Received: March 30, 2024. Revised: April 29, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Carpet Weaver Optimization: A Novel Simple and Effective Human-Inspired Metaheuristic Algorithm

Saleh Alomari[1] **Khalid Kaabneh**[2] **Ibraheem AbuFalahah**[3] **Saikat Gochhait**[4,5]
Irina Leonova[6] **Zeinab Montazeri**[7] **Mohammad Dehghani**[7]* **Kei Eguchi**[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]*Symbiosis Institute of Digital and Telecom Management,*
Constituent of Symbiosis International Deemed University, Pune 412115, India
[5]*Neuroscience Research Institute, Samara State Medical University, 89 Chapaevskaya str., 443001 Samara, Russia*
[6]*Faculty of Social Sciences, Lobachevsky University, 603950 Nizhny Novgorod, Russia*
[7]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 7155713876, Iran*
[8]*Department of Information Electronics, Fukuoka Institute of Technology, Japan*
* Corresponding author's Email: adanbax@gmail.com

Abstract: In this paper, a new human-based metaheuristic algorithm called Carpet Weaving Optimization (CWO) is introduced, which is inspired by human skills when weaving a carpet. The main source of inspiration in the design of CWO is taken from the communication between the carpet weaver and the map reader who try to weave a carpet according to the given pattern. The theory of CWO is stated and then mathematically modeled based on the simulation of the carpet weaving process. The effectiveness of CWO in optimization has been assessed across twenty-three standard benchmark functions encompassing unimodal, high-dimensional multimodal, and fixed-dimensional multimodal categories. The optimization outcomes underscore CWO's capability to yield favorable results across various optimization challenges, adeptly navigating between exploration, exploitation, and achieving a balanced search process. Comparative analysis against twelve rival algorithms demonstrates CWO's superior performance, consistently outshining competitors across all twenty-three benchmark functions and securing the top rank as the premier optimizer. Moreover, the efficacy of CWO in real-world applications has been scrutinized through its optimization of four engineering design quandaries. Simulation findings corroborate CWO's commendable performance in real-world and engineering contexts, as evidenced by its capacity to deliver superior values for design variables and objective functions compared to competing algorithms.

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

1. Introduction

Optimization applications are prominent in a wide range of sciences, mathematics, engineering, industry and the real world [1]. Optimization problems have several solvable solutions, among which the best solution is known as the global optimum [2]. Dealing with optimization problems has become a fundamental challenge for researchers to achieve the global optimal solution among all existing solutions [3]. Problem solving techniques to face optimization tasks are classified into two groups: deterministic and stochastic approaches. Although deterministic approaches are effective in dealing with linear and convex optimization problems, but for dealing with real world optimization problems which are non-linear and non-convex, they stop by getting stuck in local optima [4, 5].

Metaheuristic algorithms are one of the most prominent stochastic approaches that are able to provide suitable solutions for optimization problems without depending on derivation processes and only based on the concepts of random search in the problem solving space [6].

The important issue is that due to the random nature in the performance of metaheuristic algorithms, achieving the global optimum is not guaranteed by using these algorithms [7]. However, because these solutions are close to the global optimum, they are still acceptable as quasi-optimum [8]. Achieving better quasi-optimal solutions closer to the global optimum has been the source of the main motivation of researchers to design numerous metaheuristic algorithms so far [9]. In order to provide effective search, metaheuristic algorithms should be able to scan the problem solving space at both global and local levels with high accuracy. Global search refers to the exploration ability of the algorithm for comprehensive scanning in the problem solving space with the aim of discovering the region containing the global optimum, while local search refers to the algorithm's exploitation ability for detailed scanning near promising solutions with the aim of achieving better solutions. In addition to exploration and exploitation, the main key to the success of metaheuristic algorithms is to balance these two search strategies during algorithm iterations [10].

The main research question is that according to the existing metaheuristic algorithms, is there a need for newer algorithms or not? In response to this question, the No Free Lunch (NFL) theorem [11] explains that due to the random nature of the performance of metaheuristic algorithms, it cannot be claimed that a particular algorithm is the best optimizer for all optimization applications. Therefore, there is no presupposition of the result of implementing an algorithm on an optimization problem. The NFL theorem motivates researchers to design newer metaheuristic algorithms to achieve more effective solutions for optimization problems by managing the random search process. Motivated by the NFL theorem, the novelty and innovation aspects of this study are in designing a new metaheuristic algorithm called Carpet Weaving Optimization (CWO) to deal with optimization applications. The main contributions of this study are as follows:

- CWO is designed to simulate the efforts of a carpet weaver and map reader to weave a traditional carpet.
- The idea of CWO design is derived from the skill of a carpet weaver to weave a carpet according to a given pattern.
- The theory of CWO has been described and then mathematically modeled to be used in solving optimization problems.
- The performance of CWO has been evaluated to optimize twenty-three standard benchmark functions of unimodal and multimodal types.
- The efficiency of CWO to address real-world applications is challenged to solve four engineering design problems.
- The performance quality of CWO is analyzed in comparison with the performance of twelve well-known metaheuristic algorithms.

The structure of the paper is as follows: literature review is presented in section 2. Then Carpet Weaving Optimization (CWO) is introduced and designed in section 3. Simulation studies on CWO performance are reported in Section 4. Analysis of the performance of CWO in addressing real-world applications in engineering is presented in Section 5. Finally, conclusions and several research proposals for future studies are provided in Section 6.

2. Literature review

Metaheuristic algorithms are known as one of the most effective stochastic approaches to solving optimization problems. In their design, these algorithms are inspired by various natural phenomena, swarming behavior of living organisms in wildlife, topics of biology, physics, human behavior, games, and other evolutionary phenomena. Based on the source of inspiration in design, metaheuristic algorithms can be classified into five groups: swarm-based, evolutionary-based, physics-based, human-based, and game-based methods.

Swarm-based algorithms have been developed inspired by swarming behaviors among living organisms such as insects, birds, aquatic animals, and reptiles. Particle Swarm Optimization (PSO) [12] and Ant Colony Optimization (ACO) [13] are among the most famous algorithms of this group, which have always been of interest to researchers and have been widely used in optimization applications. PSO simulates the migration strategy of flocks of birds and fish that are searching for food sources. ACO draws inspiration in its design from the ability of ants to identify the optimal communication path between the colony and the food source. Intelligent behaviors of living organisms in nature such as hunting, foraging, digging, chasing, migration, and movement models have been a source of inspiration in designing algorithms such as: Pelican Optimization Algorithm

(POA) [14], Coati Optimization Algorithm (COA) [15], Swarm Space Hopping Algorithm (SSHA) [16], Wombat Optimization Algorithm (WOA) [17], Migration-Crossover Algorithm (MCA) [18], Walrus Optimization Algorithm (WaOA) [19], and Fennec Fox Optimization (FFO) [20].

Evolutionary-based algorithms are inspired by concepts in biology, genetics, Darwin's theory of evolution, natural selection, survival of the fittest, and other genetic concepts. Genetic Algorithm (GA) [21] and Differential Evolution (DE) [22] are among the most prominent evolutionary-based algorithms that have been widely used in science. The main idea in the design of GA and DE is derived from the reproduction process, genetic concepts and the simulation of selection, crossover, and mutation operators between chromosomes.

