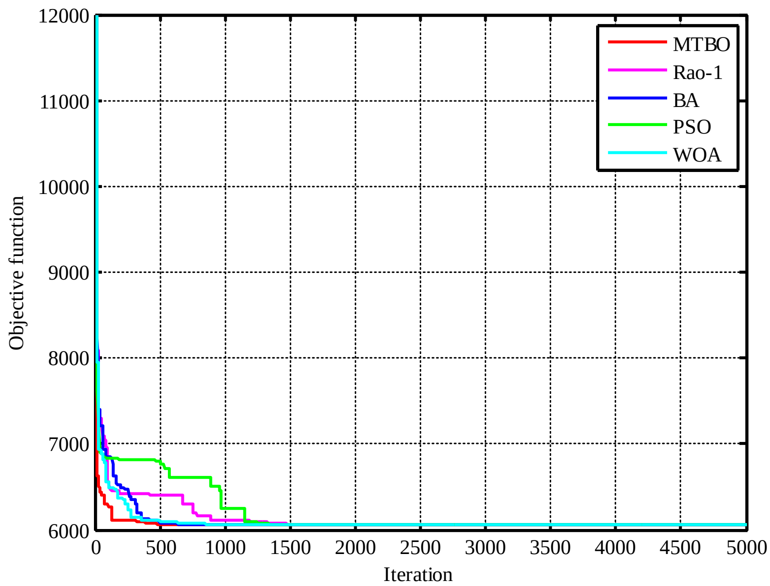


Figure 11. Convergence graphs for pressure vessel optimization problem using different algorithms.

5. Conclusions

This paper has established a novel optimization algorithm named the mountaineering team-based optimization (MTBO) algorithm based on intellectual and environmental evolution with coordinated human behavior. The proposed algorithm is formulated based on the four phases of coordinated mountaineering, the effect of natural disasters, coordinated and group effort against disasters, and the possible death of the members due to avalanches. The capability of the MTBO algorithm is investigated with different populations to identify the best values of factors considering classic functions. The performance of the MTBO is further evaluated on 23 basic functions based on unimodal, multimodal, and fixed multimodal benchmark test functions. Statistical analysis and Wilcoxon test results proved the superior and competitive performance of the MTBO algorithm in comparison with the genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), artificial bee colony (ABC), and simulated annealing (SA) algorithms (see Appendix A). Moreover, the MTBO algorithm's effectiveness has been investigated in solving the CEC 2014 test functions based on unimodal functions, simple multimodal, hybrid, and composition, which has provided very competitive results compared to the well-known Rao-1, BA, PSO, and WOA algorithms (see Appendix B). Furthermore, to evaluate the MTBO algorithm's performance, three engineering problems, including tension/compression spring design, three-bar truss design, and pressure vessel optimization, were solved, proving the MTBO is very competitive compared to the Rao-1, BA, PSO, and WOA algorithms, and proving its better performance. Hybridization of the MTBO algorithm with the well-known evolutionary algorithms is suggested for future work.

Author Contributions: Conceptualization, I.F.D. and A.P.; methodology, I.F.D.; software, A.P.; validation, I.F.D.; formal analysis, I.F.; investigation, I.F.D.; resources, I.F.D.; data curation, A.P.; writing—original draft preparation, I.F.; writing—review and editing, I.F. and I.F.D.; visualization, A.P.; supervision, M.L.N.; project administration, M.L.N.; funding acquisition, M.L.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not acceptable.

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

Appendix A. Performance Comparison of MTBO Based on Basic Test Functions

In the first part of the comparative study, to understand the proposed algorithm's power, its performance has been compared with five standard algorithms based on 23 basic functions, based on unimodal, multimodal, and fixed multimodal benchmark test functions [74–79], according to Table A1.

The performance of the MTBO algorithm is compared with the GA, DE, PSO, ABC, and SA algorithms. The control parameters of these algorithms have been selected and determined based on their reference article, according to Table A2.

Moreover, the code that the author has provided to the readers on selected sites, and no changes have been made except that for all algorithms, the number of iterations selected is 1000, and the number of the population considered is 60. Also, for this comparison, the dimension designated is 30. Visualization of some basic benchmark functions in 2D is depicted in Figure A1.

Moreover, the optimization performance of the MTBO for these benchmark functions is depicted in Figure A2.

The numerical results based on the mean value, the best value, the standard deviation, and the Wilcoxon test results are given in Tables A3–A7.

Table A1. The 23 essential functions based on unimodal, multimodal, and fixed multimodal benchmark test functions [79].

Unimodal Benchmark Functions			
Function	**Dim**	**Range**	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]$	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	$[-100, 100]$	0
$f_4(x) = \max_i \{\lvert x_1 \rvert\} \quad , 1 \le i \le n$	30	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	30	$[-30, 30]$	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	$[-100, 100]$	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + rand[0, 1)$	30	$[-1.28, 1.28]$	0
Multimodal Benchmark Functions			
Function	**Dim**	**Range**	$f_{\min}$
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert} \right)$	30	$[-500, 500]$	-418.9829×5
$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_1) + 10]$	30	$[-5.12, 5.12]$	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	30	$[-32, 32]$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right)$	30	$[-600, 600]$	0

Table A1. *Cont.*

Unimodal Benchmark Functions			
Function	**Dim**	**Range**	$f_{\min}$
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1}) + (y_n-1)^2\right]\right\} + \sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i = 1 + \frac{x_i+1}{4}$ $u(x_1,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	30	$[-50, 50]$	0
$f_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right] + (x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i,5,100,4)$	30	$[-50, 50]$	0
$f_{14}(x) = -\sum_{i=1}^{n}\sin(x_i)\cdot\left(\sin\left(\frac{i\cdot x_i^2}{\pi}\right)\right)^{2m} \quad ,m=5$	30	$[0, \pi]$	-4.687
$f_{15}(x) = \left[e^{-\sum_{i=1}^{n}\left(\frac{x_i}{\beta}\right)^{2m}} - 2e^{-\sum_{i=1}^{n}x_i^2}\right]\cdot\prod_{i=1}^{n}\cos^2 x_i \quad ,m=5$	30	$[-20, 20]$	-1
$f_{16}(x) = \left\{\left[\sum_{i=1}^{n}\sin^2(x_i)\right] - \exp\left(-\sum_{i=1}^{n}x_i^2\right)\right\}.\exp\left[-\sum_{i=1}^{n}\sin^2\sqrt{\lvert x_i\rvert}\right]$	30	$[-10, 10]$	-1
Fixed-Dimension Multimodal Benchmark Functions			
Function	**Dim**	**Range**	$f_{\min}$
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	$[-65, 65]$	1
$f_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i^2+b_i x_2\right)}{b_i^2+b_i x_3+x_4}\right]$	4	$[-5, 5]$	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1-\frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398

Table A1. *Cont.*

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

Table A2. The control parameters of different algorithms.

Algorithm	Parameter	Value
Genetic algorithm (GA) [74]	Crossover factor	0.7
	Mutation factor	0.3
(DE) [75]	Crossover probability	0.1
	Scaling factor	0.9
Particle swarm optimization (PSO) [76,77]	Constriction factor χ	0.729
	Acceleration control coefficient c_1	2.05
	Acceleration control coefficient c_2	2.05
Artificial bee colony (ABC) [68]	Onlooker number n_o	50% of the colony
	Employed bee number n_e	50% of the colony
	Scout number n_s	1
	Limit	$n_s \times D$ (dimension of the problem)
Simulated annealing (SA) [78]	Cooling rate α	0.8
	Initial temperature T_0	1

