Received: March 28, 2024. Revised: April 21, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Dollmaker Optimization Algorithm: A Novel Human-Inspired Optimizer for Solving Optimization Problems

Saleh Al omari[1] Khalid Kaabneh[2] Ibraheem AbuFalahah[3] Kei Eguchi[4]*
Saikat Gochhait[5,6] Irina Leonova[7] Zeinab Montazeri[8] Mohammad Dehghani[8]

[1]*Faculty of Science and Information Technology, Software Engineering, Jadara University, Irbid 21110, Jordan*
[2]*Faculty of Information Technology, Al-Ahliyya Amman University, Amman, 19328, Jordan*
[3]*Department of Mathematics, Faculty of Science,*
The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan
[4]*Department of Information Electronics, Fukuoka Institute of Technology, Japan*
[5]*Symbiosis Institute of Digital and Telecom Management,*
Constituent of Symbiosis International Deemed University, Pune 412115, India
[6]*Neuroscience Research Institute, Samara State Medical University, 89 Chapaevskaya str., 443001 Samara, Russia*
[7]*Faculty of Social Sciences, Lobachevsky University, 603950 Nizhny Novgorod, Russia*
[8]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 7155713876, Iran*
* Corresponding author's Email: eguti@fit.ac.jp

Abstract: In this article, a new human-based metaheuristic algorithm named Dollmaker Optimization Algorithm (DOA) is introduced, which imitates the strategy and skill of the dollmaker when making dolls. The basic inspiration of DOA is derived from two natural behaviors in the doll making process (i) making general changes to the doll-making materials and (ii) making precise small changes on the appearance characteristics of the dolls. The theory of DOA is proposed and then modeled mathematically in two phases (i) exploration based on the simulation of large changes made on doll-making materials and (ii) exploitation based on the simulation of small changes on the made dolls. The performance of DOA in optimization is evaluated on twenty-three standard benchmark functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types. The optimization results show that DOA has achieved suitable results for optimization problems with its ability in exploration, exploitation, and balance them during the search process. Comparison of DOA results with twelve competing algorithms shows that the proposed algorithm has superior performance compared to competing algorithms by providing better results in all twenty-three benchmark functions and getting the rank of the first best optimizer. In addition, the efficiency of DOA in handling real-world applications is evaluated in the optimization of four engineering design problems. Simulation results show that DOA has acceptable performance in real world and engineering applications by providing better values for design variables and objective functions compared to competing algorithms.

Keywords: Optimization algorithm, Engineering application, Human-inspired, Dollmaker, Exploration, Exploitation.

1. Introduction

Optimization, the process of refining and enhancing systems, processes, or algorithms for maximum efficiency, lies at the heart of numerous scientific, engineering, and computational endeavors [1]. It encompasses a wide spectrum of applications, from solving complex mathematical problems to fine-tuning algorithms in artificial intelligence [2]. At its core, optimization involves finding the best solution among a set of feasible alternatives, taking into account specific constraints and objectives [3]. In mathematical terms, optimization problems often entail maximizing or minimizing an objective function while adhering to given constraints. This concept extends beyond mathematics and finds application in computer science, engineering, finance, and many other fields [4, 5].

Metaheuristic algorithms, inspired by natural processes and phenomena, represent a powerful class of optimization techniques that have gained prominence across diverse domains. These algorithms leverage principles from biology, physics, and social behavior to efficiently solve complex problems where traditional optimization approaches may struggle [6].

The main research question is that given the numerous of existing metaheuristic algorithms, there remains a necessity to introduce newer algorithms? In response to this query, the No Free Lunch (NFL) theorem [7] definitively asserts that the satisfactory performance of a metaheuristic algorithm in resolving a specific set of optimization problems does not ensure analogous success in addressing other optimization challenges. The NFL theorem underscores that there is no any metaheuristic algorithm universally superior as an optimizer for all types of optimization problems. The NFL theorem, while keeping active the study field of developing metaheuristic algorithms, motivates researchers to provide more effective solutions for optimization problems by designing newer metaheuristic algorithms.

Based on the best knowledge obtained from the literature review, no metaheuristic algorithm has been designed so far inspired by the doll making process. This is while piecing together the doll-making materials and making a doll similar to the given pattern is an intelligent process that has a special potential for designing a new optimizer. In order to address this research gap, in this study, a new metaheuristic algorithm is designed based on the mathematical modeling of the doll making process.

Motivated by the NFL theorem and confirming the originality of the proposed approach based on the best knowledge obtained from the literature review, the aspects of innovation, originality, and novelty of this study are in introducing a new metaheuristic algorithm called Dollmaker Optimization Algorithm (DOA) for optimization applications. The key contributions of this study are as follows:

- DOA has been designed inspired by dollmaker strategies in the human activity of doll-making.
- The inspiration for DOA comes from the dollmaker's performance when making large changes to doll-making materials and making detailed changes to made dolls.
- The theory of DOA has been expressed and mathematically modeled in two phases (i) exploration based on the simulation of large changes made on doll-making materials and (ii) exploitation based on the simulation of small changes on the made dolls.
- The efficiency of DOA has been evaluated to optimize twenty-three standard benchmark functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types.
- The performance of DOA has been evaluated in real-world applications to optimize four engineering design problems.
- The results obtained from DOA have been compared with the performance of twelve well-known metaheuristic algorithms.

The rest of the structure of the paper is organized as follows: Literature review is presented in Section 2. The proposed Dollmaker Optimization Algorithm (DOA) approach is introduced and modeled in Section 3. The simulation studies and results are presented in Section 4. The performance of DOA in real-world applications is presented in Section 5. Finally, conclusions and several suggestions for further research are provided in Section 6.

2. Literature review

Metaheuristic algorithms are stochastic approaches that are able to provide acceptable solutions for optimization problems based on random search in the problem solving space and using random processes and trial and error [8]. The inherent stochastic nature of metaheuristic algorithms introduces an element of unpredictability, rendering no guarantees or promises of reaching the global optimum. Consequently, the outcomes obtained through these methods are often characterized as quasi-optimal [9]. In the adept implementation of metaheuristic algorithms, the interplay between exploration and exploitation emerges as pivotal for navigating the inherent randomness within problem-solving spaces [10]. Exploration denotes the algorithm's adeptness in conducting expansive searches across the problem-solving space. This capability serves as a mechanism to steer clear of local optima, enabling the algorithm to traverse the entire space and identify global optimal regions through comprehensive exploration. Conversely, exploitation signifies the algorithm's prowess in honing in on identified solutions and promising regions within the problem-solving space. This facet involves targeted searches, concentrating efforts around areas that show potential for yielding improved solutions. In the context of metaheuristic algorithms, the skillful orchestration of exploration and exploitation stands out as a critical determinant of optimization success. Achieving an optimal balance between these two facets throughout the search process is essential. This delicate equilibrium allows the algorithm to harness the advantages of both exploration and exploitation, maximizing the chances of discovering high-quality solutions while avoiding the entrapment in suboptimal regions [11]. The efforts of researchers in achieving better quasi-optimal solutions for optimization problems and more effective management of exploration and exploitation have led to the design and development of countless metaheuristic algorithms.

Metaheuristic algorithms have been developed from various natural phenomena, biological sciences, genetics, physics, human activities, and other evolutionary phenomena. Based on the source of inspiration used in the design, metaheuristic algorithms are classified into five groups: swarm-based, evolutionary-based, physics-based, human-based, and game-based approaches.

Swarm-based algorithms simulate the collective behaviors observed in birds, animals, insects, reptiles, aquatic, and other live organisms. Ant Colony Optimization (ACO) [12], Particle Swarm Optimization (PSO) [13], Artificial Bee Colony (ABC) [14], and Firefly Algorithm (FA) [15] exemplify this category. ACO mimics ant communication for efficient pathfinding, while PSO emulates group movements of fish and birds. ABC simulates hierarchical behaviors within bee colonies, and FA takes inspiration from the Information exchange through optical communication among fireflies. Natural behaviors among living organisms such as foraging, hunting, migration, digging, flight strategy, and chasing process have been sources of inspiration in designing swarm-based algorithms such as: Pufferfish Optimization Algorithm [16], Grey Wolf Optimizer (GWO) [17], Wombat Optimization Algorithm (WOA) [18], Termite Alate Optimization Algorithm (TAOA) [19], Whale Optimization Algorithm (WOA) [20], African Vultures Optimization Algorithm (AVOA) [21], Swarm Space Hopping Algorithm (SSHA) [22], Reptile Search Algorithm (RSA) [23], Marine Predator Algorithm (MPA) [24], Migration-Crossover Algorithm (MCA) [25], White Shark Optimizer (WSO) [26], and Tunicate Swarm Algorithm (TSA) [27].

