Article

GMBO: Group Mean-Based Optimizer for Solving Various Optimization Problems

Mohammad Dehghani ¹, Zeinab Montazeri ¹ and Štěpán Hubálovský ^{2,*}

¹ Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 71557-13876, Iran; M.Dehghani@sutech.ac.ir (M.D.); z.montazeri@sutech.ac.ir (Z.M.)

² Department of Applied Cybernetics, Faculty of Science, University of Hradec Králové, 500 03 Hradec Králové, Czech Republic

* Correspondence: stepan.hubalovsky@uhk.cz; Tel.: +420-49-333-2704

Abstract: There are many optimization problems in the different disciplines of science that must be solved using the appropriate method. Population-based optimization algorithms are one of the most efficient ways to solve various optimization problems. Population-based optimization algorithms are able to provide appropriate solutions to optimization problems based on a random search of the problem-solving space without the need for gradient and derivative information. In this paper, a new optimization algorithm called the Group Mean-Based Optimizer (GMBO) is presented; it can be applied to solve optimization problems in various fields of science. The main idea in designing the GMBO is to use more effectively the information of different members of the algorithm population based on two selected groups, with the titles of the good group and the bad group. Two new composite members are obtained by averaging each of these groups, which are used to update the population members. The various stages of the GMBO are described and mathematically modeled with the aim of being used to solve optimization problems. The performance of the GMBO in providing a suitable quasi-optimal solution on a set of 23 standard objective functions of different types of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal is evaluated. In addition, the optimization results obtained from the proposed GMBO were compared with eight other widely used optimization algorithms, including the Marine Predators Algorithm (MPA), the Tunicate Swarm Algorithm (TSA), the Whale Optimization Algorithm (WOA), the Grey Wolf Optimizer (GWO), Teaching–Learning-Based Optimization (TLBO), the Gravitational Search Algorithm (GSA), Particle Swarm Optimization (PSO), and the Genetic Algorithm (GA). The optimization results indicated the acceptable performance of the proposed GMBO, and, based on the analysis and comparison of the results, it was determined that the GMBO is superior and much more competitive than the other eight algorithms.

Keywords: optimization; optimization algorithms; population based; exploration; exploitation

Citation: Dehghani, M.; Montazeri, Z.; Hubálovský, Š. GMBO: Group Mean-Based Optimizer for Solving Various Optimization Problems. *Mathematics* **2021**, *9*, 1190. <https://doi.org/10.3390/math9111190>

Academic Editor:
Alessandro Nicolai

Received: 22 April 2021
Accepted: 21 May 2021
Published: 24 May 2021

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

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

1. Introduction

There are many optimization problems in the different sciences and real-life problems that should be optimized and solved using the appropriate methods. Optimization is a science that offers a feasible solution from among the many solutions that exist to an optimization problem. Each optimization problem has three main parts: decision variables, primary objectives (including constraints and limitations), and secondary objectives (including objective functions). Therefore, the decision variables of an optimization problem must be determined in such a way that the objective function is optimized according to the constraints of the problem. After an optimization problem is defined and its various parts are identified, it must be modeled mathematically. In the next step, the designed optimization problem must be solved using the appropriate method.

Optimization problem-solving techniques are generally classified into two groups: definite deterministic and stochastic methods. Deterministic methods include two groups

of gradient-based and non-gradient-based methods. Gradient-based methods, which apply information of derivatives and gradients for finding a global optimal solution, are mathematical programming techniques that consist of nonlinear and linear programming, while non-gradient-based methods use conditions other than the gradient information for finding a global optimal solution [1]. The main disadvantage of gradient-based techniques is a high probability of getting stuck in local optimal solution areas when the search space is nonlinear. On the other hand, the main disadvantage of non-gradient-based techniques is that the implementation of these methods is not easy and applying them requires a high level of mathematical readiness.

Another way to solve optimization problems is to use population-based optimization algorithms, which is a stochastic method. Population-based optimization algorithms are one of the most powerful and effective tools in solving various optimization problems. Population-based optimization algorithms use random variables and operators for scanning the problem-solving space globally, and they avoid getting caught in areas with local optimal solutions. The implementation and understanding of population-based optimization algorithms are easier than with other optimization methods. The important point is that population-based optimization algorithms do not necessarily provide global optimal solutions. However, these methods have advantages that have made them widely used and popular [2]. The quasi-optimal solutions, gradient-free nature, flexibility, and independence to the problem of these methods are important features and advantages of population-based optimization algorithms.

Optimization algorithms have been developed based on simulations of various processes and phenomena of nature, animal behaviors, plants, laws of physics, and rules of games in which there are signs of optimization and progress [3]. For example, the behavior of ants is simulated in the design of the Ant Colony Algorithm (ACO) [4], the simulation of Hooke's physical law is used in the design of the Spring Search Algorithm (SSA) [5], and the simulation of the rules of darts game is used in the design of the Darts Game Optimizer (DGO) [6]. Optimization algorithms are able to provide a suitable solution to optimization problems based on a random scan of the problem search space.

The important point is that each optimization problem has a specific basic optimal solution called the global optimal. On the other hand, the solutions obtained using optimization algorithms are not necessarily the same as the global optimal. However, these solutions are acceptable because they are close to the global optimal. Hence, the solutions provided by the optimization algorithms are called quasi-optimal solutions. According to this concept, in optimizing an optimization problem using several optimization algorithms, an algorithm that can provide a quasi-optimal solution closer to global optimization is a better algorithm. For this reason, various optimization algorithms have been developed by researchers to provide better quasi-optimal solutions. In this regard, optimization algorithms have been applied by scientists in various fields such as video game playing [7] and wireless sensor networks [8] to achieve the appropriate quasi-optimal solution. In addition, optimization techniques can be used to automate generating models that are normally tedious to do one by one. It is also good for structural engineering and mechanical problems [9].

Optimization algorithms can be divided into four main groups—swarm-based, physics-based, game-based, and evolutionary-based optimization algorithms—based on the design idea.

