Computers, Materials & Continua
DOI: 10.32604/cmc.2023.030379
Article

Tech Science Press

Skill Optimization Algorithm: A New Human-Based Metaheuristic Technique

Hadi Givi[1] and Marie Hubalovska[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: Marie Hubalovska. Email: marie.hubalovska@uhk.cz
Received: 24 March 2022; Accepted: 16 May 2022

Abstract: Metaheuristic algorithms are widely used in solving optimization problems. In this paper, a new metaheuristic algorithm called Skill Optimization Algorithm (SOA) is proposed to solve optimization problems. The fundamental inspiration in designing SOA is human efforts to acquire and improve skills. Various stages of SOA are mathematically modeled in two phases, including: (i) exploration, skill acquisition from experts and (ii) exploitation, skill improvement based on practice and individual effort. The efficiency of SOA in optimization applications is analyzed through testing this algorithm on a set of twenty-three standard benchmark functions of a variety of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types. The optimization results show that SOA, by balancing exploration and exploitation, is able to provide good performance and appropriate solutions for optimization problems. In addition, the performance of SOA in optimization is compared with ten metaheuristic algorithms to evaluate the quality of the results obtained by the proposed approach. Analysis and comparison of the obtained simulation results show that the proposed SOA has a superior performance over the considered algorithms and achieves much more competitive results.

Keywords: Optimization; human-based; skill; exploration; exploitation; metaheuristic algorithm

1 Introduction

With the advancement of science and technology, the need for optimization techniques to address the applications and challenges of optimization and achieve optimal designs has become increasingly apparent. The goal in optimization is to find the best solution to a multi-solution problem (known as the optimization problem) [1]. Metaheuristic algorithms are one of the most powerful tools for solving optimization problems. These techniques, which are considered as random optimization approaches, are based on trial-and-error methods, random search, and the use of random operators. Advantages such as simplicity of understanding, ease of implementation, being independent of the type of problem, no need to derivation process, and capability to be used in nonlinear, non-convex, and NP-hard problems have led to the widespread utilization of metaheuristic algorithms [2].

Metaheuristic algorithms have the same optimization process in that they start by generating a number of candidate solutions in the problem-solving space at random. These solutions are then improved in an iterative process. Finally, the best improved candidate solution is introduced as the solution to the problem [3]. Due to the random nature of the search process in metaheuristic algorithms, the solutions obtained by these techniques are not guaranteed to be the global optimal solution [4]. However, these solutions are close to the global optimum. Therefore, these acceptable solutions are known as quasi-optimal solutions. A quasi-optimal solution is, at best, equal to the global optimum [5]. Attempts to find better quasi-optimal solutions have led to the introduction and design of numerous metaheuristic algorithms [6–11]. Metaheuristic algorithms have wide applications in various fields of science, including: Energy carriers [12,13], feature selection problem [14], parameter estimation [15], electrical engineering [16–21], protection [22], and energy management [23–26].

The main idea in designing metaheuristic algorithms is inspired by nature, wildlife, behaviors of animal, birds, insects, aquatic animals in nature, physical concepts and laws, biological sciences, game rules, human behaviors, and any type of process that has an evolutionary nature. The birds' natural behaviors have been the main inspiration for Particle Swarm Optimization (PSO) algorithm [27]. Genetic Algorithm (GA) [28] was inspired by the reproduction process. The concept of gravity force in physics was utilized to obtain Gravitational Search Algorithm (GSA) [29]. The interactions between learners and teachers have been the main inspiration for Teaching-Learning Based Optimization (TLBO) [30], and Volleyball Premier League (VPL) method was inspired by the volleyball game [31].

The main research question is that despite the countless metaheuristic algorithms that have been designed so far, is there still a need to develop newer algorithms? There is the answer to this question in the concept of No Free Lunch (NFL) theorem [32], which states: The effective performance of an algorithm in solving a group of optimization problems does not guarantee the similar performance of that algorithm in solving other optimization problems. In fact, the NFL states that there is no presupposition as to whether or not an algorithm can be implemented successfully on an optimization problem, and that a particular algorithm cannot be claimed to be the best choice for all optimization applications. The NFL theorem motivates researchers to develop newer and more efficient algorithms in order to find better solutions for various optimization problems. The NFL theorem motivated the authors of this study to design a new algorithm to solve optimization problems more effectively to provide better quasi-optimal solutions.

Innovation and contribution of this paper is in designing a new metaheuristic algorithm called Skill Optimization Algorithm (SOA) to be utilized for solving optimization problems in various fields of science. The main inspiration employed in designing SOA is the human activities and efforts that seeks to acquire and improve skills. The steps of implementing SOA are mathematically modeled in two phases of exploration and exploitation, which are based on skill learning and individual practice, respectively. Twenty-three sets of standard benchmark functions of unimodal and multimodal types have been used for evaluating the ability of SOA to achieve optimal solutions. Moreover, the results obtained by SOA are compared with the performance of ten well-known metaheuristic algorithms.

The rest of this article is structured as follows: In Section 2, literature review is provided. The proposed SOA is introduced in Section 3. Simulation studies and results are presented in Section 4. Finally, conclusions and several suggestions for future studies are given in Section 5.

2 Lecture Review

Metaheuristic algorithms are divided into five groups based on the source of inspiration used in the 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 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 was a central idea in design of Ant Colony Optimization (ACO) technique [33]. The strategy of grey wolves in hunting and attacking prey was employed in the Grey Wolf Optimizer (GWO) design [34]. The humpback whale net-bubble hunting strategy was a major source of inspiration for designing Whale Optimization Algorithm (WOA) [35]. The strategy of pelicans in hunting and trapping prey was the main inspiration of Pelican Optimization Algorithm (POA) [36]. Some other swarm-based algorithms are: Cat and Mouse Based Optimizer (CMBO) [37], Good Bad Ugly Optimizer (GBUO) [38], Marine Predator Algorithm (MPA) [39], Tasmanian Devil Optimization (TDO) [40], Mutated Leader Algorithm (MLA) [41], Tunicate Search Algorithm (TSA) [42], Northern Goshawk Optimization (NGO) [43], Donkey Theorem Optimizer (DTO) [44], Rat Swarm Optimization (RSO) [45], All Members Based Optimizer (AMBO) [46], Red Fox Optimization (RFO) [47], Best Members Based Optimizer (MBMBO) [48], and Mixed Leader Based Optimizer (MLBO) [49].

Evolutionary-based algorithms are inspired by the concepts of natural selection, biology, and genetics. GA and Differential Evolution (DE) [50] 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 are: Average and Subtraction-Based Optimizer (ASBO) [51], Genetic Programming (GP) [52], Search Step Adjustment Based Algorithm (SSABA) [53], Evolutionary Programming (EP) [54], Selecting Some Variables to Update-Based Algorithm (SSVUBA) [55], and Artificial Immune System (AIS) [56].

Physics-based algorithms are inspired by 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 [57]. Simulation of the law of curvature and spring force was the basis of Spring Search Algorithm (SSA) [58]. Simulation of momentum and collision of objects was the main inspiration of Momentum Search Algorithm (MSA) [59]. Considering gravitational force and laws of motion between objects at different distances, GSA was proposed. Some of the other physics-based algorithms are: Multi-Verse Optimizer (MVO) [60], Binary Spring Search Algorithm (BSSA) [61], Equilibrium Optimizer (EO) [62], and Henry Gas Solubility Optimization (HGSO) [63].

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 was the main idea in the design of Football Game Based Optimization (FGBO) [64]. The players' effort to put the puzzle pieces together was the major inspiration in Puzzle Optimization Algorithm (POA) design [65]. 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 [66]. Some of other game-based algorithms are: VPL, Darts Game Optimizer (DGO) [67], Binary Orientation Search algorithm (BOSA) [68], Dice Game Optimizer (DGO) [69], Orientation Search algorithm (OSA) [70], and Ring Toss Game Based Optimizer (RTGBO) [71].

