Received: July 25, 2024. Revised: October 2, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

On the Application of Tailor Optimization Algorithm for Solving Real-World Optimization Application

Tareq Hamadneh[1] Belal Batiha[2] Omar Alsayyed[3] Zeinab Montazeri[4] Hamid Jafarabadi Ashtiani[5] Melika Jafarpour[6] Mohammad Dehghani[4]* Kei Eguchi[7]

[1]*Department of Mathematics, Al Zaytoonah University of Jordan, Amman 11733, Jordan*
[2]*Department of Mathematics, Faculty of Science and Information Technology, Jadara University, Irbid 21110, Jordan*
[3]*Department of Mathematics, Faculty of Science, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan*
[4]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 7155713876, Iran*
[5]*School of Electrical and Computer Engineering, Shiraz University, Shiraz 51154-71348, Iran*
[6]*Department of Radiology Technology, Babol University of medical sciences, Babol, Iran*
[7]*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 Tailor Optimization Algorithm (TOA) is introduced. The basic idea in TOA design is taken from the processes that a tailor makes when sewing clothes. The theory of TOA is stated and then mathematically modeled in two phases of exploration and exploitation. The exploration phase is designed based on the simulation of extensive changes on the fabrics according to the garment pattern. The exploitation phase is designed based on the simulation of small changes on the sewn garments in order to handle the details of the garments. The effectiveness of proposed TOA approach to handle optimization tasks in real-world applications is evaluated on twenty-two constrained optimization problems from the CEC 2011 test suite. The simulation results show that TOA is achieved effective solutions for CEC 2011 test suite optimization with the ability to explore, exploit, and balance them. In addition, the performance of TOA is compared with the results of twelve well-known metaheuristic algorithms. Analysis of the simulation results shows that TOA is successful in 100% of CEC 2011 test suite optimization problems in competition with the compared algorithms. The findings show that TOA with 100% success and ranking as the first best optimizer in the competition with the compared algorithms has an effective efficiency to handle real world applications.

Keywords: Optimization algorithm, Engineering, Real-world application, Human-inspired, Tailor, Exploration, Exploitation.

1. Introduction

Optimization is a process that aims to find the best or closest solution to a problem, which can involve maximizing or minimizing an objective function given a set of constraints. Optimization is of particular importance in many scientific and industrial fields, because it can increase productivity, reduce costs, and improve system performance [1, 2]. One of the common methods to solve optimization problems is to use metaheuristic algorithms. These algorithms are inspired by the concepts and principles of nature and have been highly regarded due to their high ability to search in a wide space of solutions and find solutions close to the optimum [3]. Metaheuristic algorithms are widely used in solving optimization problems in various fields.

In engineering, metaheuristic algorithms are used to optimize the design of structures, mechanical systems, and manufacturing processes. For example, Golf Optimization Algorithm (GOA) is used to optimize the use of energy resources in integrated energy systems [4]. In the field of management and economics, these algorithms are used to optimize production planning, resource allocation, supply chain management, and solving stock market problems. The Wombat Optimization Algorithm (WOA) is used to optimize supply chain management applications [5]. In computer science, metaheuristic algorithms are used to optimize computer networks, allocate tasks in distributed systems, and solve complex problems such as the traveling salesman problem (TSP). These methods can be used to improve the performance of wireless networks and data traffic management [6]. In the fields of biology and medicine, these algorithms are used to optimize biological processes, design drugs, and analyze biological data. The Particle Swarm Optimization (PSO) algorithm can be used to diagnose diseases and predict the results of treatments [7].

Metaheuristic algorithms, with their special abilities to search and find near-optimal solutions, have become a powerful tool in solving complex and large problems. Due to their high flexibility and ability to work with different types of problems, these algorithms are used in many fields and help researchers and engineers to achieve more efficient and effective solutions [8].

The concepts of exploration and exploitation are two fundamental elements in the process of random search of metaheuristic algorithms, which play a vital role in the efficiency and performance of these algorithms [9]. Exploration refers to the process of searching the vast space of solutions to explore new and unknown areas. The main goal of exploration is to increase the variety of solutions and avoid getting stuck in local optimal points. By exploring, the algorithm can better identify the search space and be directed to areas with higher potential [10]. Exploitation refers to the process of focusing on specific areas of the search space that have already been identified and are most likely to improve the solution. The purpose of exploitation is to improve the quality of current solutions and to approach the global optimal point. This process is carried out using available information about the best solutions found [11].

One of the main challenges in designing metaheuristic algorithms is to find a suitable balance between exploration and exploitation. If the algorithm explores too much, it may not focus enough on the high-yielding areas and fail to reach the optimal point. On the other hand, if the algorithm focuses too much on exploitation, it may get stuck in local optimal points and not fully explore the search space. A proper balance between exploration and exploitation can lead to increasing the efficiency of the algorithm. With enough exploration, the algorithm can identify new regions and, with proper exploitation, improve the quality of the solutions [12].

The main research question is that according to the metaheuristic algorithms introduced so far, is there still a need to design newer algorithms? The answer to this question is possible by referring to the No Free Lunch (NFL) theorem [13]. The NFL theorem in the field of optimization and machine learning states that there is no general optimization algorithm that can work equally well for all optimization problems. New metaheuristic algorithms can perform better than existing algorithms by considering the specific features and challenges of different optimization problems. Therefore, research and development in the field of optimization algorithms should be continued in order to achieve more efficient and optimal solutions to deal with optimization problems.

Motivated by the NFL theorem, the innovation and novelty of this paper is in designing a new metaheuristic algorithm called Tailor Optimization Algorithm (TOA) to deal with optimization problems. The main contributions of this paper are as follows:

- TOA is designed based on the human activity of sewing.
- The theory of TOA is stated and then mathematically modeled in two phases of exploration and exploitation.
- The exploration phase is designed with regard to making changes on the fabrics based on the dress pattern.
- The exploitation phase is designed with attention to detail and making small changes to the sewn garments.
- The performance of the proposed TOA approach to address real-world applications is challenged to address twenty-two constrained optimization problems from the CEC 2011 test suite.
- The results obtained from TOA have been compared with the performance of twelve well-known metaheuristic algorithms.

In the following, the paper is organized as follows: In section 2, the literature review is presented. The proposed approach of TOA is introduced and designed in section 3. Then in section 4, simulation studies are presented. Finally, conclusions and several research suggestions for further future work are provided in Section 5.