Swarm-based optimization algorithms have been developed based on simulations of natural phenomena, group behaviors of animals, living organisms, and other swarm-based approaches. One of the most famous algorithms in this group is Particle Swarm Optimization (PSO), which was developed based on the simulation of the behavior of a group of birds [10]. PSO has a high convergence speed and can be easily implemented in various optimization problems. However, one of the most important drawbacks of the PSO is premature convergence. This is the result of an inadequate balance mechanism between global and local searches. Modeling the social behavior of gray wolves during hunting is

based on three main hunting steps, including search for prey, siege of prey, and attack on prey, used in designing the Grey Wolf Optimizer (GWO) [11]. The GWO has a high speed in providing a quasi-optimal solution, especially for optimization problems that lack local optimal solutions. However, in complex optimization problems with local optimizations, the GWO gets stuck in local optimal solutions. The Whale Optimization Algorithm (WOA) is an optimization algorithm inspired by the bubble-net hunting method of whales. The WOA is performed into three phases: (i) encircling prey, (ii) bubble-net attack, and (iii) searching for prey [12]. In the WOA, population members rely too much on the optimal population member to find the optimal position, and if the target member is close to the local optimal, the population members are misled, and the algorithm converges to the local optimal instead of converging to the global optimal. The Tunicate Swarm Algorithm (TSA) is designed based on the imitation of jet propulsion and swarm behaviors of tunicates during navigation and foraging processes [13]. The main disadvantages of the TSA are that it has many computations and does not provide good performance for complex problems with high dimensions. The Marine Predators Algorithm (MPA) was developed based on inspiration from the widespread foraging strategy, namely, Lévy and Brownian movements in ocean predators and the optimal encounter rate policy in biological interactions between predator and prey [14]. The MPA has several regulatory and control parameters, where the determination of these parameters is an important challenge.

Physics-based optimization algorithms have been developed based on simulations of various physical laws. Simulated Annealing (SA) is one of the most famous algorithms in this group, which is based on the simulation of the process of gradual heating and cooling of metals [15]. SA offers appropriate solutions in solving simple optimization problems due to its greater focus on local search. However, the main problem of SA is that it depends a lot on the initial value of the parameters. If an inappropriate value is selected for the initial temperature parameter, this algorithm is likely to get stuck in the local optimal. The Gravitational Search Algorithm (GSA) is a physics-based optimization algorithm that was developed based on the modeling of the law of gravity between objects [16]. The high computational complexity of equations and difficulty of implementation are the most important disadvantages of the GSA. The long time required to run the GSA due to its high computations is another disadvantage of this algorithm. Simulation of the law of motion and the law of momentum in a system consisting of bullets of different weights was used in designing the Momentum Search Algorithm (MSA) [17]. Although the MSA has acceptable performance for simple optimization problems without local optimization areas, it is incapable of optimizing complex optimization problems with local optimization areas.

Game-based optimization algorithms have been developed based on the simulation of different rules of individual and group games. Football-Game-Based Optimization (FGBO) was developed based on the mathematical modeling of football league rules as well as the performance and behavior of clubs [18]. The number of population update phases in the FGBO is high, and this leads to a longer implementation time of this algorithm in solving optimization problems. Simulation of the game rules and the behavior of players in the hidden-object finding game was used in the design of Hide Objects Game Optimization (HOGO) [19].

Evolutionary-based optimization algorithms are developed based on the simulation of the laws of evolution and the concepts of genetics science. The Genetic Algorithm (GA) is the most famous and widely used evolutionary-based optimization algorithm, which was developed based on the simulation of reproduction and Darwin's theory of evolution [20]. The concepts of the GA are very simple and, therefore, can be easily implemented in solving optimization problems. The low speed of the GA is one of the biggest disadvantages of this algorithm. The inability to solve complex optimization problems and getting caught up in local optimal solutions are other disadvantages of this algorithm.

Numerous optimization algorithms have been developed by researchers and have been used to solve various optimization problems. However, the important point about these optimization algorithms is that one optimization algorithm may perform well in

solving one optimization problem but fail to provide a suitable quasi-optimal solution for another optimization problem. Therefore, a unique optimization algorithm cannot be considered the best optimizer for all optimization problems. The contribution of this paper is in designing a new optimizer that offers quasi-optimal solutions that are much more suitable than the existing methods and is very close to the optimal solutions.

In this paper, a new optimization algorithm called the Group Mean-Based Optimizer (GMBO) is presented to provide quasi-optimal solutions close to the global optimal for various optimization problems. The main idea of the GMBO is to create two composite members based on the mean of two selected good and bad groups of population members of the algorithm and then use these two composite members to update the population matrix. The various steps of the proposed GMBO are described and then mathematically modeled for implementation in optimization problems. The feature of the proposed GMBO is that it has no control parameter and, therefore, does not need to adjust the parameter. The performance of the GMBO in solving optimization problems is tested on a set of 23 standard objective functions, and then its results are compared with 8 other optimization algorithms.

The rest of the paper is organized in such a way that the proposed GMBO is introduced in Section 2. Then, simulation studies are presented in Section 3. Section 4 analyzes the simulation results. The application of the proposed GMBO in solving real-life problems is tested in Section 5. Finally, conclusions and suggestions for future studies of this research are presented in Section 6.

2. Group Mean-Based Optimizer

In this section, the proposed Group Mean-Based Optimizer (GMBO) algorithm is presented. GMBO is a population-based optimization algorithm designed based on the more effective use of the population members' information in updating the algorithm population. In each iteration of the GMBO, two groups, named the good and bad groups, with a certain number of members, are selected. The main idea in designing the proposed GMBO is to use the two obtained combined members by averaging the two selected groups, the good group and the bad group, in updating the population.

The population members in the proposed GMBO algorithm are identified using a matrix called the population matrix. The number of rows in the population matrix indicates the number of algorithm members, and the number of columns in the population matrix indicates the number of problem variables. Each row of the population matrix as a population member is actually a proposed solution to the optimization problem. The population matrix is defined in the GMBO using Equation (1).

$$X = \left[\begin{array}{c|cccc} X_1 & x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_i & x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_N & x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{array} \right]_{N \times m}, \quad (1)$$

Here, X is the population matrix, X_i is the i 'th population member, $x_{i,d}$ is the value of the d 'th variable proposed by the i 'th population member, N is the number of population members, and m is the number of problem variables.

Based on the values proposed by each population member for the problem variables, the objective function is evaluated. Thus, the values of the objective function are determined using a vector using Equation (2).

$$F = \begin{bmatrix} F_1 & F(X_1) \\ \vdots & \vdots \\ F_i & F(X_i) \\ \vdots & \vdots \\ F_N & F(X_N) \end{bmatrix}_{N \times 1}, \quad (2)$$

Here, F is the vector of objective function, and F_i is the objective function value based on the i 'th population member.