Human-based algorithms are designed based on 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 were the main inspiration of Doctor and Patient Optimization (DPO) [72]. Following the people of a society by the leader in order to develop that society was the source of inspiration in designing Following Optimization Algorithm (FOA) [73]. Some other human-based algorithms are: Teamwork Optimization Algorithm (TOA) [74], Group Optimization (GO) [75], Archery Algorithm (AA) [76], and Poor and Rich Optimization (PRO) algorithm [77].

3 Skill Optimization Algorithm

In this section, the proposed Skill Optimization Algorithm (SOA) is introduced and its mathematical modeling is presented. SOA is a population-based method, whose members are human beings who strive to learn and improve their skills. SOA population members are in fact candidate solutions to the given optimization problem. The positions of these members in the search space show the values of the problem decision variables. The positions of SOA members are randomly initialized at the start of the algorithm. The SOA population can be mathematically modeled using a matrix according to Eq. (1).

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

Here, X is the SOA's population matrix, X_i indicates the ith candidate solution, $x_{i,d}$ is the value of dth variable proposed by ith population member, N denotes the number of SOA's members, and m is the number of variables.

Each member of the population is a candidate solution to the problem. In other words, by placing each member in the problem variables, a value for the objective function is evaluated. Therefore, the values obtained for the objective function can be mathematically modeled using a vector according to Eq. (2).

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

Here, F is the vector of obtained values for the objective function and F_i represents the obtained value for the objective function based on ith candidate solution. Considering the values evaluated for the objective function, the best value identifies the best member and similarly, the worst value identifies the worst member. Given that in each iteration, the members of the population and the values of the objective function are updated, the best and worst members are also updated in each iteration.

The process of updating population members in SOA is performed in two phases of exploration and exploitation. The exploration phase is based on simulating the process of skill learning from an expert. The exploitation phase is based on simulating skill improvement through individual efforts and activities. In SOA design, the update process is performed in two phases of exploration with the

aim of global search in the problem-solving space, and the phase of exploitation with the aim of local search. In the exploration phase, SOA is designed so that SOA members move in the search space under the guidance of different members, and are prevented from moving only in the direction of the best member. This increases the algorithm's exploration power in accurately scanning the search space and identifying the original optimal area. On the other hand, in the exploitation phase, based on the local search near each member of the population, the algorithm is able to converge to better possible solutions.

3.1 Phase 1: Skill Acquisition from Experts (Exploration)

In the first phase, each SOA member strives to acquire a skill based on the guidance of a community expert member. The quality of each population member is commensurate with the objective function value obtained by that member of the population. The expert member for an SOA member is the member who has better conditions based on the value of the objective function. For each SOA member, all members that have a better objective function value than that member are considered as "experts set". Among the members of this set, one member is randomly selected as an expert to train the member in question. Therefore, the expert selected to guide the SOA member is not necessarily the best candidate solution. In fact, the best candidate solution is a permanent member of the experts set, for all SOA members. The expert member causes the member of the population to be guided to different positions in the search space by learning the skill, which means global search and exploration ability of the algorithm. The new position calculated for each member of the population is acceptable if it improves the value of the objective function. Therefore, the first phase of the update can be modeled according to the mentioned concepts using Eqs. (3) and (4).

$$X_i^{P1}: x_{i,d}^{P1} = x_{i,d} + r \times \left(E_{i,d} - I \times x_{i,d}\right),\ E_i = X_k,\\ \text{where } F_k < F_i \text{ and } k \text{ is randomly selected from } \{1, 2, \ldots, N\},\ k \neq i \tag{3}$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} < F_i \\ X_i, & else, \end{cases} \tag{4}$$

Here, X_i^{P1} is the new calculated status of ith candidate solution based on the first phase, $x_{i,d}^{P1}$ is its dth dimension, F_i^{P1} represents its objective function value, E_i is the selected expert to guide and train the ith population member, $E_{i,d}$ 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 $\{1, 2\}$.

3.2 Phase 2: Skill Improvement Based on Practice and Individual Effort (Exploitation)

In the second phase, each population member tries to improve the skills acquired in the first phase through individual practice and activity. In SOA, this concept is modeled as local search with the aim of increasing exploitation in such a way that each member, in the neighborhood of its position, seeks better conditions to improve the value of its objective function (which indicates the level of skill). Similar to previous phase, the newly calculated position in this phase is acceptable if it improves the objective function value. The concepts of this phase of SOA updating are mathematically modeled using Eqs. (5) and (6).

$$X_i^{P2}: x_{i,d}^{P2} = \begin{cases} x_{i,d} + \dfrac{1-2r}{t} \times x_{i,d}, & r < 0.5 \\ x_{i,d} + \dfrac{lb_j + r\left(ub_j - lb_j\right)}{t}, & else, \end{cases} \tag{5}$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} < F_i \\ X_i, & else, \end{cases} \tag{6}$$

Here, X_i^{P2} is the new calculated status of ith candidate solution based on the second phase, $x_{i,d}^{P2}$ indicates its dth dimension, F_i^{P2} is its objective function value, t is the iteration counter, lb_j and ub_j are the lower and upper bound of jth variable.

3.3 Repetition Process, Pseudocode, and Flowchart of SOA

The first iteration of SOA is completed after updating all members based on the first and second phases. The algorithm then enters the next iteration and the update process is repeated according to Eqs. (3) to (6). Once fully implemented, SOA puts the best candidate solution in the output. The SOA flowchart is presented in Fig. 1 and its pseudocode is presented in Alg. 1.

3.4 Computational Complexity of SOA

In this subsection, the computational complexity of SOA is studied. The computational complexity of SOA initialization to create the initial population and initial evaluation of the objective function is equal to $O(Nm)$, where N is the number of population members and m is the number of problem variables. The SOA member update process has two phases, while in each iteration and in each phase, the position of each updated member and the objective function are evaluated. Thus, the computational complexity of SOA in the upgrade process is equal to $O(2NmT)$, where T is the maximum number of iterations. Thus, the total complexity of SOA is equal to $O(Nm\ (1+2\ T))$.

Algorithm 1: Pseudocode of Skill Optimization Algorithm (SOA)

Start SOA.
1. Input the optimization problem information.
2. Set T (number of iterations) and N (number of population members).
3. For $t = 1{:}T$
5. For i $= 1{:}N$
6. Phase 1:
7. Select expert member.
8. Calculate new status of ith candidate solution based on phase 1 using Eq. (3).
9. Update ith candidate solution using Eq. (4).
10. Phase 2:
11. Calculate new status of ith candidate solution based on phase 2 using Eq. (5).
12. Update ith candidate solution using Eq. (6).
13. end
14. Save the best candidate solution so far.
15. end
16. Output the best obtained solution.

End SOA.

