Computers, Materials & Continua
DOI: 10.32604/cmc.2023.034695
Article

Tech Science Press

Billiards Optimization Algorithm: A New Game-Based Metaheuristic Approach

Hadi Givi[1,*] and Marie Hubálovská[2]

[1]Department of Electrical Engineering, Shahreza Campus, University of Isfahan, Iran
[2]Department of Technics, Faculty of Education, University of Hradec Kralove, Czech Republic
*Corresponding Author: Hadi Givi. Email: h.givi@shr.ui.ac.ir
Received: 25 July 2022; Accepted: 16 September 2022

Abstract: Metaheuristic algorithms are one of the most widely used stochastic approaches in solving optimization problems. In this paper, a new metaheuristic algorithm entitled Billiards Optimization Algorithm (BOA) is proposed and designed to be used in optimization applications. The fundamental inspiration in BOA design is the behavior of the players and the rules of the billiards game. Various steps of BOA are described and then its mathematical model is thoroughly explained. The efficiency of BOA in dealing with optimization problems is evaluated through optimizing twenty-three standard benchmark functions of different types including unimodal, high-dimensional multimodal, and fixed-dimensional multimodal functions. In order to analyze the quality of the results obtained by BOA, the performance of the proposed approach is compared with ten well-known algorithms. The simulation results show that BOA, with its high exploration and exploitation abilities, achieves an impressive performance in providing solutions to objective functions and is superior and far more competitive compared to the ten competitor algorithms.

Keywords: Optimization; game-based; billiards game; exploration; exploitation; metaheuristic algorithm

1 Introduction

Optimization is the process of finding the best solution among all feasible solutions for a problem. In fact, a problem that has more than one "feasible solution" is known as an optimization problem. With advances in science and technology, scientists encounter newer optimization applications that must be addressed using effective optimization methods. Deterministic and stochastic approaches are considered as effective tools for optimization problems. Deterministic approaches perform well in solving linear, convex, continuous, and derivative optimization problems. However, most real-world optimization applications have features such as nonlinear, non-convex, non-derivative, discrete search space, and high dimensions. The weakness of deterministic approaches to handle optimization problems with such features has led to the emergence of stochastic approaches as well as metaheuristic algorithms [1].

Metaheuristic algorithms are stochastic-based approaches that are able to provide optimal solutions to optimization problems through utilizing random search, random operators, and trial and error processes [2]. Due to the random search process of optimization algorithms, there is no guarantee that the solution achieved by these methods is exactly the global optimum [3]. However, these are acceptable candidate solutions known as quasi-optimal solutions [4]. The optimization process in metaheuristic algorithms begins with the production of a number of random candidate solutions. Then, during the iterations of the algorithm, the candidate solutions are improved so that at the end of the algorithm implementation, the best candidate solution is identified as the solution to the problem [5]. Exploration with the concept of global search and exploitation with the concept of local search are two key factors in the optimization process accomplished by metaheuristic algorithms. Balancing these two capabilities during the iterations of the algorithm plays an important role in the algorithm success to achieve the optimal solution.

Metaheuristic algorithms have been used in many scientific applications, including prediction of linear dynamical systems [6], solving systems of singular boundary value problems [7], solutions of Troesch's and Bratu's problems [8], deep learning algorithms [9–11], energy carriers [12,13], electrical engineering [14–19], protection [20], and energy management [21–24]. The design of metaheuristic algorithms is inspired by various natural phenomena, animal behavior in nature, physical laws, biological sciences, rules of games, and so on [25]. Particle Swarm Optimization (PSO) is a nature-inspired approach that is modeled based on the search behavior and strategy of fishes and birds for food [26]. Genetic Algorithm (GA) is a well-known metaheuristic algorithm based on mathematical modeling of biological sciences [27]. Gravitational Search Algorithm (GSA) is a metaheuristic algorithm designed based on modeling the laws of physics and the force of gravity [28]. Teaching-Learning Based Optimization (TLBO) is an optimization approach introduced based on the simulation of human activities [29]. Volleyball Premier League (VPL) is an optimization approach developed on the basis of modeling the rules and conditions governing volleyball league competitions [30].

The main research question is that considering various metaheuristic algorithms that have been investigated so far, is there still a need to introduce newer metaheuristic algorithms? In response to this question, No Free Lunch (NFL) theorem [31] explains that there is no guarantee for the same performance of a metaheuristic algorithm in all optimization problems. Based on the NFL theorem, a metaheuristic algorithm may provide very favorable results for a set of objective functions, but it may fail to address some other optimization problems. Hence, based on the NFL theorem, there is no assumption about the success or failure of implementing a metaheuristic algorithm on an optimization problem. The NFL theorem motivates researchers to provide suitable solutions for optimization problems through designing new metaheuristic algorithms.

The innovation and novelty of this paper is the design of a new game-based metaheuristic algorithm called Billiards Optimization Algorithm (BOA) to deal with optimization applications in different fields of science and technology. The contribution of this article is as follows. The basic idea of BOA is the rules of the billiards game and the strategy of the players during this game. Different steps of BOA implementation are described and mathematically modeled. The effectiveness of BOA in solving optimization problems is tested for twenty-three standard benchmark functions of unimodal and multimodal types. The results obtained by BOA are compared with the solutions achieved by ten well-known metaheuristic algorithms.

The rest of the article is as follows: In Section 2, literature review is provided. The proposed BOA is introduced in Section 3. Simulation studies and results are presented in Section 4. Finally, conclusions and some suggestions for future investigations are provided in Section 5.

2 Lecture Review

Metaheuristic algorithms can be divided into five groups based on the source of inspiration in their design: swarm-based, evolutionary-based, physics-based, game-based, and human-based algorithms.

Swarm-based algorithms are inspired by nature, and the swarming behaviors of animals, birds, and other living things. PSO is one of the most popular and one of the first optimization methods, which was developed based on modeling the movement of birds and fishes in search of food. The ability of ants to discover the shortest path between food sources and nests has been a central idea in Ant Colony Optimization (ACO) design [32]. The strategy of grey wolves in hunting and attacking prey is employed in the Grey Wolf Optimizer (GWO) design [33]. The humpback whale net-bubble hunting strategy has been a major source of inspiration in Whale Optimization Algorithm (WOA) [34]. The strategy of pelicans in hunting and trapping prey has been the main inspiration of Pelican Optimization Algorithm (POA) [35]. Some other swarm-based algorithms proposed so far are: Fennec Fox Optimization (FFO) [36], Reptile Search Algorithm (RSA) [37], Cat and Mouse based Optimizer (CMBO) [38], Good Bad Ugly Optimizer (GBUO) [39], Marine Predator Algorithm (MPA) [40], Tasmanian Devil Optimization (TDO) [41], Mutated Leader Algorithm (MLA) [42], Tunicate Search Algorithm (TSA) [43], Northern Goshawk Optimization (NGO) [44], Donkey Theorem Optimizer (DTO) [45], Rat Swarm Optimization (RSO) [46], All Members Based Optimizer (AMBO) [47], Red Fox Optimization (RFO) [48], Mixed Best Members Based Optimizer (MBMBO) [49], and Mixed Leader Based Optimizer (MLBO) [50].

