Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.025908

ARTICLE

Language Education Optimization: A New Human-Based Metaheuristic Algorithm for Solving Optimization Problems

Pavel Trojovský[1,*], Mohammad Dehghani[1], Eva Trojovská[1] and Eva Milkova[2]

[1]Department of Mathematics, Faculty of Science, University of Hradec Králové, Hradec Králové, 500 03, Czech Republic

[2]Department of Applied Cybernetics, Faculty of Science, University of Hradec Králové, Hradec Králové, 500 03, Czech Republic

*Corresponding Author: Pavel Trojovský. Email: pavel.trojovsky@uhk.cz

Received: 05 August 2022 Accepted: 19 October 2022

ABSTRACT

In this paper, based on the concept of the NFL theorem, that there is no unique algorithm that has the best performance for all optimization problems, a new human-based metaheuristic algorithm called Language Education Optimization (LEO) is introduced, which is used to solve optimization problems. LEO is inspired by the foreign language education process in which a language teacher trains the students of language schools in the desired language skills and rules. LEO is mathematically modeled in three phases: (i) students selecting their teacher, (ii) students learning from each other, and (iii) individual practice, considering exploration in local search and exploitation in local search. The performance of LEO in optimization tasks has been challenged against fifty-two benchmark functions of a variety of unimodal, multimodal types and the CEC 2017 test suite. The optimization results show that LEO, with its acceptable ability in exploration, exploitation, and maintaining a balance between them, has efficient performance in optimization applications and solution presentation. LEO efficiency in optimization tasks is compared with ten well-known metaheuristic algorithms. Analyses of the simulation results show that LEO has effective performance in dealing with optimization tasks and is significantly superior and more competitive in combating the compared algorithms. The implementation results of the proposed approach to four engineering design problems show the effectiveness of LEO in solving real-world optimization applications.

KEYWORDS

Optimization; language education; exploration; exploitation; metaheuristic algorithm

1 Introduction

Numerous challenges in various sciences face several possible solutions. Such challenges are known as optimization issues. Hence, the operation of finding the best solution to such problems is called optimization [1]. In order to deal with optimization problems, these problems must be modeled mathematically. This modeling defines the optimization challenge based on the three main parts of decision variables, constraints, and objective function [2].

Optimization techniques fall into two groups: deterministic and stochastic methods. Deterministic methods are efficient on optimization topics that have a linear, convex, continuous, differentiable objective function, and a continuous search space. However, as optimization problems become more complex, deterministic approaches lose their ability in real-world applications that have features, such as non-convex, discrete, nonlinear, non-differentiable objective functions, discrete search space, and high-dimensions [3]. Such difficulties in deterministic approaches have led scientists to efforts to introduce random methods that have effective performance in solving complex optimization problems. Metaheuristic algorithms, as a sub-group of stochastic methods, are efficient tools that rely on random search in the problem-solving space [4]. Metaheuristic algorithms have become very popular thanks to the following advantages: easy implementation, simple concepts, efficiency in discrete search spaces and efficiency in nonlinear, non-convex, and NP-hard problems [5].

The two most important factors influencing the performance of metaheuristic algorithms are exploration and exploitation. Exploration represents the power of the algorithm in the global search, and exploitation represents the power of the algorithm in the local search [6]. Due to the nature of random search in metaheuristic algorithms, the solutions obtained from these methods are not guaranteed to be the best solution to the problem. However, because these solutions are close to the global optimal, they can be accepted as searched solutions to optimization problems. In fact, what has prompted researchers to develop numerous metaheuristic algorithms has been the pursuit of solutions closer to the global optimal.

Natural phenomena, the behaviors of living things in nature, the laws of physics, the concepts of biology, and other evolutionary processes have been the sources of inspiration for the design of metaheuristic algorithms. The Genetic Algorithm (GA) [7], which is inspired by concepts of biology, the Particle Swarm Optimization (PSO) [8], which is inspired by bird life, the Artificial Bee Colony (ABC) [9], which is inspired by bee colony behaviors, and the Ant Colony Optimization (ACO) [10], which is inspired by ant swarm activities, are the widely used and most famous metaheuristic algorithms.

The main research question is: Despite the numerous metaheuristic algorithms introduced till now, is there still any necessity for designing newer metaheuristic algorithms? The No-Free-Lunch Theorem (NFL) [11] answers the question because it says that there is no guarantee that an algorithm with good results in solving some optimization problems will work well in solving other optimization problems. The NFL theorem is the main incentive for researchers to introduce new metaheuristic algorithms to be able to provide better solutions for optimization tasks.

The aspects of novelty and innovation of this study are in the introduction of a new human-based metaheuristic algorithm called Language Education Optimization (LEO) that is efficient in optimization tasks. The key contributions of this paper are as follows:

- LEO is introduced based on the simulation of the foreign language education process.
- The fundamental inspiration of LEO is to train students in language schools in language skills and rules.
- The LEO theory is described and then mathematically modeled in three phases.
- The performance of LEO in optimization tasks is assessed in dealing with fifty-two standard benchmark functions.
- The results of LEO are compared with the performance of ten well-known metaheuristic algorithms.

- The effectiveness of LEO in handling real-world applications is evaluated in the optimization of four engineering design problems.

The paper consists of the following sections: a literature review is provided in the section "Literature Review." The proposed Language Education Optimization (LEO) approach is introduced and modeled in the section "Language Education Optimization." LEO simulation and evaluation studies on the handling of optimization tasks are presented in the section "Simulation Studies and Results". A discussion of the results is provided in the section "Discussion." The study evaluating the ability of the proposed LEO approach in CEC 2017 test suite optimization is presented in the section "Evaluation CEC 2017 Test Suite." The analysis of LEO capabilities in real-world applications is presented in the section "LEO for Real-World Applications." Conclusions and suggestions for further studies are expressed in the section "Conclusions and Future Researches."

2 Literature Review

Metaheuristic algorithms have been developed based on mathematical simulations of various phenomena, such as genetics and biology, swarm intelligences in the life of living organisms, physical phenomena, rules of games, human activities, etc. According to the main source of inspiration resulting in the design, metaheuristic algorithms fall into the following five groups: (i) swarm-based, (ii) evolutionary-based, (iii) physics-based, (iv) human-based, and (v) game-based methods.

Modeling the swarming behaviors and social and individual lives of living organisms (birds, aquatic animals, insects, animals, etc.) has led to the development of swarm-based metaheuristic algorithms. The major algorithms belonging to this group are PSO, ABC, and ACO. PSO is based on modeling the behavior of swarm movement of groups of fish and birds in which two factors, individuals' experience and group experience, affect the population displacement of the algorithm. ABC is based on simulating the social life of bees seeking food sources and extracting nectar from these food sources. ACO is based on the behavior of the ant colony searching the optimal path between the nest and the food sources. Artificial Hummingbird Algorithm (AHA) is a swarm-based method based on the simulation of intelligent foraging strategies and special flight skills of hummingbirds in nature [12]. Beluga Whale Optimization (BWO) is a swarm-based metaheuristic algorithm based on beluga whales' behaviors, including pair swim, prey, and whale fall [13]. Starling Murmuration Optimizer (SMO) is a bio-inspired metaheuristic algorithm that is based on starlings' behaviors during their stunning murmuration [14]. Rat Swarm Optimizer (RSO) is a bio-inspired optimizer that is proposed based on the chasing and attacking behaviors of rats in nature [15]. Sooty Tern Optimization Algorithm (STOA) is a bio-inspired metaheuristic algorithm that is introduced based on the simulation of attacking and migration behaviors of sea bird sooty tern in nature [16]. Emperor Penguin Optimizer (EPO) is proposed based on huddling behavior of emperor penguins in nature [17]. Orca Predation Algorithm (OPA) is introduced based on the hunting behavior of orcas, including driving, encircling, and attacking prey [18]. The activities of living organisms in nature, such as a search for food resources, foraging, and feeding through hunting effectively, have inspired the design of well-known metaheuristic algorithms, such as Whale Optimization Algorithm (WOA) [19], African Vultures Optimization Algorithm (AVOA) [20], Marine Predator Algorithm (MPA) [21], Golden Jackal Optimization (GJO) [22], Gray Wolf Optimizer (GWO) [23], Reptile Search Algorithm (RSA) [24], Honey Badger Algorithm (HBA) [25], Spotted Hyena Optimizer (SHO) [26], and Tunicate Swarm Algorithm (TSA) [27].

Modeling of genetics and biology concepts has been the main source for evolutionary-based metaheuristic algorithms development. The reproduction process simulation, based on the concepts of Darwin's theory of evolution and natural selection, has been the main source in the design of Differential Evolution (DE) [28] and GA.