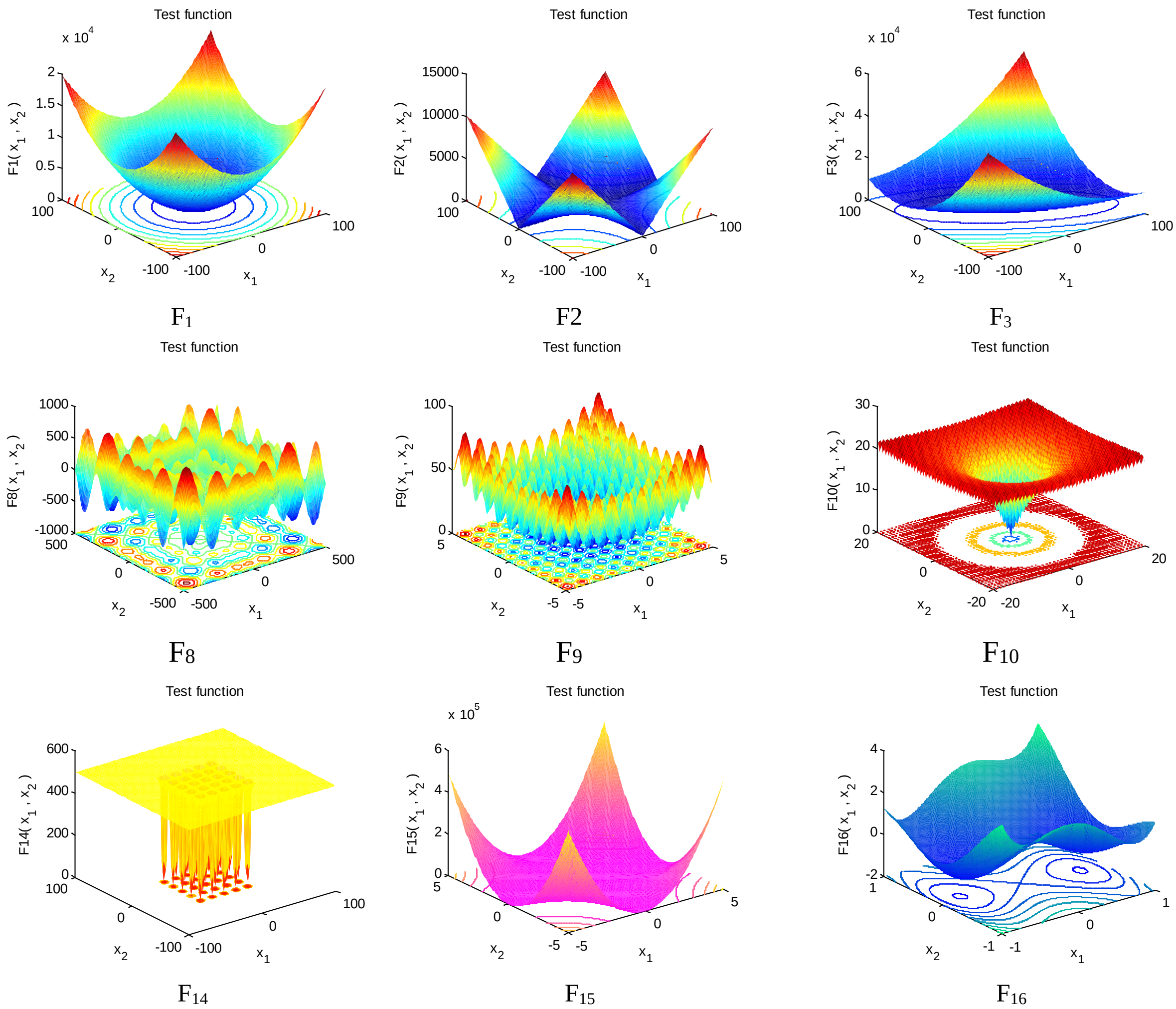
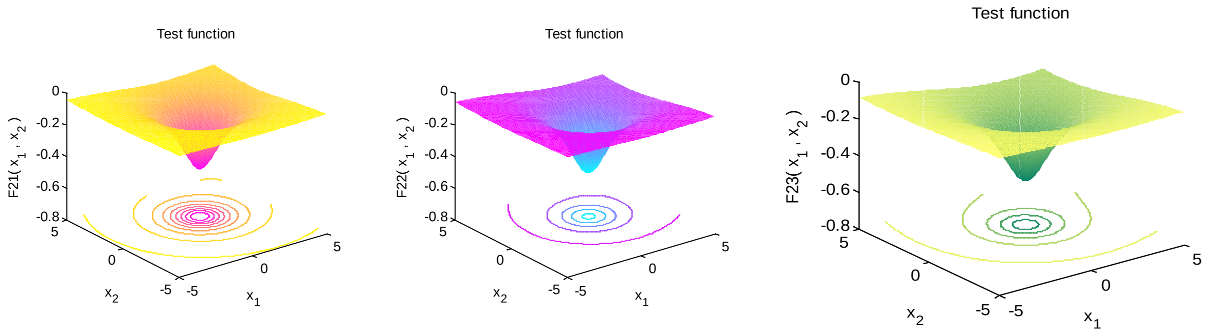
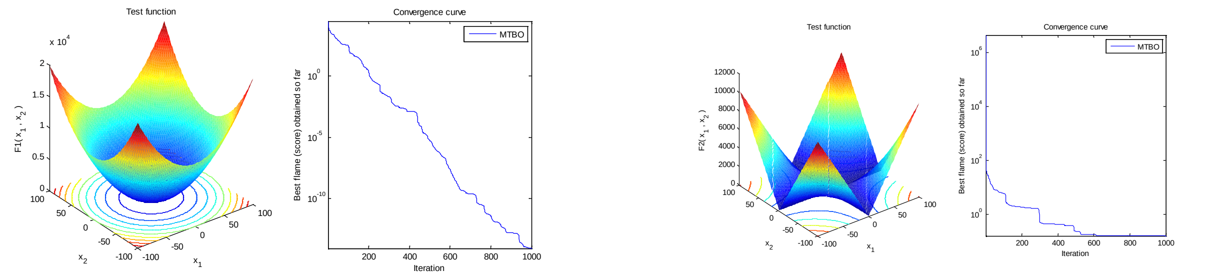


Figure A1. *Cont.*

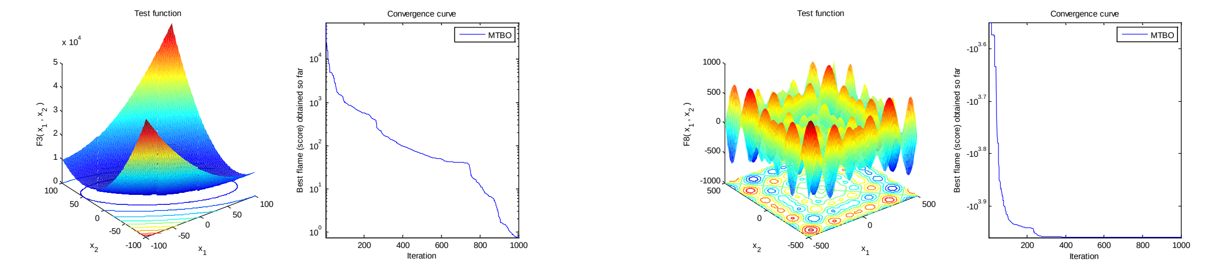


F_{21} F_{22} F_{23}

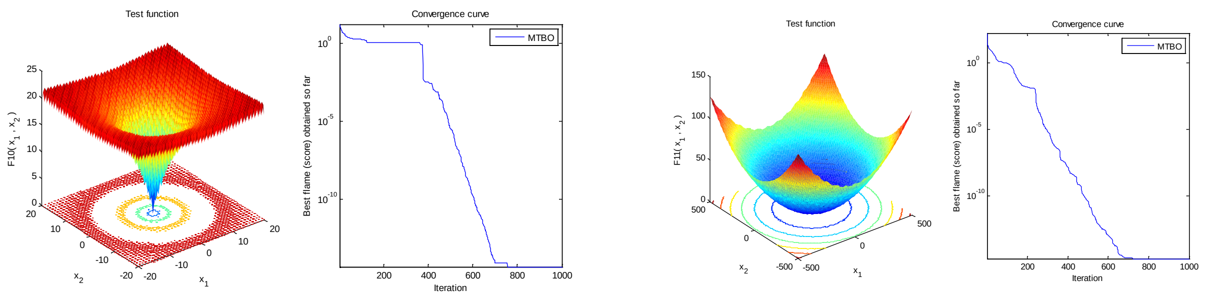
Figure A1. Visualization of some basic benchmark functions in 2D.