In this step of modeling, the good and bad groups are selected based on the values of the objective function. The good group consists of a certain number of members with the best values of the objective function, and the bad group consists of a certain number of members with the worst values of the objective function. The selection of these two groups is simulated using Equations (3)–(6).

$$F^s = \begin{bmatrix} F_1^s & \text{minimum}(F) \\ \vdots & \vdots \\ F_i^s & \vdots \\ \vdots & \vdots \\ F_N^s & \text{maximum}(F) \end{bmatrix}_{N \times 1}, \quad (3)$$

$$X^s = \begin{bmatrix} X_1^s & x_{1,1}^s & \cdots & x_{1,d}^s & \cdots & x_{1,m}^s \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_i^s & x_{i,1}^s & \cdots & x_{i,d}^s & \cdots & x_{i,m}^s \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_N^s & x_{N,1}^s & \cdots & x_{N,d}^s & \cdots & x_{N,m}^s \end{bmatrix}_{N \times m}, \quad (4)$$

$$G_{N_G \times m} = X_i^s \text{ \& } i = 1 : N_G, \quad (5)$$

$$B_{N_B \times m} = X_i^s \text{ \& } i = N - N_B + 1 : N, \quad (6)$$

Here, F^s is the sorted objective function vector based on the objective function value that is sorted from best member to worst member, X^s is the sorted population matrix based on the objective function value, G is the selected good group, B is the selected bad group, N_G is the number of good group members, and N_B is the number of bad group members.

After determining the two selected good and bad groups, in this stage, two composite members are obtained based on the averaging of these groups. This stage is simulated using Equations (7) and (8).

$$MG = \text{mean}(G_{N_G \times m}), \quad (7)$$

$$MB = \text{mean}(B_{N_B \times m}), \quad (8)$$

Here, MG is the composite member based on the mean of the good group, and MB is the composite member based on the mean of the bad group.

In this step of modeling of the proposed GMBO, the population matrix is updated in three stages based on the two composite members as well as the best member of the population. This update, based on the composite member of the good group, is simulated using Equations (9) and (10).

$$x_{i,d}^G = x_{i,d} + r \times (MG_{i,d} - x_{i,d}) \times \text{sign}(F_i - F_{MG}), \quad (9)$$

$$X_i = \begin{cases} X_i^G, & F_i^G < F_i \\ X_i, & \text{else} \end{cases}, \quad (10)$$

Here, $x_{i,d}^G$ is the new value for the d 'th problem variable, r is a random number in the interval of $[0 - 1]$, F_{MG} is the objective function value of the composite member of the good group, X_i^G is the new position of the i 'th population member, and F_i^G is its objective function value.

In the second stage, the population matrix is updated based on the composite member of the bad group, which is simulated using Equations (11) and (12).

$$x_{i,d}^B = x_{i,d} + r \times (MB_{i,d} - x_{i,d}) \times \text{sign}(F_i - F_{MB}), \quad (11)$$

$$X_i = \begin{cases} X_i^B, & F_i^B < F_i \\ X_i, & \text{else} \end{cases}, \quad (12)$$

Here, $x_{i,d}^B$ is the new value for the d 'th problem variable, r is a random number in the interval of $[0 - 1]$, F_{MB} is the objective function value of the composite member of the bad group, X_i^B is the new position of the i 'th population member, and F_i^B is its objective function value.

In the third stage, the population matrix is also updated based on the best member of the population using Equations (13) and (14).

$$x'_{i,d} = x_{i,d} + r \times (x_{i,d}^{best} - x_{i,d}), \quad (13)$$

$$X_i = \begin{cases} X'_i, & F'_i < F_i \\ X_i, & \text{else} \end{cases}, \quad (14)$$

Here, $x'_{i,d}$ is the new value for the d 'th problem variable, r is a random number in the interval of $[0 - 1]$, X'_i is the new position of the i 'th population member, and F'_i is its objective function value.

The process of updating the population matrix is repeated until the end of the algorithm. After the last iteration of the algorithm, the best quasi-optimal solution obtained with the GMBO is available as the output of the algorithm. The various steps of implementing the proposed GMBO algorithm are presented as pseudocode in Algorithm 1 and also as a flowchart in Figure 1.

Algorithm 1. Pseudocode of the GMBO algorithm.

start GMBO.

- 1: Input problem information: variables, objective function, and constraints.
 - 2: Set number of population members (N) and iterations (T).
 - 3: Generate an initial population matrix at random.
 - 4: Evaluate the objective function.
 - 5: **for** $t = 1:T$
 - 6: Sort population matrix based on Equations (3) and (4).
 - 7: Update good group based on Equation (5).
 - 8: Update bad group based on Equation (6).
 - 9: Update composite members (MG and MB) based on Equations (7) and (8).
 - 10: **for** $i = 1:N$
 - 11: Update X_i based on first stage using Equations (9) and (10).
 - 12: Update X_i based on second stage using Equations (11) and (12).
 - 13: Update X_i based on third stage using Equations (13) and (14).
 - 14: **end for**
 - 15: Save best quasi-optimal solution obtained with the GMBO so far.
 - 16: **end for**
 - 17: Output best quasi-optimal solution obtained with the GMBO.
- end GMBO.**
-

Figure 1. Flowchart of the GMBO algorithm.

3. Simulation Study and Results

In this section, the simulation studies of the proposed GMBO in terms of providing suitable quasi-optimal solutions and solving various optimization problems are presented. The performance of the GMBO in providing a quasi-suitable solution on a set of 23 standard objective functions of different types of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal functions was evaluated. Detailed information on these 23 standard objective functions is provided in Appendix A to Tables A1–A3.

In order to analyze the optimization results obtained from the GMBO, these results were compared with the performance of eight other optimization algorithms, including (i) famous methods: Genetic Algorithm (GA) [20], Particle Swarm Optimization (PSO) [10], (ii) popular methods: Gravitational Search Algorithm (GSA) [16], Teaching-Learning-Based Optimization (TLBO) [21], Grey Wolf Optimizer (GWO) [11], Whale Optimization Algorithm (WOA) [12], and (iii) recently methods: Tunicate Swarm Algorithm (TSA) [13] and Marine Predators Algorithm (MPA) [14]. The experimentation was done on MATLAB (R2020a version) using a 64-bit Core i7 processor of 3.20 GHz and 16 GB of main memory. Each of the optimization algorithms was implemented twenty independent times; the optimization results are presented as the average and the standard deviation of the best solutions as “ave” and “std”, respectively.

The values used for the main controlling parameters of the comparative algorithms are specified in Table 1.

Table 1. Parameter values for the comparative algorithms.