Modeling of the physical laws and phenomena has been used in the physics-based metaheuristic algorithms development. Material engineers use the annealing method to achieve a state in which the solid is well organized, and its energy is minimized. This method involves placing the material in a high-temperature environment and following a gradual lowering of the temperature. The Simulated Annealing (SA) method simulates this solid-state annealing process to solve the optimization problem [29]. Lichtenberg Algorithm (LA) is a physics-based optimization algorithm inspired by the Lichtenberg figures patterns and the physical phenomenon of radial intra-cloud lightning [30]. Henry Gas Solubility Optimization (HGSO) is a physics-based metaheuristic algorithm that is based on imitation of the behavior governed by Henry's law [31]. Mathematical modeling of gravitational force and Newton's laws of motion [32] is used in the Gravitational Search Algorithm (GSA) design. The development of Water Cycle Algorithm (WCA) [33] is based on the natural water cycle physical phenomenon. Archimedes Optimization Algorithm (AOA) [34], Spring Search Algorithm (SSA) [35], Multi-Verse Optimizer (MVO) [36], Equilibrium Optimizer (EO) [37], and Momentum Search Algorithm (MSA) [38] are other physics-based metaheuristic algorithms.

Modeling of human activities and interactions existing in society and individuals' life has led to the emergence of human-based metaheuristic algorithms. The educational environment of the classroom and the exchange of knowledge between the teacher and the students and also among students, have been a good inspiration source for Teaching-Learning Based Optimization (TLBO) [39]. Collaboration between members of a team and presenting teamwork applied with the aim to achieve the assigned goal set to the team is the main idea of the Teamwork Optimization Algorithm (TOA) [40]. Election Based Optimization Algorithm (EBOA) is developed based on the simulation of the election and voting process in society [41]. The War Strategy Optimization (WSO) [42] is based on the strategic movement of army troops during the war. Following Optimization Algorithm (FOA) is a human-based approach based on the simulation of the impressionability of the people of the society from the most successful person in the society who is known as the leader [43]. Human Mental Search (HMS) is a human-based method that is inspired by exploration strategies of the bid space in online auctions [44]. Examples of well-established and recently developed human-based metaheuristic algorithms are: Driving Training-Based Optimization (DTBO) [45], Chef-based Optimization Algorithm (CBOA) [46], and Poor and Rich Optimization (PRO) [47].

Modeling the game rules and behavior of players, referees, and coaches brings tremendous inspiration to game-based metaheuristic algorithms development. Football League simulations and club performances resulted in the Football Game Based Optimization (FGBO) [48], and the simulation of volleyball league matches is utilized in the Volleyball Premier League Algorithm (VPL) [49]. The Tug of war game inspired the Tug of War Optimization (TWO) [50]. Archer's strategy in shooting inspired the Archery Algorithm (AA) [51], the players' attempt to solve the puzzle inspired the Puzzle Optimization Algorithm (POA) [52], and the players' skill in throwing darts inspired the Darts Game Optimizer (DGO) [53]. Some other game-based metaheuristic algorithms are Ring Toss Game-Based Optimization (RTGBO) [54], Dice Game Optimizer (DGO) [55], and Orientation Search Algorithm (OSA) [56].

We have not found any metaheuristic algorithms simulating a foreign language education process in language schools. However, the process of teaching language skills decided by the teacher and applied to the learners is an intelligent structure with remarkable potential to be used in designing a new optimizer. In order to complete this research gap, a new human-based metaheuristic algorithm based on a simulation of the foreign language teaching process and the interactions of the people involved in it is designed and presented in this paper.

3 Language Education Optimization

In this section, the metaheuristic algorithm LEO and its mathematical model based on the simulation of human activity in foreign language education is presented.

3.1 Inspiration of LEO

One of the most important ways human beings communicate with each other is by using their ability to speak. First, human beings acquire and empirically learn the official language of their society and country. With the advancement of societies and technology, communication between different nations has increased. This reality has led to the increasing importance of learning not only the native language if people are to be able to communicate with people living in other countries. As a consequence, foreign language schools have been established.

When a person decides to learn other languages, she/he has several options for choosing a school or language teacher. Choosing the appropriate school and teacher is one of the essential steps which has a great impact on the person's success in the language learning process. After the learner chooses the language teacher, she/he also communicates with other students in the classroom environment. These learners make efforts to learn language skills from the teacher training them in the given classroom environment. Additionally, to improve their skills, the students talk and practice with each other. These interactions between students improve their level of language learning. In addition, each student improves foreign language skills by doing homework and individual practice.

There are three important phases in this intelligent process, which represent the basic specifics of human activity in foreign language teaching, which must be considered into account in the new design of the metaheuristic algorithm. These three phases are (see Figs. 1 to 3): (i) students selecting their teacher, (ii) students learning from each other, and (iii) individual practice. Mathematical modeling of foreign language education based on these three phases is utilized in the design of LEO.

3.2 Algorithm Initialization

LEO is a population-based approach that is able to provide the problem-solving process for an optimization task in an iteration-based procedure. Each member of LEO is a candidate solution of the optimization problem that proposes values for decision variables. From a mathematical point of view, each LEO member can be modeled using a vector, and the population of LEO members using a matrix according to the Eq. (1).

$$\boldsymbol{X} = \begin{bmatrix} \vec{X}_1 \\ \vdots \\ \vec{X}_i \\ \vdots \\ \vec{X}_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \tag{1}$$

where $\boldsymbol{X}$ refers to the population matrix of LEO, the vector $\vec{X}_i = (x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,m})$, $i = 1, 2, \ldots, N$, is the ith member of LEO (the ith candidate solution), $x_{i,j}$ denotes its jth component (the value of the jth problem variable), N is the size of the population matrix, m is the number of problem variables.

The initial positions of all LEO members in the search space are randomly set-up by the Eq. (2).

$$x_{i,j} = lb_j + r \cdot \left(ub_j - lb_j\right), \text{ for } i = 1, 2, \ldots, N, \; j = 1, 2, \ldots, m, \tag{2}$$

where r is a random number from the interval [0, 1], lb_j and ub_j are the lower and upper bounds of the jth problem variable, respectively. For each member X_i of LEO, $i = 1, 2, \ldots, N$, we compute the value of the objective function and we according Eq. (3) create the following vector of values of the objective function for all members of LEO:

$$\vec{F} = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F\left(\vec{X}_1\right) \\ \vdots \\ F\left(\vec{X}_i\right) \\ \vdots \\ F\left(\vec{X}_N\right) \end{bmatrix}_{N\times 1}, \tag{3}$$

where F_i refers to the value of the objective function for the ith member of LEO.

As the value of the objective function is the main criterion for measuring the goodness of a candidate solution, the minimal value in the set of values of objective function $\left\{F\left(\vec{X}_i\right) | i = 1, 2, \ldots, N\right\}$ corresponds to the best candidate solution (the best LEO member), we will call it $\vec{X}_{best}$ everywhere in the following text. By the reason that in each iteration the value of all members of LEO (candidate solutions) is updated, we have to be updated in each iteration the value of the best LEO member $\vec{X}_{best.}$

3.3 Mathematical Modelling of LEO

By initializing the algorithm, candidate solutions are generated and evaluated. These candidate solutions in LEO are updated in three different phases to improve their quality.

3.3.1 Teacher Selection and Training

Each person can choose one of the available teachers in order to learn a foreign language. In LEO, for each member of the population, members who have a better objective function value than that member are considered as suggested teachers. One of these suggested teachers is randomly selected for language teaching whose schematic is shown in Fig. 1.

This strategy leads LEO members to move to different areas of the search space, which demonstrates the global search power of LEO in exploration. In order to mathematically model this phase, the set of suggested teachers for each member of LEO, thus, for the ith member of LEO, $i = 1, 2, \ldots, N$, is at first identified the set of suggested teachers ST_i using the Eq. (4).

$$ST_i = \left\{\vec{X}_k |\ k \in \{1, 2, \ldots, N\} \ \wedge \ F_k < F_i\right\} \cup \left\{\vec{X}_{best}\right\}. \tag{4}$$

where ST_i is the set of suggested teachers for the ith student (member), $\vec{X}_k$ is a population member that has a better objective function value than $\vec{X}_i$, F_k is its objective function value, k is its row number in the population matrix, and $\vec{X}_{best}$ is the best member of the population.

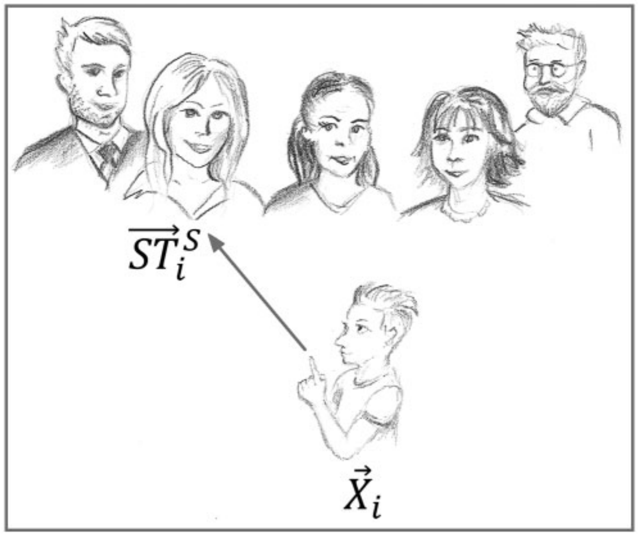

Figure 1: The first phase: Selection of a teacher

Similar to the decision of the student in language school, who chooses a teacher from among the teachers who teach in the school, in the design of LEO, this concept has also been selected for choosing a teacher. Therefore, one teacher is randomly selected among the members who have been identified as possible teachers to teach the ith student in the set ST_i.