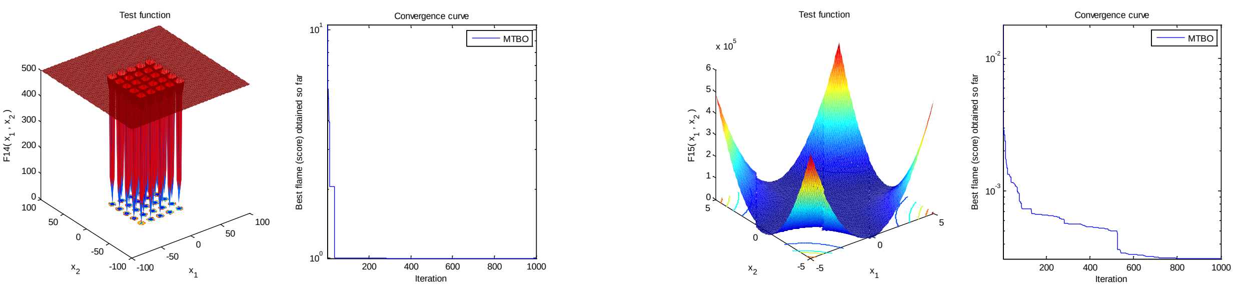
F_1 F_2



F_3 F_8



F_{10} F_{11}



F_{14} F_{15}

Figure A2. *Cont.*

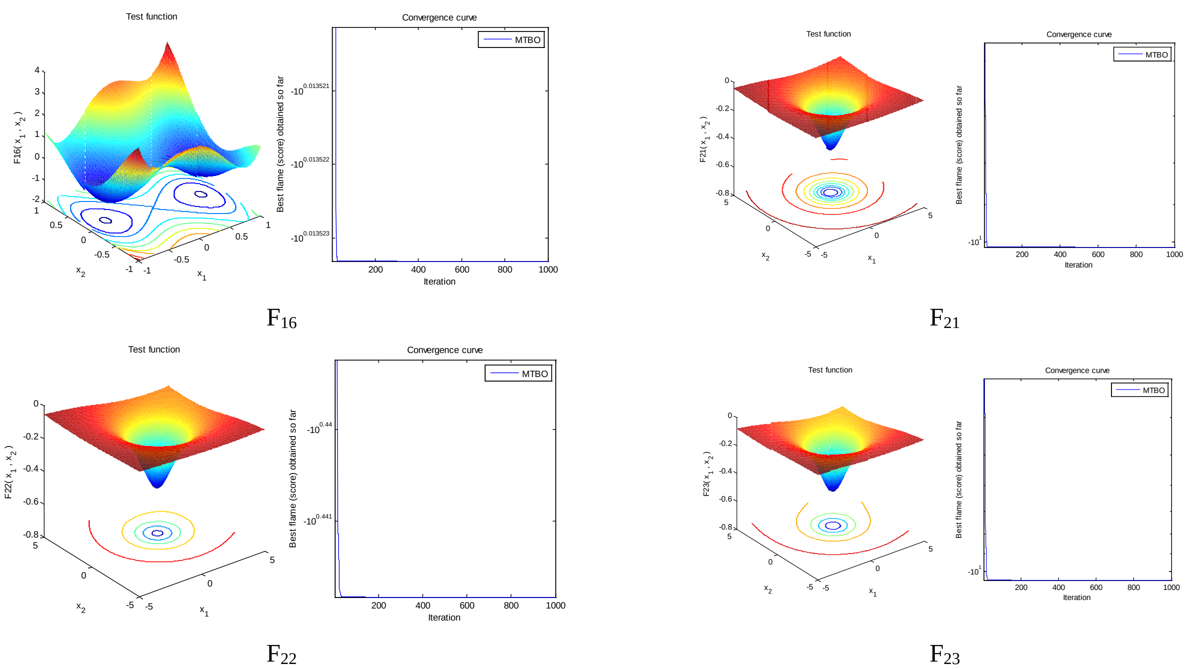


Figure A2. Optimization performance of the MTBO for some basic benchmark functions.

Table A3. Summary of the Mean results for the test functions for the classic and MTBO algorithms.

Function	GA	DE	PSO	ABC	SA	MTBO
F1	3.53×10^{-2} -	1.26×10^{-11} -	2.96×10^{-8} -	5.14×10^{-14} -	1.59×10^{-7} -	$\mathbf{7.70 \times 10^{-19}}$
F2	1.39 -	$\mathbf{7.07 \times 10^{-2}}$ +	5.59×10^{-1} -	2.38×10^{-1} -	8.95×10^{-1} -	1.72×10^{-1}
F3	$4.50 \times 10^{+2}$ -	$3.77 \times 10^{+1}$ -	$1.67 \times 10^{+2}$ -	$3.64 \times 10^{+1}$ -	$6.21 \times 10^{+1}$ -	$\mathbf{3.55 \times 10^{+1}}$
F4	$1.42 \times 10^{+1}$ -	6.19 -	9.61 -	4.42 -	6.56 -	**2.36**
F5	$9.56 \times 10^{+1}$ -	$3.80 \times 10^{+1}$ -	$3.44 \times 10^{+1}$ -	$3.07 \times 10^{+1}$ -	$5.49 \times 10^{+1}$ -	$\mathbf{2.73 \times 10^{+1}}$
F6	6.38×10^{-3} -	6.67×10^{-13} -	4.74×10^{-7} -	2.55×10^{-15} -	2.99×10^{-5} -	$\mathbf{2.71 \times 10^{-17}}$
F7	1.15×10^{-1} -	3.06×10^{-2} -	4.75×10^{-2} -	2.45×10^{-2} -	3.89×10^{-2} -	$\mathbf{1.69 \times 10^{-2}}$
F8	$-7.57 \times 10^{+3}$ -	$-8.10 \times 10^{+3}$ +	$\mathbf{-8.24 \times 10^{+3}}$ **+**	$-7.96 \times 10^{+3}$ -	$-7.51 \times 10^{+3}$ -	$-8.09 \times 10^{+3}$
F9	$4.87 \times 10^{+1}$ -	$3.21 \times 10^{+1}$ -	$3.35 \times 10^{+1}$ -	$3.30 \times 10^{+1}$ -	$3.44 \times 10^{+1}$ -	$\mathbf{2.97 \times 10^{+1}}$
F10	2.94 -	1.92 -	2.40 -	2.29 -	2.57 -	**1.34**
F11	5.80×10^{-1} -	8.32×10^{-2} -	1.72×10^{-1} -	5.02×10^{-2} -	3.97×10^{-2} -	$\mathbf{3.73 \times 10^{-2}}$
F12	4.58 -	9.51×10^{-1} -	1.96 -	1.37 -	1.21 -	$\mathbf{2.08 \times 10^{-1}}$
F13	$2.09 \times 10^{+1}$ -	$1.03 \times 10^{+1}$ -	$1.57 \times 10^{+1}$ -	$1.21 \times 10^{+1}$ -	$1.34 \times 10^{+1}$ -	**2.59**

Table A3. *Cont.*

Function	GA	DE	PSO	ABC	SA	MTBO
F14	**9.98×10^{-1}** **=**	**9.98×10^{-1}** **=**	**9.98×10^{-1}** **=**	**9.98×10^{-1}** **=**	**9.98×10^{-1}** **=**	**9.98×10^{-1}**
F15	2.55×10^{-3} -	2.66×10^{-3} -	2.50×10^{-3} -	**5.39×10^{-4}** **+**	7.85×10^{-4} -	6.73×10^{-4}
F16	**−1.03** **=**	**−1.03** **=**	**−1.03** **=**	**−1.03** **=**	**−1.03** **=**	**−1.03**
F17	**3.98×10^{-1}** **=**	**3.98×10^{-1}** **=**	**3.98×10^{-1}** **=**	**3.98×10^{-1}** **=**	**3.98×10^{-1}** **=**	**3.98×10^{-1}**
F18	**3.00** **=**	**3.00** **=**	**3.00** **=**	**3.00** **=**	**3.00** **=**	**3.00**
F19	**−3.86** **=**	**−3.86** **=**	**−3.86** **=**	**−3.86** **=**	**−3.86** **=**	**−3.86**
F20	**−3.29** **=**	−3.28 -	−3.27 -	−3.24 -	−3.27 -	**−3.29**
F21	−5.78 -	−6.28 -	−6.53 -	−7.03 -	−6.86 -	**−8.65**
F22	−6.68 -	−7.68 -	−5.53 -	−8.25 -	−7.82 -	**−8.80**
F23	−6.37 -	−8.76 -	**−9.48** **=**	−8.67 -	−7.75 -	**−9.48**
Nm	6	6	7	6	5	**20**
Final rank	3	3	2	3	6	**1**

Nm represents the number of times with a ranking higher than the mean value obtained.

The Wilcoxon test based on Refs. [80–84] is performed for different algorithms, and the proposed algorithm decisively won over all the algorithms and had a more effective performance.

Table A4. Wilcoxon's test rank summary of the statistical assessment results for the classic and MTBO algorithms.

Function	GA	DE	PSO	ABC	SA	MTBO
F1	6	3	4	2	5	**1**
F2	6	**1**	4	3	5	2
F3	6	3	5	2	4	**1**
F4	6	3	5	2	4	**1**
F5	6	4	3	2	5	**1**
F6	6	3	4	2	5	**1**
F7	6	3	5	2	4	**1**
F8	5	2	**1**	4	6	3
F9	6	2	4	3	5	**1**
F10	6	2	4	3	5	**1**
F11	6	4	5	3	2	**1**
F12	6	2	5	4	3	**1**
F13	6	2	5	3	4	**1**
F14	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**
F15	5	6	4	**1**	3	2

Table A4. *Cont.*

Function	GA	DE	PSO	ABC	SA	MTBO
F16	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**
F17	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**
F18	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**
F19	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**
F20	**1.5**	3	4.5	6	4.5	**1.5**
F21	6	5	4	2	3	**1**
F22	5	4	6	2	3	**1**
F23	6	3	**1.5**	4	5	**1.5**
Total	118	72.50	91.50	67.50	93	**40.50**
Rank mean	5.1304	3.1522	3.9783	2.9348	4.0435	**1.7609**
Final rank	6	3	4	2	5	**1**

Table A5. The competitive results of the Wilcoxon's test.

Corresponding Algorithm	MTBO versus			
	p-Values	Better	Worst	Equal
SA	1.9644×10^{-4}	18	0	5
ABC	3.2701×10^{-4}	17	1	5
PSO	0.0049	16	1	6
DE	0.0074	16	2	5
GA	2.9305×10^{-4}	17	0	6

Table A6. Summary of the best results for the test functions for the classic and MTBO algorithms.