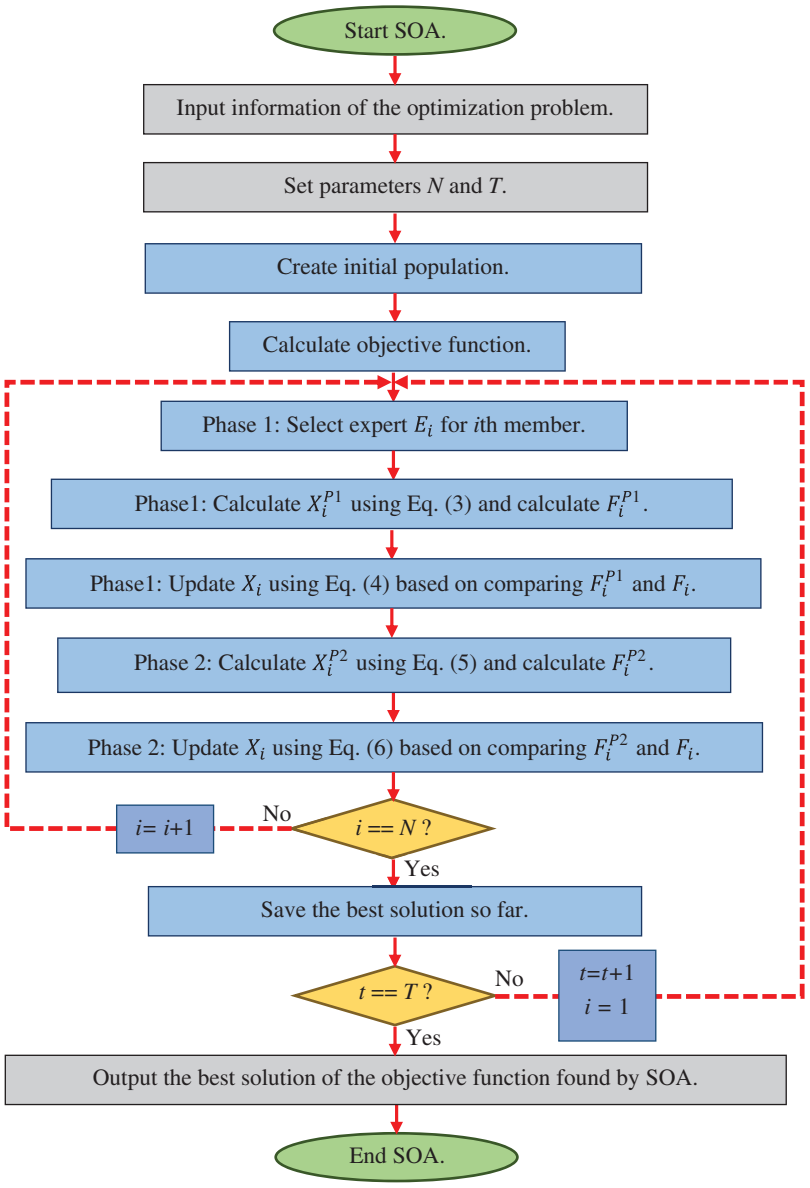

Figure 1: Flowchart of skill optimization algorithm (SOA)

4 Simulation Studies and Results

In this section, the performance of the proposed SOA in optimization applications is evaluated. For this purpose, SOA is implemented on a set of twenty-three standard objective functions. This set includes a variety of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal functions [78]. The results obtained by the proposed SOA are compared with ten well-known metaheuristic algorithms including: GA, RFO, PSO, WOA, GWO, MVO, MPA, GSA, TSA, and TLBO. The algorithms selected for comparison are among the most popular metaheuristic algorithms published. The reason for choosing these algorithms for comparison is as follows. The most popular

and widely used algorithms: GA and PSO. Highly cited algorithms: GSA, GWO, MVO, and WOA. Recently released and popular algorithms: RFO, MPA, and TSA. The setting values for the control parameters of competing algorithms are specified in Tab. 1. SOA and competing algorithms are employed in twenty independent implementations to optimize F1 to F23 functions, each of which contains 50,000 function evaluations. Optimization results are reported using six statistical indicators: Mean, best, worst, standard deviation (std), median, and rank.

Table 1: Setting values for control parameters of competing algorithms

Algorithm	Parameter	Value
RFO		
	Fox observation angle (φ_0)	$\varphi_0 \in (0,\ 2\pi)$
	Weather conditions (θ)	random value between 0 and 1
	Scaling parameter	$a \in (0,\ 0.2)$
	Population size	30
MPA		
	Constant number	$P = 0.5$
	Random vector	$R \in [0,\ 1]$
	Fish Aggregating Devices (*FADs*)	FADs = 0.2
	Binary vector	$U = 0$ or 1
	Population size	30
TSA		
	P_{min}	1
	P_{max}	4
	c_1, c_2, c_3	random numbers stand in the interval $[0,\ 1]$.
	Population size	30
WOA		
	a : Convergence parameter	Linear reduction from 2 to 0.
	r : Random vector	$r \in [0,\ 1]$.
	l : Random number	$l \in [-1,\ 1]$.
	Population size	30
GWO		
	Convergence parameter (a)	a: Linear reduction from 2 to 0.
	Population size	30
MVO		
	Wormhole existence probability (WEP)	Min (WEP) = 0.2 and Max(WEP) = 1.
	Exploitation accuracy over the iterations (p)	$p = 6$.
	Population size	60
TLBO		
	T_F: Teaching factor	$T_F = \text{round}\ [(1 + rand)]$
	random number	$rand \in [0,\ 1]$.
	Population size	30
GSA		
	Alpha	20

(Continued)

Table 1: Continued

Algorithm	Parameter	Value
	R_{power}	1
	R_{norm}	2
	G_0	100
	Population size	30
PSO		
	Topology	Fully connected
	Cognitive constant	$C_1 = 2$
	Social constant	$C_2 = 2$
	Inertia weight	Linear reduction from 0.9 to 0.1
	Velocity limit	10% of variables' dimension range
	Population size	50
GA		
	Type	Real coded
	Selection	Roulette wheel (Proportionate)
	Crossover	Whole arithmetic (Probability = 0.8, $\alpha \in [-0.5,\ 1.5]$)
	Mutation	Gaussian (Probability = 0.05)
	Population size	50

4.1 Intuitive Analysis in Two-Dimensional Search Space

In order to observe the quality of SOA, the first experiment is performed on 10 objective functions including F1, and F3 to F11 in two dimensions, while the number of SOA population members is considered equal to 5. To show the SOA behavior in optimizing these functions, convergence curves, search history curves, and trajectory curves are presented in Fig. 2. These curves show the SOA's behavior in search of the problem-solving space, solution-finding, and convergence process, how it achieves better solutions based on update processes, and decreases objective function values. Based on this analysis, it is determined that SOA, by improving the initial population, is able to increase the quality of candidate solutions and improve the accuracy of the quasi-optimal solution during the iterations of the algorithm.

4.2 Evaluation of Unimodal Functions

The first group of objective functions are selected from the unimodal type, including F1 to F7 functions. The results of implementing the proposed SOA and ten competing algorithms on these functions are reported in Tab. 2. Based on the analysis of the simulation results, it is clear that SOA has been able to provide the global optimal solution in optimizing F1, F3, F5, and F6. In optimizing F2, F4, and F7 functions, the SOA is the first best optimizer. Comparison of the optimization results shows that the proposed SOA offers superior and much more competitive performance against the ten mentioned algorithms and has the first rank of the best optimizer for optimizing F1 to F7 unimodal functions.