Physics-based algorithms are developed from laws, forces, transformations, processes, phenomena and other concepts in physics. Simulated Annealing (SA) [23] is considered one of the most widely used physics-based algorithms, which was developed with inspiration from the annealing process of metals. The simulation of physical forces and Newton's laws of motion have been a source of inspiration in designing algorithms such as: Spring Search Algorithm (SSA) [24] and Momentum Search Algorithm (MSA) [25]. Physical concepts have also been sources of inspiration in designing algorithms such as: Propagation Search Algorithm (PSA) [26], Water Cycle Algorithm (WCA) [27], and Gravitational Search Algorithm (GSA) [28].

Human-based algorithms are inspired in their design process by simulating human behavior in individual and social life such as choices, communication, decision making, thinking, and cooperation. Teaching-Learning Based Optimization (TLBO) [29] is one of the most widely used human-based algorithms that imitates the interaction and communication between students and teachers in the classroom environment. Mother Optimization Algorithm (MOA) [7] simulate the Eshrat's attentive care from her children. Doctor and Patient Optimization (DPO) [30] is inspired by the strategies of medical staff and doctors to treat patients. Human behavior in various activities has been a source of inspiration for designing algorithms such as: Driving Training-Based Optimization (DTBO) [31], Ali Baba and the Forty Thieves (AFT) [32], Election-Based Optimization Algorithm (EBOA) [33], and Teamwork Optimization Algorithm (TOA) [34].

Game-based algorithms are developed inspired by players' strategies based on the rules governing various individual and team games. Darts Game Optimizer (DGO) is one of the most popular game-based algorithms, inspired in its design by the skill of darts players to collect more points from their throws towards the board [35]. The competition between players in different games in order to win the game has been a source of inspiration in designing algorithms such as: Orientation Search Algorithm (OSA) [36], Golf Optimization Algorithm (GOA) [37], Ring toss game based optimization (RTGBO) [38], Hide Object Game Optimizer (HOGO) [39], Puzzle Optimization Algorithm (POA) [40], Dice Game Optimizer (DGO) [41], Shell game optimization (SGO) [42], and Football Game Based Optimization (FGBO) [43].

Based on the best knowledge obtained from the literature review, no metaheuristic algorithm has been designed based on the simulation of the carpet weaving process. This is while the communication between the carpet weaver and the map reader in order to weave a carpet according to the given pattern is an intelligent process that can be the basis for the design of a new optimizer. In order to address this research gap in the studies of metaheuristic algorithms, in this paper, a new metaheuristic algorithm is designed based on the simulation of the carpet weaving process, which is discussed in the next section.

3. Carpet weaving optimization

In this section, the theory and the source of inspiration used in the design of Carpet Weaving Optimization (CWO) are described first, then the steps of its implementation are mathematically modeled.

3.1 Inspiration of CWO

There is no exact information about the history of the origin of the carpet, but according to the evidence of ancient works and human discoveries, arts such as basket weaving, and felt weaving have been influential in the creation of carpet weaving art. The designs of the first carpets that had the aspect of everyday use; It was simple and primitive and broken and mostly subjective, and researchers initially thought of Egypt as the cradle of carpet weaving, but Professor Rudenko discovered the first Pazyryk carpet in the mountains of Siberia in 1949 and invalidated other theories. According to this practical example, the cradle of carpet weaving moved from the banks of the Nile, Tigris and Euphrates rivers to Central Asia, and based on this, it was determined that the cradle of carpet weaving was in Iran.

In addition to being considered an art, carpet weaving is also an attractive job to earn money. In traditional carpet weaving, the carpet weaver tries to

Received: March 30, 2024. Revised: April 29, 2024.

prepare the carpet according to the given pattern. In this carpet weaving process, a person as a map reader reads the pattern of the carpet out loud and the carpet weaver weaves the carpet according to that pattern.

Therefore, what is evident is that the art of carpet weaving is an intelligent process in which the communication between the carpet weaver and the map reader is very important for weaving a carpet according to the pattern. This skill of carpet weaving is considered as a source of basic inspiration for the design of CWO, which is discussed further.

3.2 Algorithm initialization

CWO is proposed as a population-based algorithm that is able to achieve suitable solutions for optimization problems by relying on the power of searching its members in the problem solving space. From the dualistic point of view between carpet weaving process and optimization process, each carpet as a member of CWO corresponds to a candidate solution for the problem. Considering this point of view, each CWO member is mathematically modeled using a vector. The initial position of the carpets in the solution space of the initialization problem is completely random using Eq. (1). The community of carpets together form the CWO population, which is mathematically modeled using a matrix according to Eq. (2).

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

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

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

Based on the placement of candidate solutions proposed by each CWO member in the objective function, a value is calculated. The list of evaluated values for the objective function can be represented mathematically using a vector according to Eq. (3).

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

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

3.3 Mathematical modelling of CWO

In this step, the mathematical modeling of CWO has been discussed. According to the traditional carpet weaving process, where the carpet weaver tries to weave the carpet based on the given pattern, the position of the CWO members is updated in the problem solving space. In the design of CWO, it is assumed that the carpet weaving process has two stages: (i) weaving the carpet based on the given pattern and (ii) creating creative changes in the design of the carpet simultaneously with the carpet weaving. With this point of view in the design of CWO, in each iteration the position of the population members is updated during two different phases based on the mathematical modeling of the mentioned stages. Each of these update phases is further analyzed and modeled.

3.3.1 Phase 1: Carpet weaving based on the given pattern (exploration phase)

An important step in the carpet weaving process is that the carpet weaver starts weaving the carpet according to the given pattern that is read out loud by a person called the map reader. This behavior of carpet weaving leads to extensive changes on the raw materials of carpet weaving. Simulating these extensive changes in the carpet weaving process leads to extensive changes in the position of CWO members in the problem solving space. As a result, the update phase enhances the CWO discovery capability to manage global search.

In CWO design, the position of a randomly generated member in the problem solving space is considered as a carpet weaving pattern ($X_P \ where \ x_{P,j} = lb_j + r \cdot (ub_j - lb_j)$). The advantage of this choice is that the random member can direct the population of the algorithm to different regions of the problem solving space and increase the algorithm's exploration power. Based on the modeling of extensive changes on the carpet material, a new position for each CWO member is calculated using Eq. (4). Then, if this new position provides a better value for the objective function, it replaces the previous position of the corresponding member according to Eq. (5).

$$x_{i,j}^{P1} = x_{i,j} + (1-2r) \cdot (x_{P,j} - I \cdot x_{i,j}), \quad (4)$$

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

Where, X_P is the weaving pattern, $x_{P,j}$ is its jth dimension, X_i^{P1} is the new position for the ith member based on first phase of CWO, $x_{i,j}^{P1}$ 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 creative changes to the design during carpet weaving (exploitation phase)

Although a pattern is available to prepare a carpet, carpet weavers make small changes in the design while weaving according to their own creativity in order to increase the attractiveness and beauty of the carpet. This carpet weaver's skill during carpet weaving leads to the creation of small changes on the carpet, whose simulation leads to the improvement of the exploitation ability of CWO in order to manage local search. In fact, making these small changes in the position of CWO members can lead to convergence to better solutions. In CWO design, based on the simulation of these creative changes, a new position is calculated for each member of the population according to Eq. (6). Then, if this new position is acceptable and replaces the previous position of the corresponding member, it improves the value of the objective function according to Eq. (7).

$$x_{i,j}^{P2} = (1 + \frac{(1-2\,r)}{t}) \cdot x_{i,j} \quad (6)$$

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

Where, X_i^{P2} is the new calculated position for the ith CWO member based on second phase of CWO, $x_{i,j}^{P2}$ 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.