2. Literature review

Metaheuristic algorithms are divided into four main groups based on their source of inspiration:

swarm-based, evolutionary-based, physics-based, and human-based. Next, while introducing each group, some examples of famous algorithms are given.

Swarm-based metaheuristic algorithms are designed based on the collective behaviour of living organisms such as insects, birds and fish. The particle swarm algorithm (PSO) is inspired by the group behaviour of birds and fish to search for food [14]. Ant Colony Algorithm (ACO) is designed based on the behaviour of ants in finding the shortest path between nest and food source [15]. The artificial bee algorithm (ABC) is inspired by the foraging behaviour of honey bees [16]. Different crowding behaviors in nature have been sources of inspiration in designing other algorithms such as: Walrus Optimization Algorithm (WaOA) [17], Migration-Crossover Algorithm (MCA) [18], Green Anaconda Optimization (GAO) [19], and Kookaburra Optimization Algorithm (KOA) [20].

Evolutionary-based metaheuristic algorithms are designed based on the principles of natural and genetic evolution, such as natural selection and mutation. Genetic Algorithm (GA) is inspired by the processes of natural selection and genetic recombination in natural evolution [21]. The Differential Evolution (DE) [22] algorithm is designed based on the differences between individuals of a population and using these differences to create new generations. The artificial immune system (AIS) algorithm is inspired by the body's immune system processes to identify and combat foreign agents [23].

Physics-based metaheuristic algorithms are designed based on physical laws and phenomena. Gravitational Search Algorithm (GSA) is designed based on Newton's law of gravity and gravitational interactions between particles [24]. The Simulated Annealing (SA) algorithm is inspired by the cooling process of materials in physics [25]. Electro-Magnetism Optimization (EMO) search algorithm is inspired by the laws of electromagnetism and the force of attraction and repulsion between charged particles [26]. Some other prominent physics-based algorithms are: Prism Refraction Search (PRS) [27], Momentum Search Algorithm (MSA) [28], Electromagnetic Field Optimization (EFO) [29], Spring Search Algorithm (SSA) [30], and Kepler Optimization Algorithm (KOA) [31].

Human-based metaheuristic algorithms are designed based on human behaviours and processes. The Mother Optimization Algorithm (MOA) is inspired by maternal principles of education and nurturing by mother Eshrat [32]. The Teaching-Learning Algorithm (TLBO) is inspired by the teaching and learning process in educational environments [33]. The Doctor-Patient Algorithm (DPO) is inspired by the interactions between the doctor and the patient in the diagnosis and treatment process [34]. Alibaba and the Forty Thieves (AFT) algorithm is based on the famous story of Alibaba and the Forty Thieves, where Alibaba seeks to discover the thieves' treasure [35].

Based on the best knowledge obtained from the literature review, no metaheuristic algorithm has been designed so far inspired by tailor's strategies when sewing clothes. Meanwhile, the tailor's strategies when making changes on the fabrics and also taking care of the details of the sewn clothes are intelligent processes that can be the basis for the design of a new optimizer. In order to address this research gap, in this paper, a new metaheuristic algorithm based on the simulation of tailor's strategies while sewing clothes is introduced, which is discussed in the next section.

3. Tailor optimization algorithm

In this section, the theory and the inspiration of the proposed Tailor Optimization Algorithm (TOA) approach are explained, then the its implementation steps are mathematically modeled.

3.1 Inspiration of TOA

Tailoring is one of the old and important industries of the world, which is still very influential in different societies. Tailors play an important role in the beauty and comfort of people. Sewing is also known as an art. A tailor should be able to come up with new and unique designs and use different colors and combinations to create visually appealing effects. Sewing is an art that requires technical expertise. A successful tailor must have sewing, designing, cutting and measuring skills. She/he should have a good knowledge of all kinds of fabrics and materials and be able to use the right materials according to each project. When a tailor wants to sew a dress, he must first choose a suitable pattern. By making changes on the fabric such as cutting and sewing, the tailor sews the basic design of the dress. Then, based on the details of the selected pattern, she/he takes care of the details and decorations of the clothes. Among the tailor's behaviors while sewing clothes, there are two more prominent strategies:

(i): Making changes such as cutting and sewing fabrics based on the selected pattern.

(ii): Taking care of the details and decorations of the sewn clothes according to the selected pattern. Mathematical modeling of these tailor's strategies while sewing clothes is employed in TOA design, which is discussed below.

3.2 Algorithm initialization

The proposed TOA approach is a population-based optimizer that is able to achieve suitable solutions for optimization problems in an iteration-based process based on random search in the problem solving space. In TOA, each member of the population means a candidate solution to the problem, which is mathematically modeled using a vector. Therefore, all members of the TOA population can be represented together using a matrix according to Eq. (1). The position of each member of the population in TOA is initialized completely randomly using Eq. (2).

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

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

Here, X is the TOA'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.

Each member of TOA is a candidate solution for the problem, based on which the objective function of the problem can be evaluated. The set of evaluated values for the objective function corresponding to each member of the population can be represented using a vector according to Eq. (3).

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

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

3.3 Mathematical modelling of TOA

The proposed TOA approach is an iteration-based algorithm that improves the quality of proposed candidate solutions based on updating the position of population members in the problem solving space in each iteration. In order to manage this updating process, the design of TOA is inspired by the tailor's strategies when sewing clothes. Among these two tailor strategies are more significant: (i): making changes (cutting and sewing) on the fabrics according to the pattern and (ii): dealing with the details and decorations of the sewn clothes. Therefore, in the design of TOA, the modeling of these two smart tailor strategies has been used in order to update the position of the population members in the problem solving space. Each of these strategies is modeled as a separate update phase, which is described below.

3.3.1 Phase 1: Making extensive changes to fabrics (exploration phase)

In the first phase of TOA, the position of the population members is updated based on the simulation of the tailor's strategy when cutting and sewing the fabrics. This strategy of the tailor leads to the creation of extensive changes on the fabrics, whose modeling leads to the creation of large changes in the position of the members of the population. These large displacements increase the ability of TOA exploration to manage global search in the problem-solving space. In this process, the tailor uses a suitable pattern to sew the clothes. In TOA design, it is assumed that corresponding to the sewing pattern, a position for the sewing pattern is generated in the problem solving space using Eq. (4).

$$P: p_j = x_j^{worst} + r \cdot (x_j^{best} - x_j^{worst}) \quad (4)$$

Here, P_i is the position for the sewing pattern, p_j is its jth dimension, X^{best} is best population member, x_j^{best} is its jth dimension, X^{worst} is best population member, x_j^{worst} is its jth dimension, and r is a random number within the interval $[0,1]$.

