Received: April 19, 2024. Revised: May 19, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Sculptor Optimization Algorithm: A New Human-Inspired Metaheuristic Algorithm for Solving Optimization Problems

Tareq Hamadneh[1] Khalid Kaabneh[2] Omar AlSayed[3] Gulnara Bektemyssova[4] Zeinab Montazeri[5] Mohammad Dehghani[5]* Kei Eguchi[6]

[1] *Department of Mathematics, Al Zaytoonah University of Jordan, Amman 11733, 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 Computer Engineering,*
International Information Technology University, Almaty 050000, Kazakhstan
[5]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 7155713876, Iran*
[6]*Department of Information Electronics, Fukuoka Institute of Technology, Japan*
* Corresponding author's Email: adanbax@gmail.com

Abstract: In this paper, a new metaheuristic algorithm called Sculptor Optimization Algorithm (SOA) is introduced and designed, which imitates the sculpting process. The main idea in SOA design is derived from (i) making extensive changes to the sculpture material and (ii) making small and detailed changes to the sculpture. SOA theory is expressed and then mathematically modeled in two phases of exploration and exploitation. The performance of SOA in handling optimization applications has been evaluated to optimize the CEC 2017 test suite. The optimization results show that SOA, with its high power in managing exploration and exploitation during the search process, has been able to achieve suitable solutions for optimization problems. In addition, the quality of SOA results has been compared with the performance of twelve well-known metaheuristic algorithms. Analysis of the simulation results shows that SOA has provided superior performance compared to competing algorithms by achieving better results for most of the benchmark functions. Simulation findings show that compared to competing algorithms, SOA has been successful in handling 100% of unimodal functions, multimodal functions and hybrid functions, as well as 70% of composite functions.

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

1. Introduction

Optimization is an important process in computer science, engineering, mathematics and other scientific fields that seeks to find the best solution or optimal value for a given problem [1]. This process usually involves searching the problem solving space and making multiple changes to improve the efficiency and performance of the solution [2]. Optimization problems exist in daily life and various industries, including various system design, financial and economic problems, production planning, transportation problems, resource optimization, and many others. The main goal in these problems is to obtain a solution that optimizes the desired criteria and provides the best possible result [3]. Deterministic and stochastic approaches are two categories of methods used to solve optimization problems, each of which has its own characteristics and applications [4].

In the deterministic approach, the optimization process is performed accurately and without any reliance on chance . This means that the path and process of optimization is determined in a clear and definitive way [5, 6]. One of the advantages of this approach is high predictability and reliability, because every time the algorithm is executed, it leads to a specific and repeatable result. However, in complex problems with a large search space, deterministic methods may encounter problems and need more time to solve the problem [7]. Disadvantages of deterministic approaches have led researchers to be able to deal with complex practical problems by introducing stochastic approaches [8].

In the stochastic approach, the optimization process is performed using random elements and random search. This means that at every step of the process, decisions are made based on probabilities and chance. Metaheuristic algorithms are among the most prominent stochastic approaches that have been widely used to solve optimization problems. One of the main features of metaheuristic algorithms is the ability to use creative methods and flexibility in facing complex problems and large search space. However, due to the randomness of the optimization process by metaheuristic algorithms, the obtained results may be different in each run and it is difficult to predict the results with high accuracy [9].

This means that there is no guarantee to achieve the global optimum using metaheuristic algorithms. This is why the solutions obtained from these algorithms are called quasi-optimal. The desire of researchers to achieve better solutions for optimization problems has led to the design of several metaheuristic algorithms [10].

The main research question is that according to the existing designed metaheuristic algorithms, is there still a need to design newer metaheuristic algorithms or not? In response to this question, it should be said that: Although the designed metaheuristic algorithms have had significant success in solving optimization problems, the No Free Lunch (NFL) theorem [11] shows that there is no algorithm that performs best for all optimization problems. . Therefore, according to the NFL theorem, there is still a need to design newer and innovative metaheuristic algorithms. These algorithms may provide better performance for new problems and challenges that have not been considered so far. Also, new algorithms can provide improvements in the performance and efficiency of existing algorithms and thus contribute to progress in the field of optimization.

The innovation and novelty aspects of this paper are in the design of a new metaheuristic algorithm called Sculptor Optimization Algorithm (SOA) to deal with optimization applications in different sciences. The main contributions of this paper are listed as follows:

- SOA is introduced by the inspiration of human activities in the process of sculpture.
- The main idea of SOA is to include: (i) making extensive changes to the sculpture material and (ii) making small and detailed changes to the sculpture.
- The theory of SOA is described and its implementation steps are mathematically modeled in two phases of exploration and exploitation.
- The performance of SOA is evaluated to address the CEC 2017 test suite.
- The quality of SOA in handling optimization applications is compared with the performance of twelve well-known algorithms.

The rest of the article is structured in such a way that first the literature review is presented in section 2. Then, Sculptor Optimization Algorithm (SOA) is introduced and mathematically modeled in section 3. Simulation studies and performance evaluation of SOA in optimization applications are presented in section 4. Finally, conclusions and research proposals for future studies are provided in Section 5.

2. Literature review

In recent decades, metaheuristic algorithms have attracted a lot of attention in computer science, engineering, mathematics and other scientific fields. Using concepts such as evolution, collective motivation, random search and other similar principles, these algorithms attempt to provide optimization of different problems using different methods. Metaheuristic algorithms can be divided into four groups based on the main ideas in design: swarm-based, evolutionary-based, physics-based, and human-based methods.

Swarm-based metaheuristic algorithms are designed to solve optimization problems using ideas similar to group behavior in living communities. These algorithms are inspired by the group behavior of living organisms such as ants, anteaters, bees and birds and are used to optimize various problems including routing, planning and production problems. The Particle Swarm Optimization (PSO) is a popular metaheuristic method for solving optimization problems, which is inspired by the group behavior of birds in search of food sources. In this algorithm, the optimization problem is considered as finding the best position in a multidimensional space [12]. Ant Colony Optimization (ACO) is a metaheuristic optimization method inspired by the collective behavior of ants in search of food resources. In this algorithm, ants improve their search by exchanging information on pheromones. This method is known as an efficient method in solving complex optimization problems [13]. Some other swarm based algorithms are: Walrus Optimization Algorithm (WaOA) [14], Adax Optimization Algorithm (AOA) [15], Swarm Space Hopping Algorithm (SSHA) [16], and Migration-Crossover Algorithm (MCA) [17].

Evolutionary-based algorithms are approaches that are influenced by biological sciences, genetics, the process of evolution, and evolutionary principles such as natural selection, genetic variation, and genetic inheritance. Genetic Algorithm (GA) [18] is inspired by genetic processes in nature. In this algorithm, a population of solutions (chromosomes) is generated and mutated, and natural selection is used to select parents and evolutionary operators such as crossover and genetic mutation are used to generate new generations.

Physics-based algorithms are inspired by physical principles and laws such as the laws of flow, gravity, diffusion, etc. They optimize the problems with the help of these physical principles. Simulated Annealing (SA) [19] is an optimization method inspired by the principles of metal annealing. This algorithm uses random probabilities to accept or reject changes in the search space. As the temperature decreases over time, the probability of accepting changes also decreases, which approaches a more optimal point in the search space. Laws, forces, processes, transformations, phenomena and other physical concepts have been sources of inspiration in designing algorithms such as: Electromagnetic Field Optimization (EFO) [20], Charged System Search (CSS) [21], Prism Refraction Search (PRS) [22], and Kepler Optimization Algorithm (KOA) [23].