3.4 Repetition process, pseudocode, and flowchart of CWO

The first iteration of CWO ends when all its population members are updated based on the exploration and exploitation phases. After that, with the new updated values, the algorithm enters the next iteration and the process of updating the CWO members continues until the last iteration using Eqs. (4) to (7). At the end of each iteration, the best candidate solution obtained is updated and saved. After the full implementation of the algorithm, the best solution obtained during the iterations of the algorithm is output as the proposed solution of CWO for the given problem. The pseudocode of CWO implementation steps is presented in Algorithm 1.

Algorithm 1. Pseudocode of CWO.

Start CWO.

1. Input problem information: variables, objective function, and constraints.
2. Set CWO 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: ***Carpet weaving based on the given pattern*** (exploration phase)
8. Determine carpet weaving pattern. $x_{P,j} \leftarrow lb_j + r \cdot (ub_j - lb_j)$
9. Calculate new position of ith member using Eq. (4). $x_{i,d}^{P1} \leftarrow x_{i,d} + (1-2r) \cdot \left(x_{P,j} - I \cdot x_{i,d}\right)$
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: ***Making creative changes to the design during carpet weaving*** (exploitation phase)
12. Calculate new position of ith member using Eq. (6). $x_{i,d}^{P2} \leftarrow (1 + \frac{(1-2\,r)}{t}) \cdot x_{i,j}$
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 CWO.

End CWO.

4. Simulation studies and results

This section is dedicated to the performance analysis of CWO to handle optimization tasks. For this purpose, a set of twenty-three standard benchmark functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types has been selected. A detailed description and complete information of these benchmark functions is available in [44]. In order to analyze the performance of CWO, the obtained results have been compared with the performance of twelve famous metaheuristic algorithms: GA [21], PSO [12], GSA [28], TLBO [29], MVO [45], GWO [46], WO [47], MPA [48], TSA [49], RSA [50], AVOA [51], and WSO [52]. Six statistical indicators mean, best, worst, standard deviation (std), median, and rank have been used to report the simulation results.

4.1 Evaluation of unimodal functions

Due to the lack of local optimal solutions, unimodal functions are desirable options for analyzing the exploitation ability of metaheuristic algorithms in local search. For this reason, functions F1 to F7 have been selected as unimodal. The results of the implementation of CWO and competing algorithms on functions F1 to F7 are listed in Table 1. Based on these results, CWO with its high ability in exploitation has been able to provide the global optimum for functions F1 to F6. Also, in order to handle F7, CWO has been ranked as the first best optimizer. The analysis of the simulation results shows that CWO, with its high ability in exploitation, has provided superior performance for dealing with unimodal functions in competition with the compared algorithms.

4.2 Evaluation of high-dimensional multimodal functions

High-dimensional multimodal functions challenge the exploration ability of metaheuristic algorithms due to having multiple local optima in addition to global optima.