Algorithm	Parameter	Value
MPA	Constant number	$P = 0.5$
	Random vector	R is a vector of uniform random numbers in $[0 - 1]$.
	Fish Aggregating Devices (FADs)	$FADs = 0.2$
	Binary vector	$U = 0$ or 1
TSA	$P_{\min}$ and $P_{\max}$	1, 4
	$c1, c2, c3$	random numbers lie in the range of $[0 - 1]$.
WOA	Convergence parameter (a)	a : Linear reduction from 2 to 0.
	r is a random vector in $[0 - 1]$.	
	l is a random number in $[-1, 1]$.	
GWO	Convergence parameter (a)	a : Linear reduction from 2 to 0.
TLBO	T_F : teaching factor	$T_F = \text{round}[(1 + \text{rand})]$
	random number	rand is a random number between $[0 - 1]$.
GSA	$\text{Alpha}, G_0, R_{\text{norm}}, R_{\text{power}}$	20, 100, 2, 1
PSO	Topology	Fully connected
	Cognitive and social constant	(C_1, C_2) 2, 2
	Inertia weight	Linear reduction from 0.9 to 0.1
	Velocity limit	10% of dimension range
GA	Type	Real coded
	Selection	Roulette wheel (Proportionate)
	Crossover	Whole arithmetic (Probability = 0.8, $\alpha = [-0.5, 1.5]$)
	Mutation	Gaussian (Probability = 0.05)

3.1. Evaluation Results on Unimodal Objective Functions

Seven objective functions of unimodal functions, including F1 to F7, have been selected to evaluate the performance of the GMBO in providing a quasi-optimal solution. For this purpose, the GMBO and eight other algorithms are implemented on these functions, and the optimization results are presented in Table 2. The GMBO is able to provide the basic

optimal solution for the F1, F2, F3, and F6 objective functions. Additionally, the GMBO has had an acceptable performance in providing quasi-optimal solutions for the F4, F5, and F7 objective functions. The presented results in Table 2 show that the GMBO has an acceptable performance in solving the objective functions of the unimodal type and is more suitable than the eight competing optimization algorithms.

3.2. Evaluation Results on High-Dimensional Multimodal Objective Functions

Six F8 to F13 objective functions were selected to evaluate the performance of the GMBO and eight other optimization algorithms in optimizing high-dimensional multimodal functions. The optimization results of these objective functions are presented in Table 3. Based on this table, the GMBO is able to provide the basic optimal solution for the F9 and F11 objective functions. The GMBO generates the best quasi-optimal solutions for F10 and F12 objective functions. For the F13 objective function, the GSA is the best optimizer, and the GMBO is the second-best optimizer. The optimization results show that the GMBO obtains very competitive results in the majority of high-dimensional multimodal objective functions.

3.3. Evaluation Results on Fixed-Dimensional Multimodal Objective Functions

Ten objective functions of fixed-dimensional multimodal functions, including F14 to F23, were selected to evaluate the performance of the GMBO in providing quasi-optimal solutions. The results of the implementation of the GMBO and eight other optimization algorithms on this type of objective function are presented in Table 4. Based on this table, the GMBO is able to provide the basic optimal solution for the F14, F17, F18, F19, F20, F21, F22, and F23 objective functions. The GMBO for the F15 and F16 objective functions also provides quasi-optimal solutions that are very close to the global optimal solutions. The simulation results show that the GMBO provides more acceptable and competitive performance in solving this type of optimization problem.

Tables 2–4 present the optimization results of the F1 to F23 objective functions using the GMBO and eight other compared optimization algorithms. In order to further analyze and visually compare the performance of the optimization algorithms, a boxplot of results for each algorithm and objective function is shown in Figure 2.

Table 2. Optimization results of GMBO and other optimization algorithms on unimodal objective functions.

		GA	PSO	GSA	TLBO	GWO	WOA	TSA	MPA	GMBO
F ₁	Ave	13.2405	1.7740×10^{-5}	2.0255×10^{-17}	8.3373×10^{-60}	1.09×10^{-58}	2.1741×10^{-9}	7.71×10^{-38}	3.2715×10^{-21}	0
	std	4.7664×10^{-15}	6.4396×10^{-21}	1.1369×10^{-32}	4.9436×10^{-76}	5.1413×10^{-74}	7.3985×10^{-25}	7.00×10^{-21}	4.6153×10^{-21}	0
F ₂	Ave	2.4794	0.3411	2.3702×10^{-8}	7.1704×10^{-35}	1.2952×10^{-34}	0.5462	8.48×10^{-39}	1.57×10^{-12}	0
	std	2.2342×10^{-15}	7.4476×10^{-17}	5.1789×10^{-24}	6.6936×10^{-50}	1.9127×10^{-50}	1.7377×10^{-16}	5.92×10^{-41}	1.42×10^{-12}	0
F ₃	Ave	1536.8963	589.492	279.3439	2.7531×10^{-15}	7.4091×10^{-15}	1.7634×10^{-8}	1.15×10^{-21}	0.0864	0
	std	6.6095×10^{-13}	7.1179×10^{-13}	1.2075×10^{-13}	2.6459×10^{-31}	5.6446×10^{-30}	1.0357×10^{-23}	6.70×10^{-21}	0.1444	0
F ₄	Ave	2.0942	3.9634	3.2547×10^{-9}	9.4199×10^{-15}	1.2599×10^{-14}	2.9009×10^{-5}	1.33×10^{-23}	2.6×10^{-8}	4.6×10^{-297}
	std	2.2342×10^{-15}	1.9860×10^{-16}	2.0346×10^{-24}	2.1167×10^{-30}	1.0583×10^{-29}	1.2121×10^{-20}	1.15×10^{-22}	9.25×10^{-9}	0
F ₅	Ave	310.4273	50.26245	36.10695	146.4564	26.8607	41.7767	28.8615	46.049	26.1650
	std	2.0972×10^{-13}	1.5888×10^{-14}	3.0982×10^{-14}	1.9065×10^{-14}	0	2.5421×10^{-14}	4.76×10^{-3}	0.4219	3.89×10^{-14}
F ₆	Ave	14.55	20.25	0	0.4435	0.6423	1.6085×10^{-9}	7.10×10^{-21}	0.398	0
	std	3.1776×10^{-15}	1.2564	0	4.2203×10^{-16}	6.2063×10^{-17}	4.6240×10^{-25}	1.12×10^{-25}	0.1914	0
F ₇	Ave	5.6799×10^{-3}	0.1134	0.0206	0.0017	0.0008	0.0205	3.72×10^{-4}	0.0018	3.95×10^{-6}
	std	7.7579×10^{-19}	4.3444×10^{-17}	2.7152×10^{-18}	3.87896×10^{-19}	7.2730×10^{-20}	1.5515×10^{-18}	5.09×10^{-5}	0.001	7.58×10^{-21}

Table 3. Optimization results of the GMBO and other optimization algorithms on high-dimensional test objective functions.