Human-based algorithms include approaches that use inspiration from human behavior and performance and human cognitive processes. These algorithms are usually modeled on human decision-making, learning, memory and individual development. Teaching-Learning Based Optimization (TLBO) is an optimization algorithm that is inspired by the teaching and learning process in an educational environment. This algorithm first creates a population of people (solutions), each of which is considered as a learner. Then, the optimization process is done using the communication between the teacher and the students. In this algorithm, the teacher acts as a representative of the best solutions in the population and tries to share her/his knowledge with others [24]. The principles of education and care of mother Eshrat in raising her children have been the main idea in the design of Mother Optimization Algorithm (MOA) [8]. Interactions, communications, thoughts, decisions and other human activities have been a source of inspiration in designing algorithms such as: Ali Baba and the Forty Thieves (AFT) [25], Dollmaker Optimization Algorithm (DOA) [26], and Human Mental Search (HMS) [27].

Based on the best knowledge obtained from literature review, so far, no metaheuristic algorithm has been designed inspired by human activities in art and sculpting process. Meanwhile, making changes on sculpting materials in order to make a sculpture is an intelligent process that can be the main idea in designing a new optimizer. In order to address this research gap, in this paper, a new meta-heuristic algorithm based on the mathematical modeling of sculpting is introduced and designed, which is discussed in the next section.

3. Sculptor optimization algorithm

In this section, the proposed Sculptor Optimization Algorithm (SOA) approach is introduced and mathematically modeled.

3.1 Inspiration of SOA

Sculpture, an enduring art form spanning millennia, showcases the mastery of three-dimensional expression through various mediums such as stone, metal, and clay. From the iconic works of ancient civilizations to the contemporary innovations of modern sculptors, this art form continues to captivate and inspire audiences worldwide. Renowned sculptors like Michelangelo, Rodin, and Moore have left indelible marks on the artistic landscape, pushing the boundaries of form, texture, and symbolism. Sculpture's presence in public spaces and galleries serves as a testament to its cultural significance and enduring relevance. Through the tactile and visual experience, it offers, sculpture transcends language barriers and communicates profound emotions and ideas. As an integral part of human history and expression, sculpture continues to shape and enrich our collective cultural heritage [28].

In the sculpting process, the sculptor tries to achieve a work of art by making changes on the material through carving. A wide range of materials including clay, glue, stone, metal, fabric, glass, wood, concrete, rubber and composite materials can be used in this art. In general, the two strategies of the sculptor in the sculpture process, which are more significant, are as follows: (i) making extensive changes to the sculpting material and (ii) making small precise changes to complete the sculpture.

What is evident is that sculpting is an intelligent human activity in which the sculptor's strategies to create a work of art correspond to the search process in the problem solving space in order to achieve the optimal solution. These intelligent strategies of the sculptor while making changes to the sculpting material are the main source of inspiration in the SOA design discussed below.

3.2 Algorithm initialization

The proposed SOA approach is a population-based optimizer that can achieve suitable solutions for optimization problems by benefiting from the search power of its members in the problem solving space. The proposed approach of SOA is a crowd-based optimizer that can achieve suitable solutions for optimization problems by benefiting from the search power of its members in the problem solving space. Each SOA member represents a candidate solution to the problem, which contains information on the values of the decision variables. Therefore, each SOA member is mathematically modeled using a vector where each element of this vector represents a decision variable. The SOA members

together form the SOA population, which can be mathematically modeled by the community of these vectors using a matrix according to Eq. (1). The initial position of SOA members in the problem solving space is generated completely randomly using Eq. (2).

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

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

Here, X is the SOA's population matrix, X_i is the ith member (i.e., candidate solution), $x_{i,d}$ is its dth dimension in the search space (i.e., decision variable), N is the number of population members (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.

Since each SOA member is a candidate solution for the given problem, corresponding to each SOA member the objective function can be evaluated. 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 SOA member.

3.3 Mathematical modelling of SOA

This section deals with the mathematical modeling of the proposed SOA approach. In order to update the position of SOA members in the problem solving space, it is inspired by the sculptor's strategies during the sculpting process. In this process, the sculptor uses two main strategies in order to make the sculpture: (i) making extensive changes to the sculpture material and (ii) making small and precise changes in order to finalize the sculpture. In SOA design, inspired by these strategies, the position of SOA members in the problem solving space has been updated in two phases of exploration and exploitation. Each of these SOA upgrade phases is described and modeled in detail below.

3.3.1 Phase 1: Making extensive and large changes to the sculpting materials (exploration phase)

A sculptor uses an existing model or mental image to create a sculpture. Then, based on this model, he tries to make changes on the raw materials of sculpture. Making these changes can be done through carving. In SOA design, modeling these extensive changes on sculptural materials according to the considered pattern, leads to extensive changes in the position of SOA members and as a result, increases the exploration power of the algorithm in order to manage the global search.

In SOA design, the position of the best population member is assumed as the sculpting pattern. Then corresponding to the sculpting process in which the sculptor tries to bring the shape of the raw materials closer to the intended pattern, in SOA, the position of the population members changes based on the change of movement towards the position of the best member of the population. Based on the simulation of making changes on the raw materials of sculpture, a new position in the problem solving space is calculated for each SOA member using Eq. (4). Then, if this new position improves the value of the objective function, it replaces the previous position of the corresponding member using Eq. (5).

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

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

Where, $best$ is the best member of population, $best_j$ is its jth dimension, X_i^{P1} is the new position for the ith member based on first phase of SOA, $x_{i,j}^{P1}$ 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: Making small and precise changes on the statue (exploitation phase)

Based on the considered pattern, the sculptor makes major and extensive changes on the sculpture materials. After that, the sculptor tries to make small changes on the sculpture with high precision in order to take care of the exact details of the model and complete the sculpture. In SOA design, the modeling of these small precise changes leads to the creation of small changes in the position of the SOA members and, as a result, increasing the exploitation power of the algorithm in order to manage the local search.

In the SOA design, corresponding to the sculptor's strategy that tries to achieve a better shape of the sculpture by making precise small changes and make it completely similar to the model, the position of SOA members is also improved with small changes to converge to better solutions for the given problem. Based on the simulation of this sculpting strategy, a new position has been calculated for each SOA member using equation (6). Then, if this new position improves the value of the objective function, it replaces the previous position of the corresponding member using equation (7).

$$x_{i,j}^{P2} = \frac{T-t}{T} \cdot x_{i,j} + \frac{t}{T} \cdot best_j \tag{6}$$

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

Where, X_i^{P2} is the new calculated position for the ith SOA member based on second phase of SOA, $x_{i,j}^{P2}$ is the its jth dimension, F_i^{P2} is its objective function value, T is the maximum number of iterations, and t is the iteration counter.

3.4 Repetition process, pseudocode, and flowchart of SOA

After updating all SOA members based on the first and second phases, the first iteration of the algorithm is completed. Considering that SOA is an iteration-based algorithm, the algorithm enters the next iteration with the updated values. The process of updating the position of SOA members in the problem solving space continues until the last iteration of the algorithm based on Eqs. (4) to (7). At the end of each SOA iteration, the best solution obtained is stored and updated. After the full implementation of SOA, the best solution recorded during the iterations of the algorithm is output as a solution for the given problem. The steps of SOA implementation are shown as a flowchart in Figure 1.