Evolutionary-inspired algorithms rooted in biological and genetic principles, emulate concepts such as natural selection, survival of the fittest, and genetic processes. Genetic Algorithm (GA) [28] and Differential Evolution (DE) [29] are prominent examples that simulate the natural reproduction process, incorporating genetic concepts like mutation, crossover, and selection.

Physics-based algorithms simulate laws, phenomena, transformations, forces, cycles, and other concepts in physics. Simulated Annealing (SA) [30] simulates the annealing process in metallurgy, while Gravitational Search Algorithm (GSA) [31] models gravitational forces and Newton's laws of motion. Water Cycle Algorithm (WCA) [32] mimics the transformations within the natural water cycle. Some other physics-based metaheuristic algorithms are: Lichtenberg Algorithm (LA) [33], Archimedes Optimization Algorithm (AOA) [34], and Propagation Search Algorithm (PSA) [35].

Human-based metaheuristic algorithms mimic various aspects of human life such as choices, decisions, interactions, communication, and other human activities. Teaching-Learning Based Optimization (TLBO) [36] mimics the dynamics of learning in a classroom setting. Mother Optimization Algorithm (MOA) [9] simulate the Eshrat's attentive care from her children. Ali Baba and the Forty Thieves (AFT) [37] leverages the narrative of Ali Baba and the Forty Thieves to model decision processes, creating an algorithm that adapts and learns from past experiences to make better choices over time. Some other human-inspired algorithms are: Doctor and Patient Optimization (DPO) [38], Teamwork Optimization Algorithm (TOA) [39], Driving Training-Based Optimization (DTBO) [40], and Election-Based Optimization Algorithm (EBOA) [41].

Game-based metaheuristic algorithms have been developed inspired by the strategies of players, referees, and coaches in various individual and team games under the rules of the game. One of the most cited algorithms of this group is Darts Game Optimizer (DGO), which imitates the strategy of players in tabletop darts and collecting points [42]. Simulating the skill of golf players during ball shots to guide them towards the score holes was the design idea of Golf Optimization Algorithm (GOA) [43]. The players' efforts to find the hidden object in the playground have been the inspiration for Hide Object Game Optimizer (HOGO) [44]. Some other game-inspired algorithms are: Football Game Based Optimization (FGBO) [45], Puzzle Optimization Algorithm (POA) [46], Orientation Search Algorithm (OSA) [47], Dice Game Optimizer (DGO) [48], Ring toss game based optimization (RTGBO) [49], and Shell game optimization (SGO) [50].

3. Dollmaker Optimization Algorithm

In this section, the source of inspiration and theory of the proposed Dollmaker Optimization Algorithm (DOA) approach is described, then its implementation process is mathematically modeled.

3.1 Inspiration of DOA

The doll is one of the popular children's toys series, which has different types and each one is made with different materials and sizes. The art of doll making has many fans both as a hobby and as a job and skill. From a general and simple point of view, the doll making process can be explained as follows: First, in the first step, the dollmaker must choose a pattern for the doll she wants to make. After choosing the pattern, the dollmaker must choose the materials she wants. She then sews the materials together, stuffs the doll with cotton, wool or any other type of stuffing material and then decorates it. After completing these steps, it is time to take care of the doll's details such as facial features, clothes, hair, and shoes. The dollmaker tries her best to make the doll according to the selected pattern. As it seems, the activity of doll making is an intelligent process, during which the selection of the pattern, the sewing together of the material of the doll, and the attention to the appearance details are much more significant. These special skills in the doll making process have been used as a source of inspiration in the DOA design, which is discussed further.

3.2 Algorithm Initialization

The proposed DOA approach is a population-based optimizer where dolls form the population members. Population members specify values for design variables based on their position in the problem-solving space. Therefore, each doll can be modeled as a candidate solution using a vector so that the elements of this vector represent the decision variables. The different parts of the doll correspond to the decision variables of the problem. Algorithm population can be mathematically modeled based on the community of these vectors together using a matrix according to Eq. (1). Also, the initial position of the population members is initialized completely randomly using Eq. (2).

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

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

Where, X is the DOA's population matrix, X_i is the ith doll (i.e., candidate solution), $x_{i,d}$ is its dth dimension in the search space (i.e., decision variable), N is the number of dolls (i.e., population size), m is the number of decision variables, r is a random number within the interval $[0,1]$, while lb_d and ub_d stand for the lower and upper bounds of the dth decision variable, respectively.

The objective function of the evaluable problem corresponds to the solution proposed by each member of the population. The set of evaluated values for the objective function can be represented using a vector according to Eq. (3).

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

Where, F is the vector of objective function values and F_i is the obtained objective function value based on the ith candidate solution.

The evaluated values for the objective function contain valuable information about the quality of the candidate solutions (i.e., population members). In such a way that the best evaluated value for the objective function corresponds to the best member of the population. Since DOA is an iteration-based approach, the positions of the population members and thus the objective function values are updated in each iteration. Therefore, the position of the best population member must also be updated and stored in each iteration. At the end of the DOA implementation, the position of the best member is presented as the solution for the given problem.

3.3 Mathematical modelling of DOA

The mathematical model of DOA is obtained based on the simulation of the doll making process. The position of the DOA population members is updated based on the modeling of the dollmaker's strategies during doll making. In a simple way, the process of making a doll can be considered in two parts: (i) sewing materials-doll making based on the selected pattern and (ii) beautifying the doll based on attention to appearance details such as facial features, hair, clothes, etc. In each iteration, based on the modeling of the doll-making process, the positions of the DOA members are updated in two phases (i) exploration based on the simulation of pattern

Received: March 28, 2024. Revised: April 21, 2024.

selection and sewing the doll-making materials together and (ii) exploitation based on the simulation of beautification of the doll's appearance. Each of these phases is explained and modeled mathematically.

3.3.1 Phase 1: Pattern selection and sewing (exploration phase)

The process of choosing a pattern and sewing doll-making materials leads to extensive changes in the doll's appearance. Modeling these extensive changes in the doll's appearance leads to large changes in the position of the population members and thus increases the exploration ability of the algorithm in the global search. In DOA design, the best member is considered as a doll pattern ($P = X_{best}$). On the other hand, the vector elements of each member represent the doll-making materials that must be sewn according to the selected pattern.

Based on the mathematical modeling of the doll making and sewing process according to selected pattern, a new position for each DOA member is obtained using Eq. (4). This new position replaces the previous position of the corresponding member if it improves the value of the objective function according to Eq. (5).

$$x_{i,j}^{P1} = x_{i,j} + r \cdot (P_j - I \cdot x_{i,j}), \quad (4)$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} \le F_i, \\ X_i, & else, \end{cases} \quad (5)$$

Where, P is the selected doll-making pattern, P_j is its jth dimension, X_i^{P1} is the new position for the ith member based on first phase of DOA, $x_{i,j}^{P1}$ is its jth dimension, F_i^{P1} is its objective function value, r is a random number drawn from the interval $[0, 1]$, and I is randomly selected number, taking values of 1 or 2.

3.3.2 Phase 2: Beautifying the details of the doll (exploitation phase)

The process of handling the appearance details in doll making, such as facial features, hair, and clothes, leads to small and precise changes in the appearance of the doll. Modeling these small changes in the doll's appearance leads to small changes in the position of the population members and thus increases the exploitation ability of the algorithm in local search. In DOA design, it is assumed that the dollmaker tries to bring the doll's appearance closer to the doll making pattern over time.

