

AMBO: All Members-Based Optimizer for Solving Optimization Problems

Fatemeh Ahmadi Zeidabadi¹, Sajjad Amiri Doumari¹, Mohammad Dehghani², Zeinab Montazeri², Pavel Trojovský^{3,*} and Gaurav Dhiman⁴

¹Department of Mathematics and Computer Sciences, Sirjan University of Technology, Sirjan, Iran

²Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, Iran

³Department of Mathematics, Faculty of Science, University of Hradec Králové, Hradec Králové, 50003, Czech Republic

⁴Department of Computer Science, Government Bikram College of Commerce, Patiala, Punjab, India

*Corresponding Author: Pavel Trojovský. Email: pavel.trojovsky@uhk.cz

Received: 29 April 2021; Accepted: 18 June 2021

Abstract: There are many optimization problems in different branches of science that should be solved using an appropriate methodology. Population-based optimization algorithms are one of the most efficient approaches to solve this type of problems. In this paper, a new optimization algorithm called All Members-Based Optimizer (AMBO) is introduced to solve various optimization problems. The main idea in designing the proposed AMBO algorithm is to use more information from the population members of the algorithm instead of just a few specific members (such as best member and worst member) to update the population matrix. Therefore, in AMBO, any member of the population can play a role in updating the population matrix. The theory of AMBO is described and then mathematically modeled for implementation on optimization problems. The performance of the proposed algorithm is evaluated on a set of twenty-three standard objective functions, which belong to three different categories: unimodal, high-dimensional multimodal, and fixed-dimensional multimodal functions. In order to analyze and compare the optimization results for the mentioned objective functions obtained by AMBO, eight other well-known algorithms have been also implemented. The optimization results demonstrate the ability of AMBO to solve various optimization problems. Also, comparison and analysis of the results show that AMBO is superior and more competitive than the other mentioned algorithms in providing suitable solution.

Keywords: Algorithm; all members; optimization; optimization algorithm; optimization problem; population-based algorithm

1 Introduction

Optimization is defined as finding the best solution out of all possible solutions to a problem by considering the constraints and limitations. Therefore, each optimization problem consists of three main parts: decision variables, primary objectives, and secondary objectives. The decision



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

variables are the same as the problem variables, the primary objectives represent the constraints of the problem, and the secondary objectives are the objective functions of the problem.

Population-based optimization algorithms (PBOAs) are one of the most effective methods for solving optimization problems. PBOAs are able to provide appropriate solutions to optimization problems based on random scan of the search space and through an iterative-based process [1]. Each optimization problem has a best basic solution called global optimum. On the other hand, the solution obtained by PBOAs for the optimization problem is not necessarily the global optimal solution. Hence, the solution provided by PBOAs is referred as quasi-optimal solution [2,3]. Whatever the quasi-optimal solution provided by an algorithm is closer to the global optimum, that algorithm has a better performance in solving that optimization problem. For this reason, various PBOAs have been introduced by scientists to provide quasi-optimal solutions to optimization problems. In this regard, optimization algorithms have been applied in various fields in the literature such as energy [4–7], protection [8], electrical engineering [9–14], and energy carriers [15,16] to achieve the optimal solution.

PBOAs are designed based on simulation of various natural phenomena, behavior of living organisms, physical laws, genetic sciences, rules of the games, and so on. In a general classification based on the design idea, PBOAs are categorized into 4 groups: swarm-based, physics-based, evolutionary-based, and game-based optimization algorithms.

Swarm-based optimization algorithms are designed based on simulating the behavior of living organisms such as animals, plants, and natural phenomena. Particle Swarm Optimization (PSO) is one of the most widely-used algorithms in this category, which is based on simulating behaviors of birds' swarm [17]. Ant Colony Optimization (ACO) is another swarm-based optimization technique that has been introduced based on the behavior of ants in finding the shortest route between their nest and the food source [18]. Simulation of the patient treatment process by the doctor has been used in designing doctor and patient optimizer (DPO) [19]. Some of the other swarm-based optimization algorithms are: Seagull Optimization Algorithm (SOA) [20], Whale Optimization Algorithm (WOA) [21], Firefly Algorithm (FA) [22], Artificial Bee Colony (ABC) [23], Cuckoo Search (CS) [24], Bat-inspired Algorithm (BA) [25], Spotted Hyena Optimizer (SHO) [26], Monkey Search (MS) [27], Artificial Fish-Swarm Algorithm (AFSA) [28], Group Optimization (GO) [29], Dolphin Partner Optimization (DPO) [30], Hunting Search (HS) [31], Coupled Spring Forced Bat Algorithm (SFBA) [32], Teaching-Learning-Based Optimization (TLBO) [33], Grey Wolf Optimizer (GWO) [34], Following Optimization Algorithm (FOA) [35], Moth-Flame Optimization Algorithm (MFO) [36], Grasshopper Optimization Algorithm (GOA) [37], Donkey Theorem Optimization (DTO) [38], Emperor Penguin Optimizer (EPO) [39], Multi Leader Optimizer (MLO) [40], Rat Swarm Optimizer (RSO) [41], and “The Good, the Bad, and the Ugly” Optimizer (GBUO) [42].

Physics-based optimization algorithms are introduced based on simulation of various laws of physics. Spring Search Algorithm (SSA) is one of the algorithms in this group, which was designed based on the simulation of Hooke's law in a system consisting of weights and springs [43]. Momentum Search Algorithm (MSA) is another algorithm based on the simulation of Newton's laws of motion and “momentum conservation principle” [44]. Gravitational Search algorithm (GSA) was introduced based on the law of universal gravitation between objects [45]. Some other popular physics-based optimization algorithms are: Big-Bang Big-Crunch (BBBC) [46], Galaxy-based Search Algorithm (GbSA) [47], Charged System Search (CSS) [48], Particle Collision Algorithm (PCA) [49], Simulated Annealing (SA) [50], Binary Spring Search Algorithm (BSSA) [51], Central Force Optimization (CFO) [52], Ray Optimization (RO) algorithm [53],

Curved Space Optimization (CSO) [54], Henry Gas Solubility Optimization (HGSO) [55], Small World Optimization Algorithm (SWOA) [56], and Artificial Chemical Reaction Optimization Algorithm (ACROA) [57].