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

		CWO	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
F_1	mean	0	43.66812	8.15E-48	8.15E-48	8.28E-48	4.02E-47	8.15E-48	0.099143	8.15E-48	8.15E-48	8.84E-17	0.06689	20.20947
	best	0	3.508369	2.53E-51	2.53E-51	2.46E-50	1.25E-50	2.53E-51	0.069907	2.53E-51	2.53E-51	3.55E-17	0.000322	11.87772
	worst	0	158.2929	5.79E-47	5.79E-47	5.79E-47	2.86E-46	5.79E-47	0.133372	5.79E-47	5.79E-47	2.48E-16	0.926092	37.71832
	std	0	30.11298	1.51E-47	1.51E-47	1.51E-47	7.46E-47	1.51E-47	0.01584	1.51E-47	1.51E-47	4.08E-17	0.177342	5.970485
	median	0	30.09354	7.49E-49	7.49E-49	8.52E-49	3.7E-48	7.49E-49	0.099734	7.49E-49	7.49E-49	7.5E-17	0.00644	18.68357
	rank	1	10	2	2	4	5	2	8	3	2	6	7	9
F_2	mean	0	1.417777	3.7E-29	3.7E-29	4.98E-28	1.82E-28	3.7E-29	0.171719	3.7E-29	3.7E-29	3.64E-08	0.593328	1.847484
	best	0	0.438932	3.55E-31	3.55E-31	1.26E-29	1.75E-30	3.55E-31	0.106059	3.55E-31	3.55E-31	2.31E-08	0.030002	1.156407
	worst	0	4.933104	3.19E-28	3.19E-28	3.13E-27	1.58E-27	3.19E-28	0.241511	3.19E-28	3.19E-28	8.16E-08	1.651976	2.52208
	std	0	1.012467	7.99E-29	7.99E-29	6.2E-28	3.94E-28	7.99E-29	0.035945	7.99E-29	7.99E-29	1.07E-08	0.412412	0.310876
	median	0	1.014025	3.46E-30	3.46E-30	3.08E-28	1.71E-29	3.46E-30	0.177797	3.46E-30	3.46E-30	3.4E-08	0.387045	1.81645
	rank	1	10	2	2	6	5	2	8	4	3	7	9	11
F_3	mean	0	1183.541	2.07E-11	2.07E-11	2.24E-11	1.02E-10	13224.22	10.58332	2.08E-11	2.07E-11	315.0481	257.1612	1437.086
	best	0	689.3607	2.4E-22	2.4E-22	4.62E-19	1.19E-21	1368.111	3.958327	3.6E-17	2.4E-22	162.9663	14.42282	943.6123
	worst	0	2347.532	3.42E-10	3.42E-10	3.42E-10	1.69E-09	22983.24	32.42563	3.42E-10	3.42E-10	786.0084	679.3865	2291.759
	std	0	358.2412	6.58E-11	6.58E-11	6.54E-11	3.25E-10	4883.018	6.142817	6.58E-11	6.58E-11	125.7018	164.5889	365.031
	median	0	1032.464	1.88E-14	1.88E-14	9.54E-13	9.3E-14	13466.08	7.870749	1.91E-14	1.88E-14	265.2466	194.1601	1391.844
	rank	1	10	2	2	5	6	12	7	4	3	9	8	11
F_4	mean	0	11.46044	0.000775	0.000775	0.000775	0.003826	34.33562	0.363274	0.000775	0.000775	0.819623	4.161587	1.875425
	best	0	7.894391	1.69E-05	1.69E-05	1.69E-05	8.35E-05	0.599382	0.176414	1.69E-05	1.69E-05	4.73E-05	1.517667	1.469686
	worst	0	15.79271	0.006278	0.006278	0.006278	0.030995	60.76349	0.639216	0.006278	0.006278	3.264962	8.852199	2.645543
	std	0	1.647695	0.001199	0.001199	0.001199	0.005919	16.89903	0.110405	0.001199	0.001199	0.791774	1.428045	0.265932
	median	0	11.7756	0.000258	0.000258	0.000258	0.001272	36.7226	0.352461	0.000258	0.000258	0.602242	3.898124	1.844987
	rank	1	10	2	2	3	5	11	6	4	2	7	9	8
F_5	mean	0	7160.262	4.990385	13.60277	20.44396	24.63641	23.08474	68.74312	22.60233	22.73904	34.17617	3060.683	399.4701
	best	0	897.7362	4.498638	4.498599	19.69789	22.20861	22.47026	23.36725	21.99029	21.93579	21.88627	22.47057	156.6194
	worst	0	61434.92	5.063005	24.27072	20.94502	24.99485	24.0887	255.4444	23.04111	24.0718	115.8726	59686.85	1500.501
	std	0	11451.73	0.118946	8.408498	0.265515	0.58723	0.302227	57.91832	0.303102	0.566845	25.31313	11479.28	242.5391
	median	0	3721.809	5.050883	5.05951	20.46174	24.93507	22.98594	24.92492	22.43836	22.48896	22.51296	62.02653	319.929
	rank	1	13	2	3	4	8	7	10	5	6	9	12	11
F_6	mean	0	67.50226	0.645218	4.923966	0.645218	3.185303	0.699266	0.745267	1.083072	1.480977	0.645218	0.687255	23.27003
	best	0	11.72573	0.447355	3.096115	0.447355	2.208497	0.481147	0.545819	0.653907	0.864829	0.447355	0.447715	11.09755
	worst	0	254.0519	0.838993	5.642568	0.838993	4.141928	0.939363	0.927426	1.495689	2.188011	0.838993	0.971192	42.12588
	std	0	54.51424	0.104646	0.596324	0.104646	0.516617	0.114437	0.107806	0.20176	0.31822	0.104646	0.125679	7.717934
	median	0	46.70071	0.665211	5.237235	0.665211	3.284003	0.706313	0.759652	1.102359	1.429945	0.665211	0.696413	21.61364
	rank	1	13	4	11	3	10	6	7	8	9	2	5	12
F_7	mean	2.54E-05	0.000825	0.000807	0.000785	0.001127	0.00376	0.001612	0.00846	0.001316	0.001779	0.035754	0.12277	0.007781
	best	2.35E-06	0.000287	0.000271	0.000279	0.000337	0.001293	0.000444	0.00378	0.000646	0.000961	0.011114	0.046877	0.002781
	worst	6.89E-05	0.001775	0.001758	0.001762	0.002048	0.008634	0.004726	0.016121	0.00217	0.003036	0.064422	0.272884	0.014967
	std	1.7E-05	0.000334	0.000352	0.000356	0.000388	0.001745	0.001098	0.002909	0.000375	0.000527	0.01422	0.044979	0.002701
	median	1.83E-05	0.000676	0.000673	0.000676	0.001066	0.003225	0.001023	0.008494	0.001235	0.001639	0.034643	0.118664	0.007872
	rank	1	4	3	2	5	9	7	11	6	8	12	13	10
Sum rank		7	70	17	24	30	48	47	57	34	33	52	63	72
Mean rank		1	10	2.4285714	3.4285714	4.2857143	6.8571429	6.7142857	8.1428571	4.8571429	4.7142857	7.4285714	9	10.285714
Total rank		1	12	2	3	4	8	7	10	6	5	9	11	13

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

		CWO	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
F_8	mean	-12498.6	-7774.99	-11365.7	-6704.9	-9521.61	-6996.92	-10434.4	-8292.89	-7131.21	-6812.35	-4945.83	-7441.14	-8682.84
	best	-12622.8	-9233.99	-11652.6	-6939.59	-10099.4	-8029.9	-11651.8	-9341.48	-7605.9	-7797.43	-5733.65	-8302.26	-9685.05
	worst	-11936.3	-6894.35	-10948.8	-6280.23	-8805.99	-5457.83	-8156.76	-7524.12	-6598.29	-6132.27	-4375.8	-6446.37	-7763.82
	std	163.9826	450.7563	156.4622	161.171	261.1113	548.6706	1028.308	435.8963	221.2246	352.5892	300.4616	406.2674	419.93
	median	-12577.8	-7771.36	-11355.9	-6778.08	-9442	-6974.56	-11017.7	-8239.87	-7095.81	-6719.73	-4896.69	-7478.85	-8665.19
	rank	1	7	2	12	4	10	3	6	9	11	13	8	5
F_9	mean	0	46.65731	30.33831	30.33831	30.33831	149.7737	30.33831	95.15657	30.33831	30.33831	49.22501	75.2033	66.56802
	best	0	33.03347	15.7269	15.7269	15.7269	77.64033	15.7269	57.69707	15.7269	15.7269	25.59002	45.36693	50.58438
	worst	0	68.2492	50.50149	50.50149	50.50149	249.315	50.50149	131.5287	50.50149	50.50149	67.64125	106.6014	84.684
	std	0	7.7613	7.69836	7.69836	7.69836	38.00514	7.69836	16.57629	7.69836	7.69836	9.839781	15.03763	8.956979
	median	0	46.05044	29.20825	29.20825	29.20825	144.1949	29.20825	95.76731	29.20825	29.20825	47.94703	75.61971	65.58362
	rank	1	4	2	2	2	9	2	8	3	2	5	7	6
F_{10}	mean	8.88E-16	3.723604	0.217735	0.217735	0.217735	1.074909	0.217735	0.600629	0.217735	0.217735	0.217735	2.024696	2.58646
	best	8.88E-16	2.597422	2.13E-15	2.13E-15	4.49E-15	7.04E-15	3.38E-15	0.066654	1.28E-14	4.49E-15	3.94E-09	1.122015	1.909479
	worst	8.88E-16	5.432151	0.591165	0.591165	0.591165	2.918452	0.591165	1.666468	0.591165	0.591165	0.591165	3.350624	3.635938
	std	0	0.698384	0.236881	0.236881	0.236881	1.169428	0.236881	0.416943	0.236881	0.236881	0.236881	0.539986	0.334004
	median	8.88E-16	3.732135	4.62E-15	4.62E-15	6.98E-15	1.93E-14	8.15E-15	0.599712	1.63E-14	6.98E-15	6.26E-09	2.144645	2.586359
	rank	1	12	2	2	4	9	3	8	6	5	7	10	11
F_{11}	mean	0	1.13861	0.00155	0.00155	0.00155	0.00765	0.00155	0.266359	0.002437	0.00155	4.777306	0.1243	0.977816
	best	0	0.734988	0	0	0	0	0	0.170395	0	0	1.98635	0.003314	0.855476
	worst	0	2.177837	0.003601	0.003601	0.003601	0.017776	0.003601	0.355124	0.014218	0.003601	8.37331	0.580304	1.145003
	std	0	0.309561	0.00095	0.00095	0.00095	0.004689	0.00095	0.04644	0.002764	0.00095	1.552743	0.130015	0.070753
	median	0	1.061754	0.001576	0.001576	0.001576	0.007781	0.001576	0.277646	0.001631	0.001576	4.84495	0.082142	0.961363
	rank	1	8	2	2	2	4	2	6	3	2	9	5	7
F_{12}	mean	1.57E-32	3.181511	1.01513	1.888132	1.01513	5.011477	1.028445	1.621137	1.041552	1.06239	1.154293	2.009674	1.197264
	best	1.57E-32	1.233308	0.181699	1.01238	0.181699	0.89701	0.187246	0.2185	0.209187	0.241127	0.199801	0.534041	0.261227
	worst	1.57E-32	5.63923	2.477194	3.567709	2.477194	12.22937	2.479033	3.203045	2.490027	2.53303	2.558529	3.865906	2.558285
	std	2.42E-48	1.153065	0.585662	0.607317	0.585662	2.891285	0.585506	0.731719	0.580743	0.584934	0.61057	0.881824	0.59149
	median	1.57E-32	3.104608	0.754392	1.785164	0.754392	3.724272	0.758833	1.627975	0.788177	0.800374	0.960224	2.102633	0.995756
	rank	1	12	3	10	2	13	4	9	5	6	7	11	8
F_{13}	mean	1.35E-32	2385.488	0.476108	0.476108	0.477764	2.350445	0.618297	0.497824	0.816547	1.20625	0.51365	2.866381	2.270217
	best	1.35E-32	9.531134	0.352662	0.352662	0.352662	1.741018	0.382823	0.360725	0.440976	0.914808	0.356234	0.476517	1.298185
	worst	1.35E-32	41186.32	0.650831	0.650831	0.650831	3.213012	0.855656	0.687444	1.249809	1.52361	1.075928	8.691423	3.093288
	std	2.42E-48	7905.609	0.084146	0.084146	0.084073	0.415412	0.101675	0.087893	0.180312	0.160116	0.144353	1.702002	0.412696
	median	1.35E-32	29.74058	0.444264	0.444264	0.448893	2.193234	0.625598	0.466166	0.826452	1.202909	0.448005	2.724755	2.335318
	rank	1	13	3	2	4	11	7	5	8	9	6	12	10
Sum rank		6	56	14	30	18	56	21	42	34	35	47	53	47
Mean rank		1	9.333333	2.333333	5	3	9.333333	3.5	7	5.666667	5.833333	7.833333	8.833333	7.833333
Total rank		1	11	2	5	3	11	4	8	6	7	9	10	9