		GA	PSO	GSA	TLBO	GWO	WOA	TSA	MPA	GMBO
F ₈	Ave	−8184.4142	−6908.6558	−2849.0724	−7408.6107	−5885.1172	−1663.9782	−5740.3388	−3594.16321	−5431.66
	std	833.2165	625.6248	264.3516	513.5784	467.5138	716.3492	41.5	811.32651	4.07×10^{-12}
F ₉	Ave	62.4114	57.0613	16.2675	10.2485	8.5265×10^{-15}	4.2011	5.70×10^{-3}	140.1238	0
	std	2.5421×10^{-14}	6.3552×10^{-15}	3.1776×10^{-15}	5.5608×10^{-15}	5.6446×10^{-30}	4.3692×10^{-15}	1.46×10^{-3}	26.3124	0
F ₁₀	Ave	3.2218	2.1546	3.5673×10^{-9}	0.2757	1.7053×10^{-14}	0.3293	9.80×10^{-14}	9.6987×10^{-12}	8.88×10^{-16}
	std	5.1636×10^{-15}	7.9441×10^{-16}	3.6992×10^{-25}	2.5641×10^{-15}	2.7517×10^{-29}	1.9860×10^{-16}	4.51×10^{-12}	6.1325×10^{-12}	5.29×10^{-31}
F ₁₁	Ave	1.2302	0.0462	3.7375	0.6082	0.0037	0.1189	1.00×10^{-7}	0	0
	std	8.4406×10^{-16}	3.1031×10^{-18}	2.7804×10^{-15}	1.9860×10^{-16}	1.2606×10^{-18}	8.9991×10^{-17}	7.46×10^{-7}	0	0

Table 3. Cont.

		GA	PSO	GSA	TLBO	GWO	WOA	TSA	MPA	GMBO
F ₁₂	Ave	0.047	0.4806	0.0362	0.0203	0.0372	1.7414	0.0368	0.0851	0.0140
	std	4.6547×10^{-18}	1.8619×10^{-16}	6.2063×10^{-18}	7.7579×10^{-19}	4.3444×10^{-17}	8.1347×10^{-12}	1.5461×10^{-2}	0.0052	3.1×10^{-17}
F ₁₃	Ave	1.2085	0.5084	0.002	0.3293	0.5763	0.3456	2.9575	0.4901	0.2659
	std	3.2272×10^{-16}	4.9650×10^{-17}	4.2617×10^{-14}	2.1101×10^{-16}	2.4825×10^{-16}	3.25391×10^{-12}	1.5682×10^{-12}	0.1932	6.95×10^{-16}

Table 4. Optimization results of the GMBO and other optimization algorithms on fixed-dimensional objective functions.

	E	GA	PSO	GSA	TLBO	GWO	WOA	TSA	MPA	GMBO
F ₁₄	Ave	0.9986	2.1735	3.5913	2.2721	3.7408	0.998	1.9923	0.998	0.998
	std	1.5640×10^{-15}	7.9441×10^{-16}	7.9441×10^{-16}	1.9860×10^{-16}	6.4545×10^{-15}	9.4336×10^{-16}	2.6548×10^{-7}	4.2735×10^{-16}	4.97×10^{-17}
F ₁₅	Ave	5.3952×10^{-2}	0.0535	0.0024	0.0033	0.0063	0.0049	0.0004	0.003	0.0003
	std	7.0791×10^{-18}	3.8789×10^{-19}	2.9092×10^{-19}	1.2218×10^{-17}	1.1636×10^{-18}	3.4910×10^{-18}	9.0125×10^{-4}	4.0951×10^{-15}	7.76×10^{-19}
F ₁₆	Ave	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	std	7.9441×10^{-16}	3.4755×10^{-16}	5.9580×10^{-16}	1.4398×10^{-15}	3.9720×10^{-16}	9.9301×10^{-16}	2.6514×10^{-16}	4.4652×10^{-16}	2.98×10^{-16}
F ₁₇	Ave	0.4369	0.7854	0.3978	0.3978	0.3978	0.4047	0.3991	0.3979	0.3978
	std	4.9650×10^{-17}	4.9650×10^{-17}	9.9301×10^{-17}	7.4476×10^{-17}	8.6888×10^{-17}	2.4825×10^{-17}	2.1596×10^{-16}	9.1235×10^{-15}	4.97×10^{-17}
F ₁₈	Ave	4.3592	3	3	3.0009	3	3	3	3	3
	std	5.9580×10^{-16}	3.6741×10^{-15}	6.9511×10^{-16}	1.5888×10^{-15}	2.0853×10^{-15}	5.6984×10^{-15}	2.6528×10^{-15}	1.9584×10^{-15}	2.98×10^{-16}
F ₁₉	Ave	−3.85434	−3.8627	−3.8627	−3.8609	−3.8621	−3.8627	−3.8066	−3.8627	−3.8628
	std	9.9301×10^{-17}	8.9371×10^{-15}	8.3413×10^{-15}	7.3483×10^{-15}	2.4825×10^{-15}	3.1916×10^{-15}	2.6357×10^{-15}	4.2428×10^{-15}	5.96×10^{-16}
F ₂₀	Ave	−2.8239	−3.2619	−3.0396	−3.2014	−3.2523	−3.2424	−3.3206	−3.3211	−3.322
	std	3.97205×10^{-16}	2.9790×10^{-16}	2.1846×10^{-14}	1.7874×10^{-15}	2.1846×10^{-15}	7.9441×10^{-16}	5.6918×10^{-15}	1.1421×10^{-11}	1.99×10^{-15}
F ₂₁	Ave	−4.3040	−5.3891	−5.1486	−9.1746	−9.6452	−7.4016	−5.5021	−10.1532	−10.1532
	std	1.5888×10^{-15}	1.4895×10^{-15}	2.9790×10^{-16}	8.5399×10^{-15}	6.5538×10^{-15}	2.3819×10^{-11}	5.4615×10^{-13}	2.5361×10^{-11}	1.99×10^{-15}
F ₂₂	Ave	−5.1174	−7.6323	−9.0239	−10.0389	−10.4025	−8.8165	−5.0625	−10.4029	−10.4029
	std	1.2909×10^{-15}	1.5888×10^{-15}	1.6484×10^{-12}	1.5292×10^{-14}	1.9860×10^{-15}	6.7524×10^{-15}	8.4637×10^{-14}	2.8154×10^{-11}	5.16×10^{-15}
F ₂₃	Ave	−6.5621	−6.1648	−8.9045	−9.2905	−10.1302	−10.0003	−10.3613	−10.5364	−10.5364
	std	3.8727×10^{-15}	2.7804×10^{-15}	7.1497×10^{-14}	1.1916×10^{-15}	4.5678×10^{-15}	9.1357×10^{-15}	7.6492×10^{-12}	3.9861×10^{-11}	1.96×10^{-15}

Figure 2. Cont.