Evolutionary-based algorithms are inspired by the concepts of natural selection, biology, and genetics. GA and Differential Evolution (DE) [51] are among the most widely used and well-known evolutionary algorithms, inspired by the process of reproduction, Darwin's theory of evolution, and random operators such as selection, crossover, and mutation. Some other evolutionary-based algorithms can be listed as: Average and Subtraction-Based Optimizer (ASBO) [52], Genetic Programming (GP) [53], Search Step Adjustment Based Algorithm (SSABA) [54], Evolutionary Programming (EP) [55], Selecting Some Variables to Update-Based Algorithm (SSVUBA) [56], and Artificial Immune System (AIS) technique [57].

Physics-based algorithms are arisen from simulating processes, laws, and concepts in physics. Simulated Annealing (SA) is one of the most famous optimization algorithms that is inspired by the physical process of melting and cooling metals known as annealing [58]. Simulation of the law of curvature and spring force was the main idea for designing Spring Search Algorithm (SSA) [59], simulation of momentum and collision of objects has been the main inspiration of Momentum Search Algorithm (MSA) [60], simulation of gravitational force and laws of motion between objects at different distances has been the main inspiration of GSA. Some other physics-based algorithms are: Multi-Verse Optimizer (MVO) [61], Binary Spring Search Algorithm (BSSA) [62], Equilibrium Optimizer (EO) [63], and Henry Gas Solubility Optimization (HGSO) [64].

Game-based algorithms are inspired by modeling the rules of various individual and team games, the behavior of players, coaches, and referees, as well as holding competitions. Simulating the organization of the football league as well as the behavior of the players, and the interactions of the clubs with each other has been the main idea in the design of Football Game Based Optimization (FGBO) [65]. The players' effort to put the puzzle pieces together has been the major inspiration in Puzzle Optimization Algorithm (POA) design [66]. Hide Object Game Optimizer (HOGO) is an optimization approach developed based on modeling players' behavior to find a hidden object in the hide object game [67]. Some other game-based algorithms are: VPL, Darts Game Optimizer

(DGO) [68], Binary Orientation Search algorithm (BOSA) [69], Shell Game Optimization (SGO) [70], Orientation Search algorithm (OSA) [71], and Ring Toss Game Based Optimizer (RTGBO) [72].

Human-based algorithms are inspired by human behaviors, the interactions of people in a community with each other, and the relationships between humans. TLBO is one of the most widely used optimization techniques that is designed based on modeling the space of a classroom with the presence of a teacher and a number of students in two phases of teaching and learning. Interactions between doctor and patients for examination, prevention, and treatment have been the main inspiration of Doctor and Patient Optimization (DPO) [73]. Modeling and following the people of a society by the leader in order to develop that society has been the source of inspiration in designing Following Optimization Algorithm (FOA) [74]. Some other human-based algorithms are: Teamwork Optimization Algorithm (TOA) [75], Group Optimization (GO) [76], Archery Algorithm (AA) [77], Poor and Rich Optimization (PRO) technique [78], and Skill Optimization Algorithm (SOA) [79].

Based on the best knowledge gained from the above literature review, the behavior of players in the game of billiards has not been used in the design of any metaheuristic algorithm so far. While billiards players try to place balls in pockets is an intelligence process that has a good potential to design an optimizer. In order to address this research gap, in this study, based on the mathematical modeling of this process, a new metaheuristic algorithm has been designed, which is discussed in the next section.

3 Billiards Optimization Algorithm

In this section, theory of Billiards Optimization Algorithm (BOA) is described and then its mathematical modeling for use in optimization applications is presented.

3.1 Inspiration of BOA

Billiards is a fascinating sport that has become a favorite of many people. However, different games are based on the rules and conditions of billiards. In general, in this game, players try to place the balls inside the pockets of the billiards table. The players' strategy to perform this activity is an intelligence behavior that has a potential to be designed as an optimizer. BOA is based on the simulation of players' behavior in this game, which is discussed below.

3.2 Initialization of BOA

The BOA approach is a population-based metaheuristic algorithm whose members are billiards balls. Each BOA member is a candidate solution that assigns values to problem variables based on its position in the search space. BOA members can be mathematically modeled using a matrix according to Eq. (1). BOA members are randomly initialized at the beginning of the algorithm implementation using Eq. (2).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m}, \tag{1}$$

$$X_i\colon\ x_{i,d} = lb_d + r \times (ub_d - lb_d), \tag{2}$$

Here, X is the population matrix of BOA, X_i indicates ith candidate solution, $x_{i,d}$ represents the value of dth variable proposed by ith population member, N is the number of BOA's members, m denotes the number of variables, r is a random number in interval $[0 - 1]$, lb_d and ub_d are the lower bound and upper bound of dth variable, respectively.

Given that each member of the population is a candidate solution to the problem, a value for the objective function is calculated for each member of the population. The values obtained for the objective function can be represented mathematically using a vector according to Eq. (3).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1}, \tag{3}$$

Here, F is the vector of objective function values and F_i represents the obtained value for the objective function based on ith population member.

Considering the values obtained for the objective function, the best value identifies the best member of the population. Given that in each iteration, the positions of the population members are updated and new values are evaluated for the objective function, the best member of the population must be updated in each iteration.

3.3 Mathematical Model of BOA

In BOA, members of the population are updated based on the players' behavior in placing balls in the pockets of the game table. Since the billiards table has 8 pockets, in each repetition, among the members of the population, 8 better members are considered as the pockets. The rest of the population is considered as balls that have to be moved towards these pockets. The player selects one of these pockets to hit each ball. The BOA design assumes that the pocket is chosen at random. In order to simulate this process, a proposed random position is first calculated according to Eq. (4) for each member of the population. Then, if the value of the objective function is improved in the new position, it replaces the previous position of that member according to Eq. (5).

$$X_i^{new}: x_{i,d}^{new} = x_{i,d} + r \times (P_{i,d} - I \times x_{i,d}) \tag{4}$$

$$X_i = \begin{cases} X_i^{new}, & F_i^{new} < F_i \\ X_i, & else, \end{cases} \tag{5}$$

Here, X_i^{new} is the new calculated status of ith population member, $x_{i,d}^{new}$ indicates its dth dimension, F_i^{new} represents its corresponding objective function value, P_i is the selected pocket for ith population member, $P_{i,d}$ denotes its dth dimension, r is a random number in interval $[0 - 1]$, and I is a random number, which is selected randomly from set of $\{1, 2\}$.

3.4 Repetition Process, Pseudocode, and Flowchart of BOA

After updating all members of the population, the first iteration of the BOA is completed. The algorithm then enters the next iteration based on the new positions obtained for the population members. The process of updating members of the BOA population is repeated until the end of the algorithm implementation according to Eqs. (4) and (5). After complete implementation of the BOA, the best candidate solution obtained during the algorithm iterations is the output, which is the solution to the problem. The BOA implementation steps are shown through a flowchart in Fig. 1 and the corresponding pseudo-code is presented in Algorithm 1.

Algorithm 1: Pseudo-code of Billiards Optimization Algorithm (BOA)

Start BOA.
1. Input the optimization problem information.
2. Set T (number of iterations) and N (number of population members).
3. Generate the initial position of the BOA members using Eq. (2). $x_{i,d} \leftarrow lb_d + r \times (ub_d - lb_d)$.
4. Evaluate the objective function.
5. **for** $t = 1$ to T **do**
7. Determine the position of the pockets based on eight better members.
8.   **for** $\mathrm{i} = 1$ to N **do**
9.   Select the target pocket at random among eight pockets.
10.   Calculate new status of ith candidate solution using Eq. (4). $x_{i,d}^{new} \leftarrow x_{i,d} + r \times (P_{i,d} - I \times x_{i,d})$;
11.   Update ith candidate solution using Eq. (5). $X_i \leftarrow \begin{cases} X_i^{new}, & F_i^{new} < F_i \\ X_i, & else; \end{cases}$
12.   **end for**
13.   Save the best candidate solution so far.
14. **end for**
15. Output the best obtained solution.