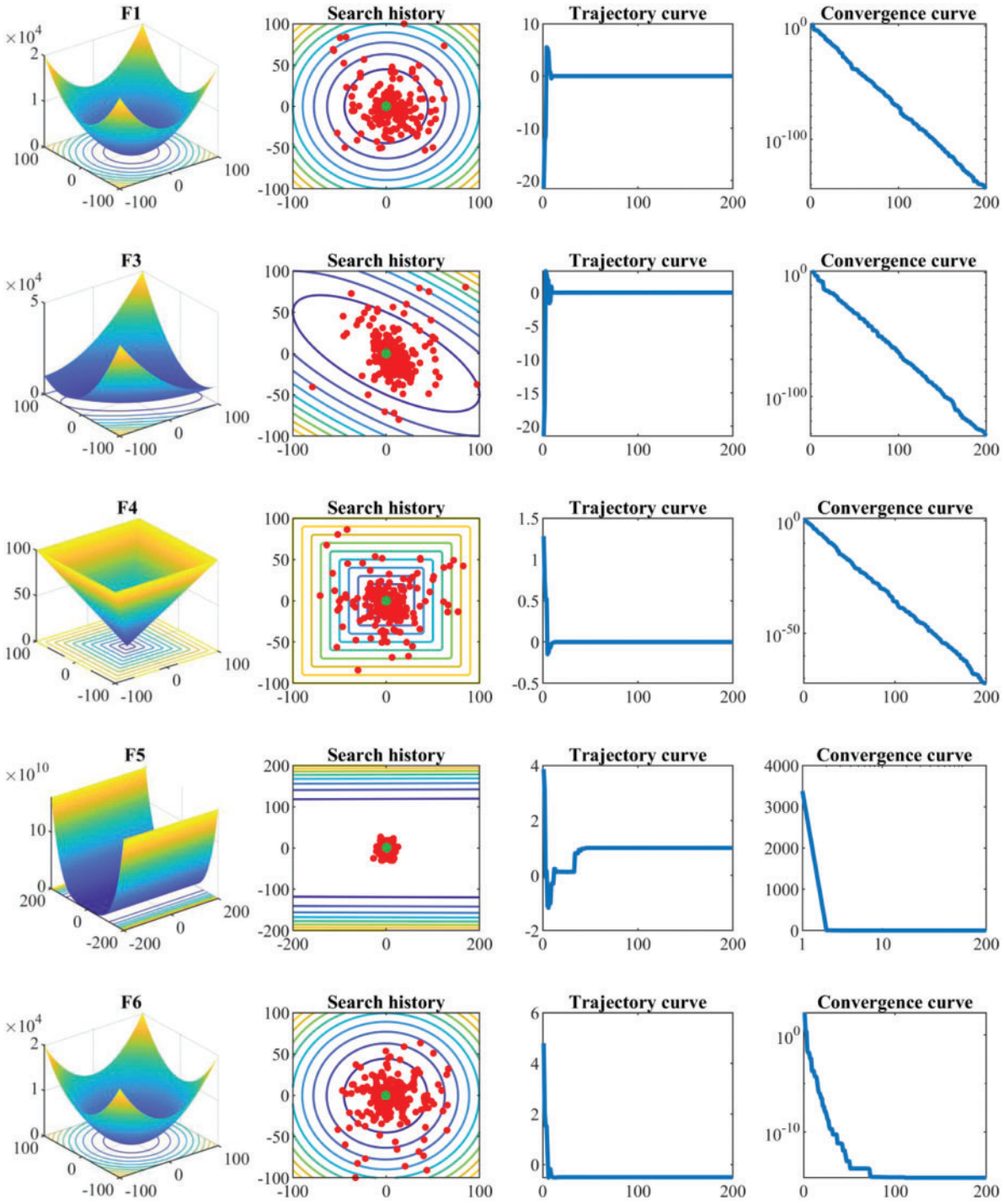

Figure 2: (Continued)

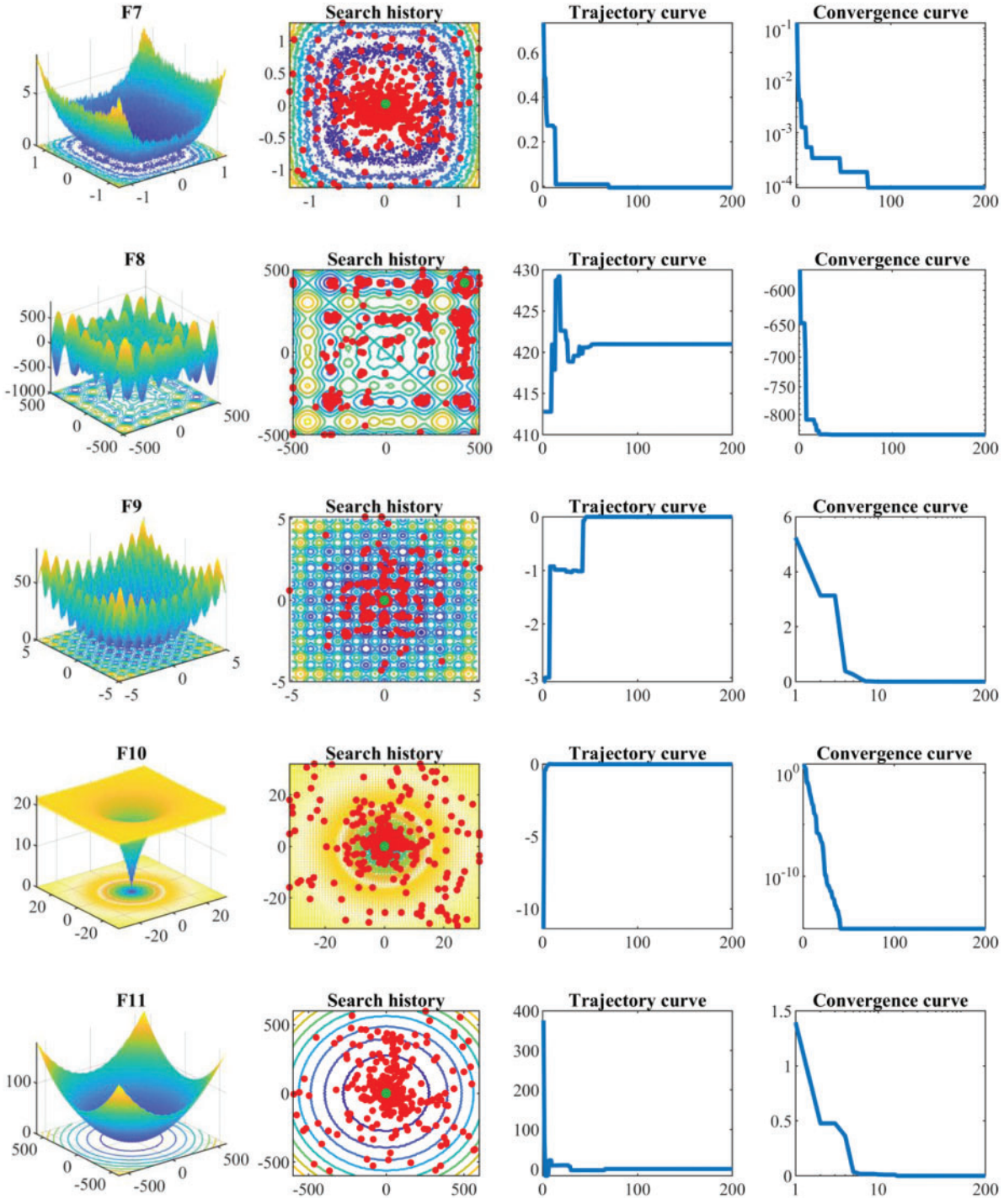

Figure 2: Search history curves, trajectory curves, and convergence curves for optimization of different objective functions using SOA