Evolutionary-based optimization algorithms are inspired by genetics and inheritance laws. Genetic Algorithm (GA) is the most famous algorithm in this group, which is designed based on simulation of reproduction process and Darwin's theory of evolution by natural selection [58]. Artificial Immune System (AIS) inspired by the mechanism of the human body and the human immune system against viruses and microbes is another evolutionary-based optimization algorithm [59]. Some other evolutionary-based optimization techniques are: Evolutionary Programming (EP) [60], Cultural Algorithm [61], Evolution Strategy (ES) [62], Differential Evolution (DE) [63], Biogeography-Based Optimizer (BBO) [64], Artificial Infectious Disease (AID) [65], Genetic Programming (GP) [66], and Improved Quantum-inspired Differential Evolution (IQDE) algorithm [67].

Game-based optimization algorithms are another POBAs, which are designed based on simulating rules of various games. Football Game-Based Optimization (FGBO) is one of the algorithms in this group, which was introduced based on simulation of football league rules and clubs' behaviors [68]. Darts Game Optimizer (DGO) inspired by the rules of the darts game and the behavior of the players in dart throwing for collecting more points is another game-based optimization technique [69]. Some of the other game-based optimization algorithms are: Orientation Search Algorithm (OSA) [70], Hide Objects Game Optimization (HOGO) [71], Shell Game Optimization (SGO) [72], Binary Orientation Search Algorithm (BOSA) [73], and Dice Game Optimizer (DGO) [74].

In this paper, a new optimization algorithm entitled All Members-Based Optimizer (AMBO) is designed to provide suitable quasi-optimal solutions for various optimization problems. In the proposed AMBO, all members of the population, regardless of their position in the search space, participate in updating the population matrix. Various steps of implementing AMBO are explained and then its mathematical formulation is presented. The performance of AMBO in providing quasi-optimal solution is evaluated for twenty-three standard objective functions of different types.

The rest of the article is organized as follows. In Section 2, the proposed AMBO algorithm is described and modeled. In Section 3, the proposed algorithm is simulated for optimizing different objective functions and the results are presented. Statistical analysis of the results is carried out in Section 4. Finally, the conclusions of this investigation and suggestions for future studies are presented in Section 5.

2 All Members-Based Optimizer

In this section, various steps and mathematical modeling of the proposed optimization algorithm are presented. All Members-Based Optimizer (AMBO) is a PBOA proposed for solving optimization problems. The main idea in designing AMBO is to make more use of the population matrix information as well as the simultaneous participation of all members of the population in updating the algorithm population. The search space for each optimization problem consists of coordinate axes equal to the number of problem variables. In most of optimization algorithms, the best population member directs the population of the algorithm along these axes. Also, in some algorithms, the worst member or several members with specific characteristics are effective in updating the algorithm population. However, an ordinary member of the population may be

more qualified to lead the population in some axes than the best member. Therefore, AMBO is designed based on this concept to use the information of all population members.

Each PBOA has a number of members called the algorithm population. The algorithm population can be displayed using a matrix called the population matrix. Each row of this matrix represents a population member and each column of this matrix represents a variable of the optimization problem. Therefore, the number of rows in the population matrix is equal to the number of population members and the number of columns in this matrix is equal to the number of the optimization problem variables.

In AMBO, the population matrix is represented using Eq. (1).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_1^1 & \cdots & x_1^d & \cdots & x_1^m \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_i^1 & \cdots & x_i^d & \cdots & x_i^m \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_N^1 & \cdots & x_N^d & \cdots & x_N^m \end{bmatrix}_{N \times m}, \quad (1)$$

where X is the population matrix, X_i is the i th population member, x_i^d is the value of d th problem variable suggested by the i th population member, N is the number of population members, and m is the number of problem variables.

In each iteration of the algorithm, the objective function of the problem is evaluated based on the suggested values of variables provided by each population member. Therefore, the values of the objective function are specified as a vector using Eq. (2).

$$OF(X) = \begin{bmatrix} OF_1 = OF(X_1) \\ \vdots \\ OF_i = OF(X_i) \\ \vdots \\ OF_N = OF(X_N) \end{bmatrix}_{N \times 1}, \quad (2)$$

where $OF(X)$ is the objective function vector and OF_i is the objective function value based on the solution suggested by i th population member.

In the proposed AMBO algorithm, population members are updated in two stages. In the first stage, each member of the population is updated based on the position of different members of the population in the search space. The important point in this process is that the new position is acceptable to a population member if it improves the value of the objective function. Otherwise, the update is not acceptable and the member remains in its previous position. The first stage is simulated using Eqs. (3) to (5).

$$N_s = \text{round} \left(N \times \left(1 - \frac{t}{T} \right) \right) \quad (3)$$

$$x_i^d = \begin{cases} x_i^d + r(x_i^d - x_j^d), & OF_i < OF_j \\ x_i^d + r(x_j^d - x_i^d), & \text{else} \end{cases} \quad \& j = 1 \text{ to } N_s \quad (4)$$

$$X_i = \begin{cases} X'_i, & OF'_i < OF_i \\ X_i, & \text{else,} \end{cases} \quad (5)$$

where N_s is the number of selected members to lead the population members, t denotes the algorithm replication counter, T is the maximum replication of the algorithm, x_i^d is the new value for d th problem variable, r is a random number in the interval $[0, 1]$, X'_i is the new position of i th population member based on the first stage, and OF'_i is its objective function value.