In language school, the teacher tries to make positive changes in the student's foreign language skill level by teaching the student. Inspired by this process, in the design of LEO, the number of changes in the position of the population members has been calculated based on the subtraction of the position of the teacher and the student to improve the position of the population members in the search space. According to this, new components of each LEO member are generated for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, m$ using Eq. (5).

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

where $\vec{ST}_i^S$ is the selected teacher to teach the ith LEO member, $ST_{i,j}^S$ is its jth component, $\vec{X}_i^{P1}$ is a new position of the ith member in the search space based on the first phase of LEO, $x_{i,j}^{P1}$ is its jth component, r is a random real number from the interval [0, 1], I is a number randomly selected from the set $\{1, 2\}$. If the value of the objective function is improved for this new position, then this new position replaces the previous position of that member based on the Eq. (6).

$$\vec{X}_i = \begin{cases} \vec{X}_i^{P1}, & F_i^{P1} < F_i; \\ \vec{X}_i, & else, \end{cases} \tag{6}$$

where F_i^{P1} is the value of the objective function of the new position of the ith member $\vec{X}_i^{P1}$.

3.3.2 Students Learning from Each Other

In the second phase of LEO, population members of LEO are updated based on modeling skills exchange between students. Students try to improve their skills based on their interactions with each other whose schematic is shown in Fig. 2.

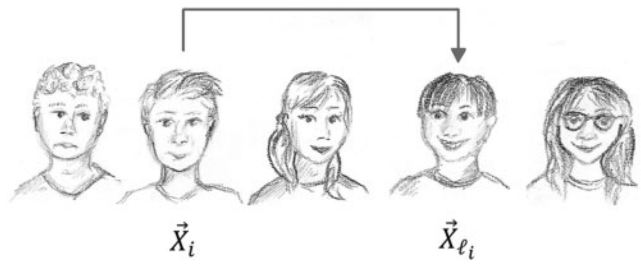

Figure 2: The second phase: Students learning in pairs

This affects the ability of LEO exploration to scan the search space. In language schools, students usually practice with each other and improve their skills. In this exercise, the student who has more skills tries to increase the scientific level of that student by teaching another student. Inspired by this interaction in language school, in LEO design, another member of the population is randomly selected for each member of the population. Then, based on the subtraction of the difference in the position of the two members, the changes in the displacement of the corresponding member are calculated. To mathematically model these interactions, for each LEO member another member of the population is randomly selected, and it is used for recomputing of its components. Thus, the new components of each member of LEO are calculated for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, m$ using the Eq. (7) and eventually updated based on the Eq. (8).

$$x_{i,j}^{P2} = \begin{cases} x_{i,j} + r \cdot \left(x_{\ell_i,j} - I \cdot x_{i,j}\right), & F_{l_i} < F_i; \\ x_{i,j} + r \cdot \left(x_{i,j} - I \cdot x_{\ell_i,j}\right), & else \end{cases} \tag{7}$$

$$\vec{X}_i = \begin{cases} \vec{X}_i^{P2}, & F_i^{P2} < F_i; \\ \vec{X}_i, & else, \end{cases} \tag{8}$$

where $\vec{X}_i^{P2}$ is the new position of the ith member in the search space based on the second phase of LEO, $x_{i,j}^{P2}$ is its jth component, F_i^{P2} is its objective function value, $\vec{X}_{\ell_i}$ is the selected student to practice and exchange language skills with the ith member of LEO, $x_{\ell_i,j}$ is its jth component, and F_{ℓ_i} is its value of the objective function, where ℓ_i is randomly selected from the set $\{1, 2, \ldots, i-1, i+1, \ldots, N\}$.

3.3.3 Individual Practice

The third phase of LEO is motivated by learning approaches that are commonly called self-learning. This is how the students make efforts to identify their own learning needs. Set learning goals, find the additional study literature and self-study online platforms. In this phase of LEO, members of LEO are updated based on simulations of individual students' practices to improve the skills they have acquired from the teacher in the first phase whose schematic is shown in Fig. 3.

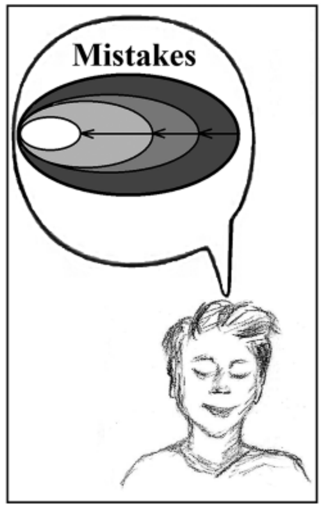

Figure 3: The third phase: Individual practice

In fact, LEO scans the search space around members based on local search, seeking better solutions. A student who goes to a language school, after participating in the class and practicing with her/his classmates, tries to improve his skills as much as possible with individual practice, which leads to small but useful changes in the student's language skills. Inspired by this student's behavior in the language learning process, in the design of LEO, the students' individual practice is modeled by making small changes in their position. To model the concepts of this LEO phase mathematically, a random position near each member is first generated using Eq. (9).

$$x_{i,j}^{P3} = x_{i,j} + \frac{lb_j + r \cdot (ub_j - lb_j)}{t}, \; i = 1, 2, \ldots, N, \; j = 1, 2, \ldots, m, \; t = 1, 2, \ldots, T, \tag{9}$$

where $x_{i,j}^{P3}$ is the jth component of the new position of the ith member $\vec{X}_i^{P3}$ based on the third phase of LEO, t is the iteration counter of the algorithm and T is total number of iterations.

Subsequently, a decision is made whether to update each LEO member $\vec{X}_i$ based on newly calculated values of $\vec{X}_i^{P3}$ according Eq. (10).

$$\vec{X}_i = \begin{cases} \vec{X}_i^{P3}, \; F_i^{P3} < F_i; \\ \vec{X}_i, \; else, \end{cases} \tag{10}$$

where F_i^{P3} is the value of the objective function of $\vec{X}_i^{P3}$.

3.3.4 Repetition Process, Pseudocode, and Flowchart of LEO

After updating all members of LEO based on all three phases, an iteration of the algorithm is completed. At the end of each iteration, the best candidate solution is updated. The iterative process of the algorithm based on Eqs. (4) to (10) continues until the end of the LEO implementation. After the LEO implementation is completed, the best candidate solution obtained during the iteration of the algorithm for the given problem is presented. The LEO pseudocode is presented in Algorithm 1 and the flowchart of its implementation is presented in Fig. 4.

Algorithm 1: Pseudocode of LEO

Start LEO.

Input problem information: variables, objective function, and constraints.

Set LEO population size (N) and the total number of iterations (T).

Generate randomly the initial population (all members of the population matrix $\boldsymbol{X}$).

Compute the vector of the values of the objective function for the initial population $\vec{F}$ and find the best member of the initial population $\vec{X}_{best}$.

For $t = 1$ to T

For $i = 1$ to N

Phase 1: Teacher selection and training

Update suggested teachers for the ith LEO member using Eq. (4).

$ST_i \leftarrow \left\{\vec{X}_k | \, k \in \{1, 2, \ldots, N\} \wedge F_k < F_i\right\} \cup \left\{\vec{X}_{best}\right\}$

Calculate a new position of the ith LEO member based on the first phase using Eq. (5).

$x_{i,j}^{P1} \leftarrow x_{i,j} + r \cdot \left(ST_{i,j}^{S} - I \cdot x_{i,j}\right)$

Update the ith LEO member using Eq. (6). $\vec{X}_i \leftarrow \begin{cases} \vec{X}_i^{P1}, & F_i^{P1} < F_i; \\ \vec{X}_i, & else. \end{cases}$

Phase 2: Students learning from each other

Calculate new position of the ith LEO member based on the second phase using Eq. (7).

$x_{i,j}^{P2} \leftarrow \begin{cases} x_{i,j} + r \cdot \left(x_{\ell_i,j} - I \cdot x_{i,j}\right), & F_l < F_i; \\ x_{i,j} + r \cdot \left(x_{i,j} - I \cdot x_{\ell_i,j}\right), & else. \end{cases}$

Update the ith LEO member using Eq. (8). $\vec{X}_i \leftarrow \begin{cases} \vec{X}_i^{P2}, & F_i^{P2} < F_i; \\ \vec{X}_i, & else. \end{cases}$

Phase 3: Simulations of students' individual practices to improve their skills

Calculate a new position of the ith LEO member based on the third phase using Eq. (9).

$x_{i,j}^{P3} \leftarrow x_{i,j} + \left(lb_j + r \cdot \left(ub_j - lb_j\right)\right) / t$

Update the ith member of LEO using Eq. (10). $\vec{X}_i \leftarrow \begin{cases} \vec{X}_i^{P3}, & F_i^{P3} < F_i; \\ \vec{X}_i, & else. \end{cases}$

End

Update $\vec{X}_{best}$.

End

Output: The best quasi-optimal solution $\vec{X}_{best}$ obtained with the LEO.