Figure 2. Boxplot of composition objective function results for different optimization algorithms.

3.4. Statistical Analysis

This section provides a statistical analysis of the performance of the GMBO and other optimization algorithms in optimizing the mentioned objective functions. Although presenting the optimization results as the average and standard deviation of the best solutions provides valuable information on the performance of the optimization algorithms, statistical analysis is important to ensure the superiority of the proposed GMBO algorithm and the nonrandomness of this superiority. Therefore, in this study, the Wilcoxon rank-sum test, which is a nonparametric statistical test, was applied to determine the significance of the results. The Wilcoxon rank-sum test was applied to specify whether the results obtained by the proposed GMBO were different from the other eight optimization algorithms in a statistically significant way.

A p -value specifies whether the given algorithm is statistically significant or not. If the p -value of the given algorithm is less than 0.05, then the corresponding algorithm is statistically significant. The results of the analysis using the Wilcoxon rank-sum test for the objective functions are shown in Table 5. It is observed from Table 5 that the GMBO is significant superiority to the other competitor algorithms based on the p -values, which are less than 0.05.

Table 5. Obtained results from the Wilcoxon test ($p \geq 0.05$).

Compared Algorithms	Unimodal	High-Multimodal	Fixed-Multimodal
GMBO vs. MPA	0.015625	0.0625	0.125
GMBO vs. TSA	0.015625	0.4375	0.007813
GMBO vs. WOA	0.015625	0.03125	0.015625
GMBO vs. GWO	0.015625	0.4375	0.015625
GMBO vs. TLBO	0.015625	0.4375	0.007813
GMBO vs. GSA	0.03125	0.15625	0.015625
GMBO vs. PSO	0.015625	0.4375	0.007813
GMBO vs. GA	0.015625	0.4375	0.003906

3.5. Sensitivity Analysis

In this section, the sensitivity of the proposed GMBO algorithm to the two parameters of the number of population members and the maximum number of iterations is analyzed.

In order to analyze the sensitivity of the proposed algorithm to the number of population members parameter, the GMBO was run independently on all 23 objective functions for the different populations with the number of 20, 30, 50, and 80 members. The results of this simulation for different numbers of population members are presented in Table 6. It is analyzed from Table 6 that for the proposed GMBO, the value of objective function decreases when the number of search agents increases. Convergence curves of GMBO for the sensitivity analysis of the number of population members are shown in Figure 3.

In order to analyze the sensitivity of the proposed GMBO algorithm to the maximum number of iterations parameter, the GMBO was run independently on all 23 objective functions for the maximum number of iterations of 100, 500, 800, and 1000. The performance of the proposed GMBO algorithm, as affected by the different maximum numbers of iterations, is presented in Table 7. The results show that GMBO converges towards the optimal solution when the maximum number of iterations is increased. Convergence curves of the GMBO for the sensitivity analysis of the maximum number of iterations are shown in Figure 4.

Figure 3. Cont.

Figure 3. Sensitivity analysis of the GMBO for the number of population members.

Table 6. Results of algorithm sensitivity analysis of the number of population members.

Objective Function	Number of Population Members			
	20	30	50	80
F ₁	0	0	0	0
F ₂	0	0	0	0
F ₃	0	0	0	0
F ₄	1.8×10^{-287}	8.8×10^{-289}	4.6×10^{-297}	1.2×10^{-299}
F ₅	28.60939	28.74985	26.16505	25.07082
F ₆	0	0	0	0
F ₇	7.74×10^{-5}	6.96×10^{-5}	3.95×10^{-6}	4.23×10^{-7}
F ₈	−3940.63	−4338.67	−5431.66	−6589.74
F ₉	0	0	0	0
F ₁₀	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}
F ₁₁	0	0	0	0
F ₁₂	0.194864	0.042659	0.0140	0.01155
F ₁₃	2.229227	1.646969	0.2659	0.1071
F ₁₄	2.982105	0.998004	0.998	0.998
F ₁₅	0.000329	0.000308	0.0003	0.0003
F ₁₆	−1.03163	−1.03163	−1.0316	−1.0316
F ₁₇	0.397904	0.397887	0.3978	0.3978
F ₁₈	3.000004	3	3	3
F ₁₉	−3.86264	−3.86278	−3.8628	−3.8628
F ₂₀	−3.32194	−3.32113	−3.322	−3.322
F ₂₁	−9.70646	−10.1528	−10.1532	−10.1532
F ₂₂	−10.3934	−10.4019	−10.4029	−10.4029
F ₂₃	−7.51071	−10.5348	−10.5364	−10.5364

Table 7. Results of algorithm sensitivity analysis of the maximum number of iterations.

Objective Function	Maximum Number of Iterations			
	100	500	800	1000
F ₁	1.3×10^{-58}	0	0	0
F ₂	1.08×10^{-30}	4×10^{-163}	9.1×10^{-262}	0
F ₃	1.58×10^{-39}	5×10^{-227}	0	0
F ₄	7.14×10^{-28}	4.5×10^{-145}	7.8×10^{-232}	4.6×10^{-297}
F ₅	27.13738	27.07678	26.88103	26.16505
F ₆	0	0	0	0
F ₇	0.000167	4.31×10^{-5}	2.5×10^{-5}	3.95×10^{-6}
F ₈	−3468.14	−4367.85	−5258.08	−5431.66
F ₉	0	0	0	0
F ₁₀	4.44×10^{-15}	4.44×10^{-15}	4.44×10^{-15}	8.88×10^{-16}
F ₁₁	0	0	0	0
F ₁₂	0.032198	0.02536	0.0166	0.0140
F ₁₃	1.1671	1.1658	1.0463	0.2659
F ₁₄	0.998	0.998	0.998	0.998
F ₁₅	0.000351	0.000308	0.000308	0.0003
F ₁₆	−1.03163	−1.03163	−1.03163	−1.0316
F ₁₇	0.397915	0.397888	0.397888	0.3978
F ₁₈	3.000018	3	3	3
F ₁₉	−3.86268	−3.86278	−3.86278	−3.8628
F ₂₀	−3.31354	−3.32177	−3.32197	−3.322
F ₂₁	−6.38282	−10.1494	−10.1531	−10.1532
F ₂₂	−10.3783	−10.4004	−10.402	−10.4029
F ₂₃	−10.5037	−10.5337	−10.5356	−10.5364

Figure 4. Cont.

Figure 4. Sensitivity analysis of the GMBO for the maximum number of iterations.