4. Simulation studies and results

This section is dedicated to the performance analysis of SOA to deal with optimization issues. With this view, CEC 2017 tests suite has been selected, which consists of thirty standard benchmark functions C17-F1 to C17-F30. These benchmark functions are classified into four types: unimodal functions of C17-F1 to C17-F3, multimodal functions of C17-F4 to C17-F10, hybrid functions of C17-F11 to C17-F20, and composite functions of C17-F21 to C17-F30. Similar to other papers, the C17-F2 functional is excluded from the simulation studies due to its unstable behavior. Comprehensive and detailed information on the CEC 2017 test suite is available at source [29]. Twelve well-known metaheuristic algorithms have been selected to compare with the performance of SOA in handling optimization problems. These competing algorithms are: GA [18], PSO [12], GSA [30], TLBO [24], MVO [31], GWO [32], WO [33], MPA [34], TSA [35], RSA [36], AVOA [37], and WSO [38]. In order to report the simulation results, six statistical indicators have been used: mean, best, worst, standard deviation (std), median, and rank.

4.1 Evaluation of unimodal functions

C17-F1 and C17-F3 functions are of unimodal type. These types of problems are able to challenge the exploitation ability of metaheuristic algorithms because they lack local optima. The outputs of SOA and competing algorithms on these functions are reported in Table 1. Based on the reported results, SOA is ranked as the first best optimizer for both unimodal functions C17-F1 and C17-F3.

Received: April 19, 2024. Revised: May 19, 2024.

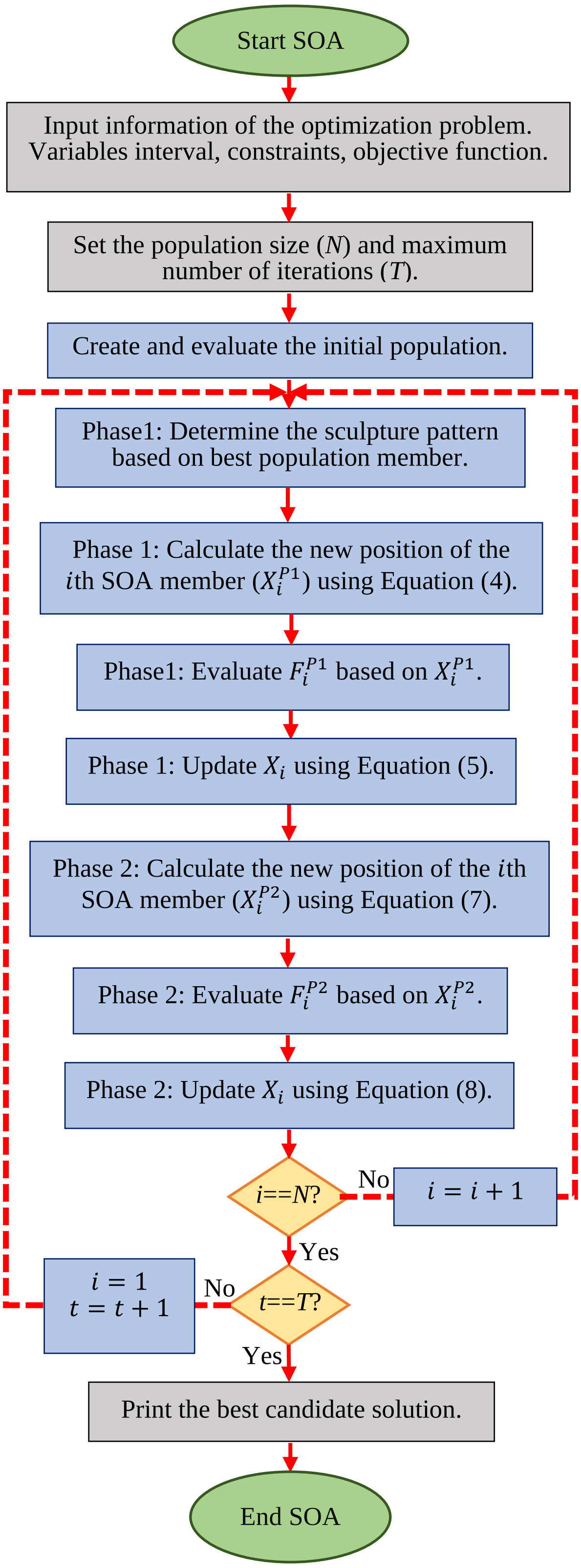

Figure. 1 flowchart of SOA

Analysis of the simulation results shows that SOA, with the benefit of high power in exploitation and management of local search, has provided superior performance compared to competing algorithms for handling unimodal functions.

4.2 Evaluation of multimodal functions

Functions C17-F4 to C17-F10 are of multimodal type. These types of problems, because they have a large number of local optima, are able to challenge the exploration ability of metaheuristic algorithms. The results of the implementation of SOA and competing algorithms on multimodal functions are reported in Table 2. Based on the results, SOA has obtained the rank of the first best optimizer for all seven multimodal functions C17-F4 to C17-F10. What is evident from the analysis of the simulation results, SOA with its capability in exploration and global search management has provided superior performance to handle multimodal functions compared to competing algorithms.

4.3 Evaluation of hybrid functions

Functions C17-F11 to C17-F20 are of hybrid type. These types of functions are complex optimization problems that challenge the ability of metaheuristic algorithms to balance exploration and exploitation. The outputs of SOA and competing algorithms for optimizing these functions are reported in Table 3. Based on the results, SOA has been ranked the first best optimizer for all ten benchmark functions C17-F11 to C17-F20. Analysis of the simulation results shows that SOA, with the ability to balance exploration and exploitation, has provided superior performance for handling hybrid functions compared to competing algorithms.

4.4 Evaluation of composite functions

Functions C17-F21 to C17-F30 are of composite type. These types of functions are complex and very challenging optimization problems. Achieving the optimal solution of these problems requires a high capability in exploration and exploitation. The results of the implementation of SOA and competing algorithms on composite functions are reported in Table 4. Based on the obtained results, SOA has obtained the rank of the first best optimizer for handling functions C17-F21, C17-F23, C17-F24, and C17-F27 to C17-F30. What is concluded from the analysis of the simulation results is that SOA has a high ability in exploration and exploitation, which has led to its superiority compared to competing algorithms for dealing with composite functions.

The performance of SOA and competing algorithms to handle the CEC 2017 test suite is plotted using boxplot diagrams in Figure 2.

Received: April 19, 2024. Revised: May 19, 2024.

Table 1. Optimization results of unimodal functions (C17-F1 and C17-F3)

		SOA	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
C17-F1	mean	100	3.75E+09	29922059	6.68E+09	52883767	1.16E+09	34117332	29924452	87329488	1.26E+08	29920043	29921604	37633141
	best	100	3.17E+09	11269.62	5.85E+09	16720.51	2.43E+08	3803052	15145.17	27519.96	43087919	9494.238	9654.111	7479419
	worst	100	4.83E+09	1.09E+08	7.93E+09	1.92E+08	2.48E+09	1.12E+08	1.09E+08	3.17E+08	2.31E+08	1.09E+08	1.09E+08	1.17E+08
	std	0	7.99E+08	57014628	1.03E+09	1.01E+08	1.05E+09	56192559	57017477	1.66E+08	90300260	57015766	57014846	57510733
	median	100	3.5E+09	5487530	6.46E+09	9692814	9.64E+08	10454058	5486941	16005629	1.14E+08	5484408	5487217	12861838
	rank	1	12	4	13	8	11	6	5	9	10	2	3	7
C17-F3	mean	300	6902.565	1240.146	7321.369	1959.553	8332.828	2169.272	1238.949	3040.511	1516.273	7718.261	1238.914	10656.29
	best	300	4876.621	716.5073	5384.348	1036.187	4775.201	990.1539	716.5156	1515.706	827.924	4787.324	716.5073	4829.611
	worst	300	8355.404	2197.303	8986.303	3648.869	10826.74	3420.841	2194.748	5830.173	2535.783	10138.86	2194.667	16260.72
	std	0	1615.175	737.8683	1772.477	1302.823	2778.987	1180.036	737.1237	2151.424	824.2112	2407.768	737.0895	6641.81
	median	300	7189.118	1023.387	7457.413	1576.577	8864.688	2133.045	1022.267	2408.083	1350.693	7973.429	1022.241	10767.41
	rank	1	9	4	10	6	12	7	3	8	5	11	2	13
Sum rank		2	21	8	23	14	23	13	8	17	15	13	5	20
Mean rank		1	10.5	4	11.5	7	11.5	6.5	4	8.5	7.5	6.5	2.5	10
Total rank		1	9	3	10	5	10	4	3	7	6	4	2	8