End BOA.

3.5 Computational Complexity

In this subsection, the computational complexity of the BOA is analyzed. BOA initialization has a computational complexity of $O(Nm)$, where N is the BOA population size and m is the number of problem variables. The process of updating the algorithm population has a computational complexity equal to $O(NmT)$, where T is the number of algorithm iterations. Accordingly, the total computational complexity of BOA is equal to $O(Nm(1 + T))$.

4 Simulation Studies and Results

In this section, the performance of BOA in solving optimization problems is evaluated. To this end, BOA has been implemented on twenty-three standard objective functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types. Comprehensive information of these test functions including the mathematical model, dimensions, constraints, optimal value of the objective function, and other details has been presented in [80]. The capability of BOA in providing solutions to these functions has been compared with the performance of ten well-known metaheuristic algorithms including GA, WOA, RSA, PSO, GWO, MPA, MVO, GSA, TLBO, and TSA. The proposed BOA approach and ten competitor algorithms are each implemented on F1 to F23 functions in twenty independent implementations, while each implementation contains 1000 iterations. The optimization results are presented using six statistical indicators of mean, best, worst, standard deviation (std), median, and rank.

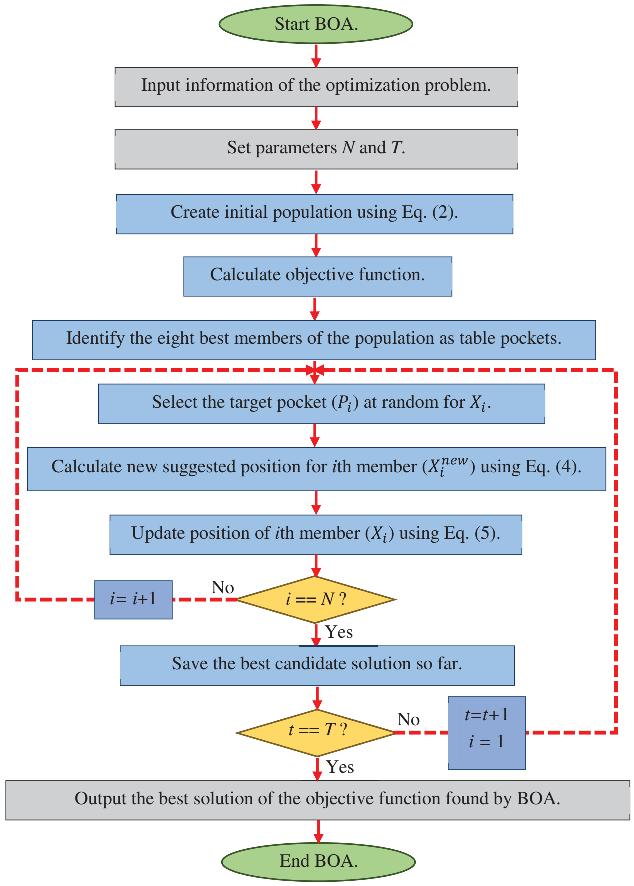

Figure 1: Flowchart of Billiards Optimization Algorithm (BOA)

4.1 Evaluation of Unimodal Functions

Unimodal functions F1 to F7 are suitable for testing local search capability or the exploitation of metaheuristic algorithms since they have only one optimal solution. The results of optimization of F1 to F7 functions using BOA and the mentioned competitor algorithms are reported in Table 1. The results show that BOA has provided the global optimum for F1 to F6 functions. Also, in solving F7, the BOA approach is the first best optimizer for this function. Based on the presented simulation results, it is clear that BOA has a high capability to solve unimodal problems and its performance is superior to competitor algorithms.

Table 1: Evaluation results for unimodal functions

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RSA	BOA
F_1	mean	34.5437	0.011351	9.92E-17	4.15E-75	8.45E-59	0.141392	6.1E-150	1.26E-47	1.88E-49	6.46E-84	0
	best	21.8428	2.18E-05	5.36E-17	1.46E-76	5.78E-61	0.085887	1.1E-172	1.9E-50	2.3E-52	9.43E-93	0
	worst	48.27454	0.097758	2.23E-16	2.18E-74	6.53E-58	0.229047	1.2E-148	9.02E-47	1.75E-48	1.19E-82	0
	std	8.103596	0.021904	4.27E-17	5.75E-75	1.47E-58	0.035688	2.7E-149	2.43E-47	4.01E-49	2.64E-83	0
	median	33.39374	0.004052	8.44E-17	1.9E-75	4.14E-59	0.134685	2.2E-160	7.09E-49	1.92E-50	3.69E-88	0
	rank	11	9	8	4	5	10	2	7	6	3	1
F_2	mean	2.893674	1.469507	5.21E-08	3.96E-39	9.56E-35	0.256823	4.8E-104	1.05E-28	5.4E-28	6.78E-46	0

(Continued)