Based on the modeling of this process, a new position has been calculated for each DOA member using Eq. (6). Then, if the value of the objective function is improved, this new position replaces the previous position of the corresponding member according to Eq. (7).

$$x_{i,j}^{P2} = x_{i,j} + \left(1 - 2\, r_{i,j}\right) \cdot \frac{ub_j - lb_j}{t} \quad (6)$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} \le F_i \\ X_i, & else \end{cases} \quad (7)$$

Where, X_i^{P2} is the new calculated position for the ith member based on second phase of DOA, $x_{i,j}^{P2}$ is the its jth dimension, F_i^{P2} is its objective function value, r is a random number drawn from the interval $[0, 1]$, and t is the iteration counter.

Algorithm 1. Pseudocode of DOA.

Start DOA.

1. Input problem information: variables, objective function, and constraints.
2. Set DOA population size (N) and iterations (T).
3. Generate the initial population matrix at random using Eq. (2). $x_{i,d} \leftarrow lb_d + r \cdot (ub_d - lb_d)$
4. Evaluate the objective function.
5. For $t = 1$ to T
6. For $i = 1$ to N
7. Phase 1: *Pattern selection and sewing* (exploration phase)
8. Determine doll-making pattern. $P \leftarrow X_{best}$
9. Calculate new position of ith member using Eq. (4). $x_{i,d}^{P1} \leftarrow x_{i,d} + r \cdot \left(P_d - I \cdot x_{i,d}\right)$
10. Update ith member using Eq. (5). $X_i \leftarrow \begin{cases} X_i^{P1}, & F_i^{P1} < F_i \\ X_i, & else \end{cases}$
11. Phase 2: *Beautifying the details of the doll* (exploitation phase)
12. Calculate new position of ith member using Eq. (6). $x_{i,d}^{P2} \leftarrow x_{i,d} + (1 - 2r) \cdot \frac{(ub_d - lb_d)}{t}$
13. Update ith member using Eq. (7). $X_i \leftarrow \begin{cases} X_i^{P2}, & F_i^{P2} < F_i \\ X_i, & else \end{cases}$
14. end
15. Save the best candidate solution so far.
16. end
17. Output the best quasi-optimal solution obtained with the DOA.

End DOA.

Received: March 28, 2024. Revised: April 21, 2024.

3.4 Repetition process, pseudocode, and flowchart of DOA

The first iteration of DOA is completed by updating the position of all population members based on the first and second phases. Then, based on the updated values, the algorithm enters the next iteration and the process of updating the position of the population members continues based on Eqs. (4) to (7) until the last iteration of the algorithm. At the end of each iteration, the position of the best member is updated and saved as the best candidate solution. After the complete implementation of DOA, the algorithm outputs the best candidate solution as a solution to the problem. The pseudocode of DOA implementation steps is presented in Algorithm 1.

4. Simulation Studies and Results

In this section, the performance of the proposed DOA approach in optimization applications is evaluated. For this purpose, a set of twenty-three standard benchmark functions has been selected from unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types [51]. Also, in order to measure the quality of DOA in the optimization process, the performance of the proposed algorithm is compared with the performance of twelve famous algorithms: GA [28], PSO [13], GSA [31], TLBO [36], MVO [52], GWO [17], WO [20], MPA [24], TSA [27], RSA [23], AVOA [21], and WSO [26]. The simulation results are reported using six statistical indicators: mean, best, worst, standard deviation (std), median, and rank.

4.1 Evaluation of Unimodal Functions

Unimodal functions are suitable criteria for measuring the exploitation ability of metaheuristic algorithms in local search. The optimization results of unimodal functions F1 to F7 using DOA and competing algorithms are reported in Table 1. Based on the results obtained, DOA is the first best optimizer for benchmark functions F1 to F7. The

Table 1. Optimization results of unimodal functions (F1 to F7)

		DOA	GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	AVOA	WSO
F_1	mean	0	27.14679	0.089852	1.19E-16	2.19E-48	2.19E-48	0.133176	2.19E-48	4.37E-47	2.36E-48	2.19E-48	2.19E-48	58.6581
	best	0	15.955	0.000433	4.77E-17	6.79E-52	6.79E-52	0.093903	6.79E-52	1.36E-50	3.04E-50	6.79E-52	6.79E-52	4.712689