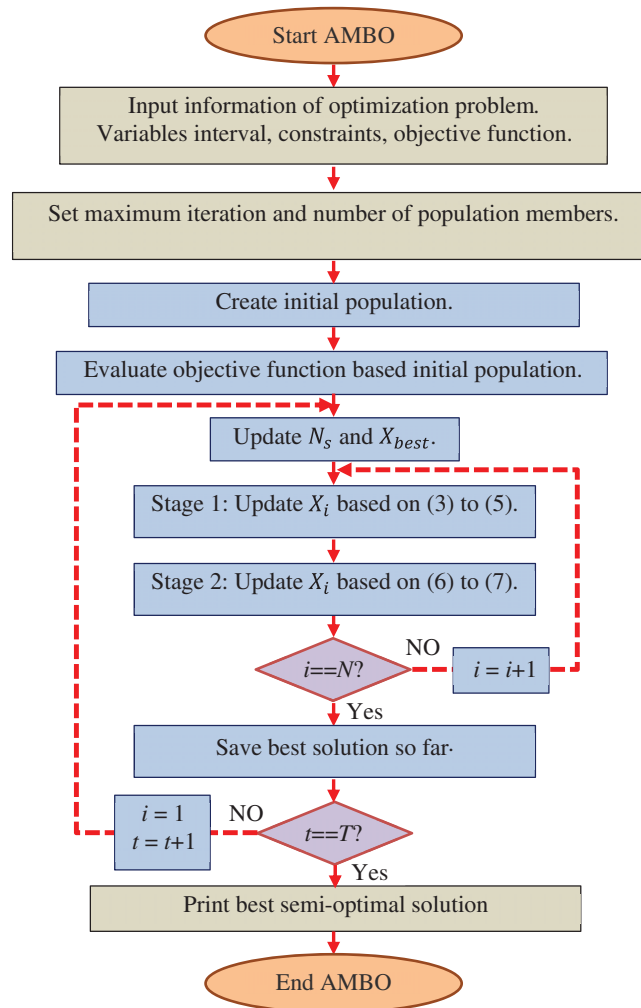


Figure 1: Flowchart of AMBO

In the second stage, the population matrix is updated based on the best member. In this stage, similar to the first stage, the new position is acceptable to a population member if it improves the objective function. The second stage of AMBO is simulated in Eqs. (6) and (7).

$$x_i''^d = x_i^d + r(x_{best}^d - x_i^d) \quad (6)$$

$$X_i = \begin{cases} X''_i, & OF''_i < OF_i \\ X_i, & \text{else,} \end{cases} \quad (7)$$

where $x_i''^d$ is the new value for d th problem variable. x_{best}^d is the best member, which achieves the best objective function value, X''_i is the new position of i th population member based on the second stage, and OF''_i is its corresponding objective function value.

For each iteration, the population matrix is updated through these two steps and this process is repeated until the algorithm stops. At the end of the algorithm iterations, AMBO provides the best obtained quasi-optimal solution to the optimization problem. The implementation process of the proposed algorithm in an optimization problem is shown as a flowchart in Fig. 1. The pseudo-code of the proposed AMBO algorithm is also presented in Algorithm 1.

Algorithm 1: Pseudo-code of the AMBO

Start AMBO.

1. Input optimization problem information.
 2. Set N and T .
 3. Create initial population matrix.
 4. Evaluate objective function.
 5. for $t = 1:T$
 6. Update N_s and X_{best} (best member)
 7. for $i = 1:N$
 8. Stage 1
 9. Update i 'th member based on (3) to (5).
 10. Stage 2
 11. Update i 'th member based on (6) to (7).
 12. End for $i = 1:N$
 13. Save the best solution so far.
 14. End for $t = 1:T$
 15. Output the best quasi-optimal solution.
- End AMBO.
-

3 Simulation Study

In this section, the ability of AMBO for solving optimization problems and providing quasi-optimal solutions is evaluated. For this purpose, AMBO is implemented on a set of twenty-three standard objective functions from three different types including unimodal, high-dimensional multimodal, and fixed-dimensional multimodal functions. Complete information of these objective functions is provided in the Appendix (Tabs. A1 to A3).

3.1 Experimental Setup

In order to analyze the performance of the proposed AMBO in providing the quasi-optimal solution, AMBO is compared with eight other optimization algorithms namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Teaching Learning-Based Optimization (TLBO), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Marine Predators Algorithm (MPA), and Tunicate Swarm Algorithm (TSA).

To solve each of the objective functions, 20 independent runs of the proposed AMBO has been performed, where each run includes 1000 iterations. The average (Ave) and standard deviation (std) of the best solutions have been used to present the results of optimization for the objective functions.

3.2 Evaluation Results for Objective Functions

The objective functions F_1 to F_7 belong to unimodal category. These functions have been selected to evaluate the performance of the optimization algorithms. The optimization results for these objective functions using the proposed AMBO and eight other optimization algorithms are presented in [Tab. 1](#). According to this table, AMBO is the best optimizer for F_1 , F_2 , F_3 , F_4 , and F_6 objective functions.

Table 1: Results of applying optimization algorithms on unimodal objective functions

	AMBO	MPA	TSA	GOA	GWO	TLBO	GSA	PSO	GA	
F_1	Ave	0	3.2715E-21	7.71E-38	2.1741E-09	1.09E-58	8.3373E-60	2.0255E-17	1.7740E-05	13.2405
	Std	0	4.6153E-21	7.00E-21	7.3985E-25	5.1413E-74	4.9436E-76	1.1369E-32	6.4396E-21	4.7664E-15
F_2	Ave	0	1.57E-12	8.48E-39	0.5462	1.2952E-34	7.1704E-35	2.3702E-08	0.3411	2.4794
	Std	0	1.42E-12	5.92E-41	1.7377E-16	1.9127E-50	6.6936E-50	5.1789E-24	7.4476E-17	2.2342E-15
F_3	Ave	0	0.0864	1.15E-21	1.7634E-08	7.4091E-15	2.7531E-15	279.3439	589.4920	1536.8963
	Std	0	0.1444	6.70E-21	1.0357E-23	5.6446E-30	2.6459E-31	1.2075E-13	7.1179E-13	6.6095E-13
F_4	Ave	0	2.6E-08	1.33E-23	2.9009E-05	1.2599E-14	9.4199E-15	3.2547E-09	3.9634	2.0942
	Std	0	9.25E-09	1.15E-22	1.2121E-20	1.0583E-29	2.1167E-30	2.0346E-24	1.9860E-16	2.2342E-15
F_5	Ave	20.1578	46.049	28.8615	41.7767	29.8607	146.4564	36.10695	50.26245	310.4273
	Std	2.22E-14	0.4219	4.76E-03	2.5421E-14	6.95E-13	1.9065E-14	3.0982E-14	1.5888E-14	2.0972E-13
F_6	Ave	0	0.398	7.10E-21	1.6085E-09	0.6423	0.4435	0	20.25	14.55
	Std	0	0.1914	1.12E-25	4.6240E-25	6.2063E-17	4.2203E-16	0	0	3.1776E-15
F_7	Ave	1.41E-05	0.0018	3.72E-04	0.0205	0.0008	0.0017	0.0206	0.1134	5.6799E-03
	Std	6.06E-21	0.0010	5.09E-05	1.5515E-18	7.2730E-20	3.87896E-19	2.7152E-18	4.3444E-17	7.7579E-19