End LEO.

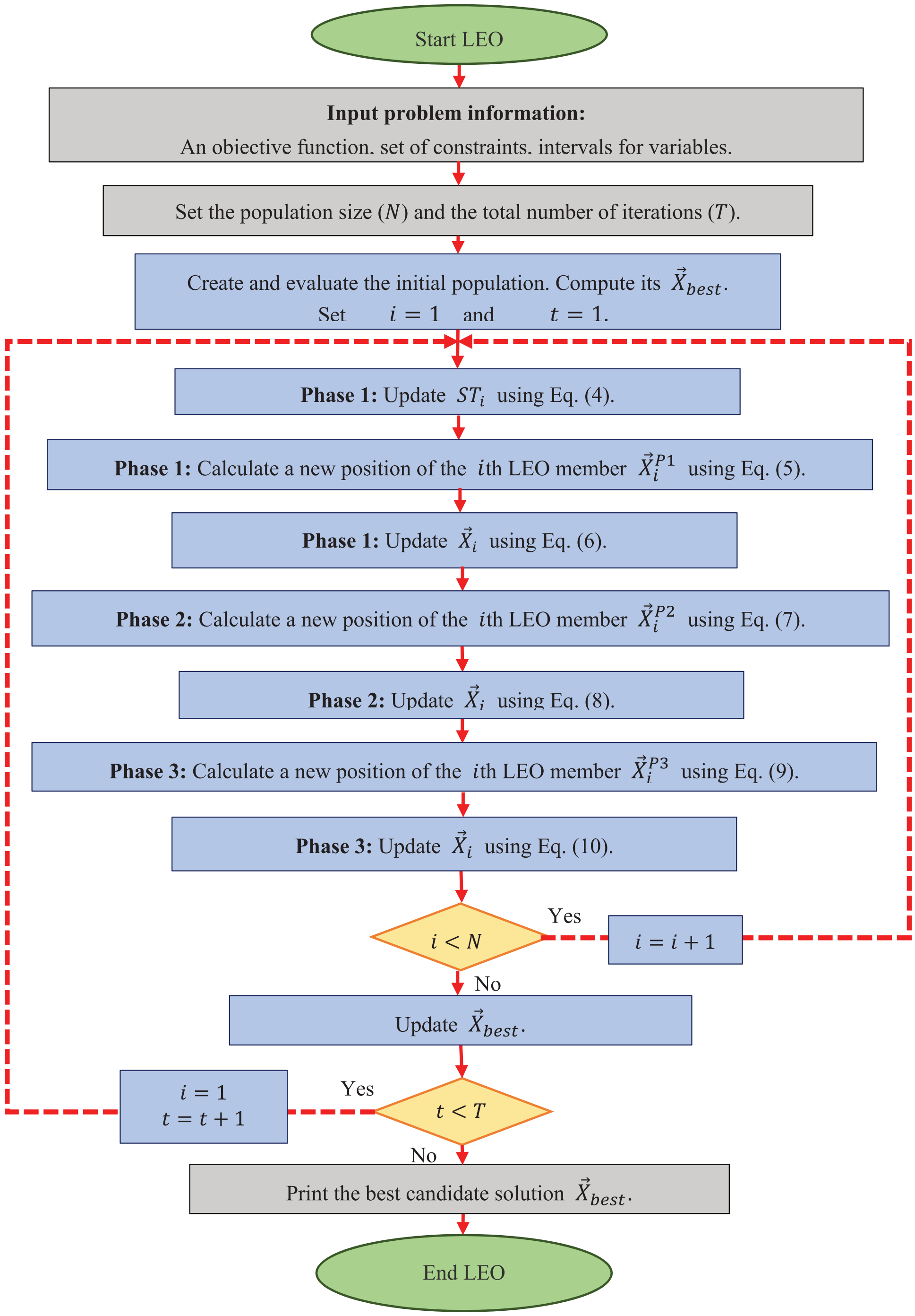

Figure 4: Flowchart of LEO

3.4 Computational Complexity of LEO

The computational complexity of LEO is analyzed in this subsection. LEO initialization has a computational complexity equal to $O(Nm)$ where N is the population size of LEO and m is the number of problem variables. In each iteration, the process of updating LEO members is carried out in three phases. Therefore, the computational complexity of the LEO population update process is equal to $O(3NmT)$, where T is the total number of iterations of the algorithm. As a result, the computational complexity of LEO is equal to $O(Nm(1+3T))$.

3.5 LEO vs. TLBO

The TLBO algorithm updates the population members of the algorithm in two phases, teacher and student. On the other hand, the proposed LEO approach updates the population members in three stages: teacher selection and training, students learning from each other, and individual practice.

In the teacher's phase of the TLBO algorithm, the best member is considered a teacher for the entire population, and the other members are considered students. But in LEO, for each member of the population, all the members with better fitness compared to that member are considered candidate teachers for the corresponding member. Among them, the teacher is randomly selected to train the corresponding member. Also, in the population update equation in TLBO, subtracting the teacher's position from the average of the entire population is used. But in the population update equation in LEO, the difference between the selected teacher's position and the corresponding member's position. In the student phase of the TLBO algorithm, the update equation is modeled based on the subtraction of the position of two students. But in the design of LEO, the subtraction of the member with better fitness than the other member multiplied by the I index is used.

Also, compared to TLBO, which only has two phases of population update, in LEO design, to increase the exploitation ability in local search, the third phase of an update called individual exercise is used.

4 Simulation Studies and Results

In this section, the capability of the proposed LEO algorithm in optimization applications is studied. A set of twenty-three objective functions including seven unimodal functions, six multimodal functions, and ten fixed-dimensional multimodal functions have been utilized to analyze LEO performance. Details and full description of these benchmark functions are provided in [57]. The reasons for selecting these benchmark functions are explained below. Unimodal functions, including F1 to F7, have only one extremum in their search space and, therefore, lack local optimal solutions. The purpose of optimizing these types of functions is to test the exploitation power of the metaheuristic algorithm in the local search and to get as close to the global optimal as possible. multimodal functions, including F8 to F13, have a number of extremums, of which only one is the main extremum and the rest are local extremums. The main purpose of optimizing this type of functions is to test the exploration power of the metaheuristic algorithm in the global search to achieve the main extremum and not get stuck in other local extremums. Multimodal functions including F14 to F23 have smaller dimensions, as well as fewer local extremums, compared to multimodal functions F8 to F13. The purpose of optimizing these functions is to simultaneously test the exploration and exploitation of the metaheuristic algorithm in local and global searches. In fact, the purpose of selecting these functions is to benchmark the ability of the metaheuristic algorithm to strike a balance between exploitation and exploration. Details of these functions are provided in Tables 1–3 [57].

Table 1: Information on unimodal objective functions

	Objective function	Range	Dimensions (m)	$F_{\min}$
1.	$F_1(x) = \sum_{i=1}^{m} x_i^2$	$[-100,\ 100]^m$	30	0
2.	$F_2(x) = \sum_{i=1}^{m} \lvert x_i \rvert + \prod_{i=1}^{m} \lvert x_i \rvert$	$[-10,\ 10]^m$	30	0
3.	$F_3(x) = \sum_{i=1}^{m} \left(\sum_{j=1}^{i} x_i\right)^2$	$[-100,\ 100]^m$	30	0
4.	$F_4(x) = \max\{\lvert x_i \rvert\ ,\ 1 \le i \le m\}$	$[-100,\ 100]^m$	30	0
5.	$F_5(x) = \sum_{i=1}^{m-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2))$	$[-30,\ 30]^m$	30	0
6.	$F_6(x) = \sum_{i=1}^{m} \lfloor x_i + 0.5 \rfloor^2$	$[-100,\ 100]^m$	30	0
7.	$F_7(x) = \sum_{i=1}^{m} i x_i^4 + random\,(0, 1]$	$[-1.28,\ 1.28]^m$	30	0

Table 2: Information on multimodal objective functions

	Objective function	Range	Dimensions (m)	$F_{\min}$
8.	$F_8(x) = \sum_{i=1}^{m} - x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500,\ 500]^m$	30	$-418.98\ m$
9.	$F_9(x) = \sum_{i=1}^{m} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$[-5.12,\ 5.12]^m$	30	0
10.	$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m} x_i^2}\right) - \exp\left(\frac{1}{m}\sum_{i=1}^{m}\cos\ (2\pi x_i)\right) + 20 + e$	$[-32,\ 32]^m$	30	0
11.	$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{m} x_i^2 - \prod_{i=1}^{m}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,\ 600]^m$	30	0
12.	$F_{12}(x) = \frac{\pi}{m}\ \left\{10\sin(\pi y_1) + \sum_{i=1}^{m-1}(y_i - 1)^2 \left(1 + 10\sin^2(\pi y_{i+1})\right) + (y_m - 1)^2\right\} + \sum_{i=1}^{m} u(x_i, 10, 100, 4)$, where $y_i = 1 + \frac{1 + x_i}{4}$, $u(x_i, a, i, n) = \begin{cases} k(x_i - a)^n,\ x_i > -a; \\ 0,\ -a \le\ x_i\ \le a; \\ k(-x_i - a)^n,\ x_i < -a. \end{cases}$	$[-50,\ 50]^m$	30	0
13.	$F_{13}(x) = 0.1\ \left\{\sin^2(3\pi x_1) + \sum_{i=1}^{m}(x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2 \left[1 + \sin^2(2\pi x_m)\right]\right\} + \sum_{i=1}^{m} u(x_i,\ 5,\ 100,\ 4)$	$[-50,\ 50]^m$	30	0