Table 1: Continued

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RSA	BOA
	best	1.771522	0.129615	3.14E-08	8.03E-40	1.9E-36	0.162297	3.4E-117	1.05E-30	1.58E-29	4.79E-49	0
	worst	4.187515	10.91586	7.28E-08	1.2E-38	2.83E-34	0.393994	9.4E-103	6.38E-28	2.66E-27	5.43E-45	0
	std	0.688131	2.427296	1.18E-08	3.01E-39	8.16E-35	0.061766	2.1E-103	1.6E-28	7.3E-28	1.51E-45	0
	median	2.928265	0.819186	4.9E-08	2.99E-39	7.25E-35	0.25831	8.9E-108	3.99E-29	2.14E-28	3.56E-47	0
	rank	11	10	8	4	5	9	2	6	7	3	1
F_3	mean	2151.287	874.6891	474.5464	1.19E-24	6.36E-15	13.69546	19397.34	1.96E-12	2.7E-12	4.76E-58	0
	best	1306.053	38.39523	191.6011	8.36E-29	7.31E-19	6.427129	1155.268	2.74E-17	1.58E-21	1.19E-69	0
	worst	3690.226	5365.03	1028.324	1.56E-23	5.42E-14	23.71885	46521.37	2.32E-11	2.68E-11	5.35E-57	0
	std	651.9863	1532.811	210.8921	3.52E-24	1.38E-14	5.422219	11275.71	5.22E-12	7.22E-12	1.3E-57	0
	median	2013.684	279.0787	413.2552	3.35E-26	1.58E-16	12.17659	22075.51	9.99E-14	1.21E-13	1.49E-61	0
	rank	10	9	8	3	4	7	11	5	6	2	1
F_4	mean	3.182379	6.409232	1.347784	4.73E-30	1.34E-14	0.575497	45.73347	0.006311	3.38E-19	1.34E-35	0
	best	2.460207	2.625176	1.93E-08	8.45E-32	7.64E-16	0.203873	0.047089	6.36E-06	3.7E-20	3.83E-40	0
	worst	4.320177	9.826729	3.852453	2.31E-29	1.1E-13	0.98408	88.30133	0.074488	8.65E-19	1.66E-34	0
	std	0.440358	2.122281	1.08719	5.62E-30	2.5E-14	0.180899	32.22163	0.016314	2.17E-19	3.82E-35	0
	median	3.158793	6.156931	1.060589	2.21E-30	4.31E-15	0.59779	42.59176	0.001452	2.94E-19	2.7E-37	0
	rank	9	10	8	3	5	7	11	6	4	2	1
F_5	mean	512.3849	4685.587	26.4173	26.85637	26.80292	237.6288	27.19238	28.27683	23.63643	27.45887	0
	best	227.6302	6.358709	25.84867	25.97881	25.29612	26.95008	26.45993	26.48034	22.44977	26.21217	0
	worst	1904.539	90133.63	27.6268	28.55723	27.93213	1709.385	28.5018	28.86228	24.27026	28.59278	0
	std	355.6761	20115.89	0.455677	0.70839	0.739022	398.9057	0.541389	0.706209	0.4456	0.72896	0
	median	440.5026	81.59334	26.32633	26.69676	27.11089	63.20481	27.04987	28.63414	23.65578	27.18532	0
	rank	10	11	3	5	4	9	6	8	2	7	1
F_6	mean	34.86707	0.060897	1.25E-16	1.078098	0.705754	0.136757	0.077372	3.848483	1.86E-09	1.54416	0
	best	16.97404	9.5E-06	4.71E-17	0.54459	1.77E-05	0.067418	0.009203	2.589849	1.03E-09	0.862897	0
	worst	73.13031	0.835192	4.43E-16	1.595351	1.724524	0.237401	0.713129	4.796125	4.73E-09	2.393213	0
	std	17.35489	0.184214	9.08E-17	0.298311	0.459725	0.03697	0.153834	0.592668	8.37E-10	0.399298	0
	median	31.4086	0.009074	8.57E-17	1.088354	0.739962	0.14036	0.036663	4.050477	1.71E-09	1.639428	0
	rank	11	4	2	8	7	6	5	10	3	9	1
F_7	mean	0.010645	0.164599	0.061373	0.001623	0.000817	0.009972	0.001116	0.005855	0.000615	0.000401	3.56E-05
	best	0.003995	0.095988	0.019791	0.000455	0.00012	0.0056	7.13E-05	0.00218	0.000215	2.99E-05	5.88E-07
	worst	0.017568	0.285254	0.112113	0.003976	0.001565	0.018206	0.003835	0.013816	0.00151	0.000953	0.000105
	std	0.004044	0.052236	0.023729	0.001106	0.000407	0.003234	0.001128	0.003043	0.000314	0.000307	3.34E-05
	median	0.00971	0.153973	0.056558	0.001292	0.000774	0.009512	0.000817	0.004586	0.000571	0.000317	2.71E-05
	rank	9	11	10	6	4	8	5	7	3	2	1
sum rank		71	64	47	33	34	56	42	49	31	28	7
mean rank		10.14286	9.142857	6.714286	4.714286	4.857143	8	6	7	4.428571	4	1
total rank		11	10	7	4	5	9	6	8	3	2	1

4.2 *Evaluation of High-Dimensional Multimodal Functions*

High-dimensional multimodal functions F8 to F13 are appropriate for testing global search capability or exploration of metaheuristic algorithms due to having multiple local optimal solutions. The results of BOA implementation and competitor algorithms on F8 to F13 functions are presented in Table 2. The optimization results show that BOA has provided the global optimum for F9 and F11 functions. In solving the functions F8, F10, F12, and F13, the proposed approach is the first best optimizer. Based on the simulation results, it is concluded that BOA has a high ability to solve high-dimensional multimodal problems and its global search power and exploration capability is superior than the considered competitor algorithms.

Table 2: Evaluation results for high-dimensional multimodal functions

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RSA	BOA
F_8	mean	−8348.04	−7174.66	−2512.77	−5400.22	−6359.7	−7991.99	−10779.6	−5925.69	−9619.16	−7548.39	−12569.5
	best	−9571.05	−8778.83	−2969.06	−6984.4	−7834.43	−9028.56	−12569.5	−7583.05	−10355.8	−9259.4	−12569.5
	worst	−6569.32	−5206.21	−2015.67	−4465	−4357.11	−7097.08	−8026.53	−4728.78	−9025.09	−5383.42	−12569.5
	std	757.8547	895.3058	260.3632	685.2514	877.8481	632.0867	1645.717	716.9043	417.5433	1154.307	1.87E-12
	median	−8603.98	−7160.36	−2498.05	−5345.39	−6523.61	−7972.62	−10889.5	−5954.46	−9629.9	−7805.26	−12569.5
	rank	4	7	11	10	8	5	2	9	3	6	1
F_9	mean	58.86846	65.44741	29.15228	0	0.980934	103.7514	0	160.3345	0	0	0
	best	16.8933	27.85958	16.9143	0	0	80.65951	0	115.0893	0	0	0
	worst	95.10693	126.4306	46.76302	0	9.129372	141.3836	0	231.9054	0	0	0
	std	19.16566	23.37939	7.881373	0	2.70233	15.81913	0	31.28116	0	0	0
	median	58.96616	62.19288	26.86388	0	0	102.5599	0	165.7714	0	0	0
	rank	4	5	3	1	2	6	1	7	1	1	1
F_{10}	mean	3.546679	2.925435	8.19E-09	4.44E-15	1.58E-14	0.44077	3.2E-15	1.468278	4.09E-15	4.54E-13	8.88E-16
	best	2.698017	1.897756	6.07E-09	4.44E-15	1.15E-14	0.087453	8.88E-16	7.99E-15	8.88E-16	8.88E-16	8.88E-16
	worst	4.12547	4.878091	1.12E-08	4.44E-15	2.22E-14	2.141281	7.99E-15	3.546227	4.44E-15	9.03E-12	8.88E-16
	std	0.430288	0.938045	1.35E-09	0	2.96E-15	0.611269	2.09E-15	1.672925	1.09E-15	2.02E-12	0
	median	3.502708	2.817299	8.05E-09	4.44E-15	1.51E-14	0.133476	4.44E-15	2.22E-14	4.44E-15	8.88E-16	8.88E-16
	rank	11	10	7	4	5	8	2	9	3	6	1
F_{11}	mean	1.582293	0.320849	8.822246	0	0.005007	0.421495	0.009403	0.006064	0	0	0
	best	1.214307	0.006832	3.255758	0	0	0.253372	0	0	0	0	0
	worst	2.154375	2.005863	18.13903	0	0.047681	0.592456	0.081925	0.017241	0	0	0
	std	0.243938	0.466655	4.109461	0	0.011945	0.079737	0.023873	0.006718	0	0	0
	median	1.535265	0.099446	9.042558	0	0	0.423356	0	0.004493	0	0	0
	rank	7	5	8	1	2	6	4	3	1	1	1
F_{12}	mean	0.157037	1.423618	0.127676	0.072387	0.037371	0.960236	0.011252	6.958465	2.31E-10	0.069238	1.62E-32
	best	0.035385	4.47E-05	3.7E-19	0.03343	0.006546	0.000691	0.001059	0.404251	7.31E-11	0.012096	1.57E-32
	worst	0.359563	3.854956	0.635088	0.178688	0.073536	3.859424	0.083424	15.28632	5.8E-10	0.179779	2.54E-32
	std	0.091936	1.221873	0.213301	0.030545	0.019632	0.990186	0.01782	4.73628	1.16E-10	0.039794	2.16E-33
	median	0.147131	1.2003	1.07E-18	0.064612	0.036212	0.641166	0.00666	7.798794	1.93E-10	0.061529	1.57E-32
	rank	8	10	7	6	4	9	3	11	2	5	1
F_{13}	mean	2.614757	3.869959	0.100869	0.983304	0.633228	0.030971	0.261863	2.988147	0.001652	1.803955	7.65E-32
	best	1.149958	0.263545	4.86E-18	0.58094	0.113218	0.016912	0.049589	2.280729	1.28E-09	1.051985	1.35E-32
	worst	4.837813	17.42028	1.222664	1.450667	1.136582	0.066663	0.519796	5.169063	0.010987	2.793816	6.07E-31
	std	1.063186	4.538492	0.28389	0.239798	0.282332	0.012548	0.141363	0.695106	0.004024	0.41072	1.61E-31
	median	2.458018	1.703829	1.46E-17	0.929117	0.638573	0.028099	0.253052	2.728009	3.35E-09	1.694537	1.35E-32
	rank	9	11	4	7	6	3	5	10	2	8	1
sum rank		43	48	40	29	27	37	17	49	12	27	6
mean rank		7.166667	8	6.666667	4.833333	4.5	6.166667	2.833333	8.166667	2	4.5	1
total rank		8	9	7	5	4	6	3	10	2	4	1