Table 2. Optimization results of multimodal functions (C17-F4 to C17-F10)

		SOA	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
C17-F4	mean	400	756.2803	407.0778	1023.244	408.364	518.8691	420.3609	406.1551	411.6275	409.9557	406.9487	417.2115	413.5689
	best	400	609.4061	402.9129	699.3513	403.6598	452.796	407.4143	403.1427	406.0323	408.1579	404.424	402.1355	410.6779
	worst	400	891.6641	412.6971	1344.147	417.0357	591.9035	449.9685	411.0114	428.0941	415.0862	412.7573	447.9672	417.2304
	std	0	141.1031	4.411591	297.7355	6.634736	75.38316	21.5205	3.658529	11.86509	3.697816	4.266461	22.4691	3.051428
	median	400	762.0256	406.3506	1024.739	406.3802	515.3884	412.0305	405.2332	406.1918	408.2894	405.3067	409.3717	413.1836
	rank	1	12	4	13	5	11	10	2	7	6	3	9	8
C17-F5	mean	501.2464	547.2709	533.4406	552.357	512.9521	546.7992	531.418	520.0626	513.0451	526.8712	539.893	522.8261	522.8996
	best	500.9951	536.3623	522.7623	541.7054	508.654	532.5737	518.8646	510.8694	508.5295	521.7144	536.1189	510.2547	519.4749
	worst	501.9917	554.6879	545.4724	563.7161	518.8112	567.246	554.3878	528.8351	520.3343	531.6904	546.0748	538.187	529.1864
	std	0.537048	8.998921	12.35993	13.06621	5.220377	17.10149	17.95214	7.931909	5.501677	4.477696	4.663814	14.2132	4.740026
	median	500.9993	549.0167	532.7639	552.0032	512.1717	543.6885	526.2097	520.2729	511.6583	527.0401	538.6891	521.4313	521.4685
	rank	1	12	9	13	2	11	8	4	3	7	10	5	6
C17-F6	mean	600	621.8597	611.8238	627.2659	601.1761	616.783	615.685	601.8073	601.132	604.9192	611.7485	605.2937	607.1626
	best	600	619.421	611.179	624.9619	600.7619	610.3021	605.318	600.5168	600.5987	603.4907	602.5172	601.486	604.9079
	worst	600	624.8138	613.3259	630.0351	601.9896	627.2811	630.0504	603.2542	601.7266	607.1037	624.07	613.123	609.9837
	std	0	2.508793	1.08938	2.402352	0.612459	7.92843	11.23482	1.344121	0.504635	1.719831	10.83326	5.771931	2.370398
	median	600	621.6021	611.3952	627.0333	600.9764	614.7744	613.6858	601.7292	601.1013	604.5412	610.2034	603.2829	606.8794
	rank	1	12	9	13	3	11	10	4	2	5	8	6	7
C17-F7	mean	711.1267	779.206	752.1955	777.8195	725.1737	793.7458	749.9011	729.2947	726.0821	743.2779	720.2082	730.5259	733.2535
	best	710.6726	770.8391	739.8917	770.848	719.4516	773.3004	739.7043	717.9447	717.4924	738.9536	716.384	723.4819	724.105
	worst	711.7995	791.1335	767.5882	785.4081	734.1237	820.3607	767.0287	748.051	743.6843	747.8681	725.9581	737.387	741.9442
	std	0.553542	9.737206	13.72203	6.774961	6.905284	22.70952	13.38803	14.06779	13.00362	4.831107	4.977967	6.433275	8.076162
	median	711.0174	777.4256	750.651	777.5109	723.5598	790.6611	746.4357	725.5916	721.5759	743.145	719.2454	730.6173	733.4825
	rank	1	12	10	11	3	13	9	5	4	8	2	6	7
C17-F8	mean	801.4928	837.962	826.0171	840.9305	813.8239	837.3567	829.4792	813.2707	815.9257	830.3692	818.5799	820.4988	816.5487
	best	800.995	831.3862	817.0271	833.1841	809.4682	826.323	819.4513	812.0803	810.5759	825.9966	815.1144	815.8128	815.15
	worst	801.9912	842.3144	838.1851	845.2605	816.7777	848.2774	837.2072	814.6034	820.9421	834.1363	821.884	825.2081	819.8687
	std	0.621323	5.293889	9.550151	5.739528	3.495746	10.38283	8.40996	1.11946	4.687493	4.555997	3.430197	4.892419	2.402956
	median	801.4926	839.0736	824.4281	842.6387	814.5249	837.4131	830.6291	813.1995	816.0924	830.6719	818.6605	820.487	815.588
	rank	1	12	8	13	3	11	9	2	4	10	6	7	5
C17-F9	mean	900	1249.748	1094.449	1278.886	907.5434	1221.684	1218.105	904.6387	911.994	911.9231	904.1092	906.912	907.4863
	best	900	1158.27	935.6922	1211.821	900.4138	1081.567	1019.533	900.2541	900.5761	906.6673	900.1974	900.8462	903.782
	worst	900	1339.42	1414.273	1377.013	920.2232	1408.205	1399.893	911.4106	933.2975	916.1862	911.4079	912.0021	913.6208
	std	0	92.66405	239.7117	75.91288	9.859639	153.9281	172.1806	5.829994	16.59145	4.361221	5.684322	5.049406	4.598963
	median	900	1250.652	1013.916	1263.355	904.7684	1198.482	1226.498	903.445	907.0513	912.4194	902.4158	907.3998	906.2712
	rank	1	12	9	13	6	11	10	3	8	7	2	4	5
C17-F10	mean	1006.179	2100.705	1756.048	2280.04	1585.158	1922.863	1917.888	1757.991	1721.75	2014.101	2083.535	1866.061	1715.376
	best	1000.284	1939.136	1499.997	2165.739	1439.603	1787.018	1538.736	1484.314	1536.19	1694.81	1898.082	1550.309	1516.019
	worst	1012.668	2213.922	2169.893	2490.567	1725.234	2046.405	2352.013	2061.386	1987.267	2293.663	2225.296	2222.758	2065.043
	std	7.194373	125.0343	331.5048	159.2575	127.1298	142.257	404.4252	272.1116	207.2191	266.9464	148.2689	298.1319	262.0966
	median	1005.882	2124.881	1677.151	2231.926	1587.897	1929.013	1890.402	1743.131	1681.772	2033.965	2105.381	1845.588	1640.22
	rank	1	12	5	13	2	9	8	6	4	10	11	7	3
Sum rank		7	84	54	89	24	77	64	26	32	53	42	44	41
Mean rank		1	12	7.714286	12.71429	3.428571	11	9.142857	3.714286	4.571429	7.571429	6	6.285714	5.857143
Total rank		1	12	9	13	2	11	10	3	4	8	6	7	5