Six high-dimensional multimodal objective functions F_8 to F_{13} have been selected to evaluate the performance of the optimization algorithms in providing a suitable quasi-optimal solution. [Tab. 2](#) shows the optimization results for these objective functions using AMBO and eight other optimization algorithms. AMBO is the best optimizer for F_8 , F_9 , and F_{10} objective functions.

The third group of objective functions, including F_{14} to F_{23} , is selected from the fixed-dimensional multimodal type. The results of the implementation of the proposed algorithm and eight other optimization algorithms on these objective functions are presented in [Tab. 3](#). The results in this table show that AMBO is the best optimizer for F_{15} , F_{16} , and F_{17} target functions.

Table 2: Results of applying optimization algorithms on High-dimensional multimodal objective functions

	AMBO	MPA	TSA	GOA	GWO	TLBO	GSA	PSO	GA	
F ₈	Ave	-9164.21	-3594.16321	-5740.3388	-1663.9782	-5885.1172	-7408.6107	-2849.0724	-6908.6558	-8184.4142
	Std	2.03E-13	811.32651	41.5	716.3492	467.5138	513.5784	264.3516	625.6248	833.2165
F ₉	Ave	0	140.1238	5.70E-03	4.2011	8.5265E-15	10.2485	16.2675	57.0613	62.4114
	Std	0	26.3124	1.46E-03	4.3692E-15	5.6446E-30	5.5608E-15	3.1776E-15	6.3552E-15	2.5421E-14
F ₁₀	Ave	4.44E-15	9.6987E-12	9.80E-14	0.3293	1.7053E-14	0.2757	3.5673E-09	2.1546	3.2218
	Std	2.65E-35	6.1325E-12	4.51E-12	1.9860E-16	2.7517E-29	2.5641E-15	3.6992E-25	7.9441E-16	5.1636E-15
F ₁₁	Ave	0	0	1.00E-07	0.1189	0.0037	0.6082	3.7375	0.0462	1.2302
	Std	0	0	7.46E-07	8.9991E-17	1.2606E-18	1.9860E-16	2.7804E-15	3.1031E-18	8.4406E-16
F ₁₂	Ave	5.13E-07	0.0851	0.0368	1.7414	0.0372	0.0203	0.0362	0.4806	0.0470
	Std	4.74E-22	0.0052	1.5461E-02	8.1347E-12	4.3444E-17	7.7579E-19	6.2063E-18	1.8619E-16	4.6547E-18
F ₁₃	Ave	9.68E-06	0.4901	2.9575	0.3456	0.5763	0.3293	0.0020	0.5084	1.2085
	Std	4.65E-18	0.1932	1.5682E-12	3.25391E-12	2.4825E-16	2.1101E-16	4.2617E-14	4.9650E-17	3.2272E-16

Table 3: Results of applying optimization algorithms on Fixed-dimensional multimodal objective functions

	AMBO	MPA	TSA	GOA	GWO	TLBO	GSA	PSO	GA	
F ₁₄	Ave	0.998004	0.9980	1.9923	0.9980	3.7408	2.2721	3.5913	2.1735	0.9986
	Std	2.23E-16	4.2735E-16	2.6548E-07	9.4336E-16	6.4545E-15	1.9860E-16	7.9441E-16	7.9441E-16	1.5640E-15
F ₁₅	Ave	0.000307	0.0030	0.0004	0.0049	0.0063	0.0033	0.0024	0.0535	5.3952E-02
	Std	5.82E-18	4.0951E-15	9.0125E-04	3.4910E-18	1.1636E-18	1.2218E-17	2.9092E-19	3.8789E-19	7.0791E-18
F ₁₆	Ave	-1.03163	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Std	7.45E-16	4.4652E-16	2.6514E-16	9.9301E-16	3.9720E-16	1.4398E-15	5.9580E-16	3.4755E-16	7.9441E-16
F ₁₇	Ave	0.397887	0.3979	0.3991	0.4047	0.3978	0.3978	0.3978	0.7854	0.4369
	Std	4.97E-17	9.1235E-15	2.1596E-16	2.4825E-17	8.6888E-17	7.4476E-17	9.9301E-17	4.9650E-17	4.9650E-17
F ₁₈	Ave	3	3	3	3	3.0000	3.0009	3	3	4.3592
	Std	1.19E-15	1.9584E-15	2.6528E-15	5.6984E-15	2.0853E-15	1.5888E-15	6.9511E-16	3.6741E-15	5.9580E-16
F ₁₉	Ave	-3.86278	-3.8627	-3.8066	-3.8627	-3.8621	-3.8609	-3.8627	-3.8627	-3.85434
	Std	2.78E-15	4.2428E-15	2.6357E-15	3.1916E-15	2.4825E-15	7.3483E-15	8.3413E-15	8.9371E-15	9.9301E-17
F ₂₀	Ave	-3.322	-3.3211	-3.3206	-3.2424	-3.2523	-3.2014	-3.0396	-3.2619	-2.8239
	Std	3.28E-15	1.1421E-11	5.6918E-15	7.9441E-16	2.1846E-15	1.7874E-15	2.1846E-14	2.9790E-16	3.97205E-16
F ₂₁	Ave	-10.1532	-10.1532	-5.5021	-7.4016	-9.6452	-9.1746	-5.1486	-5.3891	-4.3040
	Std	6.75E-15	2.5361E-11	5.4615E-13	2.3819E-11	6.5538E-15	8.5399E-15	2.9790E-16	1.4895E-15	1.5888E-15
F ₂₂	Ave	-10.4029	-10.4029	-5.0625	-8.8165	-10.4025	-10.0389	-9.0239	-7.6323	-5.1174
	Std	2.78E-15	2.8154E-11	8.4637E-14	6.7524E-15	1.9860E-15	1.5292E-14	1.6484E-12	1.5888E-15	1.2909E-15
F ₂₃	Ave	-10.5364	-10.5364	-10.3613	-10.0003	-10.1302	-9.2905	-8.9045	-6.1648	-6.5621
	Std	6.75E-15	3.9861E-11	7.6492E-12	9.1357E-15	4.5678E-15	1.1916E-15	7.1497E-14	2.7804E-15	3.8727E-15