The tailor cuts and sews the fabrics according to the sewing pattern. Inspired by this tailor's strategy, it is assumed in the design of TOA that based on making changes on the fabrics according to the sewing pattern, a new position for each member of the population can be calculated using Eq. (5). In the following, if this new position leads to an improvement in the value of the objective function, it replaces the previous position of the corresponding member according to Eq. (6).

$$x_{i,j}^{P1} = x_{i,j} + r \cdot (p_{i,j} - I \cdot x_{i,j}), \tag{5}$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} \le F_i, \\ X_i, & else, \end{cases} \tag{6}$$

Where, X_i^{P1} is the new position for the ith member based on exploration phase of TOA, $x_{i,j}^{P1}$ 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 changes to the sewn garment (exploitation phase)

In the second phase of TOA, the position of the population members is updated based on the simulation of the tailor's strategy when dealing with the details and decorations on the sewn garments, according to the tailoring pattern. In this strategy, according to the given pattern, the tailor makes small and accurate changes in different parts of the garment so that its appearance is similar to the sewing pattern. This tailor's strategy leads to the creation of small changes on the clothes, whose modeling in TOA leads to the creation of targeted small changes in the position of the population members. These small changes in the position of the population members lead to an increase in the ability to exploit TOA in order to manage the local search in the problem-solving space.

Based on this tailoring strategy, it is assumed in TOA design that a new position is generated near the position of each member using Eq. (7). Then, if the value of the objective function needs to be improved, this new position replaces the previous position of the corresponding member according to Eq. (8).

$$x_{i,j}^{P2} = x_{i,j} + r \cdot \left(\frac{x_j^{best} - x_j^{worst}}{t+1}\right) \tag{7}$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} \le F_i \\ X_i, & else \end{cases} \tag{8}$$

Here, X_i^{P2} is the new calculated position for the ith TOA member based on exploitation phase of TOA, $x_{i,j}^{P2}$ is the its jth dimension, F_i^{P2} is its objective function value, and t is the iteration counter.

3.4 Repetition process, pseudocode, and flowchart of TOA

The process of the first iteration of TOA is completed by updating all population members according to the instructions of the first and second phases. After that, with the updated values, the algorithm enters the next iteration and the process of updating the TOA population continues until the last iteration of the algorithm based on Eqs. (4) to (8). The best candidate solution is identified and updated in each iteration. After the complete execution of TOA, the best candidate solution obtained is placed as a solution in the output. The flowchart of TOA implementation steps is shown in Figure 1.

4. TOA for real-world optimization problems

One of the important applications of metaheuristic algorithms is their efficiency to handle optimization tasks in real world applications. In this study, the performance of the proposed TOA approach has been evaluated to solve twenty-two constrained optimization problems from the CEC 2011 test suite. This test suite consists of twenty-two challenging optimization problems in engineering. Detailed information, complete details, and mathematical models of these problems are available in [36].

The titles of these real-world optimization applications are as follows: parameter estimation for frequency-modulated sound waves, the Lennard-Jones potential problem, the bifunctional catalyst blend optimal control problem, optimal control of a nonlinear stirred tank reactor, the Tersoff potential for the model Si (B), the Tersoff potential for the model Si (C), spread spectrum radar polyphase code design, transmission network expansion planning problem, large-scale transmission pricing problem, circular antenna array design problem, and the electronic logging device (ELD) problems (which consist of DED instance 1, DED instance 2, ELD instance 1, ELD instance 2, ELD instance 3, ELD instance 4, ELD instance 5, hydrothermal scheduling instance 1, hydrothermal scheduling instance 2, and hydrothermal scheduling instance 3), the Messenger spacecraft trajectory optimization problem, and the Cassini 2 spacecraft trajectory optimization problem.

In order to evaluate the quality of the TOA proposed approach, its performance has been compared with twelve famous metaheuristic algorithms: Gravitational Search Algorithm (GSA) [24], Coati Optimization Algorithm (COA) [11], Kookaburra Optimization Algorithm (KOA) [20], Golf Optimization Algorithm (GOA) [4], Spring Search Algorithm (SSA) [30], Teaching-Learning Based Optimization (TLBO) [33], Grey Wolf Optimizer (GWO) [37], Marine Predator Algorithm (MPA) [38], Tunicate Search Algorithm (TSA) [39], Reptile Search Algorithm (RSA) [40], African Vultures Optimization Algorithm (AVOA) [41], and White Shark Optimizer (WSO) [42]. The simulation results are reported using six statistical indicators: mean, best, worst, median, standard deviation (std), and rank. It should be mentioned that in order to rank the metaheuristic algorithms in handling each of the optimization problems, the comparison of the mean index has been used.

The results of the implementation of TOA and competing algorithms to address the CEC 2011 test suite are reported in Table 1. In addition, the boxplot diagrams of metaheuristic algorithms are drawn in Figure 2. Findings It shows that TOA has achieved suitable solutions for optimization problems by effectively balancing exploration and exploitation during algorithm iterations. Based on the comparison of simulation results, it is evident that TOA was the first best optimization in order to solve all twenty two problems of CEC 2011 test suite. The findings show that by providing better results for the statistical index and achieving better solutions, TOA has provided superior performance for handling the CEC 2011 test suite compared to competing algorithms.

5. Conclusions and future works

In this paper, a new metaheuristic algorithm called Tailor Optimization Algorithm (TOA) was introduced to handle optimization tasks in real world applications. The main idea in the design of TOA was taken from the tailor's strategies when sewing clothes. The theory of TOA was stated and its steps were mathematically modeled in two phases of exploration and exploitation. The efficiency of TOA to handle optimization tasks in real-world applications was evaluated on twenty-two constrained optimization problems from the CEC 2011 test suite. The optimization results showed that TOA has achieved suitable solutions for the optimization problems of this test suite with its ability to explore, exploit, and balance them during the search process. In addition, the performance of TOA is compared with the performance of twelve well-known metaheuristic algorithms. The analysis of the results showed that TOA has provided superior performance compared to competing algorithms by providing better results compared to competing algorithms and getting the rank of the first best optimizer. The findings showed that TOA, with its ability in exploration and exploitation, has been more successful in 100% of CEC 2011 test suite optimization problems in competition with compared algorithms.