5. Conclusions and future works

In this paper, a new metaheuristic algorithm called Sculptor Optimization Algorithm (SOA) was proposed to deal with optimization tasks in various sciences. The main idea in SOA design includes two main steps in the sculpting process: (i) making extensive changes to the sculpting material and (ii) making small and detailed changes to the sculpture. SOA theory was stated and then mathematically modeled in two phases of exploration and exploitation in order to use it to solve optimization problems. The performance of SOA was challenged to handle the CEC 2017 tests suite. The optimization results showed that SOA with high capability in exploration and exploitation and balancing them can scan the problem solving space well at both global and local levels. The performance of SOA in handling the CEC 2017 test suite was compared with the performance of twelve competing algorithms. Analysis of the simulation results showed that SOA has provided superior performance for handling the CEC 2017 test suite by achieving better results for most of the benchmark functions compared to competing algorithms. Simulation findings showed that in competition with compared algorithms, the implementation of SOA has been successful in 100% of unimodal, multimodal, and hybrid functions, as well as in 70% of composite functions.

Table 3. Optimization results of fixed-dimensional multimodal functions (C17-F11 to C17-F20)

		SOA	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
C17-F11	mean	1100	3063.69	1150.529	3003.955	1136.505	3968.264	1152.133	1136.806	1154.953	1152.103	1144.444	1147.277	1957.253
	best	1100	2247.869	1135.091	1349.683	1115.993	3861.475	1140.04	1118.92	1121.497	1139.939	1131.482	1130.001	1153.561
	worst	1100	3834.766	1173.893	4624.487	1182.147	4009.959	1163.076	1151.891	1227.634	1173.827	1156.568	1186.227	4304.387
	std	0	779.0951	19.26855	1598.144	33.52295	77.20759	10.19939	14.68646	53.47542	17.34002	13.58487	28.33268	1691.322
	median	1100	3086.062	1146.567	3020.824	1123.939	4000.811	1152.709	1138.206	1135.341	1147.324	1144.863	1136.441	1185.531
	rank	1	12	6	11	2	13	8	3	9	7	4	5	10
C17-F12	mean	1352.959	2.32E+08	1204662	4.63E+08	855345	1164745	2026108	1157820	1410927	3794627	1152100	488695.7	879888.1
	best	1318.646	51838247	875666.2	1.03E+08	28538.66	368850.8	631599.3	524832.5	45298.15	1405172	953339.3	22586.61	373601.3
	worst	1438.176	4.05E+08	1720755	8.09E+08	1338743	1477682	3315961	2133979	2208512	6618533	1457785	765772.2	1218963
	std	61.92816	1.93E+08	433371.3	3.86E+08	624011.5	576360.4	1212636	743081	1030488	2779662	236157.3	352829.1	427970.8
	median	1327.506	2.36E+08	1111114	4.7E+08	1027049	1406223	2078435	986233	1694949	3577402	1098639	583211.9	963494.2
	rank	1	12	8	13	3	7	10	6	9	11	5	2	4
C17-F13	mean	1305.324	11271468	15534.32	22532083	7082.466	11868.18	8487.731	7930.465	10270.21	14481.78	10120.98	7860.149	39202.8
	best	1303.114	942808.2	6702.009	1872606	4664.683	8536.728	5714.575	3725.338	6490.676	12576.06	6684.297	5163.974	9163.152
	worst	1308.508	37403222	23614.33	74794117	9272.744	15450.7	13306.72	11676.69	14345.6	16017.11	14213.15	14517.59	122902.8
	std	2.456412	18869592	9642.617	37738263	2111.285	3058.152	3587.353	4078.397	3479.193	1634.369	3347.104	4839.5	60334.55
	median	1304.837	3369922	15910.48	6730804	7196.219	11742.65	7464.815	8159.916	10122.28	14666.98	9793.239	5879.518	12372.61
	rank	1	12	10	13	2	8	5	4	7	9	6	3	11
C17-F14	mean	1400.746	3486.045	2131.524	4312.816	2077.917	3026.818	1801.676	1836.361	2344.455	1848.783	4456.204	2770.391	9308.219
	best	1400	2580.615	1648.716	3573.385	1447.777	1487.574	1475.002	1443.103	1462.201	1506.081	3521.854	1446.131	2951.204
	worst	1400.995	4204.181	2801.392	5031.203	3604.628	4169.238	2714.541	3007.735	4956.032	2763.393	6655.262	4992.667	17455.83
	std	0.537676	761.9486	597.0581	861.8674	1113.631	1291.514	658.2185	844.019	1881.761	659.6514	1609.647	1806.869	6511.315
	median	1400.995	3579.693	2037.993	4323.337	1629.631	3225.229	1508.58	1447.303	1479.793	1562.829	3823.85	2321.383	8412.923
	rank	1	10	6	11	5	9	2	3	7	4	12	8	13