	worst	0	50.66591	1.243992	3.33E-16	1.55E-47	1.55E-47	0.179155	1.55E-47	3.11E-46	1.56E-47	1.55E-47	1.55E-47	212.6302
	std	0	12.52707	0.372093	8.57E-17	6.33E-48	6.33E-48	0.033234	6.33E-48	1.27E-46	6.32E-48	6.33E-48	6.33E-48	63.18205
	median	0	25.09709	0.008651	1.01E-16	2.01E-49	2.01E-49	0.13397	2.01E-49	4.02E-48	3.60E-49	2.01E-49	2.01E-49	40.42377
	rank	1	9	7	6	2	3	8	2	5	4	2	2	10
F_2	mean	0	2.481671	0.797	4.88E-08	9.91E-30	9.91E-30	0.230664	9.91E-30	1.98E-28	6.30E-28	9.91E-30	9.91E-30	1.904458
	best	0	1.553367	0.040301	3.10E-08	9.52E-32	9.52E-32	0.142466	9.52E-32	1.90E-30	1.65E-29	9.52E-32	9.52E-32	0.589605
	worst	0	3.387835	2.21905	1.10E-07	8.56E-29	8.56E-29	0.324414	8.56E-29	1.71E-27	4.19E-27	8.56E-29	8.56E-29	6.626493
	std	0	0.652269	0.865309	2.24E-08	3.35E-29	3.35E-29	0.075419	3.35E-29	6.69E-28	1.31E-27	3.35E-29	3.35E-29	2.124325
	median	0	2.439984	0.519906	4.56E-08	9.28E-31	9.28E-31	0.23883	9.28E-31	1.86E-29	3.15E-28	9.28E-31	9.28E-31	1.36211
	rank	1	11	9	7	3	4	8	2	5	6	2	2	10
F_3	mean	0	1930.395	345.4371	423.1948	5.56E-12	5.58E-12	14.21626	17763.71	1.11E-10	7.80E-12	5.56E-12	5.56E-12	1589.816
	best	0	1267.527	19.37375	218.9078	6.44E-23	4.84E-17	5.317104	1837.744	1.29E-21	5.65E-19	6.44E-23	6.44E-23	925.9979
	worst	0	3078.452	912.5999	1055.822	9.17E-11	9.17E-11	43.55639	30872.71	1.83E-09	9.17E-11	9.17E-11	9.17E-11	3153.371
	std	0	765.8958	345.3349	263.7433	2.75E-11	2.76E-11	12.88865	10245.38	5.51E-10	2.72E-11	2.75E-11	2.75E-11	751.6498
	median	0	1869.623	260.8096	356.2979	5.05E-15	5.96E-15	10.57255	18088.59	1.01E-13	8.11E-13	5.05E-15	5.05E-15	1386.878
	rank	1	11	8	9	3	4	7	12	6	5	2	2	10
F_4	mean	0	2.51837	5.589303	1.100142	0.000208	0.000208	0.487143	46.1212	0.004158	0.000208	0.000208	0.000208	15.39364
	best	0	1.972962	2.038398	1.27E-05	4.54E-06	4.54E-06	0.236733	0.805081	9.08E-05	4.54E-06	4.54E-06	4.54E-06	10.60421
	worst	0	3.553565	11.89067	4.385664	0.001684	0.001684	0.857417	81.62169	0.033678	0.001684	0.001684	0.001684	21.21382
	std	0	0.55884	2.996113	1.660907	0.000502	0.000502	0.23043	35.45728	0.010046	0.000502	0.000502	0.000502	3.45709
	median	0	2.477499	5.235566	0.807541	6.91E-05	6.91E-05	0.472794	49.32789	0.001382	6.91E-05	6.91E-05	6.91E-05	15.81772
	rank	1	8	9	7	2	4	6	11	5	3	2	2	10
F_5	mean	0	531.2315	4105.96	40.54285	25.1797	24.99605	86.97562	25.64407	26.76871	22.09678	12.90721	1.338448	9612.804
	best	0	204.9855	24.7466	24.39427	24.13091	24.10889	25.94922	25.06069	24.13078	21.61512	1.206539	1.206592	1200.463
	worst	0	2010.139	80170.14	150.2052	26.93663	25.51563	337.6917	26.91992	27.15817	22.70481	27.1591	1.357941	82518.45
	std	0	508.8431	24085.41	53.07648	1.130638	0.623396	121.4899	0.673237	0.996632	0.474651	17.65108	0.049822	24027.66
	median	0	424.5561	77.96324	24.80443	24.77036	24.70271	28.06676	25.4556	27.09322	22.08456	1.356977	1.354673	4993.98
	rank	1	11	12	9	6	5	10	7	8	4	3	2	13
F_6	mean	0	30.56429	0.229516	0.17305	1.2957	0.761205	0.307442	0.24565	3.460993	0.17305	5.920566	0.17305	89.98014
	best	0	14.09712	0.120465	0.119982	0.398002	0.351125	0.202828	0.142249	2.399644	0.119982	3.439722	0.119982	15.22048
	worst	0	56.00716	0.614451	0.225021	2.128811	1.293145	0.402021	0.451924	4.500415	0.225021	6.677521	0.225021	340.5878
	std	0	16.21709	0.177487	0.043839	0.604112	0.36869	0.071327	0.125119	0.876788	0.043839	1.23173	0.043839	114.3272
	median	0	28.36403	0.192708	0.178412	1.245799	0.804883	0.311319	0.222756	3.568235	0.178412	6.32392	0.178412	62.08455
	rank	1	12	5	2	9	8	7	6	10	3	11	4	13
F_7	mean	2.54E-05	0.009629	0.164091	0.047206	0.001567	0.000945	0.010542	0.001343	0.004084	0.000692	0.000232	0.000261	0.000286
	best	2.35E-06	0.002906	0.061734	0.013043	0.000375	0.000412	0.003843	0.000134	0.001404	0.00017	9.17E-05	8.13E-05	0.000103
	worst	6.89E-05	0.019642	0.366195	0.085356	0.002913	0.001896	0.0204	0.005114	0.009377	0.001051	0.000481	0.000507	0.000497
	std	2.66E-05	0.005741	0.094558	0.029871	0.001022	0.000534	0.006035	0.001835	0.002961	0.000312	0.000157	0.000165	0.000139
	median	1.83E-05	0.009362	0.158424	0.046212	0.001542	0.000971	0.010494	0.000888	0.0035	0.000751	0.0002	0.000232	0.00031
	rank	1	10	13	12	8	6	11	7	9	5	2	3	4
Sum rank		7	72	63	52	33	34	57	47	48	30	24	17	70
Mean rank		1	10.28571	9	7.428571	4.714286	4.857143	8.142857	6.714286	6.857143	4.285714	3.428571	2.428571	10
Total rank		1	13	11	9	5	6	10	7	8	4	3	2	12