4.3 Evaluation of Fixed-Dimensional Multimodal Functions

Fixed-dimensional multimodal functions F14 to F23 are suitable for evaluating the ability of metaheuristic algorithms to strike a balance between exploration and exploitation. The results obtained for F14 to F23 functions using BOA and the mentioned competitor algorithms are released in Table 3. BOA approach is the first best optimizer for F15 and F20 functions. Considering other functions of this group, where BOA achieves values similar to some competitor algorithms for the mean index, it provides better values for the std index. Analysis of the simulation results shows that BOA has a superior ability to balance exploration and exploitation characteristics compared to competitor algorithms.

Table 3: Evaluation results for fixed-dimensional multimodal functions

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RSA	BOA
F_{14}	mean	1.012829	3.645876	4.086478	1.196416	4.866265	0.998004	2.816492	8.893169	0.998004	4.823742	0.998004
	best	0.998004	0.998004	1.019228	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004
	worst	1.194013	11.7187	9.831309	2.982105	12.67051	0.998004	10.76318	12.67051	0.998004	12.67051	0.998004
	std	0.045865	3.732496	2.561839	0.610693	4.26894	3.16E-12	2.996068	4.787012	5.09E-17	3.851995	0
	median	0.998006	1.992031	3.970242	0.998004	2.982105	0.998004	1.495018	12.67051	0.998004	3.96825	0.998004
	rank	3	6	7	4	9	2	5	10	1	8	1
F_{15}	mean	0.00603	0.001432	0.002131	0.000453	0.005459	0.004558	0.00063	0.008535	0.000311	0.005053	0.000307
	best	0.000644	0.000307	0.001143	0.00031	0.000307	0.000308	0.000312	0.000308	0.000308	0.000307	0.000307
	worst	0.023479	0.019276	0.004428	0.001241	0.020363	0.020363	0.002178	0.020942	0.000316	0.022553	0.000307
	std	0.007244	0.00422	0.000696	0.0003	0.008834	0.008113	0.000457	0.009974	2.25E-06	0.008991	1.90E-19
	median	0.004113	0.000307	0.00203	0.000315	0.000308	0.000627	0.000481	0.001072	0.000311	0.000653	0.000307
	rank	10	5	6	3	9	7	4	11	2	8	1
F_{16}	mean	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03005	−1.03163	−0.99082	−1.03163
	best	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163
	worst	−1.03161	−1.03163	−1.03163	−1.03162	−1.03163	−1.03163	−1.03163	−1	−1.03163	−0.21546	−1.03163
	std	4.59E-06	1.35E-16	1.35E-16	1.59E-06	2.98E-09	3.18E-08	8.38E-11	0.007072	2.28E-16	0.1825	8.31E-17
	median	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163
	rank	7	1	1	6	4	5	3	8	2	9	1
F_{17}	mean	0.422023	0.65439	0.397887	0.403109	0.397898	0.397887	0.397888	0.397909	0.397887	0.397887	0.397887
	best	0.397887	0.397887	0.397887	0.397893	0.397887	0.397887	0.397887	0.397888	0.397887	0.397887	0.397887
	worst	0.832817	1.937365	0.397887	0.500697	0.3981	0.397888	0.39789	0.397946	0.397887	0.397887	0.397887
	std	0.09697	0.526896	0	0.02297	4.74E-05	7.33E-08	7.31E-07	1.7E-05	0	8.97E-16	0
	median	0.398304	0.397887	0.397887	0.397965	0.397888	0.397887	0.397887	0.397907	0.397887	0.397887	0.397887
	rank	8	9	1	7	5	3	4	6	1	2	1
F_{18}	mean	8.421245	3	3	3	7.050008	3	3.000018	14.20181	3	13.8	3
	best	3	3	3	3	3.000001	3	3	3.000001	3	3	3
	worst	30.31682	3	3	3.000002	84.00001	3.000001	3.000147	92.03579	3	84	3
	std	11.11524	2.82E-15	2.79E-15	4.58E-07	18.11215	3.24E-07	3.46E-05	26.59322	1E-15	20.3563	2.88E-16
	median	3.000586	3	3	3	3.000005	3	3.000005	3.000009	3	3	3
	rank	8	2	3	5	7	4	6	10	1	9	1
F_{19}	mean	−3.86265	−3.82413	−3.86278	−3.86051	−3.8621	−3.86278	−3.86068	−3.86225	−3.86278	−3.74604	−3.86278
	best	−3.86278	−3.86278	−3.86278	−3.8627	−3.86278	−3.86278	−3.86276	−3.86278	−3.86278	−3.86278	−3.86278
	worst	−3.86161	−3.08976	−3.86278	−3.85474	−3.8549	−3.86278	−3.8549	−3.85501	−3.86278	−3.08976	−3.86274
	std	0.000338	0.172852	1.92E-15	0.003385	0.002056	2.03E-07	0.002361	0.001753	2.28E-15	0.282864	9.02E-16
	median	−3.86278	−3.86278	−3.86278	−3.86238	−3.86277	−3.86278	−3.86162	−3.86273	−3.86278	−3.86278	−3.86278
	rank	4	9	1	8	6	3	7	5	2	10	1
F_{20}	mean	−3.19843	−3.30089	−3.322	−3.27123	−3.25578	−3.26246	−3.25729	−3.25227	−3.322	−3.19517	−3.322
	best	−3.31774	−3.322	−3.322	−3.31452	−3.32199	−3.32199	−3.32181	−3.32165	−3.322	−3.322	−3.322
	worst	−3.02507	−3.13764	−3.322	−3.15712	−3.13762	−3.20273	−3.08687	−3.08336	−3.322	−1.9217	−3.322
	std	0.081657	0.052988	3.67E-16	0.056572	0.06962	0.06108	0.091694	0.075807	3.81E-16	0.311345	1.41E-17
	median	−3.19362	−3.322	−3.322	−3.30495	−3.26241	−3.26254	−3.32111	−3.26131	−3.322	−3.322	−3.322
	rank	9	3	1	4	7	5	6	8	2	10	1
F_{21}	mean	−4.05229	−6.6523	−6.21031	−6.63717	−9.6474	−8.12535	−9.89304	−5.79327	−10.1532	−8.78928	−10.1532
	best	−7.88766	−10.1532	−10.1532	−9.30299	−10.153	−10.1532	−10.1529	−10.1034	−10.1532	−10.1532	−10.1532
	worst	−2.294	−2.63047	−2.68286	−4.07333	−5.09985	−5.05516	−5.05519	−2.62401	−10.1532	−0.88199	−10.1532
	std	2.072922	3.659234	3.702041	2.127264	1.555137	2.54809	1.138781	3.016364	2.41E-15	3.181731	2.07E-17