C17-F15	mean	1500.331	8761.324	5466.445	11092.38	4599.096	6584.602	6070.15	3002.404	5805.102	3112.191	17656.39	7893.021	4975.745
	best	1500.001	3992.192	2583.287	3893.175	3338.2	3621.253	3683.59	2224.742	3565.752	2312.567	9726.436	4245.847	3518.584
	worst	1500.5	14386.76	10645.26	22277.42	5487.521	9453.661	11099.6	3381.437	6888.372	3542.15	25610.91	11983.31	7356.81
	std	0.254447	4754.049	3843.67	8864.015	978.2623	2741.765	3684.723	572.8593	1650.07	589.8147	7999.176	3443.864	1915.577
	median	1500.413	8333.171	4318.617	9099.464	4785.332	6631.747	4748.704	3201.719	6383.143	3297.024	17644.1	7671.464	4513.793
	rank	1	11	6	12	4	9	8	2	7	3	13	10	5
C17-F16	mean	1600.76	1917.642	1781.375	1917.041	1699.167	1937.293	1873.795	1785.751	1728.2	1694.359	1954.261	1856.19	1776.64
	best	1600.356	1846.802	1704.781	1820.922	1632.794	1777.516	1758.519	1688.397	1615.792	1652.71	1833.067	1796.135	1754.912
	worst	1601.12	2037.88	1819.379	2102.949	1752.381	2064.649	1980.977	1859.349	1824.939	1763.102	2115.193	1960.028	1800.647
	std	0.341437	92.42254	56.0077	136.4773	53.43385	144.6887	104.899	76.87336	93.26789	51.70705	131.0494	77.31046	27.18145
	median	1600.781	1892.944	1800.671	1872.146	1705.747	1953.504	1877.841	1797.63	1736.035	1680.812	1934.391	1834.299	1775.501
	rank	1	11	6	10	3	12	9	7	4	2	13	8	5
C17-F17	mean	1700.099	1799.701	1756.886	1801.102	1746.884	1790.485	1816.535	1817.113	1768.4	1761.744	1819.732	1757.794	1760.169
	best	1700.02	1785.393	1730.94	1789.525	1723.993	1765.451	1762.108	1759.876	1724.415	1742.217	1741.846	1738.363	1744.522
	worst	1700.332	1833.87	1781.801	1825.248	1807.833	1825.17	1882.911	1923.016	1871.267	1801.569	1937.864	1794.39	1796.445
	std	0.1677	24.87536	27.41091	17.67651	43.9624	27.10862	54.23719	81.38981	74.4984	29.74854	102.4209	27.02606	26.30613
	median	1700.022	1789.77	1757.4	1794.817	1727.856	1785.66	1810.561	1792.78	1738.958	1751.595	1799.61	1749.212	1749.855
	rank	1	9	3	10	2	8	11	12	7	6	13	4	5
C17-F18	mean	1805.36	1875796	14553.73	3734699	14025.85	14686.23	22045.18	20500.7	19819.6	26099.67	13151	21108.7	15180.32
	best	1800.003	107012.2	10485.35	196008.3	4886.176	11897.84	7573.227	9046.944	6302.817	17861.54	7536.731	6935.599	11015.06
	worst	1820.451	5427122	20408.05	10832095	21029.91	17678.34	32579.46	32259.77	33438.36	30902.42	19219.23	34144.34	22296.33
	std	10.87647	2665637	4598.356	5328624	8932.401	2990.043	11455.21	10244.15	14872.45	6146.922	6278.492	14095.42	5307.324
	median	1800.492	984525.7	13660.77	1955346	15093.66	14584.36	24014.01	20348.04	19768.61	27817.36	12924.02	21677.43	13704.95
	rank	1	12	4	13	3	5	10	8	7	11	2	9	6
C17-F19	mean	1900.445	264272.7	6234.047	462383.2	5507.774	83967.43	24617.07	3097.611	5367.414	4917.811	28292.72	18166.37	5890.063
	best	1900.039	17477.11	2133.648	31255.23	2225.938	2587.747	6291.868	1925.566	1944.465	2008.976	11982.19	2997.515	4498.093
	worst	1901.559	555619.3	13382.69	990034.4	9922.604	164709.7	46406.16	5966.808	13752.24	8880.879	39641.77	50974.43	7138.072
	std	0.804778	249887.4	5740.249	468112	3849.614	99452.88	17900.13	2089.672	6105.018	3447.41	13270.89	24274.41	1182.04
	median	1900.09	241997.3	4709.923	414121.5	4941.277	84286.12	22885.12	2249.034	2886.474	4390.695	30773.45	9346.766	5962.044
	rank	1	12	7	13	5	11	9	2	4	3	10	8	6
C17-F20	mean	2000.312	2198.654	2169.524	2203.849	2118.339	2193.613	2193.1	2149.241	2169.102	2105.043	2223.918	2168.489	2090.788
	best	2000.312	2151.123	2104.379	2161.363	2092.235	2119.312	2148.212	2084.434	2130.289	2084.531	2170.441	2148.493	2077.153
	worst	2000.312	2232.356	2237.314	2266.054	2164.243	2254.59	2233.363	2206.523	2244.828	2132.802	2310.774	2179.706	2116.654
	std	0	40.35506	67.19923	51.80206	34.22326	60.34035	50.19287	56.12904	55.80565	22.45678	70.70774	15.07987	19.18246
	median	2000.312	2205.569	2168.201	2193.99	2108.439	2200.276	2195.413	2153.005	2150.645	2101.419	2207.229	2172.879	2084.672
	rank	1	11	8	12	4	10	9	5	7	3	13	6	2
Sum rank		10	112	64	118	33	92	81	52	68	59	91	63	67
Mean rank		1	11.2	6.4	11.8	3.3	9.2	8.1	5.2	6.8	5.9	9.1	6.3	6.7
Total rank		1	11	4	12	2	10	9	3	6	7	9	5	8