Table 3: Information on fixed-dimensional multimodal objective functions

	Objective function	Range	Dimensions (m)	$F_{\min}$
14.	$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right)^{-1}$	$[-65.53, 65.53]^2$	2	0.998
15.	$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	$[-5, 5]^4$	4	0.00030
16.	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$	2	−1.0316
17.	$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	$[-5, 10] \times [0, 15]$	2	0.398
18.	$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14(x_1 - x_2) + (x_1 + x_2)^2\right)\right] \cdot \left[30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	$[-5, 5]^2$	2	3
19.	$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij} (x_j - P_{ij})^2\right)$	$[0, 1]^3$	3	−3.86
20.	$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij} (x_j - P_{ij})^2\right)$	$[0, 1]^6$	6	−3.32
21.	$F_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i) \cdot (X - a_i)^T + 6c_i\right]^{-1}$	$[0, 10]^4$	4	−10.1532
22.	$F_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i) \cdot (X - a_i)^T + 6c_i\right]^{-1}$	$[0, 10]^4$	4	−10.4029
23.	$F_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i) \cdot (X - a_i)^T + 6c_i\right]^{-1}$	$[0, 10]^4$	4	−10.5364

LEO's ability in optimization applications is compared with the performance of ten well-known metaheuristic algorithms. The reasons for choosing these competitor algorithms are explained below. The first group includes the widely used and well-known GA and PSO algorithms. The second group includes highly cited algorithms TLBO, MVO, GSA, GWO, and WOA, which have been employed by researchers in many optimization applications. The third group includes MPA, RSA, and TSA algorithms that have recently been published and have received a lot of attention. The control parameters of the competitor algorithms are set according to Table 4.

Table 4: Control parameters values

Algorithm	Parameter	Value
GA		
	Type	Real coded
	Selection	Roulette wheel (Proportionate)
	Crossover	Whole arithmetic (Probability = 0.8, $\alpha \in [-0.5,\ 1.5]$)
	Mutation	Gaussian (Probability = 0.05)
	Population size	50

(Continued)

Table 4 (continued)

Algorithm	Parameter	Value
PSO		
	Topology	Fully connected
	Cognitive and social constant	$(C_1, C_2) = (2,\ 2)$.
	Inertia weight	Linear reduction from 0.9 to 0.1
	Velocity limit	10% of dimension range
	Population size	50
GSA		
	Alpha, G_0, R_{norm}, R_{power}	20, 100, 2, 1
	Population size	50
TLBO		
	T_F: teaching factor random number	$T_F = \text{round}\left[(1 + rand)\right]$, where *rand* is a random number from $[0,\ 1]$.
	Population size	50
GWO		
	Convergence parameter (a)	a: Linear reduction from 2 to 0.
	Population size	50
MVO		
	Wormhole existence probability (WEP)	$\min(WEP) = 0.2$ and $\max(WEP) = 1$.
	Exploitation accuracy over the iterations (p)	$p = 6$.
	Population size	50
WOA		
	Convergence parameter (a)	a: Linear reduction from 2 to 0.
	r is a random vector from $[0,\ 1]$.	
	l is a random number in $[-1,\ 1]$.	
	Population size	50
TSA		
	P_{min} and P_{max}	1, 4
	c1, c2, c3	random numbers from the interval $[0,\ 1]$.
	Population size	50
MPA		
	Constant number	$P = 0.5$
	Random vector	R is a vector of uniform random numbers from $[0,\ 1]$.
	Fish Aggregating Devices ($FADs$)	$FADs = 0.2$
	Binary vector	$U =$ 0 or 1
	Population size	50
RSA		
	Sensitive parameter	$\beta = 0.01$
	Sensitive parameter	$\alpha = 0.1$

(Continued)

Table 4 (continued)

Algorithm	Parameter	Value
	Evolutionary Sense (ES)	ES: randomly decreasing values between 2 and −2
	Population size	50
LEO		
	Population size	30

The LEO method and ten competing algorithms are each employed in twenty independent executions, while each execution contains 1000 iterations to optimize the objective functions. The results of these simulations are reported using indicators: best, mean, median, standard deviation (std), execution time (ET), and rank.

4.1 Evaluation Unimodal Objective Function

The results of recruiting LEO and competitor algorithms to handle the benchmark functions F1 to F7 are reported in Table 5. Based on the optimization results, it is inferred that LEO with a high local search capability has been able to converge to the global optimal in handling the functions of F1, F2, F3, F4, F5, and F6. Additionally, in handling F7, the proposed LEO approach ranks first as the best optimizer for this function compared to competitor algorithms. What can be deduced from the analysis of the simulation results is that LEO has a superior performance in handling the unimodal functions of F1 to F7 compared to ten competitor algorithms, by presenting much better and more competitive results.

Table 5: Optimization results on unimodal functions

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
F_1	Mean	0	0	2.91E-49	1.19E-46	3.10E-149	0.148926	6.92E-59	7.88E-75	1.42E-16	0.076198	32.8966
	best	0	0	7.27E-52	2.48E-51	7.40E-175	0.085859	5.81E-62	1.06E-77	5.40E-17	1.80E-05	21.06668
	std	0	0	6.05E-49	4.64E-46	1.40E-148	0.040488	1.22E-58	1.35E-74	9.22E-17	0.162114	11.34771
	median	0	0	4.82E-50	1.22E-48	7.30E-158	0.1438	1.34E-59	1.32E-75	1.06E-16	0.00377	28.87703
	ET	2.8937191	15.362978	4.0495352	1.1430597	0.5090984	3.0899855	1.3499048	1.7468798	4.0656678	0.5208846	0.7732775
	rank	1	1	5	6	2	9	4	3	7	8	10
F_2	mean	0	0	9.36E-28	2.73E-28	1.80E-105	0.237577	1.29E-34	7.37E-39	5.70E-08	0.748007	2.952664
	best	0	0	7.18E-32	2.63E-30	4.40E-111	0.132067	4.73E-36	2.95E-41	3.45E-08	0.034689	1.88159
	std	0	0	1.40E-27	7.71E-28	4.50E-105	0.065336	1.81E-34	7.01E-39	2.75E-08	0.491811	0.710477
	median	0	0	1.88E-28	2.57E-29	2.30E-107	0.244521	8.18E-35	4.99E-39	4.94E-08	0.738133	2.883687
	ET	2.9539787	15.467449	2.4832809	1.1412211	0.5172495	2.61118	1.3202898	1.880538	3.8135901	0.5116	0.8022538
	rank	1	1	6	5	2	8	4	3	7	9	10
F_3	mean	0	0	3.59E-12	2.33E-10	16661.72	13.9069	4.85E-16	2.62E-25	424.2214	1330.795	2369.284
	best	0	0	7.75E-18	2.28E-20	1815.858	3.460768	9.76E-20	4.02E-28	204.953	39.58518	1050.732
	std	0	0	1.18E-11	8.32E-10	9734.605	5.176504	7.98E-16	6.12E-25	179.0495	2194.333	786.8941
	median	0	0	9.15E-14	7.97E-14	16053.15	13.61052	1.62E-16	1.28E-26	379.9541	252.6239	2356.537
	ET	8.016984	17.348643	5.80953	2.8731348	2.2469567	6.2164555	3.0022204	7.1115726	5.5055033	2.2323942	2.7865939
	rank	1	1	4	5	10	6	3	2	7	8	9
F_4	mean	0	0	2.02E-19	0.008548	40.69107	0.548397	7.80E-14	2.23E-30	1.848961	5.795349	3.327428
	best	0	0	2.18E-20	7.87E-05	0.015711	0.308784	1.01E-15	1.06E-31	0.115681	3.024605	2.1599
	std	0	0	1.39E-19	0.012793	30.82539	0.128869	3.06E-13	4.42E-30	1.561592	1.597663	0.551305
	median	0	0	1.68E-19	0.005147	38.12119	0.518	4.49E-15	9.56E-31	1.708118	5.705651	3.446797

(Continued)

Table 5 (continued)