(Continued)

Table 3: Continued

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RSA	BOA
	median	−2.62469	−7.62699	−4.18158	−6.69275	−10.1527	−10.1531	−10.1516	−4.95475	−10.1532	−10.1524	−10.1532
	rank	11	7	9	8	4	6	3	10	2	5	1
F_{22}	mean	−6.76101	−8.1775	−9.6989	−8.06962	−10.4024	−8.04873	−8.69025	−7.01929	−10.4029	−8.05397	−10.4029
	best	−10.2388	−10.4029	−10.4029	−9.9566	−10.4029	−10.4029	−10.4029	−10.3661	−10.4029	−10.4029	−10.4029
	worst	−2.5174	−2.75193	−4.67391	−3.94552	−10.402	−2.76589	−2.76539	−2.68875	−10.4029	−0.90808	−10.4029
	std	3.154309	3.169789	1.757215	1.582401	0.000256	3.030069	2.723616	3.582683	3.65E-15	3.599306	1.61E-16
	median	−7.94704	−10.4029	−10.4029	−8.43317	−10.4025	−10.4029	−10.4006	−9.45916	−10.4029	−10.3962	−10.4029
	rank	11	6	4	7	3	9	5	10	2	8	1
F_{23}	mean	−8.18721	−5.6555	−10.5364	−8.03277	−10.536	−9.99793	−9.6541	−7.04655	−10.5364	−7.32853	−10.5364
	best	−10.3471	−10.5364	−10.5364	−9.58103	−10.5363	−10.5364	−10.5362	−10.5028	−10.5364	−10.5364	−10.5364
	worst	−2.66877	−2.42173	−10.5364	−3.95463	−10.5356	−5.12847	−3.83473	−2.41642	−10.5364	−1.85948	−10.5363
	std	2.385851	3.382098	1.58E-15	1.775707	0.000194	1.657273	2.150189	3.807051	2.31E-15	4.034066	1.95E-16
	median	−8.74676	−3.83543	−10.5364	−8.62434	−10.536	−10.5363	−10.5346	−10.1711	−10.5364	−10.508	−10.5364
	rank	6	10	1	7	3	4	5	9	2	8	1
sum rank		77	58	34	59	57	48	48	87	17	77	10
mean rank		7.7	5.8	3.4	5.9	5.7	4.8	4.8	8.7	1.7	7.7	1
total rank		8	6	3	7	5	4	4	9	2	8	1

The boxplot diagrams for the performance of BOA and competitor algorithms in optimizing functions F1 to F23 are shown in Fig. 2.

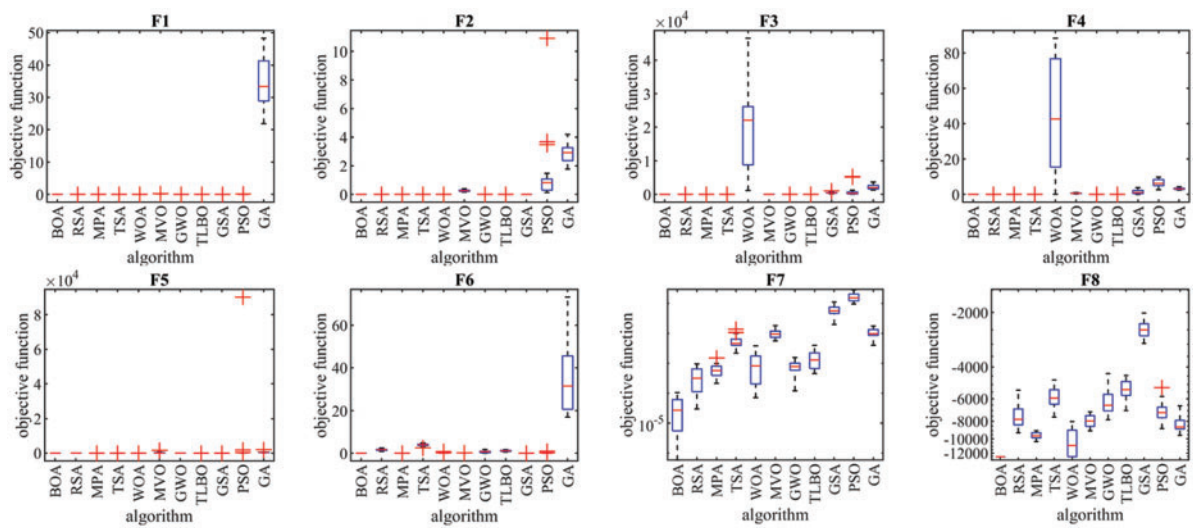

Figure 2: (Continued)