Function	GA	DE	PSO	ABC	SA	MTBO
F1	0.00	0.00	0.00	0.00	0.00	0.00
F2	0.00	0.00	0.00	0.00	0.00	0.00
F3	18.70	2.57	5.96	2.27	5.42	0.32
F4	9.26	2.73	4.90	2.16	4.07	0.27
F5	22.90	9.35	4.81	11.50	7.06	4.03
F6	0.00	0.00	0.00	0.00	0.00	0.00
F7	0.04	0.01	0.01	0.01	0.01	0.01
F8	$-9.23 \times 10^{+3}$	$-9.16 \times 10^{+3}$	$-9.14 \times 10^{+3}$	$-8.46 \times 10^{+3}$	$-9.23 \times 10^{+3}$	$-9.06 \times 10^{+3}$
F9	21.90	18.90	14.90	16.00	16.20	15.90
F10	0.00	0.00	0.00	0.00	0.00	0.00
F11	0.00	0.00	0.00	0.00	0.00	0.00
F12	0.31	0.00	0.00	0.00	0.00	0.00
F13	9.15	0.01	0.56	0.00	0.02	0.00
F14	1.00	1.00	1.00	1.00	1.00	1.00
F15	0.00	0.00	0.00	0.00	0.00	0.00
F16	−1.03	−1.03	−1.03	−1.03	−1.03	−1.03

Table A6. *Cont.*

Function	GA	DE	PSO	ABC	SA	MTBO
F17	0.40	0.40	0.40	0.40	0.40	0.40
F18	3.00	3.00	3.00	3.00	3.00	3.00
F19	−3.86	−3.86	−3.86	−3.86	−3.86	−3.86
F20	−3.32	−3.32	−3.32	−3.32	−3.32	−3.32
F21	$-1.02 \times 10^{+1}$	$-1.02 \times 10^{+1}$	$-1.02 \times 10^{+1}$	$-1.02 \times 10^{+1}$	$-1.02 \times 10^{+1}$	$-1.02 \times 10^{+1}$
F22	$-1.04 \times 10^{+1}$	$-1.04 \times 10^{+1}$	$-1.04 \times 10^{+1}$	$-1.04 \times 10^{+1}$	$-1.04 \times 10^{+1}$	$-1.04 \times 10^{+1}$
F23	$-1.05 \times 10^{+1}$	$-1.05 \times 10^{+1}$	$-1.05 \times 10^{+1}$	$-1.05 \times 10^{+1}$	$-1.05 \times 10^{+1}$	$-1.05 \times 10^{+1}$
Nb	11.00	12.00	13.00	12.00	10.00	18.00
Final rank	5.00	3.00	2.00	3.00	6.00	1.00

Nb represents the number of times with a ranking higher than the best value obtained.

Table A7. Summary of the Std. results for the test functions for the classic and MTBO algorithms.

Function	GA	DE	PSO	ABC	SA	MTBO
F1	0.08	0.00	0.00	0.00	0.00	0.00
F2	1.80	0.12	0.73	0.42	0.37	0.45
F3	407.00	199.00	202.00	58.10	46.40	60.30
F4	2.80	2.50	2.54	1.27	2.23	1.34
F5	79.00	31.90	42.40	40.50	44.20	23.00
F6	2.85	0.00	0.00	0.00	0.00	0.00
F7	0.07	0.01	0.02	0.01	0.02	0.01
F8	1080.00	533.00	684.00	346.00	1350.00	603.00
F9	14.90	9.02	9.46	14.60	16.00	6.52
F10	1.27	0.78	1.47	1.46	1.18	1.03
F11	0.95	0.09	0.24	0.05	0.05	0.03
F12	4.80	0.96	2.33	2.43	0.82	0.30
F13	12.20	9.28	11.30	12.50	12.30	5.10
F14	0.00	0.00	0.00	0.00	0.00	0.00
F15	0.01	0.01	0.01	0.00	0.00	0.00
F16	0.00	0.00	0.00	0.00	0.00	0.00
F17	0.00	0.00	0.00	0.00	0.00	0.00
F18	0.00	0.00	0.00	0.00	0.00	0.00
F19	0.00	0.00	0.00	0.00	0.00	0.00
F20	0.06	0.06	0.06	0.06	0.06	0.06
F21	3.70	3.66	3.75	3.61	3.17	2.74
F22	3.83	3.48	3.68	3.38	3.62	2.86
F23	3.89	3.16	2.58	3.33	3.57	2.58

Appendix B. Performance Comparison of MTBO Based on CEC 2014 Test Functions

This section evaluates the MTBO algorithm's performance in solving CEC 2014 test functions [80] (see Table A8), based on unimodal functions, simple multimodal, hybrid, and composition. The unimodal test functions have an optimal point to evaluate the algorithm's convergence and exploitation. Multimodal test functions have more than one optimum, which is an important challenge, as they have one global optimum, and the rest are local optimums. Multimodal functions are suitable for evaluating algorithms from exploring the local optimum to reaching the global optimum. Composition test functions are the combined or transferred versions of the unimodal and multimodal functions.

Table A8. CEC 2014 benchmark test functions.

Number		Functions	[Min, Max]
F1	Unimodal	Rotated high conditioned elliptic function	[−100, 100]
F2		Rotated Bent Cigar function	[−100, 100]
F3		Rotated discus function	[−100, 100]
F4	Simple Multimodal	Shifted and rotated Rosenbrock's function	[−100, 100]
F5		Shifted and rotated Ackley's function	[−100, 100]
F6		Shifted and rotated Weierstrass function	[−100, 100]
F7		Shifted and rotated Griewank's function	[−100, 100]
F8		Shifted Rastrigin's function	[−100, 100]
F9		Shifted and rotated Rastrigin's function	[−100, 100]
F10		Shifted Schwefel's function	[−100, 100]
F11		Shifted and rotated Schwefel's function	[−100, 100]
F12		Shifted and rotated Katsuura function	[−100, 100]
F13		Shifted and rotated HappyCat function	[−100, 100]
F14		Shifted and rotated HGBat function	[−100, 100]
F15		Shifted and rotated expanded Griewank's plus Rosenbrock's function	[−100, 100]
F16		Shifted and rotated expanded Scaffer's F6 function	[−100, 100]
F17	Hybrid	Hybrid function 1 (N = 3)	[−100, 100]
F18		Hybrid function 2 (N = 3)	[−100, 100]
F19		Hybrid function 3 (N = 4)	[−100, 100]
F20		Hybrid function 4 (N = 4)	[−100, 100]
F21		Hybrid function 5 (N = 5)	[−100, 100]
F22		Hybrid function 6 (N = 5)	[−100, 100]
F23	Composition	Composition function 1 (N = 5)	[−100, 100]
F24		Composition function 2 (N = 3)	[−100, 100]
F25		Composition function 3 (N = 3)	[−100, 100]
F26		Composition function 4 (N = 5)	[−100, 100]
F27		Composition function 5 (N = 5)	[−100, 100]
F28		Composition function 6 (N = 5)	[−100, 100]
F29		Composition function 7 (N = 3)	[−100, 100]
F30		Composition function 8 (N = 3)	[−100, 100]

To solve the CEC 2014 test functions, the effectiveness of the MTBO algorithm is compared with the Rao-1 [81], BA [82,83], PSO [75,76], and WOA [84] algorithms. The control parameters of the mentioned algorithms have been considered based on their reference article, according to Table A9. The number of iterations is selected as 5000, the population considered is 60, and the dimension is selected at 30.

The numerical results based on the mean value are given in Table A10. The results show the MTBO method's superiority and competitiveness compared to other algorithms.

Table A9. The control parameters of different algorithms for solving the CEC 2014 test functions.

Algorithm	Parameter	Value
Rao-1 [81]	Without any control parameter	-
Bat algorithm (BA) [82,83]	loudness A	0.25
	pulse rate r	0.5
	Scaling factor ε	0.1
	the minimum frequency f_{min}	0.7
	the maximum frequency f_{max}	0.9
Particle swarm optimization (PSO) [75,76]	Constriction factor χ	0.729
	Acceleration control coefficient c_1	2.05
	Acceleration control coefficient c_2	2.05
Whale optimization algorithm (WOA) [84]	Scaling factor a	[0, 2]
	Scaling factor b	1
	Scaling factor l	$[-1, 1]$

Table A10. Summary of the mean results for CEC 2014 test functions for different algorithms with D = 30.