3.3 Discussion

Two important indicators for evaluating the performance of the optimization algorithms in solving optimization problems are the exploitation index and the exploration index.

Exploitation index indicates the ability of an optimization algorithm to provide a suitable quasi-optimal solution close to the global optimum for an optimization problem. An optimization algorithm must be able to provide a suitable solution at the end of its iterations, whatever this solution is closer to the global optimum, that algorithm has higher exploitation power. Unimodal

objective functions (F_1 to F_7) have only one optimal solution. Therefore, they are suitable for evaluating the exploitation power of the optimization algorithms. The optimization results of these objective functions show that AMBO has the best performance for all F_1 to F_7 functions and has higher exploitation power than the other eight optimization algorithms.

Exploration power means the ability of an algorithm to scan the search space properly and accurately. This indicator is especially important for optimization problems that have several local optimal solutions. Therefore, an algorithm that can provide a suitable quasi-optimal solution by accurately scanning the search space and passing through local optimal solutions has high exploration power. The multimodal objective functions of the second and third groups (F_8 to F_{13} and F_{14} to F_{23}) have several local optimums. Hence, they are suitable for evaluating exploration power. The optimization results of these objective functions presented in [Tabs. 3](#) and [4](#) indicate the high exploration ability of the proposed AMBO algorithm in solving this type of objective functions.

Table 4: Results of the algorithm sensitivity analysis to the maximum number of iterations

Objective function	Maximum number of iterations			
	100	500	800	1000
F ₁	8.4E-178	0	0	0
F ₂	2.41E-90	0	0	0
F ₃	1.02E-53	5.1E-299	0	0
F ₄	6.17E-76	0	0	0
F ₅	23.9188	22.79437	21.83367	21.25926
F ₆	0	0	0	0
F ₇	0.000166	3.87E-05	2.52E-05	2.31E-05
F ₈	-6388.5	-8931.93	-9124.55	-9416.32
F ₉	0	0	0	0
F ₁₀	4.44E-15	4.44E-15	4.44E-15	4.44E-15
F ₁₁	0	0	0	0
F ₁₂	0.000106	5.11E-06	2.18E-06	7.46E-07
F ₁₃	0.100696	0.110341	2.79E-05	9.21E-06
F ₁₄	0.998004	0.998004	0.998004	0.998004
F ₁₅	0.020363	0.000307	0.000307	0.000307
F ₁₆	-1.03163	-1.03163	-1.03163	-1.03163
F ₁₇	0.397888	0.397887	0.397887	0.397887
F ₁₈	3	3	3	3
F ₁₉	-3.86278	-3.86278	-3.86278	-3.86278
F ₂₀	-3.31399	-3.32199	-3.322	-3.322
F ₂₁	-10.1519	-10.1532	-10.1532	-10.1532
F ₂₂	-10.4023	-10.4029	-10.4029	-10.4029
F ₂₃	-10.5356	-10.5364	-10.5364	-10.5364

3.4 Sensitivity Analysis

In this section, the sensitivity of the AMBO to the two parameters of maximum number of iteration and number of population members is evaluated.

In order to evaluate the sensitivity of the proposed algorithm to the maximum number of iterations, the AMBO for the maximum number of iterations of 100, 500, 800 and 1000 has been implemented independently on all objective functions. The results of these implementations are presented in [Tab. 4](#). The results show that AMBO converges towards the optimal solution when the number of iterations is increased.

Also, in order to analyze the sensitivity of the proposed algorithm to the number of population members, the AMBO has been implemented independently for different populations with number of 20, 30, 50 and 80 members. The results of this simulation for the number of different members of the population are presented in [Tab. 5](#). It is analyzed from [Tab. 5](#) that the value of fitness function decreases when number of search agents increases.

Table 5: Results of the algorithm sensitivity analysis to the number of population members

Objective function	Number of population members			
	20	30	50	80
F ₁	0	0	0	0
F ₂	2.2E-277	0	0	0
F ₃	1.4E-248	0	0	0
F ₄	1.4E-235	0	0	0
F ₅	28.73851	22.06195	20.99871	19.90636
F ₆	0	0	0	0
F ₇	0.000123	2.84E-05	1.76E-05	1.73E-05
F ₈	-5685	-8221.25	-8342.12	-8785.23
F ₉	0	0	0	0
F ₁₀	4.44E-15	4.44E-15	4.44E-15	4.44E-15
F ₁₁	0	0	0	0
F ₁₂	0.091831	2.51E-06	1.25E-06	2.1E-07
F ₁₃	1.417566	0.143312	0.110816	0.097375
F ₁₄	0.998004	0.998004	0.998004	0.998004
F ₁₅	0.000308	0.000307	0.000307	0.000307
F ₁₆	-1.03163	-1.03163	-1.03163	-1.03163
F ₁₇	0.397888	0.397887	0.397887	0.397887
F ₁₈	3	3	3	3
F ₁₉	-3.86278	-3.86278	-3.86278	-3.86278
F ₂₀	-3.32199	-3.32199	-3.322	-3.322
F ₂₁	-10.1513	-10.1531	-10.1532	-10.1532
F ₂₂	-10.4028	-10.4029	-10.4029	-10.4029
F ₂₃	-10.5364	-10.5364	-10.5364	-10.5364