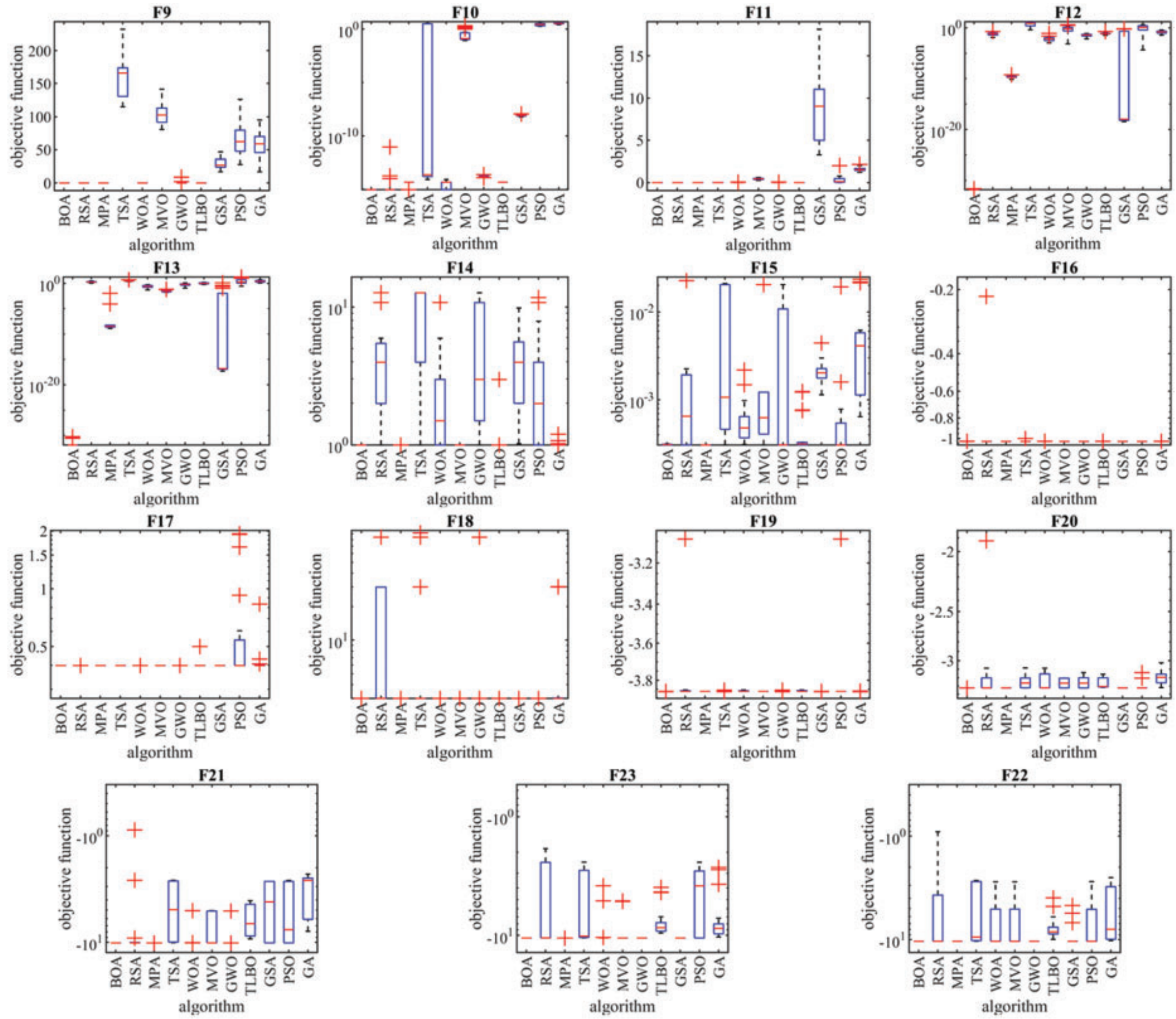

Figure 2: Boxplots of performances for Billiards Optimization Algorithm (BOA) and competitor algorithms

4.4 Statistical Analysis

In this subsection, a statistical analysis is presented to evaluate the BOA performance against the considered competitor algorithms. The Wilcoxon rank sum test [81], which is a non-parametric test, is used to accomplish this analysis. In this test, using an index called *p*-value, it is shown whether there is a significant difference between the mean of the two data samples.

The results of Wilcoxon rank sum test statistical analysis on the outputs of BOA and competitor algorithms are reported in Table 4. It can be seen from the simulation results that in cases where the *p*-value is less than 0.05, the proposed BOA has a significant statistical superiority over the corresponding competitor algorithm. Considering the obtained *p*-values and since the value of this index is less than 0.05 in all cases, it is concluded that BOA has a significant statistical superiority over all ten competitor algorithms in optimizing all three types of objective functions including unimodal, high-dimensional multimodal, and fixed-dimensional multimodal functions.

Table 4: *p*-values obtained by Wilcoxon sum rank test

Compared Algorithms	Unimodal	High-Multimodal	Fixed-Multimodal
BOA *vs.* GA	1.00E-24	1.95E-21	0.000691
BOA *vs.* PSO	1.00E-24	1.95E-21	1.93E-05
BOA *vs.* GSA	1.00E-24	1.95E-21	5.43E-11
BOA *vs.* TLBO	1.00E-24	6.91E-15	2.43E-05
BOA *vs.* GWO	1.00E-24	1.16E-16	2.11E-12
BOA *vs.* MPO	1.00E-24	1.95E-21	2.07E-18
BOA *vs.* WOA	1.00E-24	5.11E-14	6.38E-12
BOA *vs.* MPA	1.00E-24	8.64E-20	0.222474
BOA *vs.* TSA	1.00E-24	1.52E-14	1.43E-34
BOA *vs.* RSA	1.00E-24	5.12E-12	0.002713

5 Conclusions and Future Works

In this paper, a new game-based metaheuristic algorithm entitled Billiards Optimization Algorithm (BOA) was introduced. The main inspiration in BOA design is the behavior of the players in the billiards game to place the balls in the pockets of the game table. Different steps of BOA were described and mathematically modeled. The performance of the proposed BOA approach was evaluated in solving twenty-three standard benchmark functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types. The optimization results showed that BOA has a high ability in exploration, exploitation, and balancing these features during the optimization process. The performance of BOA was compared with ten well-known metaheuristic algorithms. Analysis of the simulation results showed that the proposed BOA approach has superior performance over the competitor algorithms by providing better results.

The authors also offer some proposals for future research. Design and development of binary and multi-objective versions of BOA is an attractive research topic. Application of BOA for optimization problems in various fields of science and real-world issues is also another research proposal for future investigations.

Funding Statement: The research and article are supported by Specific Research project 2022 Faculty of Education, University of Hradec Králové, Czech Republic.

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

References

[1] M. Dehghani, E. Trojovská and P. Trojovský, "A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process," *Scientific Reports*, vol. 12, no. 1, pp. 9924, 2022.

[2] M. Dehghani, Z. Montazeri, G. Dhiman, O. P. Malik, R. Morales-Menendez *et al.,* "A spring search algorithm applied to engineering optimization problems," *Applied Sciences*, vol. 10, no. 18, pp. 6173, 2020.

[3] 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, no. 4, pp. 114150, 2021.

[4] F. A. Zeidabadi, M. Dehghani and O. P. Malik, "RSLBO: Random selected leader based optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 529–538, 2021.

[5] 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.

[6] Z. Abo-Hammour, O. Alsmadi, S. Momani and O. Abu Arqub, "A genetic algorithm approach for prediction of linear dynamical systems," *Mathematical Problems in Engineering*, vol. 2013, no. 4, pp. 831657, 2013.

[7] Z. Abo-Hammour, O. Abu Arqub, O. Alsmadi, S. Momani and A. Alsaedi, "An optimization algorithm for solving systems of singular boundary value problems," *Applied Mathematics & Information Sciences*, vol. 8, no. 6, pp. 2809–2821, 2014.

[8] Z. Abo-Hammour, O. Abu Arqub, S. Momani and N. Shawagfeh, "Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm," *Discrete Dynamics in Nature and Society*, vol. 2014, pp. 401696, 2014.

[9] A. J. Moshayedi, A. S. Roy, A. Kolahdooz and Y. Shuxin, "Deep learning application pros and cons over algorithm," *EAI Endorsed Transactions on AI and Robotics*, vol. 1, pp. 1–13, 2022.

[10] A. J. Moshayedi, A. S. Roy, L. Liao and S. Li, "Raspberry Pi SCADA zonal based system for agricultural plant monitoring," in *Proc. of IEEE 6th Int. Conf. on Information Science and Control Engineering (ICISCE)*, Shanghai, China, pp. 427–433, 2019.

[11] A. J. Moshayedi, A. S. Roy, S. K. Sambo, Y. Zhong and L. Liao, "Review on: The service robot mathematical model," *EAI Endorsed Transactions on AI and Robotics*, vol. 1, pp. 1–19, 2022.

[12] 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.

[13] 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 (KBEI)*, Tehran, Iran, pp. 0539–0543, 2017.

[14] M. Dehghani, Z. Montazeri and O. P. 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.

[15] 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.

[16] 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 (KBEI)*, Tehran, Iran, pp. 0215–0219, 2017.

[17] 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.

[18] 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.

[19] 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.

[20] 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 (ICEE)*, Tehran, Iran, pp. 1372–1377, 2017.