Table 2: Evaluation results for unimodal functions

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RFO	SOA
F_1	mean	21.26981	0.00051	7.68E-17	4.29E-61	1.3E-100	0.207125	6.5E-258	3.21E-82	5.99E-86	6.46E-84	0
	best	9.075049	2.3E-09	2.87E-17	1.19E-64	5.5E-105	0.132133	5.3E-279	3.87E-85	1.49E-88	9.43E-93	0
	worst	35.6214	0.008555	1.78E-16	3.87E-60	1.8E-99	0.334305	9.6E-257	2.65E-81	4.66E-85	1.19E-82	0
	std	7.682634	0.001902	3.8E-17	9.91E-61	4.1E-100	0.049577	0	7E-82	1.31E-85	2.64E-83	0
	median	19.18395	1.98E-05	6.69E-17	3.07E-62	1.5E-101	0.200136	1.7E-264	7.98E-83	9.6E-87	3.69E-88	0
	rank	11	9	8	7	3	10	2	6	4	5	1
F_2	mean	1.569531	0.591161	3.95E-08	4.47E-32	1.8E-58	0.300354	2.2E-175	1.82E-48	2.67E-47	6.78E-46	4.6E-191
	best	0.862316	0.037644	2.93E-08	1.93E-33	1.3E-59	0.198121	9.2E-186	1.06E-50	1.11E-50	4.79E-49	2.5E-199
	worst	2.386639	2.015304	4.92E-08	3.24E-31	8.85E-58	0.478449	2.3E-174	2.08E-47	2.29E-46	5.43E-45	6.2E-190
	std	0.343698	0.564708	5.87E-09	7.24E-32	2.27E-58	0.072241	0	4.78E-48	6.43E-47	1.51E-45	0
	median	1.569694	0.403794	3.8E-08	2.2E-32	1.04E-58	0.297865	5.8E-180	1.75E-49	2.14E-48	3.56E-47	2.3E-194
	rank	11	10	8	7	3	9	2	4	5	6	1
F_3	mean	2081.243	1393.67	185.0622	1.03E-19	6.47E-29	21.44854	6629.856	3.65E-21	7.73E-23	4.76E-58	0
	best	1259.74	6.57381	74.33398	3.17E-23	4.85E-33	7.82502	69.84706	6.19E-34	6.02E-34	1.19E-69	0
	worst	2983.205	15003.24	371.1815	1.92E-18	5.88E-28	38.34412	16103.81	6.72E-20	1.51E-21	5.35E-57	0
	std	544.7972	3538.467	84.30438	4.28E-19	1.56E-28	8.370361	5355.653	1.5E-20	3.37E-22	1.3E-57	0
	median	2187.973	124.986	196.4926	2.23E-21	9.85E-31	21.26726	4491.665	9.94E-26	5.01E-26	1.49E-61	0
	rank	10	9	8	6	3	7	11	5	4	2	1
F_4	mean	2.69652	4.395579	1.05E-08	3.98E-25	9.73E-25	0.628421	35.21117	1.01E-05	1.23E-32	1.34E-35	1.9E-181
	best	1.743387	2.439263	5.04E-09	2.21E-26	7.23E-27	0.409946	0.002335	9.57E-09	3.39E-34	3.83E-40	2E-187
	worst	3.394498	7.887114	1.76E-08	1.82E-24	9.58E-24	0.907885	83.01575	6.54E-05	6.27E-32	1.66E-34	2.8E-180
	std	0.424989	1.484789	2.92E-09	4.09E-25	2.09E-24	0.149233	31.18094	1.72E-05	1.85E-32	3.82E-35	0
	median	2.824597	4.422618	1E-08	2.38E-25	4.45E-25	0.611763	24.18985	4.86E-06	5.34E-33	2.7E-37	3.6E-183
	rank	9	10	6	4	5	8	11	7	3	2	1
F_5	mean	329.9827	76.28695	31.86645	26.97201	26.63528	294.0262	26.58882	28.68167	22.01376	27.45887	0
	best	216.9837	17.87017	24.57713	25.6828	25.21945	26.37103	25.74796	27.14486	20.73023	26.21217	0
	worst	488.3518	203.2271	99.39055	28.75532	27.97638	1469.614	28.71756	28.89131	23.77688	28.59278	0
	std	92.9552	50.91239	20.58985	0.752828	0.793021	384.52	0.671657	0.41617	0.733693	0.72896	0
	median	306.1666	78.75237	25.12935	27.12385	26.21618	117.3819	26.38524	28.83166	21.9493	27.18532	0
	rank	11	9	8	5	4	10	3	7	2	6	1
F_6	mean	25.6673	0.111786	7.7E-17	1.184742	0.693371	0.209373	0.006263	3.553484	2.06E-10	1.54416	0
	best	14.76772	6.92E-09	4.64E-17	0.473779	0.245746	0.118313	0.001584	2.549842	6.13E-11	0.862897	0
	worst	40.5949	1.039352	1.61E-16	1.763981	1.502094	0.322121	0.030521	5.269283	3.36E-10	2.393213	0
	std	7.343189	0.290975	2.49E-17	0.42441	0.350938	0.062271	0.006598	0.771801	7.18E-11	0.399298	0
	median	23.57016	6.77E-06	7.13E-17	1.16059	0.748114	0.201627	0.004322	3.322299	1.88E-10	1.639428	0
	rank	11	5	2	8	7	6	4	10	3	9	1
F_7	mean	0.006177	0.175885	0.049793	0.001795	0.000355	0.012573	0.0012	0.003079	0.00036	0.000401	2.8E-05
	best	0.00217	0.068566	0.027937	0.000306	5.37E-05	0.005253	0.00011	0.001112	7.32E-05	2.99E-05	2.59E-06
	worst	0.011223	0.310016	0.086446	0.004004	0.000825	0.021717	0.004578	0.00619	0.00071	0.000953	0.000127
	std	0.002536	0.065713	0.016593	0.000897	0.000174	0.004998	0.001188	0.001172	0.00018	0.000307	2.84E-05
	median	0.005728	0.159437	0.050208	0.001609	0.000312	0.011579	0.000966	0.002885	0.000347	0.000317	2.18E-05

(Continued)

Table 2 Continued

	GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RFO	SOA
rank	8	11	10	6	2	9	5	7	3	4	1
sum rank	71	63	50	43	27	59	38	46	24	34	7
mean rank	10.1428	9	7.1428	6.1428	3.8571	8.4285	5.4285	6.5714	3.42857	4.8571	1
total rank	11	10	8	6	3	9	5	7	2	4	1

4.3 Evaluation of High-Dimensional Multimodal Functions

The second group of objective functions are selected from the high-dimensional multimodal category, including F8 to F13 functions. The optimization results of these functions using SOA and ten competing algorithms are presented in Tab. 3. The optimization results show that the proposed SOA has been able to provide the global optimal solution for F9, F10, and F11. In optimizing F8, F12, and F13 functions, the SOA is the first best optimizer. Analysis of the simulation results shows that the proposed SOA has presented superior and acceptable performance against ten compared algorithms and has the first rank of the best optimizer in solving F8 to F13 high-dimensional multimodal functions.

4.4 Evaluation of Fixed-Dimensional Multimodal Functions

The third group of objective functions is selected from the fixed-dimensional multimodal type, including F14 to F23 functions. The results of employing SOA and ten competing algorithms for optimizing these functions are given in Tab. 4. It can be inferred from the optimization results that the proposed SOA achieves the global optimal solution for F14 and F17. Moreover, SOA is the first best optimizer for F15, F21, and F22 functions. The analysis of the results also shows that in cases where the proposed SOA and some competing algorithms result in the same "mean" value, SOA has a lower value for the "std" criterion, which indicates that the performance of SOA is superior. Based on the results obtained by the proposed SOA and ten competing algorithms, one can confirm the effective ability of SOA in solving fixed-dimensional multimodal optimization problems.

The performance of competing algorithms and SOA in optimizing F1 to F23 is presented in the form of boxplots in Fig. 3.

4.5 Statistical Analysis

Reporting the results of implementing optimization algorithms on optimization problems through statistical indicators, provides useful information on the performance of these algorithms. In this subsection, a statistical analysis on the proposed algorithm and the mentioned ten competing algorithms is presented to determine whether the superiority of the proposed algorithm over competing algorithms is statistically significant or not. For this purpose, Wilcoxon rank sum test [79] which is a non-parametric statistical analysis is used. In this test, an index called "p-value" indicates whether there is a significant difference between the means of the two data samples.

The implementation results of this statistical analysis on the proposed algorithm and each of the competing algorithms are presented in Tab. 5. According to this table, in cases where the p-value is less than 0.05, the proposed SOA has a significant statistical superiority over the competing algorithm.