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
	ET	2.8576027	15.674336	2.278244	1.113035	0.4876719	2.6937605	1.2761624	1.9604716	3.7837563	0.5207023	0.7268312
	rank	1	1	3	5	10	6	4	2	7	9	8
F_5	mean	14.96979	15.775355	23.61681	28.30852	27.35899	455.0556	26.7365	26.74216	29.41177	244.4892	413.4251
	best	0	7.33E-29	22.93308	26.24623	26.5709	26.22482	26.09765	25.67999	24.60701	11.84862	196.9711
	std	12.17512	11.85089	0.446521	0.765354	0.682608	797.5517	0.720383	0.919737	13.49202	660.584	131.1147
	median	24.92267	1.03E-28	23.55474	28.64422	27.11864	34.62966	26.3345	26.37507	26.37678	81.73046	386.2732
	ET	3.5164898	15.768983	2.792387	1.3642896	0.8270417	3.1662504	1.528469	2.5528939	3.9652233	0.7641116	1.0784773
	rank	1	2	3	7	6	11	4	5	8	9	10
F_6	mean	0	6.730564	1.86E-09	3.633548	0.054408	0.152641	0.734942	1.118305	1.29E-16	0.968726	32.57338
	best	0	3.580407	7.14E-10	2.818033	0.010794	0.096246	0.251521	0.3022	6.39E-17	3.58E-05	15.90207
	std	0	0.941189	1.01E-09	0.448715	0.067093	0.03576	0.237898	0.428271	6.03E-17	3.76892	12.48423
	median	0	7.077314	1.58E-09	3.558282	0.020215	0.161175	0.751677	1.16301	1.20E-16	0.006687	29.24295
	ET	2.7485223	15.385006	2.2584292	1.1126183	0.4742653	2.7653587	1.2809443	1.9332615	3.7752441	0.5209432	0.7771004
	rank	1	10	3	9	4	5	6	8	2	7	11
F_7	mean	1.82E-05	4.91E-05	0.000687	0.005114	0.001269	0.010608	0.000741	0.001977	0.070721	0.171905	0.009377
	best	4.48E-07	5.84E-07	0.000194	0.000959	3.88E-05	0.005372	0.000129	0.000503	0.032415	0.082012	0.003451
	std	1.53E-05	5.97E-05	0.000331	0.003311	0.001309	0.003574	0.000458	0.001321	0.048188	0.07817	0.003122
	median	1.30E-05	3.46E-05	0.000668	0.004969	0.00118	0.011065	0.000684	0.001384	0.055562	0.168856	0.009215
	ET	4.8368757	16.256141	3.7829544	1.8270088	1.1957864	4.4459639	2.0153875	3.9675918	4.4816348	1.1786659	2.0212563
	rank	1	2	3	7	5	9	4	6	10	11	8
Sum rank		7	16	27	44	39	54	29	29	48	61	66
Mean rank		1	2.2857	3.857143	6.285714	5.571429	7.714286	4.142857	4.142857	6.857143	8.714286	9.428571
Total rank		1	2	3	6	5	8	4	4	7	9	10

4.2 Evaluation Multimodal Objective Function

The optimization results for the multimodal functions F8 to F13 using LEO and competitor algorithms are released in Table 6. The simulation results show the high global search power of LEO in identifying the main optimal area in the search space and providing the global optima for functions F9 and F11. In tackling the functions F10, F12, and F13, the LEO approach ranks first as the best optimizer against ten competitor algorithms. In handling F8, the proposed LEO approach is the second-best optimizer for this function after GA. Analysis of simulation results shows that LEO has an effective capability in global search and has outperformed competitor algorithms in handling functions F8 to F13.

Table 6: Optimization results of multimodal functions

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
F_8	mean	−12036.5	−5492.29	−9682.86	−6071.81	−11575	−8125.01	−5789.44	−5205.39	−2699.69	−6733.77	−8780.06
	best	−12569.5	−5680.5	−10592.5	−7029.82	−12569.4	−9232.93	−7112.15	−5843.69	−3817.06	−9667.69	−9669.95
	std	1301.688	174.523	398.4401	535.9141	1490.39	743.772	1053.789	345.395	481.9165	1106.838	534.6354
	median	−12569.5	−5532.59	−9609.38	−6133.59	−12345.4	−8199.62	−6053.67	−5126.91	−2711.14	−6652.61	−8830.14
	ET	3.3654446	15.793495	2.7434296	1.3305275	0.7168849	2.5087217	1.5015152	2.918874	4.015091	0.8559844	1.4428962
	rank	1	9	3	7	2	5	8	10	11	6	4
F_9	mean	0	0	0	161.7428	0	114.4433	8.53E-15	0	28.40606	67.1125	61.93813
	best	0	0	0	91.14753	0	79.67151	0	0	17.90926	34.825	28.30534
	std	0	0	0	46.30094	0	21.49045	2.08E-14	0	7.07584	22.56296	18.83339
	median	0	0	0	151.9278	0	110.5188	0	0	27.85884	62.19637	58.77781
	ET	2.8669459	17.212279	2.376211	1.2647991	0.5323911	3.0123371	1.3347335	2.1450715	3.8513696	0.6159458	1.0649503
	rank	1	1	1	7	1	6	2	1	3	5	4

(Continued)

Table 6 (continued)

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
F_{10}	mean	8.88E-16	8.88E-16	4.09E-15	1.195842	4.09E-15	0.670089	1.58E-14	4.26E-15	8.17E-09	3.030108	3.637002
	best	8.88E-16	8.88E-16	8.88E-16	7.99E-15	8.88E-16	0.082962	7.99E-15	8.88E-16	5.73E-09	1.963289	2.996306
	std	0	0	1.09E-15	1.525827	2.28E-15	0.570127	3.75E-15	7.94E-16	1.85E-09	0.869371	0.374535
	median	8.88E-16	8.88E-16	4.44E-15	2.22E-14	4.44E-15	0.627017	1.51E-14	4.44E-15	7.89E-09	2.836942	3.589696
	ET	3.0094849	17.125813	2.4113186	1.2760752	0.5678977	3.1039018	1.3526236	2.1633398	3.8715898	0.6312308	1.0342777
	rank	1	1	2	7	2	6	4	3	5	8	9
F_{11}	mean	0	0	0	0.005483	0	0.367801	0.000419	0	6.910967	0.085506	1.51316
	best	0	0	0	0	0	0.198274	0	0	2.816413	0.000312	1.254768
	std	0	0	0	0.006307	0	0.097641	0.001874	0	2.991608	0.112577	0.136013
	median	0	0	0	0	0	0.351523	0	0	6.936742	0.038802	1.545753
	ET	3.5200459	16.768273	2.9110024	1.4053481	0.7795856	3.6463012	1.5800646	3.6000582	4.2177367	0.8763135	1.1001583
	rank	1	1	1	3	1	5	2	1	7	4	6
F_{12}	mean	1.57E-32	1.285311	2.01E-10	6.893611	0.006199	0.796597	0.034746	0.071769	0.453499	1.072484	0.171397
	best	1.57E-32	0.894772	5.04E-11	2.099116	0.000818	0.00085	0.013112	0.028785	4.45E-19	0.1061	0.06194
	std	2.81E-48	0.270639	1.64E-10	3.433371	0.004561	0.816088	0.015606	0.018791	0.763806	0.957253	0.149226
	median	1.57E-32	1.111042	1.63E-10	7.188082	0.005082	0.535963	0.033764	0.071666	0.103669	0.920981	0.117873
	ET	9.6997483	20.339866	7.1015617	3.5621885	2.8451768	8.1251023	3.6690062	9.844877	6.0840876	2.7898242	2.9549824
	rank	1	10	2	11	3	8	4	5	7	9	6
F_{13}	mean	1.35E-32	5.86E-28	0.004674	3.170441	0.180748	0.03488	0.476456	1.0485	0.013185	5.6271	2.497715
	best	1.35E-32	6.28E-32	1.06E-09	2.163469	0.010754	0.012523	0.198365	0.60571	6.08E-18	0.169616	1.233345
	std	2.81E-48	2.62E-27	0.011071	0.54377	0.136257	0.019653	0.18628	0.230574	0.025606	5.095354	0.874627
	median	1.35E-32	5.15E-31	2.93E-09	3.19416	0.141566	0.030787	0.436048	1.081058	1.46E-17	4.537058	2.204447
	ET	9.5685345	20.25093	7.1125807	12.459267	2.8430157	7.3899077	3.6610298	8.8516304	6.1925442	2.7882383	3.1005814
	rank	1	2	3	10	6	5	7	8	4	11	9
Sum rank		6	24	12	45	15	35	27	28	37	43	38
Mean rank		1	4	2	7.5	2.5	5.833333	4.5	4.666667	6.166667	7.166667	6.333333
Total rank		1	4	2	11	3	7	5	6	8	10	9

4.3 Evaluation Fixed-Dimensional Multimodal Objective Function

The results of recruiting LEO and ten competitor algorithms to tackle fixed-dimensional multimodal functions F14 to F23 are reported in Table 7. The optimization results show that LEO is the best optimizer in handling functions F14, F19, and F20 against ten competitor algorithms. In functions where LEO has a performance similar to some of the competitor algorithms in presenting the "mean" index, it can be seen that the proposed algorithm has provided more efficient performance in handling the relevant functions by providing better values for the "std" index. What can be deduced from the analysis of the simulation results is that LEO has a superior performance compared to competitor algorithms by having the suitable ability to search globally and locally, as well as maintaining a balance between exploration and exploitation.