4 Statistical Testing

In this section, statistical analysis on the optimization results obtained by different optimization algorithms is presented. Although presenting the results in the form of average and standard deviation provides useful information about the performance of optimization algorithms, statistical analysis of the results is also important for better evaluation. For this purpose, the Wilcoxon rank sum test has been used as a non-parametric statistical test to specify the significance of the results. Wilcoxon rank test is applied to specify whether the results obtained by the proposed AMBO are different from other eight optimization algorithms in a statistically significant way.

A p -value specifies whether the given algorithm is statistically significant or not. If p -value of the given algorithm is less than 0.05, then the corresponding algorithm is statistically significant. The result of analysis using Wilcoxon rank test for the objective functions is shown in [Tab. 6](#). It is observed from [Tab. 6](#) that the p -value obtained from AMBO is much smaller than 0.05 for all the objective functions. Therefore, the proposed AMBO is statistically different from the other competitor algorithms.

Table 6: Obtained results from the Wilcoxon test ($p \geq 0.05$)

Compared algorithms	Unimodal	High-multimodal	Fixed-multimodal
AMBO vs. GA	0.015625	0.03125	0.001953
AMBO vs. PSO	0.015625	0.03125	0.003906
AMBO vs. GSA	0.03125	0.03125	0.019531
AMBO vs. TLBO	0.015625	0.03125	0.005859
AMBO vs. GWO	0.015625	0.03125	0.011719
AMBO vs. WOA	0.015625	0.03125	0.007813
AMBO vs. MPA	0.015625	0.03125	0.003906
AMBO vs. TSA	0.015625	0.0625	0.0625

5 Conclusions and Future Works

Optimization algorithms are one of the most effective and widely-used methods in solving optimization problems in various fields of science and engineering. In this paper, a new optimization algorithm called All Members-Based Optimizer (AMBO) was presented for solving optimization problems. The proposed AMBO was designed to use more information of different members of the population and to participate all members in updating the algorithm population. AMBO was mathematically modeled and implemented on a set of twenty-three standard objective functions. Also, in order to analyze the results, AMBO was compared with eight optimization algorithms including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Teaching Learning-Based Optimization (TLBO), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Marine Predators Algorithm (MPA), and Tunicate Swarm Algorithm (TSA).

The results of optimizing the unimodal objective functions showed that AMBO is more capable than other algorithms in solving such problems and therefore, it is superior considering the exploitation index. Also, the results of optimization for multimodal objective functions showed that AMBO with high exploration power is able to provide suitable quasi-optimal solutions for

this type of functions. Based on the simulation results, it can be concluded that the proposed algorithm has an acceptable ability to solve various optimization problems and is superior and much more competitive than other mentioned optimization algorithms.

The authors suggest some ideas and perspectives for future studies. Design of the binary version as well as the multi-objective version of the AMBO is an interesting potential for future investigations. Apart from this, implementing AMBO on various optimization problems and real-world optimization problems can be considered as some significant contributions, as well.

Funding Statement: PT (corresponding author) was supported by the Excellence project PřF UHK No. 2202/2020-2022 and Long-term development plan of UHK for year 2021, University of Hradec Králové, Czech Republic, <https://www.uhk.cz/en/faculty-of-science/about-faculty/official-board/internal-regulations-and-governing-acts/governing-acts/deans-decision/2020#grant-competition-of-fos-uhk-excellence-for-2020>.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. Dehghani, Z. Montazeri, A. Dehghani, O. P. Malik, R. Morales-Menendez *et al.*, “Binary spring search algorithm for solving various optimization problems,” *Applied Sciences*, vol. 11, no. 3, pp. 1286, 2021.
- [2] M. Dehghani, Z. Montazeri, A. Dehghani, H. Samet, C. Sotelo *et al.*, “DM: Dehghani method for modifying optimization algorithms,” *Applied Sciences*, vol. 10, no. 21, pp. 7683, 2020.
- [3] M. Dehghani, Z. Montazeri, G. Dhiman, O. Malik, R. Morales-Menendez *et al.*, “A spring search algorithm applied to engineering optimization problems,” *Applied Sciences*, vol. 10, no. 18, pp. 6173, 2020.
- [4] M. Dehghani, Z. Montazeri and O. P. Malik, “Energy commitment: A planning of energy carrier based on energy consumption,” *Electrical Engineering & Electromechanics*, no. 4, pp. 69–72, 2019.
- [5] M. Dehghani, M. Mardaneh, O. P. Malik, J. M. Guerrero, C. Sotelo *et al.*, “Genetic algorithm for energy commitment in a power system supplied by multiple energy carriers,” *Sustainability*, vol. 12, no. 23, pp. 10053, 2020.
- [6] M. Dehghani, M. Mardaneh, O. P. Malik, J. M. Guerrero, R. Morales-Menendez *et al.*, “Energy commitment for a power system supplied by multiple energy carriers system using following optimization algorithm,” *Applied Sciences*, vol. 10, no. 17, pp. 5862, 2020.
- [7] H. Rezk, A. Fathy, M. Aly and M. N. F. Ibrahim, “Energy management control strategy for renewable energy system based on spotted hyena optimizer,” *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2271–2281, 2021.
- [8] A. Ehsanifar, M. Dehghani and M. Allahbakhshi, “Calculating the leakage inductance for transformer inter-turn fault detection using finite element method,” in *Proc. of Iranian Conf. on Electrical Engineering*, Tehran, Iran, pp. 1372–1377, 2017.
- [9] M. Dehghani, Z. Montazeri and O. Malik, “Optimal sizing and placement of capacitor banks and distributed generation in distribution systems using spring search algorithm,” *International Journal of Emerging Electric Power Systems*, vol. 21, no. 1, pp. 20190217, 2020.
- [10] M. Dehghani, Z. Montazeri, O. P. Malik, K. Al-Haddad, J. M. Guerrero *et al.*, “A New methodology called dice game optimizer for capacitor placement in distribution systems,” *Electrical Engineering & Electromechanics*, no. 1, pp. 61–64, 2020.
- [11] S. Dehbozorgi, A. Ehsanifar, Z. Montazeri, M. Dehghani and A. Seifi, “Line loss reduction and voltage profile improvement in radial distribution networks using battery energy storage system,” in *Proc. of IEEE 4th Int. Conf. on Knowledge-Based Engineering and Innovation*, Tehran, Iran, pp. 0215–0219, 2017.