Table 3: Evaluation results for high-dimensional multimodal functions

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RFO	SOA
F_8	mean	−9724.11	−6625.71	−2631.29	−5113.2	−5936.05	−7954.47	−11359.3	−6356.99	−10196.9	−7548.39	−11359.3
	best	−10683.4	−7722.65	−4184.85	−5989.13	−8384.32	−9585.89	−12569.3	−7324.09	−11148.2	−9259.4	−12569.3
	worst	−8962.95	−5244.96	−1902.95	−4307.78	−3520.94	−6712.98	−8420.5	−5424.29	−9625.09	−5383.42	−8420.5
	std	433.4248	736.1013	574.3377	449.4007	1184.476	736.4254	1539.136	521.3239	469.5183	1154.307	1539.136
	median	−9675.75	−6606.18	−2550.52	−5024.3	−6177.85	−7929.01	−12142.7	−6401.56	−10006.2	−7805.26	−12142.7
	rank	3	6	10	9	8	4	1	7	2	5	1
F_9	mean	22.63225	65.72422	25.91867	0	0.049798	120.0591	0	161.3873	0	0	0
	best	11.16984	29.84875	15.91934	0	0	65.78826	0	111.0216	0	0	0
	worst	49.46167	97.57666	33.82859	0	0.995955	190.1506	0	222.6946	0	0	0
	std	8.335758	18.64216	4.868663	0	0.222702	34.21213	0	37.35268	0	0	0
	median	20.2806	65.16998	25.86892	0	0	106.5907	0	149.281	0	0	0
	rank	3	5	4	1	2	6	1	7	1	1	1
F_{10}	mean	2.574197	3.204465	6.23E-09	4.09E-15	1.05E-14	0.624475	4.09E-15	1.892375	3.55E-15	4.54E-13	8.88E-16
	best	1.941967	1.777997	4.57E-09	8.88E-16	7.99E-15	0.118952	8.88E-16	7.99E-15	8.88E-16	8.88E-16	8.88E-16
	worst	3.406681	5.411885	7.93E-09	4.44E-15	1.51E-14	2.76295	7.99E-15	3.437375	4.44E-15	9.03E-12	8.88E-16
	std	0.347393	0.883149	8.7E-10	1.09E-15	3.07E-15	0.711655	2.55E-15	1.442783	1.58E-15	2.02E-12	0
	median	2.586969	3.190032	6.17E-09	4.44E-15	7.99E-15	0.292485	4.44E-15	2.6303	4.44E-15	8.88E-16	8.88E-16
	rank	9	10	6	3	4	7	3	8	2	5	1
F_{11}	mean	1.319156	0.097255	0.530335	0	0.000411	0.452361	0	0.006015	0	0	0
	best	1.185729	2.8E-07	0	0	0	0.298831	0	0	0	0	0
	worst	1.585668	0.338651	1.519286	0	0.008221	0.611349	0	0.062672	0	0	0
	std	0.10427	0.10069	0.542244	0	0.001838	0.084424	0	0.014105	0	0	0
	median	1.309755	0.06711	0.204115	0	0	0.453937	0	0	0	0	0
	rank	7	4	6	1	2	5	1	3	1	1	1
F_{12}	mean	0.091632	0.60768	0.031101	0.077858	0.034503	1.025048	0.003207	5.886347	2.2E-11	0.069238	1.57E-32
	best	0.029303	8.51E-07	1.81E-19	0.038633	0.006543	0.000939	0.000173	2.777071	7.28E-12	0.012096	1.57E-32
	worst	0.227612	3.296576	0.103669	0.149053	0.058425	2.53623	0.023374	9.528924	5.47E-11	0.179779	1.57E-32
	std	0.052119	0.873867	0.048741	0.030196	0.014458	0.736736	0.005925	2.016966	1.17E-11	0.039794	2.81E-48
	median	0.073812	0.157963	4.37E-19	0.066594	0.033144	1.229275	0.000531	6.201404	1.97E-11	0.061529	1.57E-32
	rank	8	9	4	7	5	10	3	11	2	6	1
F_{13}	mean	1.811361	2.18539	0.001099	1.026993	0.462331	0.034237	0.078718	2.85759	0.000549	1.803955	1.35E-32
	best	0.727944	2.22E-05	3.93E-18	0.447815	0.201031	0.010031	0.006429	1.936112	7.26E-11	1.051985	1.35E-32
	worst	3.593348	11.52197	0.010987	1.426866	0.949111	0.112463	0.506975	4.883946	0.010987	2.793816	1.35E-32
	std	0.680912	2.930781	0.003382	0.249937	0.174519	0.025383	0.128829	0.805392	0.002457	0.41072	2.81E-48
	median	1.728505	1.073705	7.64E-18	1.045832	0.462085	0.028196	0.029947	2.736892	3.01E-10	1.694537	1.35E-32
	rank	9	10	3	7	6	4	5	11	2	8	1
sum rank		39	44	33	28	27	36	14	47	10	26	6
mean rank		6.5	7.3333	5.5	4.6666	4.5	6	2.3333	7.83333	1.6666	4.3333	1
total rank		9	10	7	6	5	8	3	11	2	4	1

Table 4: Evaluation results for fixed-dimensional multimodal functions

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RFO	SOA
F_{14}	mean	22.63225	65.72422	25.91867	0	0.049798	120.0591	0	161.3873	0	0	0
	best	11.16984	29.84875	15.91934	0	0	65.78826	0	111.0216	0	0	0
	worst	49.46167	97.57666	33.82859	0	0.995955	190.1506	0	222.6946	0	0	0
	std	8.335758	18.64216	4.868663	0	0.222702	34.21213	0	37.35268	0	0	0
	median	20.2806	65.16998	25.86892	0	0	106.5907	0	149.281	0	0	0
	rank	3	5	4	1	2	6	1	7	1	1	1
F_{15}	mean	2.574197	3.204465	6.23E-09	4.09E-15	1.05E-14	0.624475	4.09E-15	1.892375	3.55E-15	4.54E-13	8.88E-16
	best	1.941967	1.777997	4.57E-09	8.88E-16	7.99E-15	0.118952	8.88E-16	7.99E-15	8.88E-16	8.88E-16	8.88E-16
	worst	3.406681	5.411885	7.93E-09	4.44E-15	1.51E-14	2.76295	7.99E-15	3.437375	4.44E-15	9.03E-12	8.88E-16
	std	0.347393	0.883149	8.7E-10	1.09E-15	3.07E-15	0.711655	2.55E-15	1.442783	1.58E-15	2.02E-12	0
	median	2.586969	3.190032	6.17E-09	4.44E-15	7.99E-15	0.292485	4.44E-15	2.6303	4.44E-15	8.88E-16	8.88E-16
	rank	9	10	6	3	4	7	3	8	2	5	1
F_{16}	mean	1.319156	0.097255	0.530335	0	0.000411	0.452361	0	0.006015	0	0	0
	best	1.185729	2.8E-07	0	0	0	0.298831	0	0	0	0	0
	worst	1.585668	0.338651	1.519286	0	0.008221	0.611349	0	0.062672	0	0	0
	std	0.10427	0.10069	0.542244	0	0.001838	0.084424	0	0.014105	0	0	0
	median	1.309755	0.06711	0.204115	0	0	0.453937	0	0	0	0	0
	rank	7	4	6	1	2	5	1	3	1	1	1
F_{17}	mean	0.091632	0.60768	0.031101	0.077858	0.034503	1.025048	0.003207	5.886347	2.2E-11	0.069238	1.57E-32
	best	0.029303	8.51E-07	1.81E-19	0.038633	0.006543	0.000939	0.000173	2.777071	7.28E-12	0.012096	1.57E-32
	worst	0.227612	3.296576	0.103669	0.149053	0.058425	2.53623	0.023374	9.528924	5.47E-11	0.179779	1.57E-32
	std	0.052119	0.873867	0.048741	0.030196	0.014458	0.736736	0.005925	2.016966	1.17E-11	0.039794	2.81E-48
	median	0.073812	0.157963	4.37E-19	0.066594	0.033144	1.229275	0.000531	6.201404	1.97E-11	0.061529	1.57E-32
	rank	8	9	4	7	5	10	3	11	2	6	1
F_{18}	mean	1.811361	2.18539	0.001099	1.026993	0.462331	0.034237	0.078718	2.85759	0.000549	1.803955	1.35E-32
	best	0.727944	2.22E-05	3.93E-18	0.447815	0.201031	0.010031	0.006429	1.936112	7.26E-11	1.051985	1.35E-32
	worst	3.593348	11.52197	0.010987	1.426866	0.949111	0.112463	0.506975	4.883946	0.010987	2.793816	1.35E-32
	std	0.680912	2.930781	0.003382	0.249937	0.174519	0.025383	0.128829	0.805392	0.002457	0.41072	2.81E-48
	median	1.728505	1.073705	7.64E-18	1.045832	0.462085	0.028196	0.029947	2.736892	3.01E-10	1.694537	1.35E-32
	rank	9	10	3	7	6	4	5	11	2	8	1
F_{19}	mean	0.99977	2.529701	2.859906	1.196416	4.812725	0.998004	2.371342	8.947953	0.998004	4.823742	0.998004
	best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004
	worst	1.032145	6.903336	5.581779	2.982105	12.67051	0.998004	10.76318	15.50382	0.998004	12.67051	0.998004
	std	0.007623	1.959635	1.552188	0.610693	4.533176	2.69E-12	2.980135	4.420183	3.58E-13	3.851995	0
	median	0.99801	0.998004	2.311345	0.998004	2.982105	0.998004	0.998004	10.76318	0.998004	3.96825	0.998004
	rank	4	7	8	5	9	3	6	11	2	10	1
F_{20}	mean	0.006239	0.00069	0.002236	0.003453	0.005321	0.002629	0.000597	0.004376	0.000532	0.005053	0.000314
	best	0.000588	0.000307	0.001047	0.000313	0.000307	0.000312	0.000309	0.000308	0.000308	0.000307	0.000307
	worst	0.02135	0.00525	0.006158	0.020364	0.020363	0.020363	0.001271	0.020363	0.00196	0.022553	0.000328
	std	0.006294	0.00112	0.001326	0.007107	0.00891	0.00607	0.00036	0.008202	0.000371	0.008991	7.37E-06
	median	0.003551	0.000307	0.001846	0.000328	0.000307	0.000654	0.000428	0.000308	0.000406	0.000653	0.00031