Function		Rao-1	BA	PSO	WOA	MTBO
		Mean	Mean	Mean	Mean	Mean
F1	Unimodal	$1.77 \times 10^{+7}$ -	$3.71 \times 10^{+7}$ -	$3.02 \times 10^{+7}$ -	$2.94 \times 10^{+7}$ -	$\mathbf{3.30 \times 10^{+6}}$
F2	Unimodal	$8.05 \times 10^{+3}$ -	$1.91 \times 10^{+7}$ -	$1.23 \times 10^{+6}$ -	$4.81 \times 10^{+6}$ -	$\mathbf{1.99 \times 10^{+1}}$
F3	Unimodal	$3.04 \times 10^{+4}$ -	$5.12 \times 10^{+4}$ -	$2.87 \times 10^{+4}$ -	$3.32 \times 10^{+4}$ -	$\mathbf{1.50 \times 10^{+3}}$
F4	Simple Multimodal	$1.04 \times 10^{+2}$ -	$1.88 \times 10^{+2}$ -	$1.74 \times 10^{+2}$ -	$1.84 \times 10^{+2}$ -	$\mathbf{4.47 \times 10^{+1}}$
F5	Simple Multimodal	$2.09 \times 10^{+1}$ =	$2.11 \times 10^{+1}$ -	$2.09 \times 10^{+1}$ =	$\mathbf{2.04 \times 10^{+1}}$ **+**	$2.09 \times 10^{+1}$
F6	Simple Multimodal	$2.83 \times 10^{+1}$ -	$3.60 \times 10^{+1}$ -	$3.45 \times 10^{+1}$ -	$3.50 \times 10^{+1}$ -	$\mathbf{2.29 \times 10^{+1}}$
F7	Simple Multimodal	6.33×10^{-2} -	1.08 -	8.78×10^{-1} -	9.74×10^{-1} -	$\mathbf{3.14 \times 10^{-2}}$
F8	Simple Multimodal	$1.75 \times 10^{+2}$ -	$1.66 \times 10^{+2}$ -	$1.93 \times 10^{+2}$ -	$1.90 \times 10^{+2}$ -	$\mathbf{9.52 \times 10^{+1}}$
F9	Simple Multimodal	$2.16 \times 10^{+2}$ -	$2.61 \times 10^{+2}$ -	$2.18 \times 10^{+2}$ -	$2.36 \times 10^{+2}$ -	$\mathbf{1.02 \times 10^{+2}}$
F10	Simple Multimodal	$6.05 \times 10^{+3}$ -	$3.60 \times 10^{+3}$ -	$3.95 \times 10^{+3}$ -	$4.07 \times 10^{+3}$ -	$\mathbf{1.99 \times 10^{+3}}$
F11	Simple Multimodal	$6.90 \times 10^{+3}$ -	$5.67 \times 10^{+3}$ -	$\mathbf{4.78 \times 10^{+3}}$ **+**	$4.85 \times 10^{+3}$ -	$5.37 \times 10^{+3}$
F12	Simple Multimodal	2.42 -	2.73 -	2.88 -	**1.67** **+**	2.35
F13	Simple Multimodal	4.63×10^{-1} -	5.20×10^{-1} -	4.91×10^{-1} -	5.02×10^{-1} -	$\mathbf{4.31 \times 10^{-1}}$
F14	Simple Multimodal	5.44×10^{-1} -	3.90×10^{-1} -	3.43×10^{-1} -	$\mathbf{2.39 \times 10^{-1}}$ **+**	3.03×10^{-1}

Table A10. *Cont.*

Function		Rao-1	BA	PSO	WOA	MTBO
		Mean	Mean	Mean	Mean	Mean
F15		$1.79 \times 10^{+1}$ -	$6.85 \times 10^{+1}$ -	$5.71 \times 10^{+1}$ -	$9.34 \times 10^{+1}$ -	$\mathbf{1.63 \times 10^{+1}}$
F16		$1.28 \times 10^{+1}$ -	$1.29 \times 10^{+1}$ -	$1.24 \times 10^{+1}$ -	$1.27 \times 10^{+1}$ -	$\mathbf{1.13 \times 10^{+1}}$
F17	Hybrid	$1.76 \times 10^{+6}$ -	$5.77 \times 10^{+6}$ -	$2.48 \times 10^{+6}$ -	$3.58 \times 10^{+6}$ -	$\mathbf{2.06 \times 10^{+5}}$
F18	Hybrid	$4.73 \times 10^{+6}$ -	$5.81 \times 10^{+3}$ -	$4.45 \times 10^{+3}$ -	$3.51 \times 10^{+4}$ -	$\mathbf{2.18 \times 10^{+3}}$
F19	Hybrid	**8.99** **+**	$4.57 \times 10^{+1}$ -	$3.18 \times 10^{+1}$ -	$4.48 \times 10^{+1}$ -	$1.20 \times 10^{+1}$
F20	Hybrid	$5.61 \times 10^{+3}$ -	$4.15 \times 10^{+4}$ -	$2.24 \times 10^{+4}$ -	$2.22 \times 10^{+4}$ -	$\mathbf{7.36 \times 10^{+2}}$
F21	Hybrid	$5.23 \times 10^{+5}$ -	$8.17 \times 10^{+5}$ -	$9.41 \times 10^{+5}$ -	$1.05 \times 10^{+6}$ -	$\mathbf{1.18 \times 10^{+5}}$
F22	Hybrid	$4.55 \times 10^{+2}$ -	$7.21 \times 10^{+2}$ -	$7.50 \times 10^{+2}$ -	$7.23 \times 10^{+2}$ -	$\mathbf{3.82 \times 10^{+2}}$
F23	Composition	$\mathbf{3.15 \times 10^{+2}}$ **=**	$3.44 \times 10^{+2}$ -	$3.31 \times 10^{+2}$ -	$3.30 \times 10^{+2}$ -	$\mathbf{3.15 \times 10^{+2}}$
F24	Composition	$2.40 \times 10^{+2}$ -	$2.37 \times 10^{+2}$ -	$2.34 \times 10^{+2}$ -	$\mathbf{2.08 \times 10^{+2}}$ **+**	$2.31 \times 10^{+2}$
F25	Composition	$\mathbf{2.09 \times 10^{+2}}$ **+**	$2.12 \times 10^{+2}$ +	$2.25 \times 10^{+2}$ -	$2.20 \times 10^{+2}$ -	$2.13 \times 10^{+2}$
F26	Composition	$1.01 \times 10^{+2}$ -	$\mathbf{1.00 \times 10^{+2}}$ **=**	$1.01 \times 10^{+2}$ -	$\mathbf{1.00 \times 10^{+2}}$ **=**	$\mathbf{1.00 \times 10^{+2}}$
F27	Composition	$\mathbf{9.23 \times 10^{+2}}$ **-**	$1.15 \times 10^{+3}$ -	$1.08 \times 10^{+3}$ -	$9.78 \times 10^{+2}$ -	$\mathbf{8.44 \times 10^{+2}}$
F28	Composition	$\mathbf{1.51 \times 10^{+3}}$ **-**	$2.31 \times 10^{+3}$ -	$2.26 \times 10^{+3}$ -	$2.33 \times 10^{+3}$ -	$\mathbf{1.40 \times 10^{+3}}$
F29	Composition	$\mathbf{1.89 \times 10^{+6}}$ **+**	$4.02 \times 10^{+6}$ -	$5.79 \times 10^{+6}$ -	$4.88 \times 10^{+6}$ -	$3.52 \times 10^{+6}$
F30	Composition	$\mathbf{4.09 \times 10^{+3}}$ **+**	$9.09 \times 10^{+4}$ -	$6.90 \times 10^{+4}$ -	$9.76 \times 10^{+4}$ -	$7.06 \times 10^{+3}$
Nm		5	1	1	5	**21**
Final rank		2.5	4.5	4.5	2.5	**1**

Nm represents the number of times with a ranking higher than the mean value obtained.

The Wilcoxon's test is implemented based on Ref. [79], and the results are presented in Tables A11–A14. The resssults are clear that the MTBO algorithm has obtained the best results among 30 execution times decisively compared to all the other algorithms.

Table A11. Wilcoxon's test rank summary of the statistical assessment results for the CEC 2014 test functions for the different algorithms with D = 30.

Function		Rao-1	BA	PSO	WOA	MTBO
F1	Unimodal	2	5	4	3	**1**
F2	Unimodal	2	5	3	4	**1**
F3	Unimodal	3	5	2	4	**1**

Table A11. *Cont.*

Function		Rao-1	BA	PSO	WOA	MTBO
F4	Simple Multimodal	2	5	3	4	**1**
F5		3	5	3	1	3
F6		2	5	3	4	**1**
F7		2	5	3	4	**1**
F8		3	2	5	4	**1**
F9		2	5	3	4	**1**
F10		5	2	3	4	**1**
F11		5	4	**1**	2	3
F12		3	4	5	**1**	2
F13		2	5	3	4	**1**
F14		5	4	3	**1**	2
F15		2	4	3	5	**1**
F16		4	5	2	3	**1**
F17	Hybrid	2	5	3	4	**1**
F18		5	3	2	4	**1**
F19		**1**	5	3	4	2
F20		2	5	4	3	**1**
F21		2	3	4	5	**1**
F22		2	3	5	4	**1**
F23	Composition	**1.5**	5	4	3	**1.5**
F24		5	4	3	**1**	2
F25		**1**	2	5	4	3
F26		4.5	**2**	4.5	**2**	**2**
F27		2	5	4	3	**1**
F28		2	3	4	5	**1**
F29		**1**	3	5	4	2
F30		**1**	4	3	5	2
Total		79	122	102.5	103	**43.5**
Rank mean		2.6333	4.0667	3.4167	3.4333	**1.4500**
Final rank		2	5	3	4	**1**

Table A12. The competitive results of the Wilcoxon's test.

Corresponding Algorithm	MTBO versus			
	***p*-Values**	**Better**	**Worst**	**Equal**
WOA	9.1900×10^{-8}	25	4	1
PSO	2.8088×10^{-9}	28	1	1
BA	2.4151×10^{-10}	28	1	1
Rao-1	5.0240×10^{-5}	24	4	2

Table A13. Summary of the best results for the CEC 2014 test functions for the different algorithms with D = 30.