Received: April 19, 2024. Revised: May 19, 2024.

Table 4. Optimization results of fixed-dimensional multimodal functions (C17-F21 to C17-F30)

		SOA	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
C17-F21	mean	2200	2300.103	2247.697	2282.604	2276.106	2320.609	2310.583	2273.452	2312.833	2303.926	2348.855	2316.437	2302.93
	best	2200	2268.515	2241.656	2254.262	2273.04	2252.377	2250.52	2237.227	2308.642	2241.012	2335.977	2309.724	2257.741
	worst	2200	2318.128	2264.12	2297.238	2279.462	2351.274	2338.081	2308.933	2317.783	2327.82	2360.02	2321.726	2324.178
	std	0	24.50981	11.85089	21.13804	2.873362	50.19924	43.67406	43.28334	4.062778	45.51693	10.75618	6.500953	32.90509
	median	2200	2306.884	2242.506	2289.459	2275.96	2339.392	2326.866	2273.824	2312.454	2323.435	2349.713	2317.149	2314.901
	rank	1	6	2	5	4	12	9	3	10	8	13	11	7
C17-F22	mean	2300.073	2601.921	2308.822	2706.422	2306.216	2574.088	2318.53	2293.618	2308.572	2315.761	2302.94	2311.631	2314.683
	best	2300	2517.529	2307.673	2569.484	2301.051	2398.146	2313.336	2254.203	2301.262	2309.889	2300.433	2300.851	2310.279
	worst	2300.29	2704.55	2310.511	2811.582	2313.785	2714.991	2323.412	2309.771	2322.334	2328.166	2307.652	2330.589	2319.052
	std	0.156805	95.07119	1.483627	109.5114	5.823536	151.7362	4.909165	28.50917	10.475	9.071844	3.58869	14.28141	4.609822
	median	2300	2592.804	2308.553	2722.312	2305.013	2591.608	2318.686	2305.249	2305.345	2312.494	2301.838	2307.542	2314.701
	rank	2	12	6	13	4	11	10	1	5	9	3	7	8
C17-F23	mean	2600.919	2670.881	2632.344	2670.742	2614.104	2685.727	2636.706	2618.002	2613.721	2632.656	2730.608	2633.8	2641.583
	best	2600.003	2642.534	2622.744	2649.694	2610.479	2628.799	2626.728	2610.936	2607.797	2627.811	2687.469	2627.05	2630.781
	worst	2602.87	2690.437	2646.295	2699.677	2618.169	2713.095	2647.931	2627.89	2620.421	2637.415	2819.652	2639.835	2645.289
	std	1.427016	23.35791	11.0774	24.23006	4.127968	41.65202	12.19392	8.139761	7.035086	5.537488	66.93686	5.847385	7.78478
	median	2600.403	2675.278	2630.169	2666.799	2613.885	2700.506	2636.083	2616.59	2613.334	2632.7	2707.655	2634.157	2645.131
	rank	1	11	5	10	3	12	8	4	2	6	13	7	9
C17-F24	mean	2630.488	2764.442	2760.943	2814.948	2671.057	2695.758	2756.352	2705.621	2748.586	2753.216	2747.725	2759.602	2731.744
	best	2516.677	2718.766	2744.698	2793.26	2653.37	2606.178	2730.601	2577.696	2724.159	2736.761	2592.422	2749.135	2614.231
	worst	2732.32	2820.165	2772.087	2854.87	2679.713	2789.462	2776.12	2755.024	2763.782	2761.838	2850.265	2772.633	2788.86
	std	125.9143	53.2863	12.93159	29.61994	13.05055	111.1898	20.43821	92.38059	19.35181	12.30563	118.6428	10.54539	86.00429
	median	2636.477	2759.418	2763.494	2805.83	2675.573	2693.697	2759.343	2744.883	2753.202	2757.133	2774.107	2758.319	2761.943
	rank	1	12	11	13	2	3	9	4	7	8	6	10	5
C17-F25	mean	2932.639	3091.214	2922.151	3160.397	2925.042	3066.509	2918.25	2927.799	2938.682	2935.293	2927.911	2928.608	2947.566
	best	2898.047	3023.747	2909.466	3117.872	2916.384	2919.626	2818.682	2908.163	2921.353	2917.995	2909.224	2914.043	2939.97
	worst	2945.793	3243.112	2947.951	3211.861	2931.109	3411.791	2953.696	2944.46	2945.886	2949.426	2943.58	2945.608	2956.956
	std	24.95556	110.3379	18.8832	42.91059	6.731305	250.96	71.8001	20.06702	12.53453	17.24481	18.54097	16.81046	9.100078
	median	2943.359	3048.999	2915.594	3155.927	2926.337	2967.31	2950.31	2929.286	2943.744	2936.876	2929.42	2927.391	2946.668
	rank	7	12	2	13	3	11	1	4	9	8	5	6	10
C17-F26	mean	2900	3499.239	3077.316	3586.787	3098.186	3497.864	3210.621	3025.019	3264.622	3226.114	3655.909	3027.585	3023.095
	best	2900	3290.128	2903.087	3407.781	2918.499	3083.912	3059.341	2923.799	2969.095	2972.016	2903.086	2964.107	2797.308
	worst	2900	3611.292	3412.915	3747.131	3502.65	3866.17	3379.118	3244.484	3905.265	3884.648	4195.208	3183.389	3138.854
	std	4.01E-13	163.1586	258.985	150.1993	293.4519	355.2769	151.1654	159.5869	465.8576	475.4231	586.8473	113.2766	167.3004
	median	2900	3547.768	2996.631	3596.119	2985.797	3520.687	3202.013	2965.896	3092.064	3023.897	3762.671	2981.422	3078.11
	rank	1	11	5	12	6	10	7	3	9	8	13	4	2
C17-F27	mean	3089.518	3176.115	3118.615	3191.527	3108.568	3157.673	3167.774	3099.996	3116.061	3115.392	3188.183	3129.16	3144.862
	best	3089.518	3136.384	3096.366	3115.935	3092.989	3101.039	3161.634	3091.385	3094.428	3096.39	3172.912	3096.172	3110.771
	worst	3089.518	3215.615	3179.335	3311.515	3148.421	3199.53	3178.092	3119.485	3176.616	3144.959	3203.84	3180.956	3177.487
	std	2.84E-13	36.71208	43.77795	91.65186	28.80773	45.89268	8.00007	14.13792	43.67616	25.46987	14.95387	39.61884	31.19824
	median	3089.518	3176.231	3099.379	3169.329	3096.431	3165.06	3165.686	3094.558	3096.599	3110.11	3187.991	3119.756	3145.596
	rank	1	11	6	13	3	9	10	2	5	4	12	7	8
C17-F28	mean	3100	3536.129	3272.859	3628.791	3261.347	3502.346	3306.083	3274.576	3344.184	3331.172	3413.525	3318.443	3279.572
	best	3100	3472.133	3199.401	3590.591	3176.206	3342.466	3230.851	3132.423	3194.399	3281.266	3365.55	3249.938	3161.754
	worst	3100	3583.114	3322.616	3680.388	3300.575	3655.191	3390.01	3389.676	3411.068	3389.833	3441.345	3372.465	3470.321
	std	0	56.43295	63.68835	42.42686	61.98403	166.1419	72.81263	140.3759	108.7724	48.34495	36.13458	54.87407	144.199
	median	3100	3544.634	3284.71	3622.092	3284.304	3505.863	3301.736	3288.102	3385.634	3326.794	3423.603	3325.685	3243.106
	rank	1	12	3	13	2	11	6	4	9	8	10	7	5
C17-F29	mean	3132.241	3302.666	3277.673	3337.173	3224.236	3245.982	3319.907	3223.962	3264.978	3230.496	3317.931	3265.55	3246.637
	best	3130.076	3258.467	3205.419	3266.735	3176.255	3174.204	3255.906	3160.855	3189.818	3208.521	3223.27	3177.522	3188.879
	worst	3134.841	3355.244	3353.532	3412.602	3290.52	3281.839	3390.594	3289.231	3379.008	3284.469	3517.91	3308.876	3293.624
	std	2.682921	48.97186	70.24474	81.67165	57.37855	52.66018	61.83371	60.92928	97.26732	39.21587	146.5461	64.55893	46.79051
	median	3132.023	3298.477	3275.871	3334.678	3215.085	3263.943	3316.563	3222.881	3245.543	3214.496	3265.272	3287.902	3252.022
	rank	1	10	9	13	3	5	12	2	7	4	11	8	6
C17-F30	mean	3418.734	1757198	511217.3	2720237	589631.2	720143.8	966863	516542.4	930061.9	358271.9	830025.2	571668.8	1316497
	best	3394.682	1243980	115045.6	1001872	21866.23	527744.3	144694.8	16318.64	33403.47	30595.59	664488.9	16668.65	797896.3
	worst	3442.907	2225191	849270.2	3804713	854300	876607.3	2908208	1215635	1346024	512460.2	972325.8	955365.3	2284690
	std	30.01454	443117.9	326126.2	1312600	418466.3	189309.3	1406078	546372.8	666565.6	240141.9	137056.9	460082.5	756714.6
	median	3418.673	1779811	540276.8	3037181	741179.3	738111.9	407274.4	417108	1170410	445015.9	841643	657320.5	1091700
	rank	1	12	3	13	6	7	10	4	9	2	8	5	11
Sum rank		17	109	52	118	36	91	82	31	72	65	94	72	71
Mean rank		1.7	10.9	5.2	11.8	3.6	9.1	8.2	3.1	7.2	6.5	9.4	7.2	7.1
Total rank		1	11	4	12	2	10	9	3	6	7	9	5	8

 DOI: 10.22266/ijies2024.0831.43

Received: April 19, 2024. Revised: May 19, 2024.