According to this feature, functions F8 to F13 are selected from the high-dimensional multimodal type. The performance of CWO and competing algorithms for optimizing functions F8 to F13 is reported in Table 2. Based on these results, CWO with high exploration ability has been able to achieve the global optimum for F9 and F11 functions. Also, CWO was able to get the rank of the first best optimizer for F8, F10, F12, and F13 functions. Analysis of the simulation results shows that CWO with high power in exploration and local search has provided superior performance in dealing with high-dimensional multimodal functions in comparison to competing algorithms.

4.3 Evaluation of fixed-dimensional multimodal functions

Fixed-dimensional multimodal functions are a type of optimization challenges that are suitable for simultaneous evaluation of exploration and exploitation abilities in metaheuristic algorithms. With this point of view, functions F14 to F23 are selected from fixed-dimensional multimodal type. The performance of CWO and competing algorithms on functions F14 to F23 is reported in Table 3. These results show that CWO is able to get the rank of the first best optimizer in handling all ten benchmark functions F14 to F23. Analysis of the simulation results shows that CWO, with its high ability to balance exploration and exploitation, has been able to provide superior performance for handling fixed-dimensional multimodal functions compared to competing algorithms.

The performance of CWO and competing algorithms for optimizing benchmark functions F1 to F23 are drawn as boxplot diagrams in Fig. 1.

5. CWO for real-world engineering applications

The efficiency of metaheuristic algorithms in tackling real-world and engineering challenges stands as a paramount objective. To this end, the effectiveness of CWO and rival algorithms has been assessed across four engineering design domains: pressure vessel design (PV) [53], speed reducer design (SR) [54], welded beam design (WB) [47],

Received: March 30, 2024. Revised: April 29, 2024.