Function		Rao-1	BA	PSO	WOA	MTBO
		Best	Best	Best	Best	Best
F1	Unimodal	$1.080 \times 10^{+7}$	$1.220 \times 10^{+7}$	$1.160 \times 10^{+7}$	$1.050 \times 10^{+7}$	$3.130 \times 10^{+5}$
F2		$2.680 \times 10^{+2}$	$2.080 \times 10^{+6}$	$3.670 \times 10^{+5}$	$1.140 \times 10^{+6}$	0.58
F3		$1.510 \times 10^{+4}$	$2.040 \times 10^{+4}$	$1.000 \times 10^{+4}$	$1.090 \times 10^{+4}$	99.80
F4	Simple Multimodal	3.71	104.00	114.00	117.00	0.01
F5		20.80	20.30	20.50	20.30	20.20
F6		17.00	32.20	29.00	27.90	5.74
F7		0.00	0.99	0.79	0.80	0.00
F8		130.00	114.00	131.00	141.00	40.70
F9		198.00	182.00	129.00	187.00	70.60
F10		$5.410 \times 10^{+3}$	$2.220 \times 10^{+3}$	$3.370 \times 10^{+3}$	$3.560 \times 10^{+3}$	712.00
F11		$6.620 \times 10^{+3}$	$4.490 \times 10^{+3}$	$2.900 \times 10^{+3}$	$3.130 \times 10^{+3}$	$1.690 \times 10^{+3}$
F12		1.87	1.43	1.30	0.91	1.83
F13		0.36	0.38	0.40	0.33	0.23
F14		0.25	0.23	0.16	0.18	0.21
F15		16.50	27.20	31.50	49.20	4.69
F16		12.30	12.10	11.60	12.00	10.50
F17	Hybrid	$1.070 \times 10^{+6}$	$1.560 \times 10^{+6}$	$5.550 \times 10^{+5}$	$8.400 \times 10^{+5}$	$2.620 \times 10^{+4}$
F18		25,600.00	596.00	394.00	384.00	78.90
F19		5.80	16.70	17.50	17.70	8.27
F20		$2.520 \times 10^{+3}$	$1.330 \times 10^{+4}$	$6.790 \times 10^{+3}$	$5.590 \times 10^{+3}$	184.00
F21		$3.450 \times 10^{+5}$	$3.480 \times 10^{+5}$	$5.250 \times 10^{+4}$	$2.880 \times 10^{+5}$	$1.010 \times 10^{+4}$
F22		290.00	487.00	512.00	502.00	47.50
F23	Composition	315.00	322.00	322.00	322.00	315.00
F24		220.00	203.00	200.00	201.00	200.00
F25		205.00	200.00	200.00	200.00	200.00
F26		100.00	100.00	100.00	100.00	100.00
F27		415.00	440.00	417.00	420.00	408.00
F28		750.00	1740.00	200.00	1630.00	1020.00
F29		$2.930 \times 10^{+3}$	$3.420 \times 10^{+4}$	$9.360 \times 10^{+3}$	$1.130 \times 10^{+4}$	$1.330 \times 10^{+3}$
F30		$1.500 \times 10^{+3}$	$2.290 \times 10^{+4}$	$2.870 \times 10^{+4}$	$2.070 \times 10^{+4}$	$1.140 \times 10^{+3}$
Nm		3	3.00	2.00	5.00	3.00
Final rank		3.5	3.50	5.00	2.00	3.50

Nm represents the number of times with a ranking higher than the best value obtained.

Table A14. Summary of the Std. results for the CEC 2014 test functions for the different algorithms with D = 30.

Function		Rao-1	BA	PSO	WOA	MTBO
		STD	STD	STD	STD	STD
F1	Unimodal	$7.81 \times 10^{+6}$	$2.350 \times 10^{+7}$	$1.230 \times 10^{+7}$	$1.220 \times 10^{+7}$	$6.20 \times 10^{+6}$
F2		6490.00	$6.720 \times 10^{+6}$	$8.060 \times 10^{+5}$	$6.080 \times 10^{+6}$	26.20
F3		9110.00	$3.920 \times 10^{+4}$	$2.270 \times 10^{+4}$	$2.330 \times 10^{+4}$	1480.00
F4	Simple Multimodal	56.70	69.60	48.80	65.10	30.40
F5		0.06	0.20	0.15	0.19	0.04
F6		7.43	5.16	3.43	4.06	2.82
F7		0.11	0.10	0.07	0.10	0.03
F8		29.00	32.50	50.80	30.10	23.80
F9		15.50	82.40	57.20	56.10	19.10
F10		478.00	702.00	315.00	568.00	491.00
F11		251.00	838.00	1140.00	802.00	1940.00
F12		0.31	0.59	0.35	0.56	0.28
F13		0.07	0.24	0.07	0.16	0.08
F14		0.30	0.06	0.05	0.05	0.05
F15		0.96	22.80	14.80	32.80	7.51
F16		0.29	0.67	0.50	0.44	0.54
F17	Hybrid	469,000.00	$2.27 \times 10^{+6}$	$1.33 \times 10^{+6}$	$1.52 \times 10^{+6}$	$1.94 \times 10^{+5}$
F18		8,490,000.00	94,400.00	5260.00	95,100.00	3110.00
F19		2.13	31.60	19.00	40.60	2.81
F20		2080.00	14,100.00	22,100.00	10,800.00	910.00
F21		175,000.00	$4.450 \times 10^{+5}$	$7.120 \times 10^{+5}$	$8.790 \times 10^{+5}$	$1.200 \times 10^{+5}$
F22		130.00	299.00	134.00	215.00	144.00
F23	Composition	0.00	5.08	7.27	7.16	0.00
F24		8.36	9.25	2.66	4.96	7.99
F25		2.77	18.30	21.10	16.50	3.45
F26		0.08	0.24	0.14	0.11	0.12
F27		206.00	600.00	360.00	387.00	228.00
F28		331.00	282.00	858.00	684.00	300.00
F29		3,990,000.00	$6.80 \times 10^{+6}$	$5.03 \times 10^{+6}$	$5.16 \times 10^{+6}$	$4.95 \times 10^{+6}$
F30		2360.00	$3.740 \times 10^{+5}$	32,200.00	$1.19 \times 10^{+5}$	12,300.00

Figure A3 also shows the convergence comparisons of the algorithms for multimodal functions.