4. Discussion

Population-based optimization algorithms have been developed with the aim of providing a suitable quasi-optimal solution that is close to the global optimal solution. Exploitation power and exploration power are two important indicators in evaluating the performance of optimization algorithms.

Exploitation power demonstrates the ability of an optimization algorithm to provide a quasi-optimal solution to an optimization problem. In fact, based on the definition of exploitation power, an algorithm must be able to provide a quasi-optimal solution to an optimization problem after complete implementation and at the end of iterations. This indicator is especially evident in optimization problems that have only one basic optimal solution. The F1 to F7 objective functions, which are unimodal functions, have only one basic optimal solution and are, therefore, suitable for analyzing the exploitation power of optimization algorithms. The performance results of the GMBO and eight other optimization algorithms for optimizing these objective functions are presented in Table 2. These results show that the GMBO has high exploitation power and is able to provide the basic optimal solution to the problem for objective functions F1, F2, F3, F4,

and F6. The GMBO also provides an acceptable quasi-optimal solution for the F5 and F7 objective functions.

Exploratory power demonstrates the ability of an optimization algorithm to accurately scan the search space of an optimization problem. According to the definition of exploratory power, an optimization algorithm should be able to explore different areas of the problem search space during implementation and successive iterations. The exploration index is especially important for optimization problems that have several local optimal solutions in addition to the basic optimal solution. High-dimensional multimodal objective functions, including F8 to F13, and fixed-dimensional multimodal objective functions, including F14 to F23, are functions that have several local optimal solutions in addition to the basic optimal solution and are, therefore, suitable for analyzing the exploration power of optimization algorithms. The optimization results of these objective functions are presented using the GMBO and eight other optimization algorithms in Tables 3 and 4. The results show the high exploration power of the optimization algorithm in solving optimization problems with several local optimal solutions. Therefore, the GMBO is able to scan different areas of the search space well and pass through the optimal local areas with its high exploration power.

The results of the statistical analysis obtained from the Wilcoxon rank test presented in Table 5 ensure that the superiority of the GMBO over other algorithms, as well as its acceptable exploitation and exploration power, is not random.

5. Real-Life Application

In this section, the performance of the GMBO in solving problems in real-life applications is evaluated. For this purpose, the GMBO is implemented in an optimization problem, namely, pressure vessel design.

The mathematical model of this problem was adapted from [22]. Figure 5 shows the schematic view of the pressure vessel problem. Tables 8 and 9 report the performance of the GMBO and other algorithms. The GMBO provides an optimal solution at (0.778201, 0.384739, 40.320910, 200.00000), with a corresponding objective function value of 5886.0424. Figure 6 presents the convergence analysis of the GMBO for the pressure vessel design optimization problem.

Figure 5. Schematic view of the pressure vessel problem.

Table 8. Comparison results for the pressure vessel design problem.

Algorithm	Optimum Variables				Optimum Cost
	T_s	T_h	R	L	
GMBO	0.778201	0.384739	40.320910	200.00000	5886.0424
MPA	0.779035	0.384660	40.327793	199.65029	5889.3689
TSA	0.8303737	0.4162057	42.75127	169.3454	6048.7844
WOA	0.778961	0.384683	40.320913	200.00000	5891.3879
GWO	0.845719	0.418564	43.816270	156.38164	6011.5148
TLBO	0.817577	0.417932	41.74939	183.57270	6137.3724
GSA	1.085800	0.949614	49.345231	169.48741	11,550.2976
PSO	0.752362	0.399540	40.452514	198.00268	5890.3279
GA	1.099523	0.906579	44.456397	179.65887	6550.0230

Table 9. Statistical results for the pressure vessel design problem.

Algorithm	Best	Mean	Worst	SD	Median
GMBO	5886.0424	5884.1401	5891.3099	024.341	5888.0424
MPA	5889.3689	5891.5247	5894.6238	013.910	5890.6497
TSA	6048.7844	6052.6241	6071.2496	002.893	6050.2282
WOA	5891.3879	6531.5032	7394.5879	534.119	6416.1138
GWO	6011.5148	6477.3050	7250.9170	327.007	6397.4805
TLBO	6137.3724	6326.7606	6512.3541	126.609	6318.3179
GSA	11,550.2976	23,342.2909	33,226.2526	5790.625	24,010.0415
PSO	5890.3279	6264.0053	7005.7500	496.128	6112.6899
GA	6550.0230	6643.9870	8005.4397	657.523	7586.0085

Figure 6. Convergence analysis of the GMBO for the pressure vessel design optimization problem.

6. Conclusions and Future Works

Optimization algorithms have a special application in solving optimization problems. In this study, a new optimization algorithm called the Group Mean-Based Optimizer (GMBO) is presented to be used in solving optimization problems in various sciences. The proposed GMBO algorithm is based on the more effective use of the population members' information by creating two new composite members by averaging two selected groups of good and bad members in updating the algorithm population. The various steps of the GMBO have been described and then modeled mathematically to apply to solving optimization problems. The performance of the GMBO in optimization was evaluated on a set of 23 standard objective functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types. The results of the optimization of unimodal objective functions, including F1 to F7, showed the high power of the proposed GMBO in the exploitation index. Additionally, multimodal objective functions, including two groups of F8 to F13 and F14 to F23, showed that the GMBO has a good exploration index in solving optimization problems that also have several local optimal solutions. The optimization results obtained from the GMBO were compared with eight other optimization algorithms,

including the Marine Predators Algorithm (MPA), the Tunicate Swarm Algorithm (TSA), the Whale Optimization Algorithm (WOA), the Grey Wolf Optimizer (GWO), Teaching-Learning-Based Optimization (TLBO), the Gravitational Search Algorithm (GSA), Particle Swarm Optimization (PSO), and the Genetic Algorithm (GA). The optimization results show that the proposed GMBO algorithm has an acceptable ability to solve various optimization problems and is much more competitive than other algorithms. In addition, the efficiency and effectiveness of the GMBO were also demonstrated by applying it to an engineering design problem, namely, pressure vessel design. From the experimental results, it can be concluded that the GMBO is applicable to real-life studies of problems with unknown search spaces.

We propose several perspectives and ideas for future works and studies. The design of a binary version and a multiobjective version of the GMBO is an interesting potential topic for future investigations. Apart from this, applying the GMBO to real-life optimization problems and various other optimization problems can be considered significant contributions as well.

The important issue of all population-based optimization algorithms is that since population-based optimization algorithms provide a quasi-optimal solution to an optimization problem based on random search space searches, this solution may be different from the global optimal solution. Therefore, although the proposed GMBO in this paper has performed well in optimizing optimization problems, it is still possible to introduce new optimization algorithms that can provide more appropriate quasi-optimal solutions that are closer to global optimization.