Table 7: Optimization results for fixed-dimensional multimodal function

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
F_{14}	Mean	0.998004	4.717972	0.998004	9.568298	1.34532	0.998004	4.91193	0.998004	2.75256	2.622134	0.998599
	Best	0.998004	1.002247	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	1.135726	0.998004	0.998004
	Std	0	3.552871	0	4.734825	0.739327	2.16E-12	4.46652	5.07E-07	1.072373	2.683317	0.002326
	Median	0.998004	2.982105	0.998004	10.76318	0.998004	0.998004	2.982105	0.998004	2.916217	0.998004	0.998004
	ET	17.7462491	8.0923956	11.635424	5.4224719	5.4019958	11.709339	5.3290457	17.22723	6.2926096	5.2216184	5.5117843
	Rank	1	8	1	10	5	2	9	3	7	6	4
F_{15}	Mean	0.000307	0.001291	0.000307	0.008292	0.000622	0.004623	0.00637	0.003568	0.002911	0.003688	0.005315
	Best	0.000307	0.000682	0.000307	0.000308	0.000316	0.000308	0.000307	0.000308	0.001046	0.000307	0.000716
	std	1.07E-30	8.10E-15	2.15E-30	1.44E-13	2.93E-15	8.08E-14	9.40E-14	7.25E-14	1.72E-14	7.20E-14	6.55E-14
	median	0.000307	0.000977	0.000307	0.000481	0.000579	0.000721	0.000308	0.000372	0.002269	0.000672	0.002452
	ET	2.77643102	3.1187178	1.6154843	0.4930209	0.4477875	1.2104194	0.5224414	1.8548539	1.619234	0.3907623	0.6492235

(Continued)

Table 7 (continued)

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
	rank	1	4	2	11	3	8	10	6	5	7	9
F_{16}	mean	−1.03163	−1.02994	−1.03163	−1.02847	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163
	best	−1.03163	−1.03161	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163
	std	2.28E-27	2.65E-14	2.22E-27	9.74E-14	8.93E-22	2.62E-19	3.80E-20	1.74E-17	1.44E-27	1.44E-27	4.35E-17
	median	−1.03163	−1.03092	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163
	ET	2.48395666	1.8550119	1.5035697	0.4205441	0.4086445	1.0698828	0.4390887	1.6525417	1.4601808	0.2923966	0.5743327
	rank	1	7	1	8	2	4	3	5	1	1	6
F_{17}	mean	0.397887	0.424048	0.397887	0.397917	0.397888	0.397887	0.397888	0.397975	0.397887	0.727026	0.43026
	best	0.397887	0.398061	0.397887	0.397888	0.397887	0.397887	0.397887	0.397888	0.397887	0.397887	0.397887
	std	0	6.52E-13	0	3.52E-16	5.45E-18	1.20E-18	7.38E-18	1.17E-15	0	6.97E-12	1.43E-12
	median	0.397887	0.405146	0.397887	0.397902	0.397887	0.397887	0.397888	0.397934	0.397887	0.397887	0.397904
	ET	2.27247994	1.9477333	1.5244566	0.3927561	0.3839855	1.0321453	0.4026287	1.5180712	1.4767552	0.2382841	0.5140409
	rank	1	7	1	5	3	2	4	6	1	9	8
F_{18}	mean	3	3.000035	3	4.35002	3.000008	3.000001	3.000007	3.000001	3	3	4.384222
	best	3	3	3	3.000001	3	3	3	3	3	3	3
	std	3.06E-27	6.45E-16	9.50E-27	6.04E-11	1.41E-16	5.94E-18	7.51E-17	1.30E-17	3.94E-26	2.83E-26	6.13E-11
	median	3	3.000017	3	3.000005	3.000002	3	3.000005	3.000001	3	3	3.000653
	ET	2.32502184	1.6320922	1.3984629	0.3793201	0.3472154	0.9753871	0.369263	1.4492838	1.3355731	0.2360337	0.4987929
	rank	1	8	1	9	7	4	6	5	3	2	10
F_{19}	mean	−3.86278	−3.83729	−3.86278	−3.86274	−3.85965	−3.86278	−3.8625	−3.86168	−3.86278	−3.86278	−3.86233
	best	−3.86278	−3.86107	−3.86278	−3.86278	−3.86278	−3.86278	−3.86278	−3.86269	−3.86278	−3.86278	−3.86278
	std	2.28E-26	2.45E-13	2.28E-26	2.90E-16	3.95E-14	6.19E-19	8.28E-15	2.35E-14	1.97E-26	1.99E-26	1.64E-14
	median	−3.86278	−3.84502	−3.86278	−3.86275	−3.86192	−3.86278	−3.86276	−3.86243	−3.86278	−3.86278	−3.86277
	ET	6.82789298	2.4637427	1.6723552	0.5636936	0.5091525	1.3600367	0.5640221	2.1025086	1.6371261	0.4054243	0.7175955
	rank	1	8	1	3	7	2	4	6	1	1	5
F_{20}	mean	−3.322	−2.68989	−3.322	−3.23413	−3.2744	−3.25053	−3.21956	−3.26134	−3.322	−3.25512	−3.20621
	best	−3.322	−3.08344	−3.322	−3.32134	−3.32193	−3.322	−3.32199	−3.31429	−3.322	−3.322	−3.31857
	std	4.56E-27	3.95E-12	4.20E-27	2.00E-12	8.27E-13	5.99E-13	7.69E-13	5.76E-13	4.08E-27	7.90E-13	1.00E-12
	median	−3.322	−2.84683	−3.322	−3.32021	−3.32126	−3.20308	−3.20286	−3.30078	−3.322	−3.322	−3.20919
	ET	3.2087629	4.3023587	1.7909341	0.6611978	0.5274411	1.3858897	0.644364	2.2403645	1.8011172	0.4276574	0.702935
	rank	1	9	1	6	2	5	7	3	1	4	8
F_{21}	mean	−10.1532	−5.0552	−10.1532	−4.87294	−8.6648	−8.26233	−9.90012	−7.0017	−5.43537	−5.02652	−4.87562
	best	−10.1532	−5.0552	−10.1532	−10.0977	−10.153	−10.1532	−10.1531	−9.34076	−10.1532	−10.1532	−8.95019
	std	3.58E-26	3.06E-18	2.31E-26	2.83E-11	2.77E-11	3.04E-11	1.13E-11	1.65E-11	3.41E-11	3.49E-11	2.35E-11
	median	−10.1532	−5.0552	−10.1532	−4.39643	−10.1511	−10.1531	−10.1528	−7.33007	−3.37362	−2.68286	−4.62892
	ET	3.18573257	3.0584338	2.0165565	0.7852542	0.6399693	1.6992735	0.7098701	2.4809084	1.7481269	0.5359069	0.8200472
	rank	1	7	1	10	3	4	2	5	6	8	9
F_{22}	mean	−10.4029	−5.08767	−10.4029	−7.5522	−7.72926	−8.58232	−10.4025	−7.294	−10.2257	−6.86804	−7.26614
	best	−10.4029	−5.08767	−10.4029	−10.3694	−10.4027	−10.4029	−10.4028	−10.239	−10.4029	−10.4029	−10.1895
	std	2.97E-26	7.73E-18	3.65E-26	3.41E-11	3.11E-11	2.92E-11	2.96E-15	1.70E-11	7.93E-12	3.70E-11	2.20E-11
	median	−10.4029	−5.08767	−10.4029	−10.0003	−10.3963	−10.4029	−10.4025	−7.67937	−10.4029	−7.76588	−7.7724
	ET	3.49341681	3.4076357	2.1893865	0.8162337	0.7114147	1.6880477	0.8373461	2.7606977	1.8845902	0.6828267	0.9331872
	rank	1	10	1	6	5	4	2	7	3	9	8
F_{23}	mean	−10.5364	−5.12847	−10.5364	−5.74677	−8.85422	−8.95456	−10.5359	−7.90185	−10.5364	−5.99368	−7.44674
	best	−10.5364	−5.12848	−10.5364	−10.4859	−10.5363	−10.5364	−10.5363	−9.55081	−10.5364	−10.5364	−10.219
	std	1.82E-26	1.71E-17	2.61E-26	3.61E-11	3.01E-11	2.87E-11	1.86E-15	1.60E-11	1.63E-26	3.87E-11	1.94E-11
	median	−10.5364	−5.12847	−10.5364	−4.34441	−10.5325	−10.5363	−10.5359	−8.31455	−10.5364	−3.83543	−8.09458
	ET	3.85466265	3.4715726	2.4489502	0.9250013	0.833995	1.9249851	0.9267729	3.3723531	1.9306753	0.7938576	1.0732158
	rank	1	10	2	9	5	4	3	6	2	8	7
Sum rank		10	78	12	77	42	39	50	52	30	55	74
Mean rank		1	7.8	1.2	7.7	4.2	3.9	5	5.2	3	5.5	7.4
Total rank		1	11	2	10	5	4	6	7	3	8	9

Boxplot diagrams of the proposed LEO and competitor algorithms to handle the functions F1 to F23 are shown in Fig. 5.