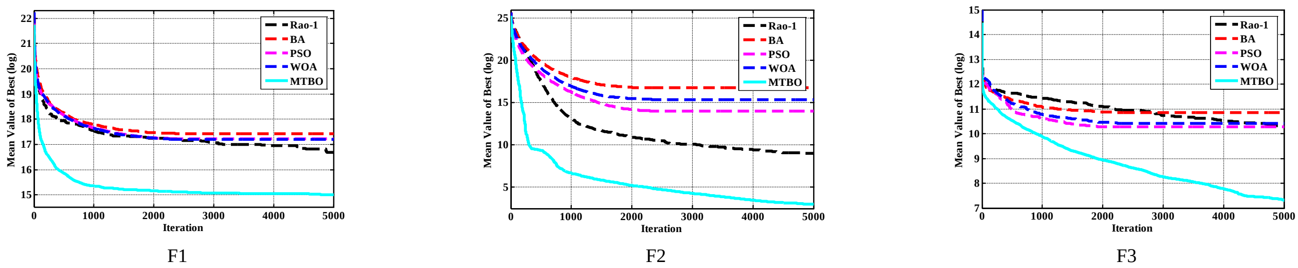


Figure A3. *Cont.*

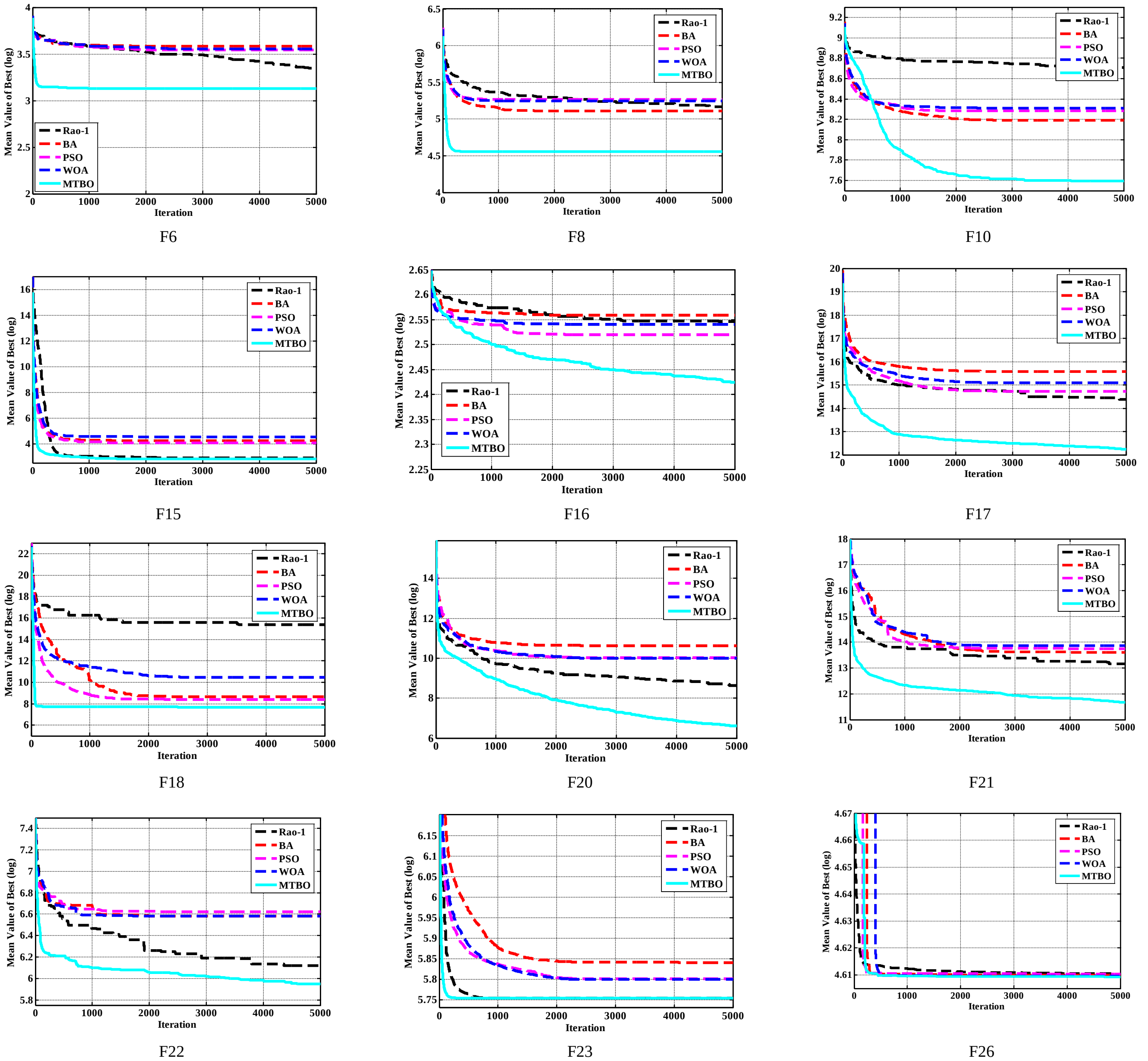


Figure A3. Convergence comparisons of the algorithms for multimodal functions.

References

1. Abualigah, L.; Elaziz, M.A.; Khasawneh, A.M.; Alshinwan, M.; Ibrahim, R.A.; Al-qaness, M.A.; Mirjalili, S.; Sumari, P.; Gandomi, A.H. Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis, and results. *Neural Comput. Appl.* **2022**, *34*, 1–30. [CrossRef]
2. Yang, X.S.; He, X. Nature-inspired optimization algorithms in engineering: Overview and applications. *Nat. Inspired Comput. Eng.* **2016**, *637*, 1–20.
3. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [CrossRef]
4. Rechenberg, I. Evolutionsstrategien. In *Simulationsmethoden in Der Medizin und Biologie*; Springer: Berlin/Heidelberg, Germany, 1978; pp. 83–114.
5. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
6. Koza, J.R.; Poli, R. Genetic programming. In *Search Methodologies*; Springer: Boston, MA, USA, 2005; pp. 127–164.
7. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948.
8. Al Salami, N.M. Ant colony optimization algorithm. *UbiCC J.* **2009**, *4*, 823–826.
9. Karaboga, D. Artificial bee colony algorithm. *Scholarpedia* **2010**, *5*, 6915. [CrossRef]

10. Krishnanand, K.N.; Ghose, D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intell.* **2009**, *3*, 87–124. [CrossRef]
11. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
12. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *arXiv* **2013**, arXiv:1308.3898. [CrossRef]
13. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [CrossRef]
14. Mehmood, K.; Chaudhary, N.I.; Khan, Z.A.; Cheema, K.M.; Raja, M.A.Z.; Milyani, A.H.; Azhari, A.A. Nonlinear Hammerstein System Identification: A Novel Application of Marine Predator Optimization Using the Key Term Separation Technique. *Mathematics* **2022**, *10*, 4217. [CrossRef]
15. Mehmood, K.; Chaudhary, N.I.; Khan, Z.A.; Cheema, K.M.; Raja, M.A.Z.; Milyani, A.H.; Azhari, A.A. Dwarf Mongoose Optimization Metaheuristics for Autoregressive Exogenous Model Identification. *Mathematics* **2022**, *10*, 3821. [CrossRef]
16. Mehmood, K.; Chaudhary, N.I.; Khan, Z.A.; Raja, M.A.Z.; Cheema, K.M.; Milyani, A.H. Design of Aquila Optimization Heuristic for Identification of Control Autoregressive Systems. *Mathematics* **2022**, *10*, 1749. [CrossRef]
17. Malik, N.A.; Chang, C.-L.; Chaudhary, N.I.; Khan, Z.A.; Raja, M.A.Z.; Kiani, A.K.; Milyani, A.H.; Azhari, A.A. Parameter estimation of harmonics arising in electrical instruments of smart grids using cuckoo search heuristics. *Front. Energy Res.* **2022**, *10*, 1733. [CrossRef]
18. Chaudhary, N.I.; Raja, M.A.Z.; Khan, Z.A.; Mehmood, A.; Shah, S.M. Design of fractional hierarchical gradient descent algorithm for parameter estimation of nonlinear control autoregressive systems. *Chaos Solitons Fractals* **2022**, *157*, 111913. [CrossRef]
19. Khan, W.U.; He, Y.; Raja, M.A.Z.; Chaudhary, N.I.; Khan, Z.A.; Shah, S.M. Flower Pollination Heuristics for Nonlinear Active Noise Control Systems. *CMC-Comput. Mater. Contin.* **2021**, *67*, 815–834.
20. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
21. Van Laarhoven, P.J.; Aarts, E.H. Simulated annealing. In *Simulated Annealing: Theory and Applications*; Springer: Dordrecht, The Netherlands, 1987; pp. 7–15.
22. Zhao, W.; Wang, L.; Zhang, Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl. Based Syst.* **2019**, *163*, 283–304. [CrossRef]
23. Yadav, A. AEFA: Artificial electric field algorithm for global optimization. *Swarm Evol. Comput.* **2019**, *48*, 93–108.
24. Rezaee Jordehi, A. A chaotic-based big bang–big crunch algorithm for solving global optimisation problems. *Neural Comput. Appl.* **2014**, *25*, 1329–1335. [CrossRef]
25. Du, H.; Wu, X.; Zhuang, J. Small-world optimization algorithm for function optimization. In Proceedings of the International Conference on Natural Computation, Xi'an, China, 24–28 September 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 264–273.
26. Shah-Hosseini, H. Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation. *Int. J. Comput. Sci. Eng.* **2011**, *6*, 132–140.
27. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [CrossRef]
28. Doğan, B.; Ölmez, T. A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Inf. Sci.* **2015**, *293*, 125–145. [CrossRef]
29. Birbil, Ş.İ.; Fang, S.C. An electromagnetism-like mechanism for global optimization. *J. Glob. Optim.* **2003**, *25*, 263–282. [CrossRef]
30. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **2011**, *43*, 303–315. [CrossRef]
31. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
32. Glover, F. Tabu search—Part I. *ORSA J. Comput.* **1989**, *1*, 190–206. [CrossRef]
33. He, S.; Wu, Q.H.; Saunders, J.R. Group search optimizer: An optimization algorithm inspired by animal searching behavior. *IEEE Trans. Evol. Comput.* **2009**, *13*, 973–990. [CrossRef]
34. Xing, B.; Gao, W.J. Imperialist competitive algorithm. In *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*; Springer: Cham, Switzerland, 2014; pp. 203–209.
35. Kashan, A.H. League championship algorithm: A new algorithm for numerical function optimization. In Proceedings of the International Conference of Soft Computing and Pattern Recognition, Malacca, Malaysia, 4–7 December 2009; IEEE: New York, NY, USA, 2009; pp. 43–48.
36. Tan, Y.; Zhu, Y. Fireworks algorithm for optimization. In Proceedings of the International Conference in Swarm Intelligence, Brussels, Belgium, 8–10 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 355–364.
37. Moosavian, N.; Roodsari, B.K. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm Evol. Comput.* **2014**, *17*, 14–24. [CrossRef]
38. Dai, C.; Zhu, Y.; Chen, W. Seeker optimization algorithm. In Proceedings of the International Conference on Computational and Information Science, Reading, UK, 28–31 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 167–176.
39. Ghorbani, N.; Babaei, E. Exchange market algorithm. *Appl. Soft Comput.* **2014**, *19*, 177–187. [CrossRef]
40. Eita, M.A.; Fahmy, M.M. Group counseling optimization. *Appl. Soft Comput.* **2014**, *22*, 585–604. [CrossRef]
41. Dehghani, M.; Trojovská, E.; Trojovský, P. A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. *Sci. Rep.* **2022**, *12*, 9924. [CrossRef]