 DOI: 10.22266/ijies2024.0630.63

analysis of the simulation results shows that DOA has provided superior performance for handling unimodal functions by providing better results compared to competing algorithms.

4.2 Evaluation of High-dimensional Multimodal Functions

High-dimensional multimodal functions F8 to F13 are suitable criteria for measuring the exploration ability of metaheuristic algorithms in global search. The results of DOA implementation and competing algorithms on functions F8 to F13 are reported in Table 2. Based on the optimization results, DOA is the first best optimizer for functions F8 to F13. Comparison of simulation results shows that DOA with high ability in exploration has provided superior performance for handling high-dimensional multimodal functions compared to competing algorithms.

4.3 Evaluation of Fixed-dimensional Multimodal Functions

Fixed-dimensional multimodal F14 to F23 are suitable criteria for measuring the ability of metaheuristic algorithms in balancing exploration and exploitation during the search process. The optimization results of F14 to F23 using DOA and competing algorithms are reported in Table 2. Based on the obtained results, DOA is the first best optimizer for functions F14 to F23. Analysis of simulation results shows that DOA by balancing exploration and exploitation has provided superior performance compared to competing algorithms for fixed-dimensional multimodal optimization.

The convergence curve obtained from the performance of DOA and competing algorithms is drawn in Figure 1.

5. DOA for real-World engineering applications

One of the most important tasks of metaheuristic algorithms is their efficiency in handling real world and engineering applications. For this purpose, the performance of DOA and competing algorithms has been evaluated to address four engineering design issues: pressure vessel design (PV) [53], speed reducer design (SR) [54], welded beam design (WB) [20], and tension/compression spring design (TCS) [20]. Mathematical models of these problems are available in the mentioned references.

Table 2. Optimization results of high-dimensional multimodal functions (F8 to F13)

		DOA	GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	AVOA	WSO
F_8	mean	-12498.6	-8571.09	-6903.15	-3551.27	-6058.52	-6486.83	-8047.28	-10923.9	-6520.74	-9697.79	-5914.18	-12174.9	-7351.6
	best	-12622.8	-9738.69	-8334.05	-4614.84	-7343.22	-7160.27	-9296.19	-12322.8	-7635.27	-10415.6	-6125.52	-12324	-9132.8
	worst	-11936.3	-7334.28	-5528.37	-2972.82	-5131.71	-5611.35	-7190.47	-7945.2	-4853.87	-9079.55	-5421.48	-11655.8	-6442.85
	std	256.1381	785.8267	879.9175	594.8246	725.7787	544.7651	876.6688	2091.028	926.1619	455.0936	267.1615	235.5707	888.3561
	median	-12577.8	-8543.41	-7039.91	-3479.65	-6028.22	-6471.94	-7954.19	-11776.6	-6487.72	-9703.24	-5966.46	-12243.8	-7296.1
	rank	1	5	8	13	11	10	6	3	9	4	12	2	7
F_9	mean	0	56.80313	68.40265	33.50678	8.136836	8.136836	95.20529	8.136836	162.7367	8.136836	8.136836	8.136836	30.05767
	best	0	32.4179	42.15101	16.78524	4.218008	4.218008	53.07454	4.218008	84.36015	4.218008	4.218008	4.218008	19.64073
	worst	0	77.48891	110.062	51.49186	13.54467	13.54467	140.0999	13.54467	270.8934	13.54467	13.54467	13.54467	47.62651
	std	0	15.7339	23.75593	11.82388	3.225068	3.225068	30.46679	3.225068	64.50135	3.225068	3.225068	3.225068	9.821795
	median	0	55.42619	66.18456	31.85476	7.833751	7.833751	94.19074	7.833751	156.675	7.833751	7.833751	7.833751	29.10191
	rank	1	6	7	5	2	3	8	2	9	2	2	2	4
F_{10}	mean	8.88E-16	3.240236	2.485635	0.058397	0.058397	0.058397	0.572728	0.058397	1.167943	0.058397	0.058397	0.058397	4.767728
	best	8.88E-16	2.564946	1.50717	5.30E-09	4.38E-15	1.42E-14	0.089534	1.56E-15	7.57E-15	4.38E-15	1.22E-15	1.22E-15	3.157463
	worst	8.88E-16	4.281638	4.500794	0.158552	0.158552	0.158552	2.238518	0.158552	3.171046	0.158552	0.158552	0.158552	7.296848
	std	0	0.48896	1.030056	0.099236	0.099236	0.099236	0.799656	0.099236	1.984723	0.099236	0.099236	0.099236	1.449472
	median	8.88E-16	3.275591	2.544806	8.41E-09	5.05E-15	1.89E-14	0.268439	6.63E-15	2.09E-14	5.05E-15	1.89E-15	1.89E-15	4.758482
	rank	1	11	10	7	5	6	8	3	9	4	2	2	12
F_{11}	mean	0	1.311805	0.165302	6.415549	0.000416	0.001608	0.356127	0.000416	0.008312	0.000416	0.000416	0.000416	1.527795
	best	0	1.14695	0.002575	2.666538	0	0	0.22673	0	0	0	0	0	0.983417
	worst	0	1.536421	0.779505	11.24762	0.000966	0.017221	0.477028	0.000966	0.019314	0.000966	0.000966	0.000966	2.923811
	std	0	0.148333	0.27341	3.257752	0.000398	0.005398	0.09789	0.000398	0.007958	0.000398	0.000398	0.000398	0.649631
	median	0	1.289042	0.109185	6.50714	0.000423	0.000438	0.371221	0.000423	0.008454	0.000423	0.000423	0.000423	1.425144
	rank	1	7	5	9	2	3	6	2	4	2	2	2	8
F_{12}	mean	1.57E-32	0.516917	1.608203	0.459194	0.335744	0.307753	1.086292	0.290147	5.445224	0.272261	1.444939	0.272261	3.182297
	best	1.57E-32	0.152587	0.161499	0.073048	0.12856	0.085656	0.098165	0.056183	0.974647	0.048732	0.991714	0.048732	1.009694
	worst	1.57E-32	0.940286	4.754493	1.120546	0.739394	0.681629	3.600025	0.666861	13.28784	0.664392	2.129247	0.664392	6.775412
	std	3.78E-48	0.297123	1.529211	0.441376	0.244852	0.235959	1.368371	0.248742	4.907015	0.245351	0.434558	0.245351	2.183355
	median	1.57E-32	0.518172	1.49152	0.320247	0.268527	0.255313	0.69286	0.221576	4.04661	0.202331	1.458074	0.202331	2.807788
	rank	1	8	11	7	6	5	9	4	13	2	10	3	12
F_{13}	mean	1.35E-32	2.537667	3.338477	0.178122	1.108471	0.584994	0.156863	0.318692	2.553878	0.129917	0.127694	0.127694	3203.844
	best	1.35E-32	1.268438	0.134622	0.095543	0.664537	0.118305	0.105416	0.13126	1.891704	0.094585	0.094585	0.094585	12.3843
	worst	1.35E-32	3.602348	11.2958	0.971217	1.497775	0.979144	0.24964	0.728344	3.4911	0.174555	0.174555	0.174555	55323.73
	std	3.78E-48	0.893195	3.616823	0.256327	0.281016	0.316012	0.05014	0.206751	0.705027	0.03575	0.035251	0.035251	16587.25
	median	1.35E-32	2.681348	3.085504	0.123316	1.111873	0.576816	0.152954	0.273687	2.38306	0.121349	0.119153	0.119153	39.48181
	rank	1	10	12	6	9	8	5	7	11	4	2	3	13
Sum rank		6	47	53	47	35	35	42	21	55	18	30	14	56
Mean rank		1	7.833333	8.833333	7.833333	5.833333	5.833333	7	3.5	9.166667	3	5	2.333333	9.333333
Total rank		1	8	9	8	6	6	7	4	10	3	5	2	11