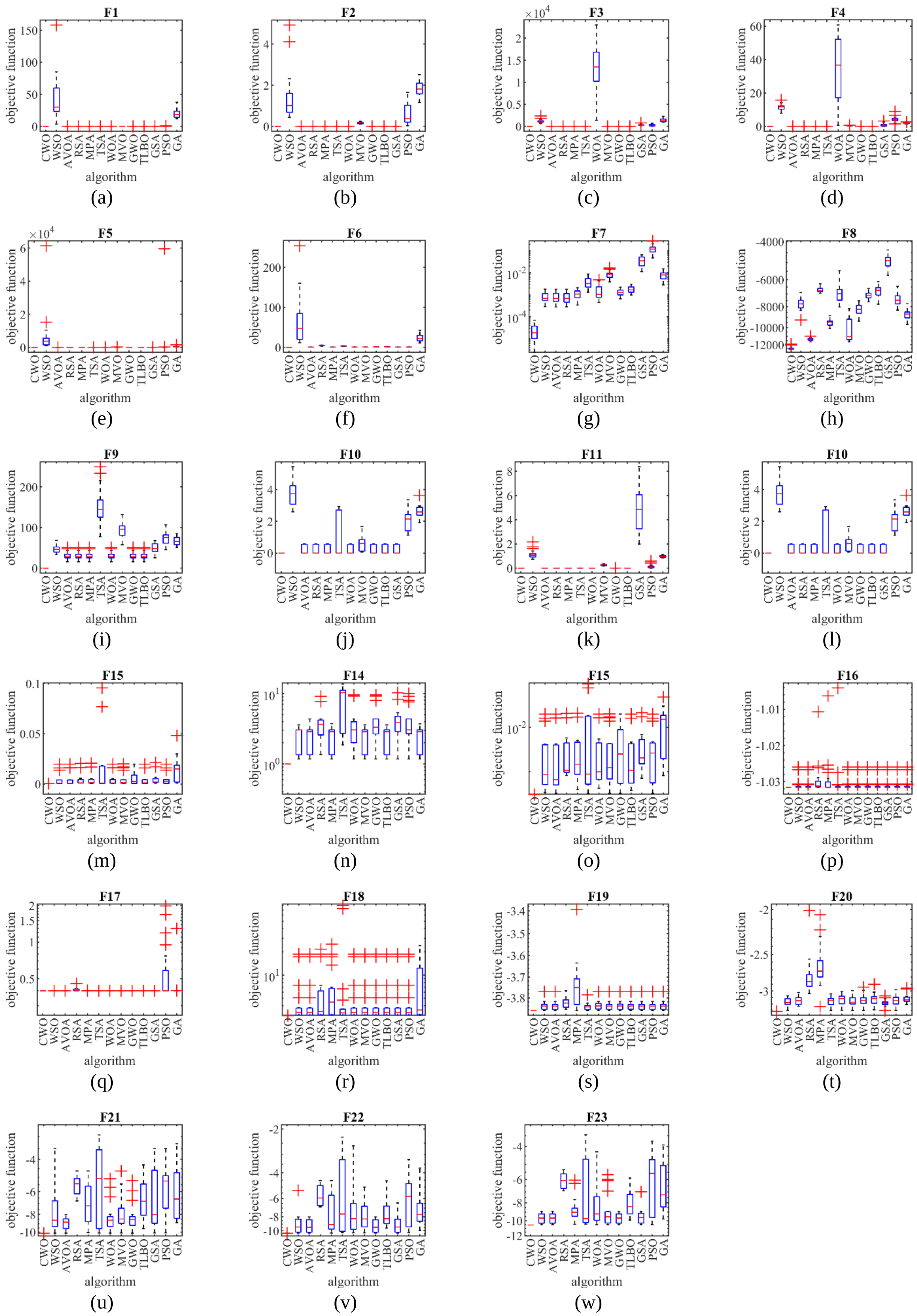

Figure. 1 boxplot diagrams of benchmark functions: (a)F1, (b)F2, (c)F3, (d)F4, (e)F5, (f)F6, (g)F7, (h)F8, (i)F9, (j)F10, (k)F11, (l)F12, (m)F13, (n)F14, (o)F15, (p)F16, (q)F17, (r)F18, (s)F19, (t)F20, (u)F21, (v)F22, and (w)F23

 DOI: 10.22266/ijies2024.0831.18

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

		CWO	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
F_{14}	mean	0.998004	2.406166	2.406035	3.737824	2.348115	7.616808	3.381686	2.340305	4.127349	2.340306	4.038658	4.061502	2.373873
	best	0.998004	1.172197	1.172197	1.358984	1.172197	1.85796	1.172197	1.172197	1.172197	1.172197	1.172225	1.172197	1.172197
	worst	0.998004	3.5782	4.358085	9.079454	3.734221	13.57908	9.51353	3.5782	9.51354	3.5782	10.24679	10.04823	3.702104
	std	0	0.763645	0.849186	1.55558	0.775126	3.765621	1.960999	0.764752	2.362973	0.764751	1.886152	2.236951	0.798074
	median	0.998004	3.043495	2.876375	3.651804	2.876375	10.27112	3.043495	2.876375	3.310847	2.876375	3.899936	3.043495	2.880222
	rank	1	7	6	9	4	13	8	2	12	3	10	11	5
F_{15}	mean	0.000307	0.003974	0.00331	0.003818	0.003874	0.014374	0.00361	0.004828	0.005304	0.003468	0.004633	0.00473	0.01327
	best	0.000307	0.00034	0.000324	0.000821	0.00039	0.000322	0.00042	0.000324	0.000324	0.000377	0.000708	0.000324	0.000812
	worst	0.000307	0.019801	0.019803	0.02044	0.020707	0.095633	0.019857	0.019985	0.019821	0.019807	0.02117	0.019801	0.048177
	std	2.19E-19	0.004926	0.004557	0.00457	0.004657	0.022386	0.004551	0.0055	0.005753	0.004516	0.00464	0.005177	0.010404
	median	0.000307	0.000855	0.000657	0.001068	0.001467	0.00088	0.000988	0.00124	0.002504	0.001075	0.002038	0.002627	0.014993
	rank	1	7	2	5	6	13	4	10	11	3	8	9	12
F_{16}	mean	-1.03163	-1.03097	-1.03097	-1.0295	-1.02942	-1.02994	-1.03097	-1.03097	-1.03097	-1.03097	-1.03097	-1.03097	-1.03097
	best	-1.03163	-1.03163	-1.03163	-1.03161	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
	worst	-1.03163	-1.02595	-1.02595	-1.01067	-1.00631	-1.00415	-1.02595	-1.02595	-1.02595	-1.02595	-1.02595	-1.02595	-1.02595
	std	1.58E-16	0.001402	0.001402	0.004102	0.004967	0.00529	0.001402	0.001402	0.001402	0.001402	0.001402	0.001402	0.001402
	median	-1.03163	-1.03162	-1.03162	-1.03126	-1.03161	-1.03162	-1.03162	-1.03162	-1.03162	-1.03162	-1.03162	-1.03162	-1.03162
	rank	1	6	2	10	11	9	3	5	4	8	2	2	7
F_{17}	mean	0.397887	0.397977	0.397977	0.406396	0.398318	0.397989	0.397978	0.397977	0.397978	0.398025	0.397977	0.62772	0.443121
	best	0.397887	0.397888	0.397888	0.398494	0.397888	0.397893	0.397888	0.397889	0.397889	0.397899	0.397888	0.397888	0.397899
	worst	0.397887	0.398421	0.398421	0.455791	0.400585	0.398345	0.398421	0.398421	0.398421	0.398428	0.398421	1.983607	1.295193
	std	0	0.000132	0.000132	0.01108	0.000678	0.000115	0.000132	0.000132	0.000132	0.000132	0.000132	0.404721	0.172732
	median	0.397887	0.397908	0.397908	0.401824	0.397966	0.397932	0.397909	0.397909	0.39791	0.397978	0.397908	0.397956	0.397991
	rank	1	4	2	10	9	7	5	3	6	8	2	12	11
F_{18}	mean	3	5.002669	5.00267	6.841139	7.097465	10.78153	5.002687	5.002669	5.002678	5.00267	5.002669	5.002669	7.853609
	best	3	3.002261	3.002262	3.003682	3.011493	3.001886	3.002261	3.002261	3.002264	3.002262	3.002261	3.002261	3.003693
	worst	3	18.61844	18.61844	21.77546	25.26957	80.03973	18.61847	18.61844	18.61844	18.61844	18.61844	18.61844	24.28014
	std	1E-15	3.90627	3.90627	5.850069	5.487039	19.41267	3.906271	3.90627	3.906268	3.90627	3.90627	3.90627	6.538782
	median	3	3.288222	3.288223	3.369787	4.465574	3.239682	3.288223	3.288223	3.288233	3.288223	3.288222	3.288222	3.522608
	rank	1	2	6	10	11	13	9	5	8	7	4	3	12
F_{19}	mean	-3.86278	-3.84034	-3.84034	-3.82321	-3.74893	-3.84383	-3.83877	-3.84034	-3.83933	-3.83961	-3.84034	-3.84034	-3.84023
	best	-3.86278	-3.86276	-3.86276	-3.85602	-3.86276	-3.86268	-3.86275	-3.86276	-3.86276	-3.86239	-3.86276	-3.86276	-3.86275
	worst	-3.86278	-3.77036	-3.77036	-3.76512	-3.39291	-3.78586	-3.77007	-3.77036	-3.77028	-3.77019	-3.77036	-3.77036	-3.77014
	std	1.96E-15	0.019159	0.019159	0.021656	0.097578	0.015803	0.018873	0.019159	0.019074	0.018795	0.019159	0.019159	0.019238
	median	-3.86278	-3.84055	-3.84055	-3.82541	-3.74975	-3.84427	-3.83941	-3.84055	-3.83989	-3.84025	-3.84055	-3.84055	-3.8405
	rank	1	3	5	11	12	2	10	6	9	8	4	4	7
F_{20}	mean	-3.322	-3.17041	-3.14678	-2.81334	-2.65919	-3.15761	-3.13445	-3.15065	-3.1405	-3.12972	-3.18222	-3.14421	-3.12015
	best	-3.322	-3.30584	-3.23727	-3.03781	-3.24146	-3.30692	-3.30207	-3.30584	-3.30583	-3.29394	-3.30584	-3.30584	-3.20985
	worst	-3.322	-3.0722	-3.02238	-2.00933	-2.05295	-3.04337	-3.00993	-2.99341	-2.94347	-2.89958	-3.0722	-2.98164	-2.95721
	std	3.82E-16	0.050862	0.057717	0.195133	0.236904	0.0575	0.066193	0.06531	0.070944	0.075213	0.045586	0.069519	0.053932
	median	-3.322	-3.17917	-3.14898	-2.86078	-2.71753	-3.14247	-3.11366	-3.17201	-3.14235	-3.11401	-3.19093	-3.14217	-3.13445
	rank	1	3	6	12	13	4	9	5	8	10	2	7	11
F_{21}	mean	-10.1532	-7.83418	-8.99147	-5.61373	-7.27249	-6.14553	-8.48278	-8.15154	-8.48603	-6.80471	-7.0309	-5.99047	-6.41214
	best	-10.1532	-10.1317	-10.1317	-6.75395	-10.0608	-10.1182	-10.1305	-10.1075	-10.1311	-9.57949	-10.1075	-10.0911	-8.87297
	worst	-10.1532	-3.48276	-8.01005	-4.63231	-4.63535	-2.94362	-5.10491	-4.63534	-5.21998	-4.29046	-3.48276	-3.49838	-3.28593
	std	1.79E-15	1.963909	0.601265	0.601265	1.603656	2.456913	1.182708	1.283426	1.107686	1.401884	2.078981	1.688889	1.733226
	median	-10.1532	-8.60794	-8.81858	-5.44083	-7.18006	-5.08843	-8.6075	-8.47889	-8.55018	-6.77785	-8.01157	-5.2604	-6.59672
	rank	1	6	2	13	7	11	4	5	3	9	8	12	10
F_{22}	mean	-10.4029	-9.15769	-9.41115	-5.88945	-7.87848	-7.04697	-7.89095	-8.10706	-9.4108	-7.7858	-9.22982	-6.74764	-7.40288
	best	-10.4029	-10.3738	-10.3738	-6.85213	-10.3722	-10.2959	-10.331	-10.3337	-10.3737	-9.5917	-10.3738	-10.2616	-9.96781
	worst	-10.4029	-5.26446	-8.03899	-4.51729	-4.51729	-2.27182	-2.60641	-5.00887	-8.03804	-4.53832	-6.44241	-3.24529	-3.7102
	std	3.02E-15	1.066158	0.740802	0.740801	1.849192	2.765269	1.837465	1.353916	0.740858	1.333847	0.922119	2.096623	1.375889
	median	-10.4029	-9.4264	-9.47442	-5.95272	-9.05467	-7.67297	-8.25403	-8.29804	-9.47412	-8.26052	-9.4264	-5.79634	-7.72592
	rank	1	5	2	13	8	11	7	6	3	9	4	12	10
F_{23}	mean	-10.5364	-9.7651	-9.7651	-6.18201	-8.84878	-7.64949	-8.47114	-9.05314	-9.76479	-8.14164	-9.6001	-7.03827	-6.99808
	best	-10.5364	-10.4978	-10.4978	-6.91472	-10.4401	-10.4189	-10.4971	-10.4978	-10.4976	-9.91103	-10.4978	-10.4713	-9.78384
	worst	-10.5364	-8.8854	-8.8854	-5.30231	-6.06232	-3.46242	-4.26889	-5.67105	-8.88525	-5.88482	-6.99038	-3.74017	-3.92563
	std	2.38E-15	0.487289	0.487289	0.487289	0.982752	2.529413	1.809806	1.363166	0.487204	1.065584	0.711335	2.253917	1.640685
	median	-10.5364	-9.66341	-9.66341	-6.08032	-9.00013	-9.72502	-9.18625	-9.51455	-9.66323	-8.39273	-9.60166	-5.57259	-7.25669
	rank	1	2	3	13	7	10	8	6	4	9	5	11	12
Sum rank		10	45	36	106	88	93	67	53	68	74	49	83	97
Mean rank		1	4.5	3.6	10.6	8.8	9.3	6.7	5.3	6.8	7.4	4.9	8.3	9.7
Total rank		1	3	2	13	10	11	6	5	7	8	4	9	12