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

Funding: The research was supported by the Project of Specific Research, PrF UHK No. 2101/2021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The authors declare to honor the Principles of Transparency and Best Practice in Scholarly Publishing about Data.

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

Appendix A

Information on the 23 objective functions is provided in Tables A1–A3.

Table A1. Unimodal test functions.

Objective Function	Range	Dimensions	F _{min}
$F_1(x) = \sum_{i=1}^m x_i^2$	$[-100, 100]$	30	0
$F_2(x) = \sum_{i=1}^m x_i + \prod_{i=1}^m x_i $	$[-10, 10]$	30	0
$F_3(x) = \sum_{i=1}^m (\sum_{j=1}^i x_j)^2$	$[-100, 100]$	30	0
$F_4(x) = \max\{ x_i , 1 \leq i \leq m\}$	$[-100, 100]$	30	0
$F_5(x) = \sum_{i=1}^{m-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	30	0
$F_6(x) = \sum_{i=1}^m ([x_i + 0.5])^2$	$[-100, 100]$	30	0
$F_7(x) = \sum_{i=1}^m ix_i^4 + \text{random}(0, 1)$	$[-1.28, 1.28]$	30	0

Table A2. High-dimensional multimodal test functions.

Objective Function	Range	Dimensions	F _{min}
$F_8(x) = \sum_{i=1}^m -x_i \sin(\sqrt{ x_i })$	$[-500, 500]$	30	-12569
$F_9(x) = \sum_{i=1}^m [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	30	0
$F_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{m} \sum_{i=1}^m x_i^2}\right) - \exp\left(\frac{1}{m} \sum_{i=1}^m \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	30	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^m x_i^2 - \prod_{i=1}^m \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	30	0
$F_{12}(x) = \frac{\pi}{m} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^m (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^m u(x_i, 10, 100, 4)$ $u(x_i, a, i, n) = \begin{cases} k(x_i - a)^n & x_i > -a \\ 0 & -a < x_i < a \\ k(-x_i - a)^n & x_i < -a \end{cases}$	$[-50, 50]$	30	0
$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^m (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_m)] \right\}$ $+ \sum_{i=1}^m u(x_i, 5, 100, 4)$	$[-50, 50]$	30	0

Table A3. Fixed-dimensional multimodal test functions.

Objective Function	Range	Dimensions	F _{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	$[-65.53, 65.53]$	2	0.998
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$[-5, 5]$	4	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]$	2	-1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	$[-5, 10] \times [0, 15]$	2	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-5, 5]$	2	3
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - P_{ij})^2\right)$	$[0, 1]$	3	-3.86
$F_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - P_{ij})^2\right)$	$[0, 1]$	6	-3.22
$F_{21}(x) = -\sum_{i=1}^5 \left[(X - a_i)(X - a_i)^T + 6c_i \right]^{-1}$	$[0, 10]$	4	-10.1532
$F_{22}(x) = -\sum_{i=1}^7 \left[(X - a_i)(X - a_i)^T + 6c_i \right]^{-1}$	$[0, 10]$	4	-10.4029
$F_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + 6c_i \right]^{-1}$	$[0, 10]$	4	-10.5364

References

1. Dhiman, G. SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. *Knowl. Based Syst.* **2021**, *222*, 106926. [\[CrossRef\]](#)
2. Dhiman, G.; Kaur, A. STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Eng. Appl. Artif. Intell.* **2019**, *82*, 148–174. [\[CrossRef\]](#)
3. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [\[CrossRef\]](#)
4. Dorigo, M.; Stützle, T. Ant colony optimization: Overview and recent advances. In *Handbook of Metaheuristics*; Springer International Publishing: Cham, Switzerland, 2019; pp. 311–351.
5. Dehghani, M.; Montazeri, Z.; Dehghani, A.; Seifi, A. Spring Search Algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law. In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEL)*; IEEE: Tehran, Iran, 2017; pp. 210–214.

6. Dehghani, M.; Montazeri, Z.; Givi, H.; Guerrero, J.M.; Dhiman, G. Darts game optimizer: A new optimization technique based on darts game. *Int. J. Intell. Eng. Syst* **2020**, *13*, 286–294. [\[CrossRef\]](#)
7. Gaina, R.D.; Devlin, S.; Lucas, S.M.; Perez, D. Rolling horizon evolutionary algorithms for general video game playing. *IEEE Trans. Games* **2021**. [\[CrossRef\]](#)
8. Niccolai, A.; Grimaccia, F.; Mussetta, M.; Gandelli, A.; Zich, R. Social network optimization for WSN routing: Analysis on problem codification techniques. *Mathematics* **2020**, *8*, 583. [\[CrossRef\]](#)
9. Roy, K.; Ting, T.C.H.; Lau, H.H.; Lim, J.B. Nonlinear behaviour of back-to-back gapped built-up cold-formed steel channel sections under compression. *J. Constr. Steel Res.* **2018**, *147*, 257–276. [\[CrossRef\]](#)
10. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; IEEE Service Center: Piscataway, NJ, USA, 1992; p. 1948.
11. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
12. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
13. Kaur, S.; Awasthi, L.K.; Sangal, A.; Dhiman, G. Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103541. [\[CrossRef\]](#)
14. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [\[CrossRef\]](#)
15. Van Laarhoven, P.J.; Aarts, E.H. Simulated annealing. In *Simulated Annealing: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 1987; pp. 7–15.
16. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [\[CrossRef\]](#)
17. Dehghani, M.; Samet, H. Momentum search algorithm: A new meta-heuristic optimization algorithm inspired by momentum conservation law. *SN Appl. Sci.* **2020**, *2*, 1–15. [\[CrossRef\]](#)
18. Dehghani, M.; Mardaneh, M.; Guerrero, J.M.; Malik, O.; Kumar, V. Football game based optimization: An application to solve energy commitment problem. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 514–523. [\[CrossRef\]](#)
19. Dehghani, M.; Montazeri, Z.; Saremi, S.; Dehghani, A.; Malik, O.P.; Al-Haddad, K.; Guerrero, J.M. HOGO: Hide Objects Game Optimization. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 216–225. [\[CrossRef\]](#)
20. Bose, A.; Biswas, T.; Kuila, P. A novel genetic algorithm based scheduling for multi-core systems. In *Smart Innovations in Communication and Computational Sciences*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 45–54.
21. 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. [\[CrossRef\]](#)
22. Kannan, B.; 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. [\[CrossRef\]](#)