Received: March 28, 2024. Revised: April 21, 2024.

Table 3. Optimization results of fixed-dimensional multimodal functions (F14 to F23)

		DOA	GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	AVOA	WSO
F_{14}	mean	0.998004	1.403333	3.670276	3.63959	1.358244	3.758727	1.358243	2.757099	8.18865	1.368734	3.235489	1.446536	1.446712
	best	0.998004	1.044723	1.044723	1.04476	1.044723	1.044723	1.044723	1.044723	1.932389	1.044723	1.109103	1.044723	1.044723
	worst	0.998004	2.431301	11.47978	11.22258	1.694612	10.23762	1.694612	10.23762	14.6476	1.904192	11.47978	3.312462	2.341651
	std	0	0.453039	4.522659	3.378872	0.320831	4.523842	0.320833	3.595382	6.388951	0.346885	3.526134	0.669671	0.456907
	median	0.998004	1.506957	2.180353	3.216621	1.501789	2.857107	1.501789	1.546613	11.07371	1.501789	2.638739	1.501789	1.546611
	rank	1	5	11	10	3	12	2	8	13	4	9	6	7
F_{15}	mean	0.000307	0.014544	0.003072	0.002941	0.001377	0.003843	0.003204	0.001568	0.015513	0.001922	0.001847	0.001164	0.002056
	best	0.000307	0.000824	0.000314	0.000829	0.000332	0.000314	0.000314	0.000361	0.000314	0.000352	0.000786	0.000314	0.000329
	worst	0.000307	0.060619	0.018243	0.007401	0.00557	0.019213	0.019109	0.005638	0.103766	0.006779	0.006421	0.005564	0.018243
	std	3.42E-19	1.96E-02	7.22E-03	2.55E-03	1.88E-03	8.72E-03	7.36E-03	1.98E-03	3.80E-02	2.20E-03	2.01E-03	1.92E-03	5.46E-03
	median	0.000307	0.01325	0.001285	0.002181	0.001199	0.001238	0.001334	0.001168	0.00087	0.001618	0.001247	0.000435	0.000796
	rank	1	12	9	8	3	11	10	4	13	6	5	2	7
F_{16}	mean	-1.03163	-1.03141	-1.03141	-1.03141	-1.03141	-1.03141	-1.03141	-1.03141	-1.03	-1.02932	-1.02943	-1.03141	-1.03141
	best	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03161	-1.03163	-1.03163
	worst	-1.03163	-1.02969	-1.02969	-1.02969	-1.02969	-1.02969	-1.02969	-1.02969	-1.00185	-1.00238	-1.00348	-1.02969	-1.02969
	std	2.47E-16	0.000718	0.000718	0.000718	0.000718	0.000718	0.000718	0.000718	0.008933	0.008842	0.008339	0.000718	0.000718
	median	-1.03163	-1.03163	-1.03163	-1.03163	-1.03162	-1.03163	-1.03163	-1.03163	-1.03163	-1.0316	-1.03119	-1.03163	-1.03163
	rank	1	7	2	2	8	4	5	3	9	11	10	2	6
F_{17}	mean	0.397887	0.458562	0.706528	0.397921	0.397986	0.397922	0.397922	0.397922	0.397953	0.398379	0.40923	0.397921	0.397922
	best	0.397887	0.397893	0.397888	0.397888	0.397898	0.397888	0.397888	0.397888	0.397892	0.397888	0.398647	0.397888	0.397888
	worst	0.397887	1.603208	2.527926	0.398094	0.398142	0.398095	0.398094	0.398095	0.398187	0.401001	0.475655	0.398094	0.398094
	std	0	0.362446	0.84922	8.05E-05	0.000104	8.04E-05	8.05E-05	8.05E-05	0.000107	0.001225	0.023275	8.05E-05	8.05E-05
	median	0.397887	0.397962	0.397913	0.397895	0.39797	0.397896	0.397895	0.397896	0.397915	0.397972	0.403142	0.397895	0.397895
	rank	1	11	12	2	8	6	3	5	7	9	10	2	4
F_{18}	mean	3	7.428352	3.598769	3.598769	3.59877	3.59878	3.598769	3.598792	11.18138	6.412647	6.068332	3.598769	3.598769
	best	3	3.002614	3.000878	3.000878	3.000879	3.000882	3.000878	3.000878	3.000843	3.013279	3.00179	3.000879	3.000878
	worst	3	31.47844	7.190825	7.190825	7.190826	7.190831	7.190826	7.19087	86.69911	28.73222	28.20614	7.190825	7.190825
	std	1.57E-15	12.39497	1.654736	1.654736	1.654735	1.654731	1.654736	1.654737	33.05035	8.045268	10.12022	1.654736	1.654736
	median	3	3.203033	3.11195	3.11195	3.11195	3.111964	3.11195	3.11195	3.106629	4.693452	3.162393	3.111951	3.11195
	rank	1	12	3	4	7	8	5	9	13	11	10	6	2
F_{19}	mean	-3.86278	-3.85393	-3.85407	-3.85407	-3.85309	-3.85272	-3.85407	-3.85197	-3.85413	-3.73129	-3.83106	-3.85407	-3.85407
	best	-3.86278	-3.86276	-3.86278	-3.86278	-3.86256	-3.86278	-3.86278	-3.86276	-3.86268	-3.86277	-3.85373	-3.86278	-3.86278
	worst	-3.86278	-3.82658	-3.82689	-3.82689	-3.82665	-3.82678	-3.82689	-3.82649	-3.82851	-3.31987	-3.77948	-3.82689	-3.82689
	std	3.06E-15	1.18E-02	1.16E-02	1.16E-02	1.11E-02	1.18E-02	1.16E-02	1.14E-02	1.09E-02	1.76E-01	2.83E-02	1.16E-02	1.16E-02
	median	-3.86278	-3.85408	-3.85415	-3.85415	-3.85374	-3.8533	-3.85415	-3.8529	-3.85419	-3.73218	-3.83449	-3.85415	-3.85415
	rank	1	6	3	3	7	8	5	9	2	11	10	4	3
F_{20}	mean	-3.322	-3.18573	-3.21805	-3.26912	-3.19858	-3.21307	-3.2267	-3.20495	-3.21171	-2.56654	-2.77361	-3.2215	-3.25324
	best	-3.322	-3.26324	-3.31577	-3.31577	-3.29979	-3.31576	-3.31577	-3.31071	-3.31403	-3.22929	-3.05177	-3.2891	-3.31577
	worst	-3.322	-2.98585	-3.0732	-3.22501	-2.96297	-3.02701	-3.11916	-3.05619	-3.06896	-1.85588	-1.76712	-3.13146	-3.15744
	std	5.97E-16	0.089041	0.100858	0.027776	0.108921	0.102318	0.084727	0.103642	0.087284	0.430915	0.381955	0.078192	0.057732
	median	-3.322	-3.19871	-3.2533	-3.27476	-3.21363	-3.23417	-3.26502	-3.23027	-3.2049	-2.62387	-2.84781	-3.23417	-3.2678
	rank	1	11	6	2	10	7	4	9	8	13	12	5	3
F_{21}	mean	-10.1532	-6.32629	-5.75988	-7.15746	-6.85362	-9.11209	-8.66278	-9.10772	-6.02322	-7.48198	-5.25381	-9.79103	-8.23648
	best	-10.1532	-9.30974	-10.1328	-10.1378	-9.44556	-10.1445	-10.1378	-10.1438	-10.1245	-10.0502	-5.60812	-10.1453	-10.1453
	worst	-10.1532	-2.67925	-2.91651	-2.94365	-3.61378	-5.16941	-4.94259	-5.07451	-2.75776	-4.94259	-4.94247	-9.4797	-2.94365
	std	2.8E-15	3.323573	3.429165	4.169174	2.534973	2.191722	2.644992	2.234237	4.118955	2.668872	0.287213	0.287213	3.818368
	median	-10.1532	-6.84722	-5.19094	-9.47976	-7.12446	-9.66389	-9.65576	-9.68701	-5.06923	-7.7078	-5.23938	-9.77661	-9.6876
	rank	1	10	12	8	9	3	5	4	11	7	13	2	6
F_{22}	mean	-10.4029	-7.39418	-6.51403	-9.84827	-7.90855	-10.0914	-8.3401	-8.0498	-6.95672	-8.03305	-5.36124	-10.0918	-9.75137
	best	-10.4029	-9.88697	-10.3587	-10.3951	-9.63881	-10.3949	-10.3784	-10.3748	-10.32	-10.3929	-5.66449	-10.3951	-10.3951
	worst	-10.4029	-3.18596	-2.94388	-5.40846	-4.28686	-9.66401	-3.48283	-2.11701	-2.02811	-4.93469	-4.93469	-9.66528	-3.5691
	std	4.72E-15	2.368753	4.176814	1.449131	2.106412	0.356691	3.178741	3.638883	4.539533	2.819459	0.356579	0.356579	1.987714
	median	-10.4029	-7.88166	-5.12907	-10.1313	-8.20216	-10.1349	-9.74675	-9.71899	-7.64722	-9.0997	-5.40504	-10.1356	-10.1313
	rank	1	10	12	4	9	3	6	7	11	8	13	2	5
F_{23}	mean	-10.5364	-6.58572	-6.6397	-10.0809	-8.12183	-10.3022	-9.34622	-8.56444	-7.51932	-9.07171	-5.48951	-10.3026	-10.3026
	best	-10.5364	-9.87879	-10.513	-10.5244	-9.7604	-10.5241	-10.5243	-10.5234	-10.4352	-10.4468	-5.7113	-10.5244	-10.5244
	worst	-10.5364	-3.21988	-2.92646	-6.01233	-4.83532	-10.0579	-5.33925	-2.54813	-2.88379	-5.37894	-5.24503	-10.0581	-10.0581
	std	3.72E-15	3.151682	4.612978	1.304319	1.995703	0.208558	2.662232	3.874934	4.349108	1.820117	0.208735	0.208734	0.208734
	median	-10.5364	-7.096	-4.36246	-10.2535	-8.78108	-10.2708	-10.2221	-10.1502	-10.1095	-9.44035	-5.45793	-10.271	-10.271
	rank	1	12	11	5	9	4	6	8	10	7	13	3	2
Sum rank		10	96	81	48	73	66	51	66	97	87	105	34	45
Mean rank		1	9.6	8.1	4.8	7.3	6.6	5.1	6.6	9.7	8.7	10.5	3.4	4.5
Total rank		1	10	8	4	7	6	5	6	11	9	12	2	3