[21] 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.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] 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.
[27] D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” *Machine Learning*, vol. 3, no. 2/3, pp. 95–99, 1988.
[28] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, “GSA: A gravitational search algorithm,” *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
[29] 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.
[30] R. Moghdani and K. Salimifard, “Volleyball premier league algorithm,” *Applied Soft Computing*, vol. 64, no. 5, pp. 161–185, 2018.
[31] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
[32] H. Givi, M. A. Noroozi, B. Vahidi, J. S. Moghani and M. A. V. Zand, “A novel approach for optimization of z-matrix building process using ant colony algorithm,” *Journal of Basic and Applied Scientific Research*, vol. 2, no. 9, pp. 8932–8937, 2012.
[33] S. Mirjalili, S. M. Mirjalili and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
[34] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, no. 12, pp. 51–67, 2016.
[35] P. Trojovský and M. Dehghani, “Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications,” *Sensors*, vol. 22, no. 3, pp. 855, 2022.
[36] E. Trojovská, M. Dehghani and P. Trojovský, “Fennec fox optimization: A new nature-inspired optimization algorithm,” *IEEE Access*, vol. 10, pp. 84417–84443, 2022.
[37] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem and A. H. Gandomi, “Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer,” *Expert Systems with Applications*, vol. 191, no. 11, pp. 116158, 2022.
[38] M. Dehghani, Š. Hubálovský and P. Trojovský, “Cat and mouse based optimizer: A new nature-inspired optimization algorithm,” *Sensors*, vol. 21, no. 15, pp. 5214, 2021.
[39] H. Givi, M. Dehghani, Z. Montazeri, R. Morales-Menendez, R. A. Ramirez-Mendoza *et al.,* “The good, the bad, and the ugly optimizer,” *Applied Sciences*, vol. 11, no. 5, pp. 2042, 2021.
[40] A. Faramarzi, M. Heidarinejad, S. Mirjalili and A. H. Gandomi, “Marine predators algorithm: A nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, no. 4, pp. 113377, 2020.
[41] M. Dehghani, Š. Hubálovský and P. Trojovský, “Tasmanian devil optimization: A new bio-inspired optimization algorithm for solving optimization algorithm,” *IEEE Access*, vol. 10, pp. 19599–19620, 2022.

[42] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, Z. Montazeri, P. Trojovský *et al.,* "MLA: A new mutated leader algorithm for solving optimization problems," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 5631–5649, 2022.
[43] S. Kaur, L. K. Awasthi, A. L. Sangal and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, no. 2, pp. 103541, 2020.
[44] M. Dehghani, Š. Hubálovský and P. Trojovský, "Northern Goshawk Optimization: A new swarm-based algorithm for solving optimization problems," *IEEE Access*, vol. 9, pp. 162059–162080, 2021.
[45] M. Dehghani, M. Mardaneh, O. P. Malik and S. M. NouraeiPour, "DTO: Donkey theorem optimization," in *Proc. of Iranian Conf. on Electrical Engineering (ICEE)*, Yazd, Iran, pp. 1855–1859, 2019.
[46] G. Dhiman, M. Garg, A. Nagar, V. Kumar and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 8, pp. 8457–8482, 2021.
[47] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, Z. Montazeri, P. Trojovský *et al.,* "AMBO: All members-based optimizer for solving optimization problems," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2905–2921, 2022.
[48] D. Połap and M. Woźniak, "Red fox optimization algorithm," *Expert Systems with Applications*, vol. 166, no. 10, pp. 114107, 2021.
[49] S. A. Doumari, F. A. Zeidabadi, M. Dehghani and O. P. Malik, "Mixed best members based optimizer for solving various optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 384–392, 2021.
[50] F. A. Zeidabadi, S. A. Doumari, M. Dehghani and O. P. Malik, "MLBO: Mixed leader based optimizer for solving optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 472–479, 2021.
[51] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
[52] M. Dehghani, Š. Hubálovský and P. Trojovský, "A new optimization algorithm based on average and subtraction of the best and worst members of the population for solving various optimization problems," *PeerJ Computer Science*, vol. 8, no. 10, pp. e910, 2022.
[53] W. Banzhaf, P. Nordin, R. E. Keller and F. D. Francone, "Genetic programming: An introduction: On the automatic evolution of computer programs and its applications," in *Library of Congress Cataloging-in-Publication Data*, 1st ed., vol. 27, San Francisco, CA, USA: Morgan Kaufmann Publishers, pp. 1–398, 1998.
[54] F. A. Zeidabadi, A. Dehghani, M. Dehghani, Z. Montazeri and Š. Hubálovský, "SSABA: Search step adjustment based algorithm," *Computers, Materials & Continua*, vol. 71, no. 3, pp. 4237–4256, 2022.
[55] D. B. Fogel, *Artificial Intelligence through Simulated Evolution*, Hoboken, New Jersey, U.S. United States: Wiley-IEEE Press, pp. 227–296, 1998.
[56] M. Dehghani and P. Trojovský, "Selecting some variables to update-based algorithm for solving optimization problems," *Sensors*, vol. 22, no. 5, pp. 1795, 2022.
[57] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.
[58] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[59] 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 (KBEI)*, Tehran, Iran, pp. 0210–0214, 2017.
[60] 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.

[61] S. Mirjalili, S. M. Mirjalili and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[62] 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 (KBEI)*, Tehran, Iran, pp. 0220–0224, 2017.

[63] A. Faramarzi, M. Heidarinejad, B. Stephens and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, pp. 105190, 2020.

[64] 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, no. 4, pp. 646–667, 2019.

[65] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. P. Malik and V. Kumar, "Football game based optimization: An application to solve energy commitment problem," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 514–523, 2020.

[66] F. A. Zeidabadi and M. Dehghani, "POA: Puzzle optimization algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 1, pp. 273–281, 2022.

[67] 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.

[68] 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, no. 5, pp. 286–294, 2020.

[69] 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.

[70] M. Dehghani, Z. Montazeri, 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.

[71] 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.

[72] S. A. Doumari, H. Givi, M. Dehghani and O. P. Malik, "Ring toss game-based optimization algorithm for solving various optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 545–554, 2021.

[73] 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.

[74] M. Dehghani, M. Mardaneh and O. P. 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.

[75] M. Dehghani and P. Trojovský, "Teamwork optimization algorithm: A new optimization approach for function minimization/maximization," *Sensors*, vol. 21, no. 13, pp. 4567, 2021.

[76] M. Dehghani, Z. Montazeri, A. Dehghani and O. P. Malik, "GO: Group optimization," *Gazi University Journal of Science*, vol. 33, no. 2, pp. 381–392, 2020.

[77] F. A. Zeidabadi, M. Dehghani, P. Trojovský, Š. Hubálovský, V. Leiva *et al.,* "Archery nlgorithm: A novel stochastic optimization algorithm for solving optimization problems," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 399–416, 2022.

[78] S. H. S. Moosavi and V. K. Bardsiri, "Poor and rich optimization algorithm: A new human-based and multi populations algorithm," *Engineering Applications of Artificial Intelligence*, vol. 86, no. 12, pp. 165–181, 2019.

[79] H. Givi and M. Hubalovska, "Skill optimization algorithm: A new human-based metaheuristic technique," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 179–202, 2023.
[80] X. Yao, Y. Liu and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary computation*, vol. 3, no. 2, pp. 82–102, 1999.
[81] F. Wilcoxon, "Individual comparisons by ranking methods," *Breakthroughs in Statistics*, vol. 1, no. 6, pp. 196–202, 1992.