42. Alba, E.; Dorronsoro, B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Trans. Evol. Comput.* **2005**, *9*, 126–142. [CrossRef]
43. Lin, L.; Gen, M. Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation. *Soft Comput.* **2009**, *13*, 157–168. [CrossRef]
44. Chen, J.; Xin, B.; Peng, Z.; Dou, L.; Zhang, J. Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2009**, *39*, 680–691. [CrossRef]
45. Macready, W.G.; Wolpert, D.H. Bandit problems and the exploration/exploitation tradeoff. *IEEE Trans. Evol. Comput.* **1998**, *2*, 2–22. [CrossRef]
46. Sadoughi, M.K.; Hu, C.; MacKenzie, C.A.; Eshghi, A.T.; Lee, S. Sequential exploration-exploitation with dynamic trade-off for efficient reliability analysis of complex engineered systems. *Struct. Multidiscip. Optim.* **2018**, *57*, 235–250. [CrossRef]
47. Xiong, N.; Molina, D.; Ortiz, M.L.; Herrera, F. A walk into metaheuristics for engineering optimization: Principles methods and recent trends. *Int. J. Comput. Intell. Syst.* **2015**, *8*, 606–636. [CrossRef]
48. Sun, H.; Ebadi, A.G.; Toughani, M.; Nowdeh, S.A.; Naderipour, A.; Abdullah, A. Designing framework of hybrid photovoltaic-biowaste energy system with hydrogen storage considering economic and technical indices using whale optimization algorithm. *Energy* **2022**, *238*, 121555. [CrossRef]
49. Naderipour, A.; Nowdeh, S.A.; Saftjani, P.B.; Abdul-Malek, Z.; Mustafa, M.W.B.; Kamyab, H.; Davoudkhani, I.F. Deterministic and probabilistic multi-objective placement and sizing of wind renewable energy sources using improved spotted hyena optimizer. *J. Clean. Prod.* **2021**, *286*, 124941. [CrossRef]
50. Naderipour, A.; Abdul-Malek, Z.; Noorden, Z.A.; Davoudkhani, I.F.; Nowdeh, S.A.; Kamyab, H.; Chelliapan, S.; Ghiasi, S.M.S. Carrier wave optimization for multi-level photovoltaic system to improvement of power quality in industrial environments based on Salp swarm algorithm. *Environ. Technol. Innov.* **2021**, *21*, 101197. [CrossRef]
51. Babanezhad, M.; Nowdeh, S.A.; Abdelaziz, A.Y.; AboRas, K.M.; Kotb, H. Reactive power based capacitors allocation in distribution network using mathematical remora optimization algorithm considering operation cost and loading conditions. *Alex. Eng. J.* **2022**, *61*, 10511–10526. [CrossRef]
52. Alanazi, A.; Alanazi, M.; Nowdeh, S.A.; Abdelaziz, A.Y.; El-Shahat, A. An optimal sizing framework for autonomous photo-voltaic/hydrokinetic/hydrogen energy system considering cost, reliability and forced outage rate using horse herd optimization. *Energy Rep.* **2022**, *8*, 7154–7175. [CrossRef]
53. Moghaddam, M.J.H.; Kalam, A.; Nowdeh, S.A.; Ahmadi, A.; Babanezhad, M.; Saha, S. Optimal sizing and energy management of stand-alone hybrid photovoltaic/wind system based on hydrogen storage considering LOEE and LOLE reliability indices using flower pollination algorithm. *Renew. Energy* **2019**, *135*, 1412–1434. [CrossRef]
54. Nowdeh, S.A.; Davoudkhani, I.F.; Moghaddam, M.H.; Najmi, E.S.; Abdelaziz, A.Y.; Ahmadi, A.; Razavi, S.; Gandoman, F.H. Fuzzy multi-objective placement of renewable energy sources in distribution system with objective of loss reduction and reliability improvement using a novel hybrid method. *Appl. Soft Comput.* **2019**, *77*, 761–779. [CrossRef]
55. Jahannoosh, M.; Nowdeh, S.A.; Naderipour, A.; Kamyab, H.; Davoudkhani, I.F.; Klemeš, J.J. New hybrid meta-heuristic algorithm for reliable and cost-effective designing of photovoltaic/wind/fuel cell energy system considering load interruption probability. *J. Clean. Prod.* **2021**, *278*, 123406. [CrossRef]
56. Jafar-Nowdeh, A.; Babanezhad, M.; Arabi-Nowdeh, S.; Naderipour, A.; Kamyab, H.; Abdul-Malek, Z.; Ramachandaramurthy, V.K. Meta-heuristic matrix moth–flame algorithm for optimal reconfiguration of distribution networks and placement of solar and wind renewable sources considering reliability. *Environ. Technol. Innov.* **2020**, *20*, 101118. [CrossRef]
57. Jahannoush, M.; Nowdeh, S.A. Optimal designing and management of a stand-alone hybrid energy system using meta-heuristic improved sine–cosine algorithm for Recreational Center case study for Iran country. *Appl. Soft Comput.* **2020**, *96*, 106611. [CrossRef]
58. Naderipour, A.; Abdul-Malek, Z.; Vahid, M.Z.; Seifabad, Z.M.; Hajivand, M.; Arabi-Nowdeh, S. Optimal reliable and cost-effective framework of photovoltaic-wind-battery energy system design considering outage concept using grey wolf optimizer algorithm—Case study for Iran. *IEEE Access* **2019**, *7*, 182611–182623. [CrossRef]
59. Davoodkhani, F.; Arabi Nowdeh, S.; Abdelaziz, A.Y.; Mansoori, S.; Nasri, S.; Alijani, M. A new hybrid method based on gray wolf optimizer-crow search algorithm for maximum power point tracking of photovoltaic energy system. In *Modern Maximum Power Point Tracking Techniques for Photovoltaic Energy Systems*; Springer: Cham, Switzerland, 2020; pp. 421–438.
60. Arabi Nowdeh, S.; Moghaddam, M.J.H.; Nasri, S.; Abdelaziz, A.Y.; Ghanbari, M.; Faraji, I. A new hybrid moth flame optimizer-perturb and observe method for maximum power point tracking in photovoltaic energy system. In *Modern Maximum Power Point Tracking Techniques for Photovoltaic Energy Systems*; Springer: Cham, Switzerland, 2020; pp. 401–420.
61. Shakarami, M.R.; Davoudkhani, I.F. Wide-area power system stabilizer design based on grey wolf optimization algorithm considering the time delay. *Electr. Power Syst. Res.* **2016**, *133*, 149–159. [CrossRef]
62. Ghasemi, M.; Davoudkhani, I.F.; Akbari, E.; Rahimnejad, A.; Ghavidel, S.; Li, L. A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent Flow of Water-based Optimization (TFWO). *Eng. Appl. Artif. Intell.* **2020**, *92*, 103666. [CrossRef]
63. Hadidian-Moghaddam, M.J.; Arabi-Nowdeh, S.; Bigdeli, M.; Azizian, D. A multi-objective optimal sizing and siting of distributed generation using ant lion optimization technique. *Ain Shams Eng. J.* **2018**, *9*, 2101–2109. [CrossRef]
64. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]

65. Bedogni, V.; Manes, A. A constitutive equation for the behaviour of a mountaineering rope under stretching during a climber's fall. *Procedia Eng.* **2011**, *10*, 3353–3358. [CrossRef]
66. Landrø, M.; Pfuhl, G.; Engeset, R.; Jackson, M.; Hetland, A. Avalanche decision-making frameworks: Classification and description of underlying factors. *Cold Reg. Sci. Technol.* **2020**, *169*, 102903. [CrossRef]
67. Wickens, C.D.; Keller, J.W.; Shaw, C. Human factors in high-altitude mountaineering. *J. Hum. Perform. Extrem. Environ.* **2015**, *12*, 1. [CrossRef]
68. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697. [CrossRef]
69. Belegundu, A.D.; Arora, J.S. A study of mathematical programming methods for structural optimization. Part I: Theory. *Int. J. Numer. Methods Eng.* **1985**, *21*, 1583–1599. [CrossRef]
70. Nowacki, H. Optimization in pre-contract ship design. In Proceedings of the International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design, Tokyo, Japan, 28–30 August 1973.
71. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [CrossRef]
72. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [CrossRef]
73. Kannan, B.K.; Kramer, S.N. 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]
74. Goldberg, D.E.; Holland, J.H. Genetic algorithms and machine learning. *Mach. Learn.* **1988**, *3*, 95–99. [CrossRef]
75. 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]
76. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
77. Ghasemi, M.; Akbari, E.; Rahimnejad, A.; Razavi, S.E.; Ghavidel, S.; Li, L. Phasor particle swarm optimization: A simple and efficient variant of PSO. *Soft Comput.* **2019**, *23*, 9701–9718. [CrossRef]
78. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
79. Wang, H.; Rahnamayan, S.; Sun, H.; Omran, M.G. Gaussian bare-bones differential evolution. *IEEE Trans. Cybern.* **2013**, *43*, 634–647. [CrossRef]
80. Liang, J.J.; Qu, B.Y.; Suganthan, P.N. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Rep. Nanyang Technol. Univ. Singap.* **2013**, *635*, 490.
81. Rao, R. Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. *Int. J. Ind. Eng. Comput.* **2020**, *11*, 107–130. [CrossRef]
82. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Cruz, C., Gonzalez, J., Krasnogor, N., Terraza, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74, SCI 284.
83. Yang, X.-S.; Gandomi, A.H. Bat Algorithm: A Novel Approach for Global Engineering Optimization. *Eng. Comput.* **2012**, *29*, 464–483. [CrossRef]
84. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]