The implementation results of DOA and competing algorithms on four engineering design problems are reported in Table 4. Based on the optimization results, DOA is the first best optimizer in handling all four engineering design problems. Analysis of simulation results shows that DOA has an effective performance to tackle optimization tasks in real world and engineering applications by providing better results compared to competing algorithms.

6. Conclusions and future Works

In this paper, a new human-inspired metaheuristic algorithm named Dollmaker Optimization Algorithm (DOA) was introduced, which imitates the doll making process.

Received: March 28, 2024. Revised: April 21, 2024.

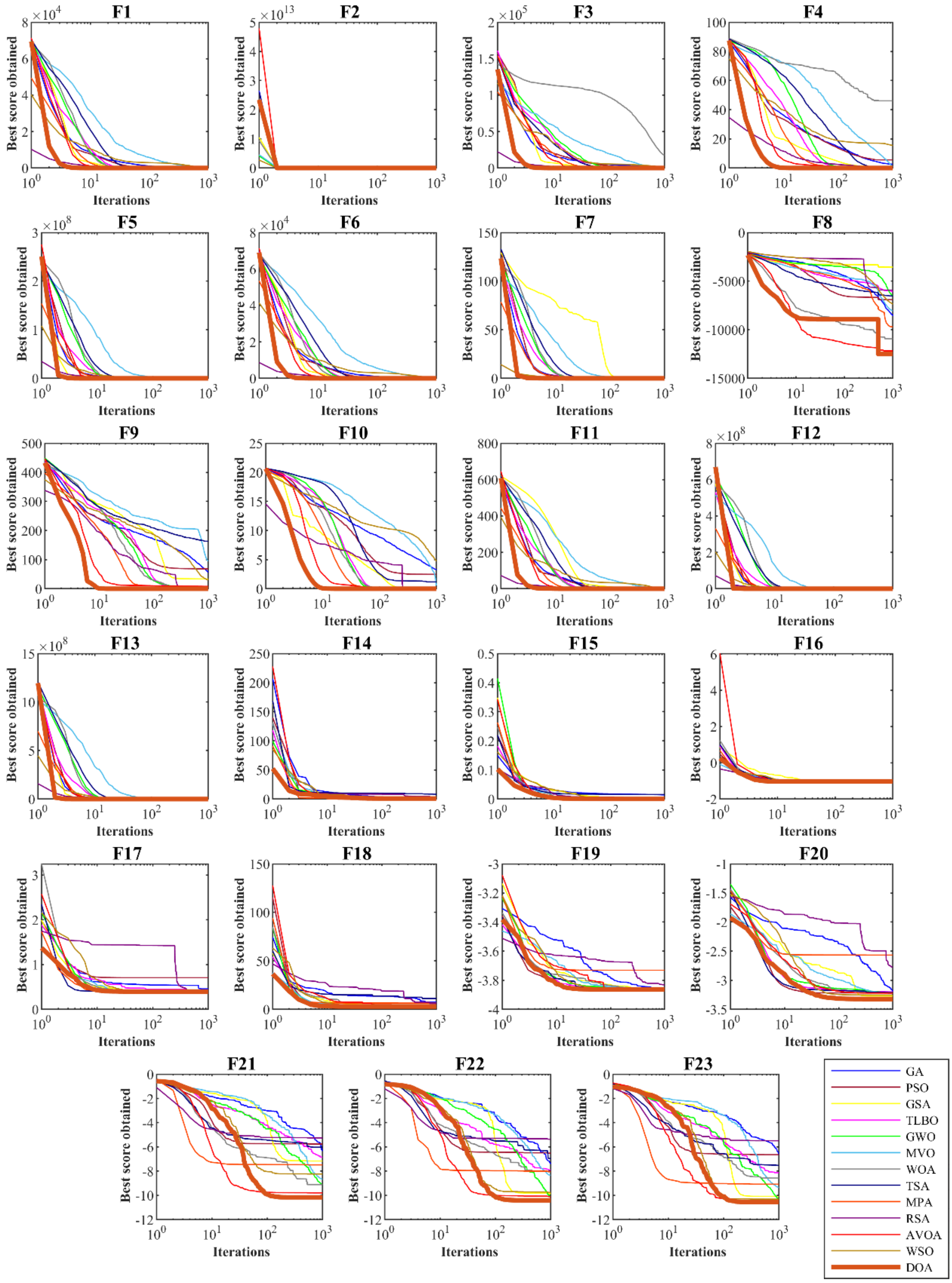

Figure. 1 Convergence curves of benchmark functions (F1 to F23)

The basic inspiration of DOA is derived from the strategies and skills of the dollmaker when making large changes to the doll-making materials and making precise changes to the made doll in order to bring its shape closer to the pattern. The theory of DOA was expressed and mathematically modeled in two phases (i) exploration based on the simulation of large changes made on doll-making materials and (ii) exploitation based on the simulation of small changes on the made dolls.

Received: March 28, 2024. Revised: April 21, 2024.

Table 4. Optimization results of engineering applications

		DOA	GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	AVOA	WSO
PV	mean	5882.901	31918.61	42099.39	21123.16	28408.74	6061.177	6463.371	7722.295	6214.961	5882.901	10512.38	6242.74	5882.913
	best	5882.901	12949.24	15010.37	6759.244	13525.85	5889.198	5926.998	6346.685	5909.136	5882.901	6593.516	5882.908	5882.901
	worst	5882.901	57355.81	88182.59	45002.15	43088.14	7061.063	7144.982	10167.43	7243.051	5882.901	19104.28	7187.374	5883.139
	std	2.94E-12	16500.98	32344.02	15934.73	13579.14	537.6844	530.3106	1879.148	617.8341	4.61E-05	4375.98	593.5255	0.083873
	median	5882.901	30708.89	35346.85	20194.28	27622.94	5905.27	6437.443	7272.22	5981.248	5882.901	10137.87	6171.503	5882.901
	rank	1	12	13	10	11	4	7	8	5	2	9	6	3
SR	mean	2996.348	8.93E+13	1.32E+14	3511.18	4.95E+13	3004.496	3031.932	3251.509	3029.562	2996.348	3230.905	3001.394	2996.35
	best	2996.348	4330.813	4772.934	3269.576	4521.608	2999.1	3005.025	3008.508	3012.594	2996.348	3094.299	2996.351	2996.348
	worst	2996.348	6.33E+14	6.39E+14	4138.362	2.27E+14	3011.336	3064.544	4517.361	3046.542	2996.348	3330.88	3008.48	2996.367
	std	1.47E-12	2.26E+14	2.86E+14	337.977	9.40E+13	5.250691	25.75427	647.0683	13.02429	1.26E-05	92.45705	6.220521	0.006835
	median	2996.348	4.98E+13	3.74E+13	3475.09	2.57E+13	3004.427	3033.202	3130.586	3029.147	2996.348	3220.991	3001.319	2996.349
	rank	1	12	13	10	11	5	7	9	6	2	8	4	3
WB	mean	1.724852	5.66E+12	6.79E+13	2.307414	2.54E+13	1.727077	1.745007	2.397287	1.74245	1.724852	2.265817	1.745101	1.724852
	best	1.724852	2.564353	2.66433	1.770318	1.977066	1.725508	1.729279	1.791787	1.732598	1.724852	1.918369	1.724895	1.724852
	worst	1.724852	1.10E+14	8.22E+14	2.583019	4.29E+14	1.730919	1.776049	4.300487	1.748876	1.724852	3.804345	1.798654	1.724852
	std	1.08E-15	3.88E+13	3.07E+14	0.315295	1.51E+14	0.002512	0.02089	1.155755	0.008069	3.72E-08	0.630209	0.034926	3.38E-09
	median	1.724852	4.975513	5.083608	2.319265	4.800336	1.726406	1.741559	2.033636	1.743335	1.724852	2.181651	1.736944	1.724852
	rank	1	11	13	9	12	4	6	10	5	3	8	7	2
TCS	mean	0.012665	0.023632	3.62E+13	0.019486	0.017921	0.012717	0.016715	0.013413	0.01291	0.012665	0.017366	0.012987	0.012666
	best	0.012665	0.01796	0.017315	0.014172	0.01738	0.012688	0.012893	0.012688	0.012712	0.012665	0.013034	0.012667	0.012665
	worst	0.012665	0.03219	3.62E+14	0.024328	0.018478	0.012736	0.017603	0.015233	0.013282	0.012665	0.086405	0.014007	0.012671
	std	1.54E-18	0.005797	1.76E+14	0.005154	0.000518	1.71E-05	0.002238	0.001355	0.000224	4.84E-09	0.025867	0.000595	1.91E-06
	median	0.012665	0.022861	0.017315	0.019165	0.017865	0.012721	0.017343	0.013096	0.012918	0.012665	0.013213	0.012839	0.012665
	rank	1	12	13	11	10	4	8	7	5	2	9	6	3
Sum rank		4	47	52	40	44	17	28	34	21	9	34	23	11
Mean rank		1	11.75	13	10	11	4.25	7	8.5	5.25	2.25	8.5	5.75	2.75
Total rank		1	11	12	9	10	4	7	8	5	2	8	6	3

The performance of DOA was evaluated in the optimization of twenty-three standard benchmark functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types. The optimization findings showed that DOA has achieved good results with its high ability in exploration, exploitation, and balance them during the search process in the solution space. In order to measure the quality of DOA in the optimization process, the obtained results were compared with the performance of twelve well-known algorithms. The simulation findings showed that DOA has provided superior performance by providing better results and getting the rank of the first best optimizer compared to competing algorithms. In addition, the performance of DOA was evaluated in real-world applications to address four engineering design issues The simulation findings showed that DOA has an effective performance in optimizing real-world and engineering applications by achieving better values for design variables and objective functions compared to competing algorithms.

The introduction of DOA raises several research proposals for future studies. The development of binary and multi-objective versions of DOA is one of the most prominent research proposals of this paper. Using DOA to solve optimization problems in different sciences and real-world applications are among the other suggestions of this paper for further studies.

Conflicts of Interest

"The authors declare no conflict of interest."

Author Contributions

Conceptualization, S.A.O, K.K, and I.A.F; methodology, TH, M.D, and K.E; software, K.E, S.G, I.L, K.K, and I.A.F; validation, K.E, M.D, S.G, and I.L; formal analysis, Z.M, M.D, K.E, and S.G; investigation, K.K, Z.M, I.A.F, and I.L; resources, S.A.O, Z.M and K.K; data curation, K.E and I.A.F; writing—original draft preparation, M.D, S.A.O, S.G, and I.L; writing—review and editing, I.A.F Z.M, and K.K; visualization, K.E; supervision, M.D; project administration, K.E, S.A.O, and S.G; funding acquisition, K.E.

References

[1] A. A. Al-Nana, I. M. Batiha, and S. Momani, "A numerical approach for dealing with fractional boundary value problems", *Mathematics,* Vol. 11, No. 19, pp. 4082, 2023.