and tension/compression spring design (TCS) [47].

Table 4 presents the implementation outcomes of both CWO and alternative algorithms across four distinct engineering design challenges. According to the optimization outcomes, CWO emerges as the top-performing optimizer across all four engineering

Table 4. Optimization results of engineering applications

		CWO	WSO	AVOA	RSA	MPA	TSA	WOA	GWO	MVO	TLBO	GSA	PSO	GA
PV	mean	5882.901	5882.91	6140.345	9195.018	5882.901	6120.471	7198.879	6298.193	6010.447	21998.8	16786.4	31793.66	24509.91
	best	5882.901	5882.901	5882.906	6391.304	5882.901	5901.671	6214.711	5914.45	5887.406	11350.98	6509.873	12413.07	10938.45
	worst	5882.901	5883.071	6816.175	15342.01	5882.901	6856.008	8948.232	6785.845	6725.806	32501.05	33870.41	64763.47	42708.73
	std	2.94E-12	0.060005	424.6324	3130.755	4.62E-05	442.0238	1344.419	379.4059	384.6814	9715.071	11400.35	23140.24	11805.47
	median	5882.901	5882.901	6089.378	8927.077	5882.901	5953.262	6876.877	6279.643	5898.905	21436.61	16121.84	26962.61	23644.43
	rank	1	3	6	9	2	5	8	7	4	11	10	13	12
SR	mean	2996.348	2996.35	2999.959	3164.16	2996.348	3020.11	3178.901	3021.806	3002.177	3.54E+13	3364.68	9.44E+13	6.39E+13
	best	2996.348	2996.348	2996.35	3066.426	2996.348	3007.971	3005.048	3002.556	2998.317	4087.581	3191.827	4267.39	3951.079
	worst	2996.348	2996.362	3005.028	3235.686	2996.348	3032.259	4084.543	3045.138	3007.071	1.62E+14	3813.392	4.57E+14	4.53E+14
	std	1.47E-12	0.00489	4.450415	66.14756	1.27E-05	9.318109	462.9391	18.42566	3.75656	6.73E+13	241.8025	2.05E+14	1.61E+14
	median	2996.348	2996.349	2999.904	3157.067	2996.348	3019.814	3092.387	3022.715	3002.128	1.84E+13	3338.86	2.68E+13	3.56E+13
	rank	1	3	4	8	2	6	9	7	5	11	10	13	12
WB	mean	1.724852	1.724852	1.739339	2.11188	1.724852	1.737443	2.205939	1.739271	1.726444	1.82E+13	2.141641	4.86E+13	4.05E+12
	best	1.724852	1.724852	1.724883	1.863302	1.724852	1.730394	1.77274	1.728019	1.725322	1.905296	1.75738	2.396993	2.325465
	worst	1.724852	1.724852	1.777653	3.212607	1.724852	1.74204	3.567566	1.76148	1.729193	3.07E+14	2.33882	5.88E+14	7.87E+13
	std	1.08E-15	1.07E-08	0.024987	0.450877	3.73E-08	0.005773	0.826874	0.014945	0.001797	1.08E+14	0.225575	2.2E+14	2.77E+13
	median	1.724852	1.724852	1.733503	2.051665	1.724852	1.738076	1.945769	1.736805	1.725964	3.925179	2.150119	4.127844	4.050508
	rank	1	2	7	8	3	5	10	6	4	12	9	13	11
TCS	mean	0.012665	0.012665	0.012895	0.016028	0.012665	0.012841	0.0132	0.015563	0.012702	0.016425	0.017545	2.59E+13	0.020511
	best	0.012665	0.012665	0.012666	0.012929	0.012665	0.012699	0.012681	0.012828	0.012681	0.016039	0.013743	0.015992	0.016453
	worst	0.012665	0.012669	0.013625	0.065421	0.012665	0.013107	0.014502	0.016198	0.012716	0.016824	0.021009	2.59E+14	0.026634
	std	1.54E-18	1.37E-06	0.000426	0.018507	4.85E-09	0.00016	0.00097	0.001601	1.22E-05	0.000371	0.003687	1.26E+14	0.004147
	median	0.012665	0.012665	0.012789	0.013057	0.012665	0.012846	0.012974	0.016012	0.012705	0.016386	0.017316	0.015992	0.019959
	rank	1	3	6	9	2	5	7	8	4	10	11	13	12
Sum rank		4	11	23	34	9	21	34	28	17	44	40	52	47
Mean rank		1	2.75	5.75	8.5	2.25	5.25	8.5	7	4.25	11	10	13	11.75
Total rank		1	3	6	8	2	5	8	7	4	110	9	12	11