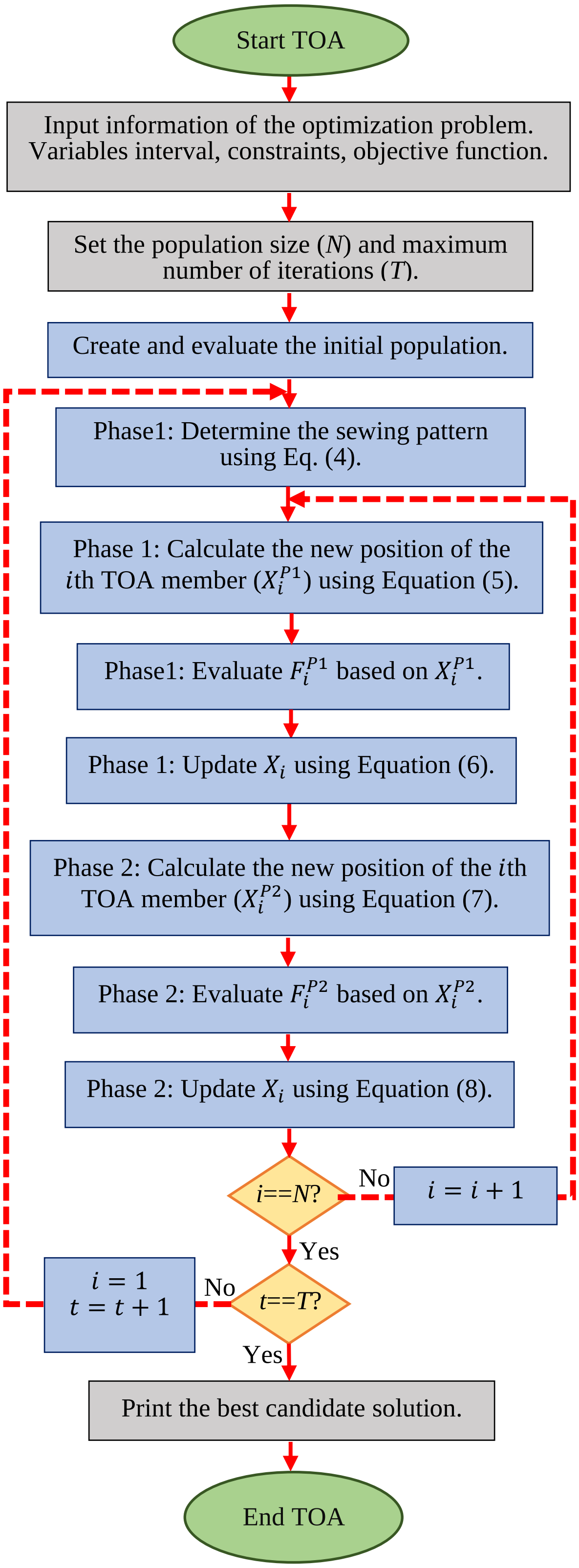

Figure. 1 flowchart of TOA

Table 1. Optimization results of CEC 2011 test suite

		TOA	WSO	AVOA	RSA	MPA	TSA	COA	GOA	GWO	TLBO	GSA	SSA	KOA
C11-F1	mean	5.860902	9.287021	9.891289	13.29474	10.1855	20.63383	23.14364	23.34298	19.91985	14.93192	14.87855	11.74373	13.27816
	best	1.98E-10	3.171166	3.907239	9.712726	6.077739	19.47566	22.26689	22.24808	18.40361	10.7694	12.71635	6.794303	10.03214
	worst	12.183	13.3693	13.58012	16.46845	14.39696	22.40084	24.46245	25.11439	20.66189	18.2254	16.97651	16.61256	16.98022
	std	9.191179	6.352624	5.968245	4.830647	6.199248	1.703963	1.286596	1.719834	1.443034	4.427814	2.677296	5.861266	5.035423
	median	5.630304	10.30381	11.0389	13.49889	10.13364	20.3294	22.92262	23.00472	20.30695	15.36643	14.91067	11.78402	13.05015
	rank	1	2	3	7	4	11	12	13	10	9	8	5	6
C11-F2	mean	-26.0547	-22.9078	-22.2314	-20.0246	-21.6955	-11.1491	-12.6357	-9.88293	-9.61488	-16.6161	-8.9137	-22.3596	-19.6285
	best	-26.7969	-23.5809	-22.9092	-22.2022	-22.3824	-12.6185	-14.2803	-10.3648	-13.1049	-19.7995	-10.6697	-22.9283	-20.0253
	worst	-25.1785	-21.7379	-20.9752	-18.6178	-20.9942	-9.80541	-10.6985	-9.43348	-7.50894	-12.916	-7.56156	-21.839	-19.0965
	std	0.943765	1.14408	1.203911	2.092314	0.834146	1.918396	2.013763	0.689155	3.546263	4.856384	1.885268	0.604342	0.534156
	median	-26.1218	-23.1562	-22.5207	-19.6392	-21.7027	-11.0863	-12.782	-9.86673	-8.92286	-16.8744	-8.71176	-22.3356	-19.6962
	rank	1	2	4	6	5	10	9	11	12	8	13	3	7
C11-F4	mean	1.14E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05
	best	1.14E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05
	worst	1.14E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05
	std	2.56E-19	1.64E-14	2.03E-14	1.9E-14	2.38E-14	3.04E-11	6.96E-09	6.07E-11	2.62E-14	1.53E-14	1.1E-12	1.9E-14	1.9E-14
	median	1.14E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05
	rank	1	3	7	5	8	11	13	12	9	2	10	4	6
C11-F4	mean	13.74389	16.60183	15.44964	15.97763	15.51112	17.01622	16.53715	19.43001	16.18183	16.40418	18.20501	15.82447	15.19664
	best	13.67721	14.91731	14.33044	15.56605	13.85534	15.41725	14.2065	15.41725	15.0278	15.00959	13.81928	14.82544	13.8763
	worst	13.79425	18.45619	15.99284	16.62162	16.40696	18.77527	19.19848	22.04658	17.5138	17.92265	20.73299	16.95542	15.95173
	std	0.075233	2.563979	1.033515	0.669226	1.545418	1.8717	3.149354	3.848467	1.426721	2.174394	4.229717	1.2755	1.240616
	median	13.75204	16.5169	15.73764	15.86143	15.89109	16.93618	16.37181	20.12811	16.09285	16.34223	19.13389	15.75851	15.47927
	rank	1	10	3	6	4	11	9	13	7	8	12	5	2
C11-F5	mean	-33.7862	-29.8605	-28.9174	-27.6151	-26.8704	-20.6622	-17.7802	-17.1668	-23.9684	-24.4412	-24.5107	-22.6104	-22.8496
	best	-34.4019	-30.5022	-29.7488	-28.9541	-27.0341	-21.757	-19.0306	-19.0859	-28.5447	-24.592	-28.5758	-23.6503	-23.2995
	worst	-33.0524	-28.7602	-27.7238	-26.6708	-26.7988	-19.6941	-16.7838	-14.9575	-18.6094	-24.2428	-22.424	-22.0519	-22.4085
	std	0.753531	1.029249	1.156526	1.382968	0.148854	1.453936	1.352856	2.775349	5.53429	0.197182	3.83941	0.990097	0.513434
	median	-33.8452	-30.0898	-29.0985	-27.4178	-26.8244	-20.5988	-17.6532	-17.3119	-24.3597	-24.4649	-23.5215	-22.3697	-22.8453
	rank	1	2	3	4	5	11	12	13	8	7	6	10	9
C11-F6	mean	-23.8708	-19.5347	-18.6034	-19.2995	-17.5487	-10.07	-9.9947	-10.3197	-5.11704	-16.8751	-8.36861	-14.1672	-14.4194
	best	-27.1555	-22.2434	-21.1858	-20.1783	-19.8225	-10.91	-13.466	-11.0447	-13.7294	-19.8362	-15.1011	-16.8062	-15.8426
	worst	-22.7759	-18.4262	-17.5194	-18.1892	-16.6256	-9.19519	-7.58452	-9.44867	-2.11095	-10.3342	-2.11095	-13.2875	-13.2875
	std	2.969416	2.480261	2.37342	1.180767	2.079838	0.950513	3.440836	1.071312	7.790004	6.084039	9.512731	2.385682	1.801424