(Continued)

Table 4 Continued

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RFO	SOA
	rank	11	4	5	7	10	6	3	8	2	9	1
F_{21}	mean	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.02688	−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	−1.03162	−1.03163	−1.03163	−1.03162	−1.03163	−1.03163	−1.03163	−1	−1.03163	−0.21546	−1.03163
	std	2.56E-06	1.53E-16	1.02E-16	2.37E-06	3.68E-09	6.21E-08	4.97E-12	0.011587	1.22E-10	0.1825	2.28E-16
	median	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163
	rank	6	1	1	7	4	5	2	8	3	9	1
F_{22}	mean	0.533562	0.681058	0.397887	0.401505	0.397904	0.397887	0.397887	0.397898	0.397887	0.397887	0.397887
	best	0.397887	0.397887	0.397887	0.397895	0.397887	0.397887	0.397887	0.397888	0.397887	0.397887	0.397887
	worst	2.098616	2.38985	0.397887	0.466757	0.398223	0.397888	0.397889	0.39793	0.397887	0.397887	0.397887
	std	0.400668	0.637721	0	0.015362	7.49E-05	1.13E-07	2.57E-07	1.13E-05	2.5E-09	8.97E-16	0
	median	0.397906	0.397887	0.397887	0.397985	0.397887	0.397887	0.397887	0.397893	0.397887	0.397887	0.397887
	rank	9	10	1	8	7	4	5	6	3	2	1
F_{23}	mean	7.140989	3	3	3.000002	3.000002	7.05	3.000003	3.000004	3	13.8	3
	best	3	3	3	3	3	3	3	3.000001	3	3	3
	worst	30.56093	3	3	3.000023	3.000009	84	3.000011	3.000019	3	84	3
	std	10.05281	2.67E-15	1.91E-15	5.04E-06	2.72E-06	18.11215	3.64E-06	3.82E-06	2.03E-12	20.3563	5.29E-16
	median	3.005672	3	3	3.000001	3.000001	3	3.000001	3.000003	3	3	3
	rank	10	2	3	5	6	9	7	8	4	11	1
sum rank		71	61	33	67	60	56	52	74	26	82	10
mean rank		7.1	6.1	3.3	6.7	6	5.6	5.2	7.4	2.6	8.2	1
total rank		9	7	3	8	6	5	4	10	2	11	1

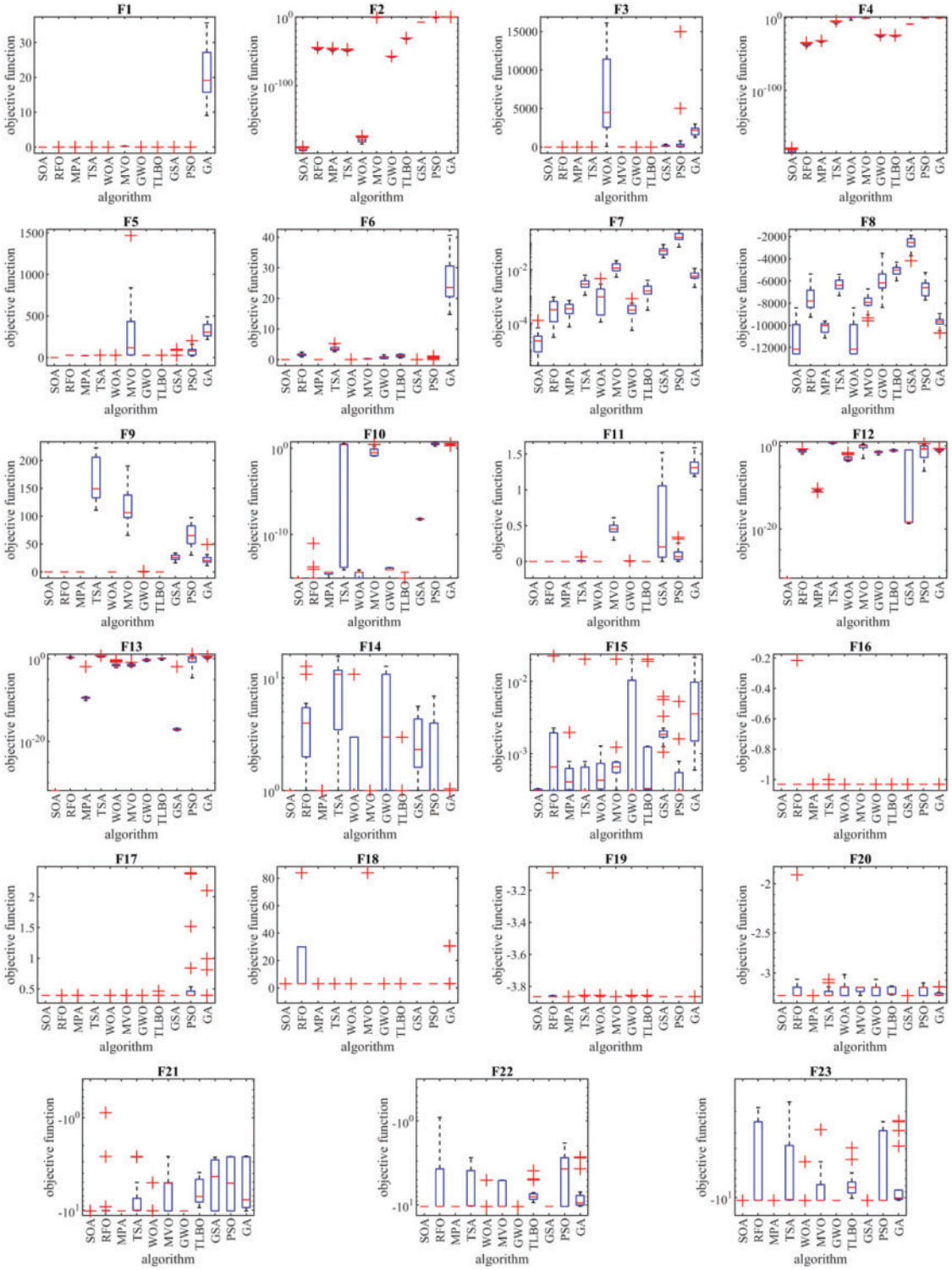

Figure 3: Boxplots of performances for skill optimization algorithm (SOA) and competing algorithms