design scenarios. Examination of the simulation findings reveals that CWO demonstrates remarkable efficacy in addressing optimization endeavors within real-world and engineering contexts, consistently outperforming its competitors.

6. Conclusions and future works

In this paper, a new metaheuristic algorithm called Carpet Weaving Optimization (CWO) was introduced to solve optimization problems.

The main source of inspiration in the design of CWO comes from the human interaction between carpet weavers, map readers when weaving a rug based on a given pattern in the traditional carpet weaving process. The theory of CWO was stated and then modeled mathematically based on the simulation of the carpet weaving process.

The evaluation of CWO's performance extended to optimizing twenty-three standard benchmark functions, spanning unimodal, high-dimensional multimodal, and fixed-dimensional multimodal varieties. The optimization results underscored CWO's proficiency in exploration, exploitation, and maintaining a balanced approach throughout the search process within the solution space, yielding commendable outcomes. To gauge CWO's optimization prowess, its results were juxtaposed against those of twelve established algorithms. Simulation findings revealed CWO's superior performance, securing the top rank as the foremost optimizer by consistently outperforming its competitors and delivering superior results. Additionally, CWO's efficacy in real-world applications was assessed across four engineering design challenges, affirming its effectiveness in optimizing real-world and engineering scenarios by yielding superior values for design variables and objective functions compared to rival algorithms.

The introduction of CWO sparks several avenues for future research endeavors. Notably, there's a call for developing binary and multi-objective variants of CWO, representing significant research proposals outlined in this paper. Moreover, leveraging CWO to tackle optimization challenges across diverse scientific disciplines and real-world applications stands as another avenue for future exploration and study.

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.

Acknowledgments

Acknowledgments are to show that the article is supported by what organization. For example, “This work was supported by the National Nature Science Foundation under Grant No. 405”.

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, 2024/03/15/, 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] 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, 2023/06/26, 2023.

[8] H. Givi, M. Dehghani, and Š. Hubálovský, “Red Panda Optimization Algorithm: An effective bio-inspired metaheuristic algorithm for solving engineering optimization problems”, *IEEE Access*, 2023.

[9] M. Dehghani, Z. Montazeri, A. Dehghani, H. Samet, C. Sotelo, D. Sotelo, A. Ehsanifar, O. P. Malik, J. M. Guerrero, and G. Dhiman, “DM: Dehghani Method for Modifying Optimization Algorithms”, *Applied Sciences*, Vol. 10, No. 21, pp. 7683, 2020.

[10] M. Dehghani, Š. Hubálovský, and P. Trojovský, “Tasmanian Devil Optimization: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm”, *IEEE Access*, 2022.

[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, “Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications”, *Sensors*, Vol. 22, No. 3, pp. 855, 2022.

[15] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, “Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems”, *Knowledge-Based Systems*, Vol. 259, pp. 110011, 2023/01/10/, 2023.

[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] 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.

[18] 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.

[19] 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.

[20] E. Trojovská, M. Dehghani, and P. Trojovský, "Fennec Fox Optimization: A New Nature-Inspired Optimization Algorithm", *IEEE Access*, 2022.
[21] D. E. Goldberg, and J. H. Holland, "Genetic Algorithms and Machine Learning", *Machine Learning*, Vol. 3, No. 2, pp. 95-99, 1988.
[22] 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.
[23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science*, Vol. 220, No. 4598, pp. 671-680, 1983.
[24] M. Dehghani, Z. Montazeri, G. Dhiman, O. Malik, R. Morales-Menendez, R. A. Ramirez-Mendoza, A. Dehghani, J. M. Guerrero, and L. Parra-Arroyo, "A spring search algorithm applied to engineering optimization problems", *Applied Sciences*, Vol. 10, No. 18, pp. 6173, 2020.
[25] M. Dehghani, and H. Samet, "Momentum search algorithm: a new meta-heuristic optimization algorithm inspired by momentum conservation law", *SN Applied Sciences*, Vol. 2, No. 10, pp. 1-15, 2020.
[26] 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.
[27] 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.
[28] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm", *Information Sciences*, Vol. 179, No. 13, pp. 2232-2248, 2009.
[29] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems", *Computer-Aided Design*, Vol. 43, No. 3, pp. 303-315, 2011.
[30] 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.
[31] 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, 2022/06/15, 2022.
[32] 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.
[33] 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.
[34] M. Dehghani, and P. Trojovský, "Teamwork Optimization Algorithm: A New Optimization Approach for Function Minimization/Maximization", *Sensors*, Vol. 21, No. 13, pp. 4567, 2021.
[35] 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.
[36] 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.
[37] 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.
[38] 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.
[39] 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.
[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer”, *Advances in Engineering Software,* Vol. 69, pp. 46-61, 2014.

[47] S. Mirjalili, and A. Lewis, “The whale optimization algorithm”, *Advances in engineering software,* Vol. 95, pp. 51-67, 2016.

[48] 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.

[49] 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, 2020/04/01/, 2020.

[50] 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.

[51] 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.

[52] 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.

[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.