- [12] Z. Montazeri and T. Niknam, "Optimal utilization of electrical energy from power plants based on final energy consumption using gravitational search algorithm," *Electrical Engineering & Electromechanics*, no. 4, pp. 70–73, 2018.
- [13] M. Dehghani, M. Mardaneh, Z. Montazeri, A. Ehsanifar, M. J. Ebadi *et al.*, "Spring search algorithm for simultaneous placement of distributed generation and capacitors," *Electrical Engineering & Electromechanics*, no. 6, pp. 68–73, 2018.
- [14] M. Premkumar, R. Sowmya, P. Jangir, K. S. Nisar and M. Aldhaifallah, "A new metaheuristic optimization algorithms for brushless direct current wheel motor design problem," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2227–2242, 2021.
- [15] M. Dehghani, Z. Montazeri, A. Ehsanifar, A. R. Seifi, M. J. Ebadi *et al.*, "Planning of energy carriers based on final energy consumption using dynamic programming and particle swarm optimization," *Electrical Engineering & Electromechanics*, no. 5, pp. 62–71, 2018.
- [16] Z. Montazeri and T. Niknam, "Energy carriers management based on energy consumption," in *Proc. of IEEE 4th Int. Conf. on Knowledge-Based Engineering and Innovation*, Tehran, Iran, pp. 0539–0543, 2017.
- [17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of ICNN'95-Int. Conf. on Neural Networks*, Perth, WA, Australia, pp. 1942–1948, 1995.
- [18] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics*, vol. 146, pp. 311–351, 2019.
- [19] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. P. Malik, R. A. Ramirez-Mendoza *et al.*, "A new "doctor and patient" optimization algorithm: An application to energy commitment problem," *Applied Sciences*, vol. 10, no. 17, pp. 5791, 2020.
- [20] G. Dhiman, K. K. Singh, M. Soni, A. Nagar, M. Dehghani *et al.*, "MOSOA: A new multi-objective seagull optimization algorithm," *Expert Systems with Applications*, vol. 167, pp. 114150, 2020.
- [21] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [22] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Arxiv*, vol. 2, no. 2, pp. 78–84, 2010.
- [23] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," In: P. Melin, O. Castillo, L. T. Aguilar, J. Kacprzyk, and W. Pedrycz (eds.), *Foundations of Fuzzy Logic and Soft Computing. IFSA 2007. Lecture Notes in Computer Science*, vol. 4529, Berlin, Heidelberg: Springer, 2007. https://doi.org/10.1007/978-3-540-72950-1_77.
- [24] A. H. Gandomi, X.-S. Yang and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.
- [25] X. S. Yang, "A new metaheuristic bat-inspired algorithm," In: J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor (eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence*, vol. 284, Berlin, Heidelberg: Springer, 2010. https://doi.org/10.1007/978-3-642-12538-6_6.
- [26] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Engineering Software*, vol. 114, pp. 48–70, 2017.
- [27] A. Mucherino and O. Seref, "Monkey search: A novel metaheuristic search for global optimization," in *AIP Conf. Proc. American Institute of Physics*, vol. 953, no. 1, pp. 162–173, 2007.
- [28] M. Neshat, G. Sepidnam, M. Sargolzaei and A. N. Toosi, "Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 965–997, 2014.
- [29] M. Dehghani, Z. Montazeri, A. Dehghani, and O. P. Malik, "GO: Group optimization," *Gazi University Journal of Science*, vol. 33, pp. 381–392, 2020.
- [30] Y. Shiqin, J. Jianjun and Y. Guangxing, "A dolphin partner optimization," in *2009 WRI Global Congress on Intelligent Systems*, Xiamen, China, vol. 1, pp. 124–128, 2009.
- [31] R. Oftadeh, M. Mahjoob and M. Shariatpanahi, "A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search," *Computers & Mathematics with Applications*, vol. 60, no. 7, pp. 2087–2098, 2010.