Table 5: *p*-values obtained by Wilcoxon sum rank test

Compared Algorithms	Unimodal	High-Multimodal	Fixed-Multimodal
SOA *vs.* GA	1.01E-24	3.49E-16	5.19E-04
SOA *vs.* PSO	1.01E-24	1.97E-21	0.012161
SOA *vs.* GSA	1.01E-24	2.88E-21	2.32E-11
SOA *vs.* TLBO	1.01E-24	1.54E-14	0.0156238
SOA *vs.* GWO	5.71E-24	3.59E-15	2.67E-09
SOA *vs.* MPO	1.01E-24	1.97E-21	5.06E-08
SOA *vs.* WOA	1.01E-24	1.57E-10	4.68E-13
SOA *vs.* MPA	1.01E-24	2.73E-19	0.0188501
SOA *vs.* TSA	1.01E-24	3.32E-09	1.08E-32
SOA *vs.* RFO	5.84E-24	5.17E-12	0.014673

5 Conclusions and Future Works

In this paper, a new human-based metaheuristic algorithm called Skill Optimization Algorithm (SOA) was designed. Human effort to learn and improve skills is the main inspiration of SOA. The steps of implementing and updating SOA in two phases of exploration and exploitation were described and then mathematically modeled. A set of twenty-three standard objective functions of different types of unimodal and multimodal were considered to evaluate the optimization performance of SOA. The results of optimizing unimodal functions indicated the exploitation ability of SOA in local search. The results obtained by the optimization of high-dimensional multimodal functions showed the exploration ability of SOA in global search. The results of optimizing fixed-dimensional multimodal functions showed that SOA has an acceptable ability to balance the indicators of exploration and exploitation. In order to analyze the quality of the results achieved by SOA, its performance was compared with ten well-known metaheuristic algorithms. The simulation results confirmed the superiority of SOA performance over mentioned algorithms based on exploration and exploitation criteria. In other words, SOA can provide more effective solutions to optimization problems compared to the considered competing algorithms.

The proposed SOA can activate several research tasks for further studies. Development of binary and multi-objective versions of SOA are among the main proposals of this research. Employment of SOA for optimization purposes in various fields and real-world applications are other suggestions of the authors for future investigations.

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

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, 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.
[3] 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.
[4] 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, 2021.
[5] 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, 2022.
[6] M. Premkumar, P. Jangir, R. Sowmya and R. M. Elavarasan, "Many-objective gradient-based optimizer to solve optimal power flow problems: Analysis and validations," *Engineering Applications of Artificial Intelligence*, vol. 106, pp. 104479, 2021.
[7] S. Kumar, P. Jangir, G. G. Tejani, M. Premkumar and H. H. Alhelou, "MOPGO: A new physics-based multi-objective plasma generation optimizer for solving structural optimization problems," *IEEE Access*, vol. 9, pp. 84982–85016, 2021.
[8] R. -M. Devi, M. Premkumar, P. Jangir, M. -A. Elkotb, R. -M. Elavarasan *et al.,* "IRKO: An improved runge-kutta optimization algorithm for global optimization problems," *Computers, Materials \& Continua*, vol. 70, no. 3, pp. 4803–4827, 2022.
[9] M. Premkumar, P. Jangir and R. Sowmya, "MOGBO: A new multiobjective gradient-based optimizer for real-world structural optimization problems," *Knowledge-Based Systems*, vol. 218, pp. 106856, 2021.
[10] M. Premkumar, P. Jangir, R. Sowmya, H. H. Alhelou, A. A. Heidari *et al.,* "MOSMA: Multi-objective slime mould algorithm based on elitist non-dominated sorting," *IEEE Access*, vol. 9, pp. 3229–3248, 2021.
[11] S. Kumar, P. Jangir, G. G. Tejani and M. Premkumar, "MOTEO: A novel physics-based multiobjective thermal exchange optimization algorithm to design truss structures," *Knowledge-Based Systems*, vol. 242, pp. 108422, 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. vol. 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] R. -M. Devi, M. Premkumar, P. Jangir, B. -S. Kumar, D. Alrowaili *et al.,* "BHGSO: Binary hunger games search optimization algorithm for feature selection problem," *Computers, Materials \& Continua*, vol. 70, no. 1, pp. 557–-579, 2022.
[15] M. Premkumar, P. Jangir, R. Sowmya, R. M. Elavarasan and B. S. Kumar, "Enhanced chaotic JAYA algorithm for parameter estimation of photovoltaic cell/modules," *ISA Transactions*, vol. 116, pp. 139–166, 2021/10/01/, 2021.
[16] 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.
[17] 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. vol. 1, pp. 61–64, 2020.
[18] 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.
[19] 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. vol. 4, pp. 70–73, 2018.

[20] 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. vol. 6, pp. 68–73, 2018.
[21] 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.
[22] 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.
[23] 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. https://doi.org/10.20998/2074-272X.2019.4.10.
[24] 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.
[25] 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.
[26] 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.
[27] 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.
[28] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.
[29] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "GSA: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
[30] 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.
[31] R. Moghdani and K. Salimifard, "Volleyball premier league algorithm," *Applied Soft Computing*, vol. 64, pp. 161–185, 2018.
[32] 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.
[33] 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.
[34] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
[35] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
[36] P. Trojovský and M. Dehghani, "Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications," *Sensors*, vol. 22, no. 3, pp. 855, 2022.
[37] 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.
[38] 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.
[39] A. Faramarzi, M. Heidarinejad, S. Mirjalili and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, pp. 113377, 2020.

[40] 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. https://doi.org.10.1109/ACCESS.2022.3151641.
[41] 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.
[42] 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, pp. 103541, 2020.
[43] 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. https://doi.org.10.1109/ACCESS.2021.3133286.
[44] M. Dehghani, M., Mardaneh, O. P. Malik and S. M. NouraeiPour, "DTO: Donkey theorem optimization," in *Proc. of Iranian Conference on Electrical Engineering (ICEE)*, Yazd, Iran, IEEE, pp. 1855–1859, 2019.
[45] 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, pp. 1–26, 2020.
[46] 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.
[47] D. Połap and M. Woźniak, "Red fox optimization algorithm," *Expert Systems with Applications*, vol. 166, pp. 114107, 2021.
[48] 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.
[49] 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.
[50] 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.
[51] 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, pp. e910, 2022.
[52] 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.
[53] F. -A. Zeidabadi, A. Dehghani, M. Dehghani, Z. Montazeri, Š Hubálovský *et al.,* "SSABA: Search step adjustment based algorithm," *Computers, Materials \& Continua*, vol. 71, no. 2, pp. 4237–4256, 2022.
[54] L. J. Fogel, A. J. Owens and M. J. Walsh, "Artificial intelligence through simulated evolution," 1966.
[55] M. Dehghani and P. Trojovský, "Selecting some variables to update-based algorithm for solving optimization problems," *Sensors*, vol. 22, no. 5, pp. 1795, 2022.
[56] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.
[57] S. Kirkpatrick C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[58] 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.
[59] 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.

[60] 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.
[61] 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.
[62] A. Faramarzi, M. Heidarinejad, B. Stephens and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, pp. 105190, 2020.
[63] 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.
[64] 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, no. 5, pp. 514–523, 2020.
[65] 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.
[66] 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.
[67] 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.
[68] 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.
[69] 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.
[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] 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.
[72] 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.
[73] 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.
[74] M. Dehghani and P. Trojovský, "Teamwork optimization algorithm: A New optimization approach for function minimization/Maximization," *Sensors*, vol. 21, no. 13, pp. 4567, 2021.
[75] 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.
[76] F. -A. Zeidabadi, M. Dehghani, P. Trojovský, Š Hubálovský, V. Leiva *et al.,* "Archery algorithm: A novel stochastic optimization algorithm for solving optimization problems," *Computers*," *Materials \& Continua*, vol. 72, no. 1, pp. 399–416, 2022.
[77] 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, pp. 165–181, 2019.

[78] X. Yao, Y. Liu and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
[79] F. Wilcoxon, "Individual comparisons by ranking methods," *Breakthroughs in Statistics*, vol. 1, no. 6, pp. 196–202, 1992.

Appendix

The source codes of the proposed SOA are available at:

https://www.mathworks.com/matlabcentral/fileexchange/110675-skill-optimization-algorithm-soa.