[2] W. G. Alshanti, I. M. Batiha, M. m. A. Hammad, and R. Khalil, "A novel analytical approach for solving partial differential equations via a tensor product theory of Banach spaces", *Partial Differential Equations in Applied Mathematics,* Vol. 8, pp. 100531, 2023.

[3] A. Taheri, K. RahimiZadeh, A. Beheshti, J. Baumbach, R. V. Rao, S. Mirjalili, and A. H. Gandomi, "Partial reinforcement optimizer: An evolutionary optimization algorithm", *Expert Systems with Applications,* Vol. 238, pp. 122070, 2024.

[4] N. Singh, X. Cao, S. Diggavi, and T. Başar, "Decentralized multi-task stochastic

optimization with compressed communications," *Automatica*, Vol. 159, pp. 111363, 2024.
[5] T. Hamadneh, M. Ali, and H. AL-Zoubi, "Linear Optimization of Polynomial Rational Functions: Applications for Positivity Analysis", *Mathematics*, Vol. 8, No. 2, pp. 283, 2020.
[6] G. Kumar, R. Saha, M. Conti, T. Devgun, and R. Thomas, "GREPHRO: Nature-inspired optimization duo for Internet-of-Things", *Internet of Things*, pp. 101067, 2024.
[7] 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.
[8] P. D. Kusuma, and A. Dinimaharawati, "Extended stochastic coati optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 482-494, 2023, doi: 10.22266/ijies2023.0630.38.
[9] I. Matoušová, P. Trojovský, M. Dehghani, E. Trojovská, and J. Kostra, "Mother optimization algorithm: a new human-based metaheuristic approach for solving engineering optimization", *Scientific Reports*, Vol. 13, No. 1, pp. 10312, 2023.
[10] P. D. Kusuma, and A. Dinimaharawati, "Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 2, 2024, doi: 10.22266/ijies2024.0430.31.
[11] M. Qaraad, S. Amjad, N. K. Hussein, M. Farag, S. Mirjalili, and M. A. Elhosseini, "Quadratic interpolation and a new local search approach to improve particle swarm optimization: Solar photovoltaic parameter estimation", *Expert Systems with Applications*, vol. 236, pp. 121417, 2024.
[12] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26, No. 1, pp. 29-41, 1996.
[13] J. Kennedy, and R. Eberhart, "Particle swarm optimization", In: *Proc. of ICNN'95-international conference on neural networks*, Vol. 4, pp. 1942-1948, 1995.
[14] D. Karaboga, and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems", In: *Proc. of International fuzzy systems association world congress*, pp. 789-798. Berlin, 2007.
[15] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation", *International journal of bio-inspired computation*, Vol. 2, No. 2, pp. 78-84, 2010.
[16] O. Al-Baik, S. Alomari, O. Alssayed, S. Gochhait, I. Leonova, U. Dutta, O. P. Malik, Z. Montazeri, and M. Dehghani, "Pufferfish Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics*, Vol. 9, No. 2, pp. 65, 2024.
[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.
[18] Z. Benmamoun, K. Khlie, M. Dehghani, and Y. Gherabi, "WOA: Wombat Optimization Algorithm for Solving Supply Chain Optimization Problems", *Mathematics*, Vol. 12, No. 7, pp. 1059, 2024.
[19] A. Majumder, "Termite alate optimization algorithm: a swarm-based nature inspired algorithm for optimization problems", *Evolutionary Intelligence*, Vol. 16, No. 3, pp. 997-1017, 2023.
[20] S. Mirjalili, and A. Lewis, "The whale optimization algorithm", *Advances in engineering software*, Vol. 95, pp. 51-67, 2016.
[21] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems", *Computers & Industrial Engineering*, Vol. 158, pp. 107408, 2021.
[22] P. D. Kusuma, and M. Kallista, "Swarm Space Hopping Algorithm: A Swarm-based Stochastic Optimizer Enriched with Half Space Hopping Search", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 2, 2024, doi: 10.22266/ijies2024.0430.54.
[23] 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, pp. 116158, 2022.
[24] 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.
[25] P. D. Kusuma, and M. Kallista, "Migration-Crossover Algorithm: A Swarm-based Metaheuristic Enriched with Crossover Technique and Unbalanced Neighbourhood Search", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 1, 2024, doi: 10.22266/ijies2024.0229.59.

[26] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, “White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems”, *Knowledge-Based Systems*, pp. 108457, 2022.

[27] 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.

[28] D. E. Goldberg, and J. H. Holland, “Genetic Algorithms and Machine Learning”, *Machine Learning*, Vol. 3, No. 2, pp. 95-99, 1988.

[29] 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.

[30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing”, *Science*, Vol. 220, No. 4598, pp. 671-680, 1983.

[31] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “GSA: a gravitational search algorithm”, *Information Sciences*, Vol. 179, No. 13, pp. 2232-2248, 2009.

[32] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, “Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems”, *Computers & Structures*, Vol. 110, pp. 151-166, 2012.

[33] J. L. J. Pereira, M. B. Francisco, C. A. Diniz, G. A. Oliver, S. S. Cunha Jr, and G. F. Gomes, “Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization”, *Expert Systems with Applications*, Vol. 170, pp. 114522, 2021.

[34] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, “Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems”, *Applied Intelligence*, Vol. 51, No. 3, pp. 1531-1551, 2021.

[35] M. H. Qais, H. M. Hasanien, S. Alghuwainem, and K. H. Loo, “Propagation Search Algorithm: A Physics-Based Optimizer for Engineering Applications”, *Mathematics*, Vol. 11, No. 20, pp. 4224, 2023.

[36] 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.

[37] M. Braik, M. H. Ryalat, and H. Al-Zoubi, “A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves”, *Neural Computing and Applications*, Vol. 34, No. 1, pp. 409-455, 2022.

[38] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. P. Malik, R. A. Ramirez-Mendoza, J. Matas, J. C. Vasquez, and L. Parra-Arroyo, “A new “Doctor and Patient” optimization algorithm: An application to energy commitment problem”, *Applied Sciences*, Vol. 10, No. 17, pp. 5791, 2020.

[39] M. Dehghani, and P. Trojovský, “Teamwork Optimization Algorithm: A New Optimization Approach for Function Minimization/Maximization”, *Sensors*, Vol. 21, no. 13, pp. 4567, 2021.

[40] 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.

[41] P. Trojovský, and M. Dehghani, “A new optimization algorithm based on mimicking the voting process for leader selection”, *PeerJ Computer Science*, Vol. 8, pp. e976, 2022.

[42] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman, “Darts game optimizer: A new optimization technique based on darts game”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, pp. 286-294, 2020, doi: 10.22266/ijies2020.1031.26.

[43] Z. Montazeri, T. Niknam, J. Aghaei, O. P. Malik, M. Dehghani, and G. Dhiman, “Golf Optimization Algorithm: A New Game-Based Metaheuristic Algorithm and Its Application to Energy Commitment Problem Considering Resilience”, *Biomimetics*, 2023.

[44] M. Dehghani, Z. Montazeri, S. Saremi, A. Dehghani, O. P. Malik, K. Al-Haddad, and J. M. Guerrero, “HOGO: Hide objects game optimization”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 10, 2020, doi: 10.22266/ijies2020.0831.19.

[45] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. Malik, and V. Kumar, “Football game based optimization: An application to solve energy commitment problem,” *International Journal of Intelligent Engineering and Systems*, Vol. 13, pp. 514-523, 2020, doi: 10.22266/ijies2020.1031.45.

[46] 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, doi: 10.22266/ijies2022.0228.25.

[47] 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.

[48] 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.

[49] 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, doi: 10.22266/ijies2021.0630.46.

[50] 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, doi: 10.22266/ijies2020.0630.23.

[51] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster", *IEEE Transactions on Evolutionary computation,* Vol. 3, No. 2, pp. 82-102, 1999.

[52] 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.

[53] B. Kannan, and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design", *Journal of mechanical design,* Vol. 116, No. 2, pp. 405-411, 1994.

[54] A. H. Gandomi, and X.-S. Yang, "Benchmark problems in structural optimization", *Computational optimization, methods and algorithms,* pp. 259-281, 2011.