	median	-22.7759	-18.7346	-17.8543	-19.4153	-16.8733	-10.0874	-9.46413	-10.3928	-2.31391	-18.665	-8.13121	-13.2875	-14.2738
	rank	1	2	4	3	5	10	11	9	13	6	12	8	7
C11-F7	mean	0.852092	1.048561	1.085758	1.089457	1.123918	1.780284	1.887722	1.988212	1.405854	1.822112	1.007369	1.094874	1.26534
	best	0.576444	0.897991	0.952339	0.883893	0.896865	1.720403	1.61784	1.719307	1.264521	1.725317	0.924828	1.003848	1.003106
	worst	1.014777	1.115357	1.161838	1.169358	1.282126	1.884012	1.994796	2.169424	1.708546	1.991815	1.081874	1.168011	1.421776
	std	0.270131	0.137052	0.124419	0.186252	0.221654	0.097734	0.244618	0.25918	0.276654	0.16576	0.091709	0.111061	0.249484
	median	0.908573	1.090449	1.114427	1.152289	1.15834	1.758361	1.969127	2.032058	1.325174	1.785658	1.011387	1.103818	1.318238
	rank	1	3	4	5	7	10	12	13	9	11	2	6	8
C11-F8	mean	217.8	222.6991	223.0314	228.8892	221.1376	289.7481	264.1854	320.2845	255.8959	264.1854	224.0871	289.9628	221.4613
	best	217.8	220	220	220	220	260.9466	227.7112	280.9186	220	243.9048	220	227.7877	220
	worst	217.8	226.6322	227.6037	245.6912	223.8561	328.5745	301.8937	361.8864	349.7032	309.6049	233.8802	320.149	223.0819
	std	0	4.441596	5.043875	16.44641	2.497317	40.14035	43.15727	45.11147	85.26604	41.35467	8.992579	58.2482	2.294899
	median	217.8	222.082	222.2609	224.9327	220.3471	284.7357	263.5684	319.1665	226.9401	251.616	221.2341	305.9573	221.3817
	rank	1	4	5	7	2	11	9	13	8	10	6	12	3
C11-F9	mean	8701.393	79089.9	94987.57	335626.4	238273.7	653448.1	856537.4	1066351	123741.1	415750.8	175512.8	382426.9	621184.2
	best	5403.097	65546.69	78590.21	323532.6	220539.6	473118.8	761169	734578.4	99979.15	257058.4	115555.8	311431.3	589824.2
	worst	13901.86	100033.1	120001.2	353770.9	263657.4	736172	919593.3	1234383	146296.5	651308.7	251478.6	474672	639172.8
	std	4967.243	20577.92	24424.39	17447.67	25287.82	166807	91987.03	306971.7	26409.97	238454.7	76298.26	107266.9	29669.32
	median	7750.305	75389.91	90679.44	332601	234448.9	702250.8	872693.7	1148222	124344.4	377318	167508.5	371802.2	627870
	rank	1	2	3	7	6	11	12	13	4	9	5	8	10
C11-F10	mean	-21.274	-17.851	-17.2872	-16.9944	-18.3533	-11.9494	-10.2307	-11.238	-13.2345	-11.8002	-14.171	-16.7613	-16.6794
	best	-21.6116	-18.1864	-17.6145	-17.3299	-18.6005	-13.1993	-10.3777	-11.5765	-17.441	-12.4864	-19.6989	-16.9472	-16.8805
	worst	-20.58	-17.5009	-16.9822	-16.5453	-17.8385	-11.2707	-10.0925	-10.9768	-10.9748	-11.3386	-11.4067	-16.3563	-16.2652
	std	0.63683	0.475941	0.467312	0.510803	0.47298	1.190768	0.18028	0.341248	3.937268	0.660305	5.075219	0.371758	0.380606
	median	-21.4523	-17.8584	-17.2762	-17.0512	-18.4872	-11.6638	-10.2263	-11.1994	-12.2611	-11.688	-12.7891	-16.8709	-16.7859
	rank	1	3	4	5	2	10	13	12	9	11	8	6	7
C11-F11	mean	565995.2	2198529	2452508	1706489	2054692	6837040	2165140	9109225	6353489	1881902	1900098	2736779	2986780
	best	258229.6	2100385	2354043	1539264	1829047	6565080	2093116	8854810	5417762	1783748	1309884	2560298	2803298
	worst	820275.3	2316404	2552711	1836317	2238476	7202544	2224515	9267550	7517254	2013820	3111228	2882056	3127837
	std	333248.4	142576.8	123544.9	202116.6	241632.9	380895.3	76797.68	240749.8	1176166	130081.3	1109404	192433.5	188712.3
	median	592738	2188663	2451639	1725187	2075623	6790267	2171465	9157270	6239471	1865019	1589640	2752381	3007992
	rank	1	7	8	2	5	12	6	13	11	3	4	9	10
C11-F12	mean	1187807	3249665	3748991	5093441	5228348	10833650	7813616	14564001	6806356	7548715	3294660	3988704	7363500
	best	1144377	3189639	3657615	4953196	5064104	10570391	7683112	13775380	6419487	7147561	3162570	3809036	7232072
	worst	1236860	3284035	3805859	5264854	5360183	11023184	7944198	15214953	7043848	7847732	3449042	4090228	7467847
	std	60229.43	56676.18	86427.01	188279.3	165990.5	272688.1	153300.6	804237.7	376688.7	429410	200206.2	171673.7	137107