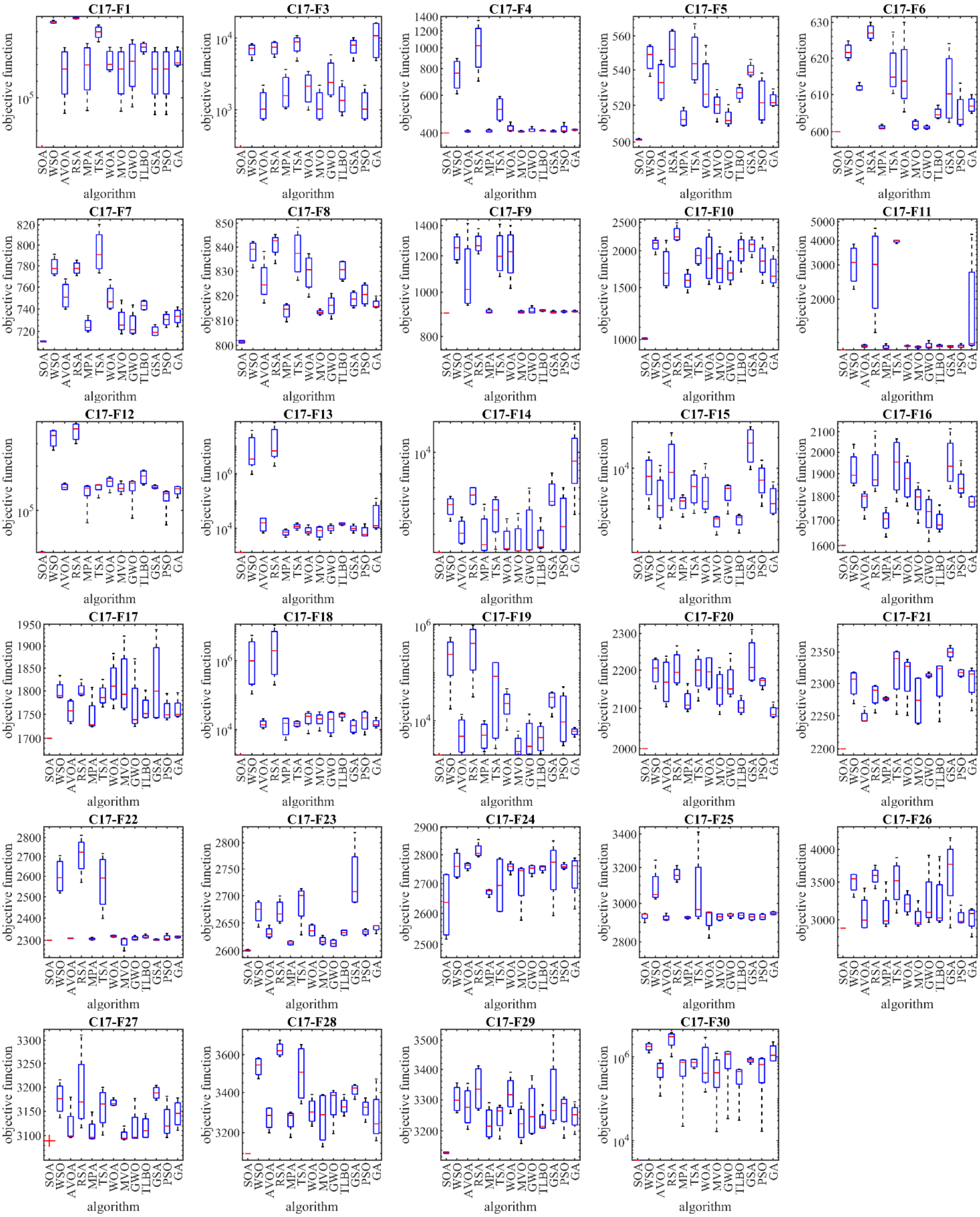

Figure. 2 boxplot diagrams of CEC 2017 test suite (C17-F1 to C17-F30)

By introducing SOA, several research proposals can be proposed for further studies in the future. Among the most prominent of these proposals is the design of binary and multi-objective versions of SOA. The implementation of SOA to address various optimization applications in science and the real-world tasks is one of the other research proposals of this study.

Conflicts of Interest

"The authors declare no conflict of interest."

Author Contributions

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

References

[1] R. Sowmya, M. Premkumar, and P. Jangir, "Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems", *Engineering Applications of Artificial Intelligence*, Vol. 128, pp. 107532, 2024.

[2] H. Jia, and C. Lu, "Guided learning strategy: A novel update mechanism for metaheuristic algorithms design and improvement", *Knowledge-Based Systems*, Vol. 286, pp. 111402, 2024.

[3] S. Zhao, T. Zhang, L. Cai, and R. Yang, "Triangulation topology aggregation optimizer: A novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications", *Expert Systems with Applications*, Vol. 238, pp. 121744, 2024.

[4] R. Rani, S. Jain, and H. Garg, "A review of nature-inspired algorithms on single-objective optimization problems from 2019 to 2023", *Artificial Intelligence Review*, Vol. 57, No. 5, pp. 1-51, 2024.

[5] 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.

[6] H. Qawaqneh, A. Zafar, M. Raheel, A. A. Zaagan, E. H. M. Zahran, A. Cevikel, and A. Bekir, "New soliton solutions of M-fractional Westervelt model in ultrasound imaging via two analytical techniques", *Optical and Quantum Electronics*, Vol. 56, No. 5, pp. 737, 2024.

[7] 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.

[8] 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.

[9] P. Sharma, and S. Raju, "Metaheuristic optimization algorithms: A comprehensive overview and classification of benchmark test functions", *Soft Computing*, Vol. 28, No. 4, pp. 3123-3186, 2024.

[10] 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.

[11] 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.

[12] J. Kennedy, and R. Eberhart, "Particle swarm optimization", In: *Proc. of ICNN'95-International Conference on Neural Networks*, Vol. 4, ed: IEEE, pp. 1942-1948, 1995.

[13] 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.

[14] P. Trojovský, and M. Dehghani, "A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior", *Scientific Reports*, Vol. 13, No. 1, pp. 8775, 2023.

[15] T. Hamadneh, K. Kaabneh, O. Alssayed, K. Eguchi, S. Gochhait, I. Leonova, and M. Dehghani, "Addax Optimization Algorithm: A Novel Nature-Inspired Optimizer for Solving Engineering Applications", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 3, pp. 732-743, 2024, doi: 10.22266/ijies2024.0630.57.

[16] 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 & Systems*, Vol. 17, No. 2, 2024, doi: 10.22266/ijies2024.0430.54.

[17] 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 & Systems*, Vol. 17, No. 1, 2024, doi: 10.22266/ijies2024.0229.59.

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

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

[20] H. Abedinpourshotorban, S. M. Shamsuddin, Z. Beheshti, and D. N. Jawawi, "Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm", *Swarm and Evolutionary Computation*, Vol. 26, pp. 8-22, 2016.

[21] A. Kaveh, and S. Talatahari, "A novel heuristic optimization method: charged system search", *Acta Mechanica*, Vol. 213, No. 3, pp. 267-289, 2010.

[22] R. Kundu, S. Chattopadhyay, S. Nag, M. A. Navarro, and D. Oliva, "Prism refraction search: a novel physics-based metaheuristic algorithm", *The Journal of Supercomputing*, pp. 1-50, 2024.

[23] M. Abdel-Basset, R. Mohamed, S. A. A. Azeem, M. Jameel, and M. Abouhawwash, "Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion", *Knowledge-Based Systems*, Vol. 268, pp. 110454, 2023.

[24] 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.

[25] 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.

[26] S. A. omari, K. Kaabneh, I. AbuFalahah, K. Eguchi, S. Gochhait, I. Leonova, Z. Montazeri, and M. Dehghani, "Dollmaker Optimization Algorithm: A Novel Human-Inspired Optimizer for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 3, pp. 816-828, 2024. doi: 10.22266/ijies2024.0630.63.

[27] S. J. Mousavirad, and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm", *Applied Intelligence*, Vol. 47, No. 3, pp. 850-887, 2017.

[28] J. C. Rich, *The materials and methods of sculpture: Courier Corporation*, 1988.

[29] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, and P. Definitions, "Evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization", *Technology Report*, 2016.

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

[31] S. Mirjalili, and A. Lewis, "The whale optimization algorithm", *Advances in Engineering Software*, Vol. 95, pp. 51-67, 2016.

[32] 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.

[33] X.-S. Yang, "A new metaheuristic bat-inspired algorithm", In: *Proc. of Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, pp. 65-74, 2010.

[34] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems", *Information Sciences*, Vol. 183, No. 1, pp. 1-15, 2012.

[35] X.-S. Yang, and S. Deb, "Cuckoo search via Lévy flights", In: *Proc. of 2009 World congress on nature & biologically inspired computing (NaBIC)*, pp. 210-214, 2009.

[36] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization", *IEEE transactions on Evolutionary Computation*, Vol. 13, No. 2, pp. 398-417, 2008.

[37] A. T. Azar, and V. S. Snášel, "Advances in swarm intelligence for optimizing problems in computer science", *Springer*, 2023.

[38] F. Glover, and M. Laguna, *Tabu search*, Handbook of combinatorial optimization, pp. 3261-3362, 1998.