- [32] H. Zhang and Q. Hui, "A coupled spring forced bat searching algorithm: Design, analysis and evaluation," in *2020 American Control Conf.*, Denver, CO, USA, pp. 5016–5021, 2020.
- [33] R. V. Rao, V. J. Savsani and D. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.
- [34] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [35] M. Dehghani, M. Mardaneh and O. Malik, "FOA: 'Following' optimization algorithm for solving power engineering optimization problems," *Journal of Operation and Automation in Power Engineering*, vol. 8, no. 1, pp. 57–64, 2020.
- [36] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [37] S. Saremi, S. Mirjalili and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
- [38] M. Dehghani, M. Mardaneh, O. P. Malik and S. M. NouraeiPour, "DTO: Donkey theorem optimization," in *Proc. of Iranian Conf. on Electrical Engineering*, Yazd, Iran, pp. 1855–1859, 2019.
- [39] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowledge-Based Systems*, vol. 159, pp. 20–50, 2018.
- [40] M. Dehghani, Z. Montazeri, A. Dehghani, R. R. Mendoza, H. Samet *et al.*, "MLO: Multi leader optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 364–373, 2020.
- [41] G. Dhiman, M. Garg, A. K. Nagar, V. Kumar and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *Journal of Ambient Intelligence and Humanized Computing*, 2020. <https://doi.org/10.1007/s12652-020-02580-0>.
- [42] H. Givi, M. Dehghani, Z. Montazeri, R. Morales-Menendez, R. A. Ramirez-Mendoza *et al.*, "GBUO: The good, the bad, and the ugly, optimizer," *Applied Sciences*, vol. 11, no. 5, pp. 2042, 2021.
- [43] M. Dehghani, Z. Montazeri, A. Dehghani and A. Seifi, "Spring search algorithm: A new meta-heuristic optimization algorithm inspired by hooke's law," in *Proc. of IEEE 4th Int. Conf. on Knowledge-Based Engineering and Innovation*, Tehran, Iran, pp. 0210–0214, 2017.
- [44] M. Dehghani and H. Samet, "Momentum search algorithm: A new meta-heuristic optimization algorithm inspired by momentum conservation law," *SN Applied Sciences*, vol. 2, no. 10, pp. 1–15, 2020.
- [45] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "GSA: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [46] O. K. Erol and I. Eksin, "A new optimization method: Big bang–big crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, 2006.
- [47] H. Shah-Hosseini, "Principal components analysis by the galaxy-based search algorithm: A novel meta-heuristic for continuous optimisation," *International Journal of Computational Science and Engineering*, vol. 6, no. 1–2, pp. 132–140, 2011.
- [48] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta Mechanica*, vol. 213, no. 3–4, pp. 267–289, 2010.
- [49] W. F. Sacco and C. Oliveira, "A new stochastic optimization algorithm based on a particle collision metaheuristic," in *6th World Congresses of Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 30 May–03 June, 2005.
- [50] P. J. Van Laarhoven and E. H. Aarts, "Simulated Annealing," in *Simulated Annealing: Theory and Applications*, Dordrecht, Netherlands: Springer, pp. 7–15, 1987.
- [51] M. Dehghani, Z. Montazeri, A. Dehghani, N. Nouri and A. Seifi, "BSSA: Binary spring search algorithm," in *Proc. of IEEE 4th Int. Conf. on Knowledge-Based Engineering and Innovation*, Tehran, Iran, pp. 0220–0224, 2017.
- [52] R. A. Formato, "Central force optimization: A new nature inspired computational framework for multidimensional search and optimization," *Nature Inspired Cooperative Strategies for Optimization*, Berlin, Heidelberg: Springer, pp. 221–238, 2008.

- [53] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: Ray optimization," *Computers & Structures*, vol. 112, pp. 283–294, 2012.
- [54] F. F. Moghaddam, R. F. Moghaddam and M. Cheriet, "Curved space optimization: A random search based on general relativity theory," Arxiv Preprint Arxiv: 1208.2214, 2012.
- [55] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.
- [56] H. Du, X. Wu and J. Zhuang, "Small-world optimization algorithm for function optimization," in *Proc. of Int. Conf. on Natural Computation*, Berlin, Heidelberg, pp. 264–273, 2006.
- [57] B. Alatas, "ACROA: Artificial chemical reaction optimization algorithm for global optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13170–13180, 2011.
- [58] A. Bose, T. Biswas and P. Kuila, "A novel genetic algorithm based scheduling for multi-core systems," in *Smart Innovations in Communication and Computational Sciences*, Singapore, Springer, pp. 45–54, 2019.
- [59] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.
- [60] L. J. Fogel, A. J. Owens and M. J. Walsh, "Artificial intelligence through simulated evolution," in *Evolutionary Computation: The Fossil Record*, IEEE, pp. 227–296, 1998. <https://doi.org/10.1109/9780470544600.ch7>.
- [61] R. G. Reynolds, "An introduction to cultural algorithms," in *Proc. of the third Annual Conf. on Evolutionary Programming*, river edge, NJ: World Scientific, vol. 24, pp. 131–139, 1994.
- [62] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.
- [63] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [64] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [65] G. Huang, "Artificial infectious disease optimization: A SEIQR epidemic dynamic model-based function optimization algorithm," *Swarm and Evolutionary Computation*, vol. 27, pp. 31–67, 2016.
- [66] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, no. 2, pp. 87–112, 1994.
- [67] W. Deng, H. Liu, J. Xu, H. Zhao and Y. Song, "An improved quantum-inspired differential evolution algorithm for deep belief network," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7319–7327, 2020.
- [68] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. Malik and V. Kumar, "Football game based optimization: An application to solve energy commitment problem," *International Journal of Intelligent Engineering and Systems*, vol. 13, pp. 514–523, 2020.
- [69] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero and G. Dhiman, "Darts game optimizer: A new optimization technique based on darts game," *International Journal of Intelligent Engineering and Systems*, vol. 13, pp. 286–294, 2020.
- [70] M. Dehghani, Z. Montazeri, O. P. Malik, A. Ehsanifar and A. Dehghani, "OSA: Orientation search algorithm," *international journal of industrial electronics,* *Control and Optimization*, vol. 2, no. 2, pp. 99–112, 2019.
- [71] M. Dehghani, Z. Montazeri, S. Saremi, A. Dehghani, O. P. Malik *et al.*, "HOGO: Hide objects game optimization," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 4, pp. 216–225, 2020.
- [72] D. Mohammad, M. Zeinab, O. P. Malik, H. Givi and J. M. Guerrero, "Shell game optimization: A novel game-based algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 3, pp. 246–255, 2020.
- [73] M. Dehghani, Z. Montazeri, O. P. Malik, G. Dhiman and V. Kumar, "BOSA: Binary orientation search algorithm," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 5306–5310, 2019.

[74] M. Dehghani, Z. Montazeri and O. P. Malik, “DGO: Dice game optimizer,” *Gazi University Journal of Science*, vol. 32, no. 3, pp. 871–882, 2019.

Appendix A. Information of objective function

Information on the twenty-three objective functions used in the simulations is presented in [Tabs. A1](#) to [A3](#).

Table A1: Unimodal objective functions

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

Table A2: High-dimensional objective functions

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

Table A3: Fixed-dimensional objective functions

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