Received: July 25, 2024. Revised: October 2, 2024.

	median	1184995	3262492	3766246	5077858	5244552	10870513	7813577	14632835	6881045	7599784	3283513	4027776	7377041
	rank	1	2	4	6	7	12	11	13	8	10	3	5	9
C11-F13	mean	15289.76	15534.37	15554.17	50326.53	16574.2	15958.19	15526.3	16337.51	15561.73	15604.56	15572.45	15549.56	19574.4
	best	15289.75	15486.13	15494.25	39310.43	15503.14	15714.52	15482.38	15914.75	15506.07	15526.6	15520.52	15487.01	15535.8
	worst	15289.77	15619.41	15660.38	65146.5	19722.54	16401.72	15609.41	17280.8	15658.9	15707.1	15642.17	15653.41	31683.11
	std	0.011611	80.48495	100.5251	15779.75	2846.292	422.804	78.68115	872.4631	93.92871	123.8388	80.27942	98.61077	10946.43
	median	15289.76	15515.97	15531.02	48424.59	15535.56	15858.26	15506.71	16077.25	15540.97	15592.27	15563.56	15528.92	15539.35
	rank	1	3	5	13	11	9	2	10	6	8	7	4	12
C11-F14	mean	18112.4	62853.41	73888.97	73845.83	87550.4	162838.2	63064.25	262035.2	63756.73	63467.61	63544.48	73830.29	73826.75
	best	18059.17	20436.54	20904.85	20881.55	21245.28	106873	20673.72	177589.2	21123.26	21057.02	21087.72	20876.3	20832.75
	worst	18204.2	106655.1	128582.7	128473.7	155930.4	236610.2	106656	383688.6	108056.7	107362.1	107390	128469.4	128544.9
	std	91.44637	56121.07	70089.61	70099.26	87687.53	81607.58	55972.12	120240.4	56511.95	56152.28	56153.46	70068.11	70132.86
	median	18093.11	62161.01	73034.17	73014.03	86512.96	153934.7	62463.63	243431.5	62923.49	62725.67	62850.09	72987.74	72964.66
	rank	1	2	10	9	11	12	3	13	6	4	5	8	7
C11-F15	mean	32554.75	2364474	2944188	3026447	4226628	3296624	2524096	4146183	2384934	2540259	2364603	2944240	5138066
	best	32454.34	517543.3	638037.6	727572.9	1129036	927052.7	831286.5	1254197	517660.5	723279.9	517717.5	638089.2	1987617
	worst	32626.9	3512239	4377329	4456996	6414863	4568137	3788076	6872227	3593508	3777295	3512397	4377391	8143784
	std	98.27628	1846453	2305560	2296595	3093337	2205460	1776798	3124625	1878016	1813267	1846439	2305567	3414391
	median	32568.88	2714057	3380693	3460609	4681306	3845653	2738510	4229154	2714285	2830230	2714149	3380741	5210432
	rank	1	2	7	9	12	10	5	11	4	6	3	8	13
C11-F16	mean	132214.5	13586934	16931393	22661059	26465027	14447888	13586900	15302504	13594393	13591588	13590470	38978589	38098743
	best	130060.5	13241913	16500946	19389293	24904965	13822025	13246269	14293247	13252545	13249206	13248313	34731435	33601123
	worst	134947.7	13974576	17415344	27404186	28558455	15717826	13974766	17942251	13983864	13973804	13970973	43160482	44502204
	std	3055.305	415702.5	518999	4619008	2065407	1164413	414598.8	2391873	415748.9	411120	411747.2	4829680	6240280
	median	131924.9	13565624	16904641	21925379	26198343	14125850	13563283	14487259	13570582	13571672	13571297	39011219	37145822
	rank	1	3	9	10	11	7	2	8	6	5	4	13	12
C11-F17	mean	1907349	3.38E+09	4.22E+09	7.68E+09	6.81E+09	1.29E+10	8.31E+09	1.81E+10	4.59E+09	1.26E+10	3.38E+09	1E+10	1.03E+10
	best	1897783	3.25E+09	4.06E+09	7.24E+09	6.58E+09	1.14E+10	7.99E+09	1.38E+10	4.47E+09	1.01E+10	3.25E+09	9.3E+09	9.9E+09
	worst	1923258	3.53E+09	4.41E+09	7.95E+09	6.97E+09	1.41E+10	8.75E+09	2.15E+10	4.74E+09	1.55E+10	3.53E+09	1.11E+10	1.1E+10
	std	15330.85	1.54E+08	1.93E+08	4.38E+08	2.64E+08	1.61E+09	5.09E+08	4.45E+09	1.62E+08	3.08E+09	1.55E+08	1.14E+09	7.04E+08
	median	1904178	3.37E+09	4.21E+09	7.77E+09	6.84E+09	1.31E+10	8.25E+09	1.85E+10	4.58E+09	1.24E+10	3.37E+09	9.81E+09	1.01E+10
	rank	1	2	4	7	6	12	8	13	5	11	3	9	10
C11-F18	mean	932636.9	5522160	6656833	9771228	15980890	63061435	17454979	1.17E+08	6546008	13674962	5536107	43813641	38167225
	best	929032	4529168	5420393	8146881	14446535	43800928	11368122	81050688	5896599	7533019	4571385	38109890	36910176
	worst	935259.8	5955727	7189835	11149528	16657236	71459900	27822590	1.33E+08	7020436	20911517	5899421	47097930	39645160
	std	3543.117	904995.5	1126155	1785094	1410359	17591332	10326288	32954602	644250.2	7554659	874966.9	5644412	1673264
	median	933128	5801873	7008553	9894253	16409893	68492455	15314602	1.26E+08	6633498	13127656	5836811	45023373	38056781
	rank	1	2	5	6	8	12	9	13	4	7	3	11	10
C11-F19	mean	1015087	6358962	7669509	9182333	17152058	62821069	18259444	1.15E+08	7624691	14967162	6647696	55217810	39186390
	best	958248.4	4683137	5591066	7309535	14399738	52754668	16013437	98509514	5785337	7872773	4971051	52420101	36352611
	worst	1155471	7662559	9307147	11431069	19207195	79812893	22648780	1.45E+08	8892751	24156744	8393476	60654271	40835792
	std	127304.5	1731638	2164165	2752734	2784007	16278531	4109518	28809041	1969243	10761747	1937439	5012496	2649421
	median	973315.2	6545075	7889911	8994363	17500649	59358357	17187779	1.09E+08	7910339	13919566	6613129	53898433	39778578
	rank	1	2	5	6	8	12	9	13	4	7	3	11	10
C11-F20	mean	931837.9	6053537	7322001	11415953	16874721	66332954	16585159	1.24E+08	6875478	12039988	6064161	51266630	39037633
	best	926781.7	5935857	7174895	9901564	16444699	58849977	15079406	1.09E+08	6718872	11748200	5945179	47389243	37471730
	worst	937397.9	6178536	7478237	13851216	17364795	77641130	18300628	1.46E+08	7049586	12589574	6198044	54933824	40376597
	std	6403.284	136583.8	170666.9	2364674	512817.4	10827834	1804943	21555912	187716.2	532498.2	141674.9	5656040	1632186
	median	931585.9	6049878	7317436	10955517	16844694	64420355	16480300	1.2E+08	6866727	11911089	6056710	51371726	39151102
	rank	1	2	5	6	9	12	8	13	4	7	3	11	10
C11-F21	mean	12.58728	28.79524	32.34681	38.00116	39.81853	66.04013	45.07744	86.42775	42.25558	50.74794	38.68537	55.19432	54.3253
	best	9.874464	21.83702	23.33594	29.73812	25.38953	54.16408	35.47676	68.39954	35.39282	40.93961	28.11511	42.09047	33.03518
	worst	14.82524	34.93995	40.30472	47.13971	51.86213	76.7729	52.32495	105.1987	46.38737	59.03366	48.40893	61.61081	68.61014
	std	3.081446	7.702247	9.752454	9.718627	14.87106	16.59538	9.526881	26.52925	6.463794	10.75871	11.69897	12.12544	20.49512
	median	12.82471	29.202	32.87329	37.56341	41.01123	66.61178	46.25402	86.05636	43.62106	51.50925	39.10871	58.53801	57.82794
	rank	1	2	3	4	6	12	8	13	7	9	5	11	10
C11-F22	mean	15.96388	31.75672	35.23327	41.92937	41.94376	61.94204	53.36482	74.0998	44.40511	57.85303	43.02135	57.63744	53.73153
	best	11.38632	25.34323	27.60183	36.79969	31.93872	57.51068	43.9191	60.06303	40.24824	51.04782	31.12226	51.36268	45.75694
	worst	19.35733	36.4718	40.44158	47.23651	48.91765	69.0072	62.31853	84.73294	49.35287	63.8842	49.1952	66.14209	59.58751
	std	5.361405	6.889546	7.814412	6.056502	10.46694	6.886286	11.47779	14.10166	5.633708	9.114878	10.99954	9.788351	8.430633
	median	16.55593	32.60592	36.44483	41.84064	43.45933	60.62514	53.61082	75.80162	44.00967	58.24005	45.88397	56.5225	54.79084
	rank	1	2	3	4	5	12	8	13	7	11	6	10	9
Sum rank		22	64	108	137	147	240	191	268	161	169	131	177	187
Mean rank		1.00E+00	2.91E+00	4.91E+00	6.23E+00	6.68E+00	1.09E+01	8.68E+00	1.22E+01	7.32E+00	7.68E+00	5.95E+00	8.05E+00	8.50E+00
Total rank		1	2	3	5	6	12	11	13	7	8	4	9	10
Wilcoxon: *p*-value			6.72E-14	5.03E-15	8.80E-16	3.66E-15	1.88E-15	8.80E-16	2.06E-12	3.66E-15	2.76E-15	4.39E-15	1.31E-15	2.76E-15

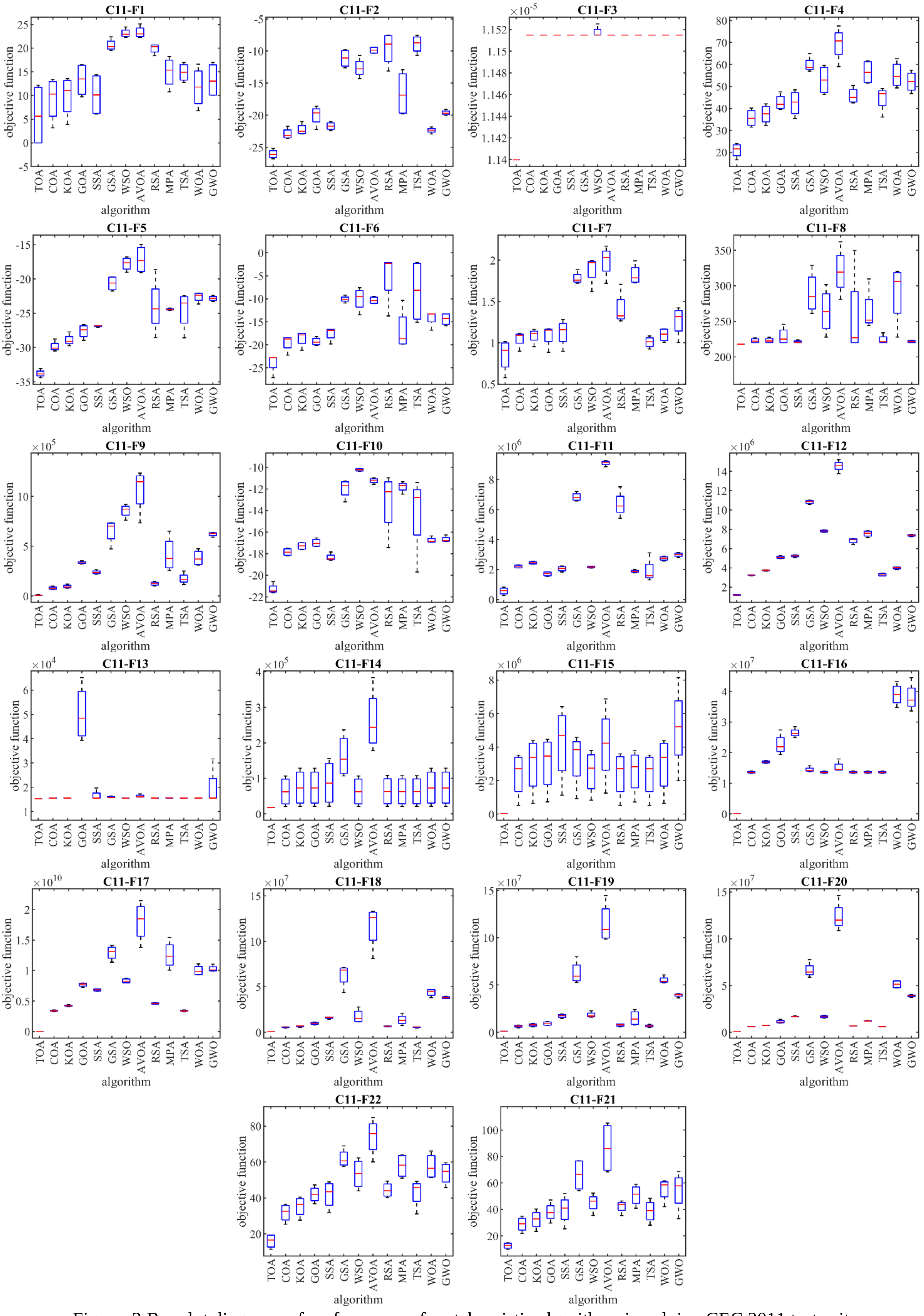

Figure. 2 Boxplot diagrams of performance of metaheuristic algorithms in solving CEC 2011 test suite

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

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

References

[1] H. Qawaqneh, "New contraction embedded with simulation function and cyclic (α, β)-admissible in metric-like spaces", *International Journal of Mathematics and Computer Science*, vol. 15, No. 4, pp. 1029-1044, 2020.

[2] T. Hamadneh, A. Hioual, O. Alsayyed, Y. A. Al-Khassawneh, A. Al-Husban, and A. Ouannas, "The FitzHugh–Nagumo Model Described by Fractional Difference Equations: Stability and Numerical Simulation", *Axioms*, Vol. 12, No. 9, pp. 806, 2023.

[3] 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.

[4] 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.

[5] Z. Benmamoun, K. Khlie, M. Dehghani, and Y. Gherabi, "WOA: Wombat Optimization Algorithm for Solving Supply Chain Optimization Problems", *Mathematics*, vol. 12, No. 7, p. 1059, 2024.

[6] R. Priyadarshi, "Energy-efficient routing in wireless sensor networks: A meta-heuristic and artificial intelligence-based approach: A comprehensive review", *Archives of Computational Methods in Engineering*, pp. 1-29, 2024.

[7] E. Elsedimy, S. M. AboHashish, and F. Algarni, "New cardiovascular disease prediction approach using support vector machine and quantum-behaved particle swarm optimization", *Multimedia Tools and Applications*, vol. 83, No. 8, pp. 23901-23928, 2024.

[8] 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.

[9] S. A. omari *et al.*, "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.

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

[11] 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, p. 110011, 2023/01/10/ 2023, doi: https://doi.org/10.1016/j.knosys.2022.110011.

[12] E. Trojovská, M. Dehghani, and P. Trojovský, "Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm", *IEEE Access*, vol. 10, pp. 49445-49473, 2022.

[13] 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.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 27 Nov.-1 Dec. 1995 1995, vol. 4: IEEE, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[15] 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.

[16] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems", in *International fuzzy systems association world congress*, 2007: Springer, pp. 789-798.

[17] 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, p. 8775, 2023.
[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.
[19] M. Dehghani, P. Trojovský, and O. P. Malik, "Green Anaconda Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics,* vol. 8, No. 1, p. 121, 2023. [Online]. Available: https://www.mdpi.com/2313-7673/8/1/121.
[20] M. Dehghani, Z. Montazeri, G. Bektemyssova, O. P. Malik, G. Dhiman, and A. E. Ahmed, "Kookaburra Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics,* vol. 8, No. 6, p. 470, 2023. [Online]. Available: https://www.mdpi.com/2313-7673/8/6/470.
[21] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning", *Machine Learning,* vol. 3, No. 2, pp. 95-99, 1988/10/01 1988, doi: 10.1023/A:1022602019183.
[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] L. N. De Castro and J. I. Timmis, "Artificial immune systems as a novel soft computing paradigm", *Soft computing,* vol. 7, No. 8, pp. 526-544, 2003.
[24] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm", *Information sciences,* vol. 179, No. 13, pp. 2232-2248, 2009.
[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *science,* vol. 220, No. 4598, pp. 671-680, 1983.
[26] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, and H. Sossa, "Circle detection using electro-magnetism optimization", *Information Sciences,* vol. 182, No. 1, pp. 40-55, 2012.
[27] 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.
[28] 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.
[29] 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.
[30] M. Dehghani *et al.*, "A spring search algorithm applied to engineering optimization problems", *Applied Sciences,* vol. 10, No. 18, p. 6173, 2020.
[31] 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, p. 110454, 2023.
[32] 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, p. 10312, 2023/06/26 2023, doi: 10.1038/s41598-023-37537-8.
[33] 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.
[34]M. Dehghani *et al.*, "A new "Doctor and Patient" optimization algorithm: An application to energy commitment problem", *Applied Sciences,* vol. 10, No. 17, p. 5791, 2020.
[35] 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.
[36] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems", *Jadavpur University, Nanyang Technological University, Kolkata,* pp. 341-359, 2010.
[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software,* vol. 69, pp. 46-61, 2014/03/01/ 2014, doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.
[38] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic", *Expert Systems with Applications,* vol. 152, p. 113377, 2020.
[39] 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, p. 103541, 2020/04/01/ 2020, doi: https://doi.org/10.1016/j.engappai.2020.103541.

[40] 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, p. 116158, 2022.

[41] 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, p. 107408, 2021.

[42] 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,* p. 108457, 2022.