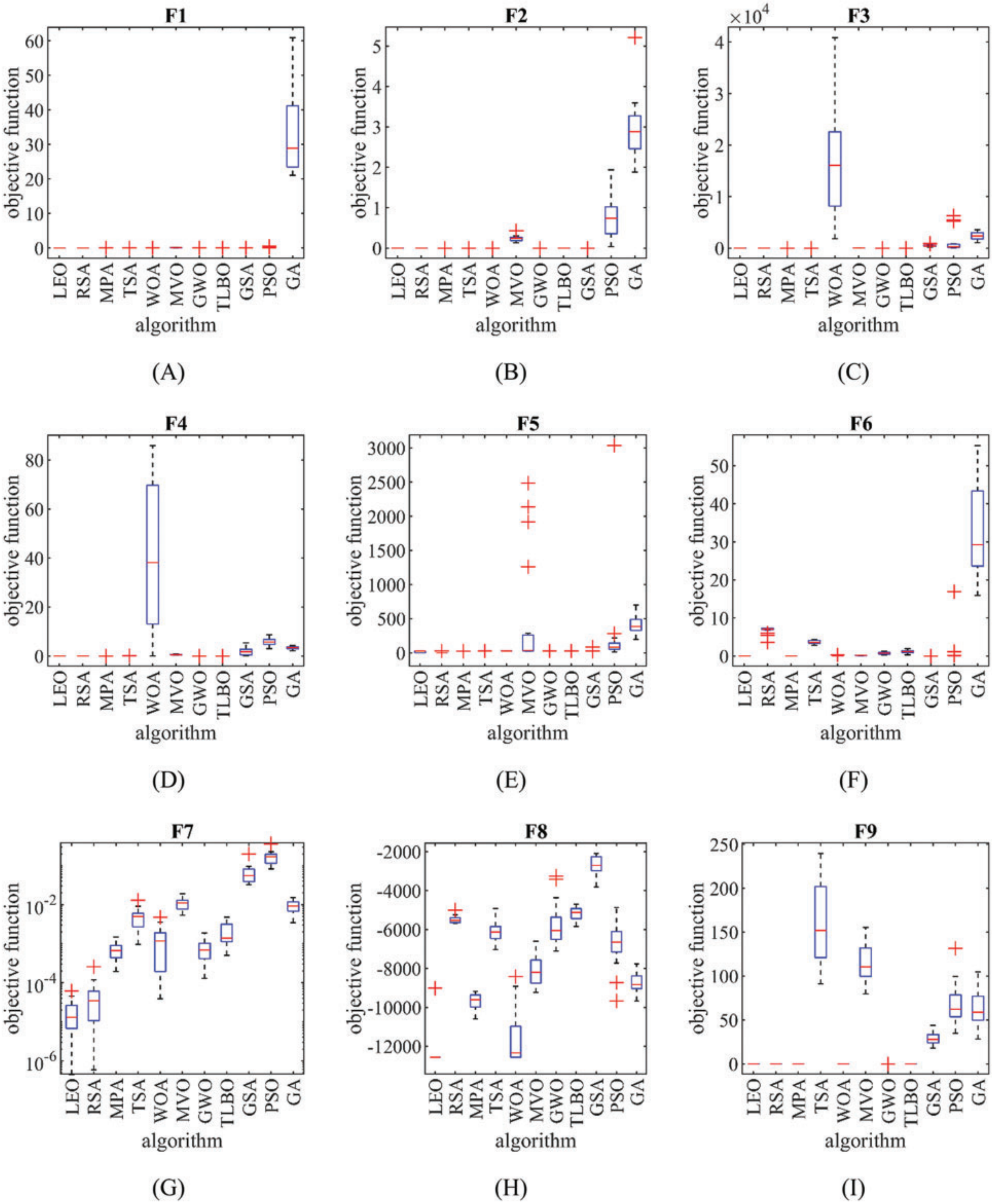

Figure 5: (Continued)

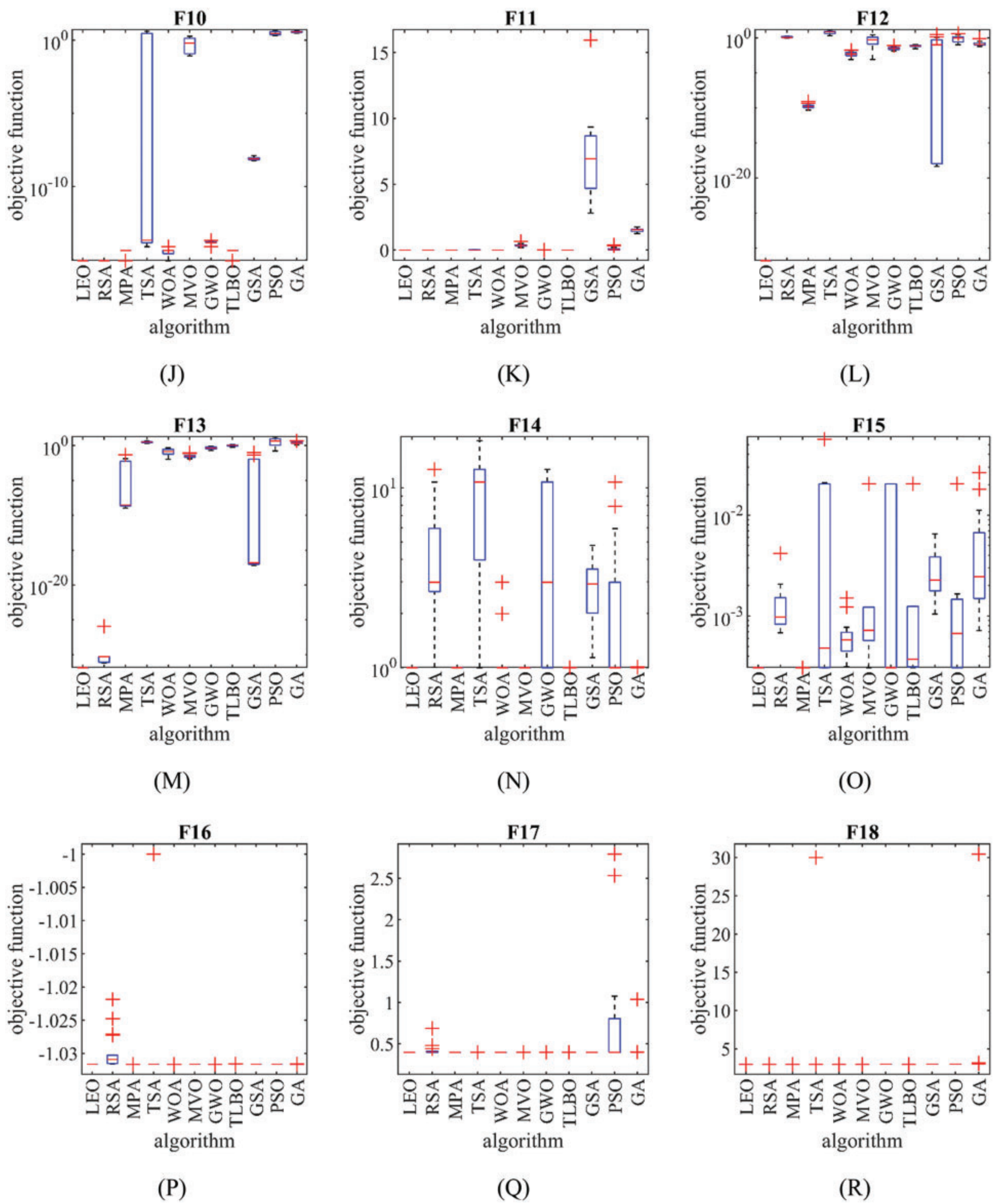

Figure 5: Boxplot diagrams of LEO and competitor algorithms performances on F1 to F23

4.4 Statistical Analysis

This subsection is devoted to statistical analysis to determine whether LEO has a statistically significant superiority over competitor algorithms. To this end, the non-parametric Wilcoxon rank sum test is utilized to determine this issue [58]. As usual, a "*p*-value" was used in this test to detect a significant difference between the mean of two data samples (specifically, *p* equal to 5% was chosen).

The results obtained from the statistical analysis using the Wilcoxon rank sum test of the performance of the LEO and competitor algorithms are presented in Table 8. What can be deduced from the results of statistical analysis is that in cases where the *p*-value is less than 0.05, LEO has a significant statistical superiority over the compared algorithm. Consequently, it is observed that LEO has a statistically significant advantage over all competitor algorithms.

Table 8: Results of Wilcoxon rank sum test

Compared algorithm	Objective function type		
	Unimodal	multimodal	Fixed-dimensional multimodal
LEO *vs.* RSA	0.054684	1.63E-11	1.44E-34
LEO *vs.* MPA	1.56E-13	1.15E-11	0.014404
LEO *vs.* TSA	1.01E-24	1.28E-19	1.44E-34
LEO *vs.* WOA	1.01E-24	1.26E-11	1.44E-34
LEO *vs.* MVO	1.01E-24	1.97E-21	1.44E-34
LEO *vs.* GWO	1.01E-24	1.66E-15	1.44E-34
LEO *vs.* TLBO	8.73E-24	1.04E-14	1.44E-34
LEO *vs.* GSA	6.63E-24	1.97E-21	6.79E-14
LEO *vs.* PSO	2.6E-23	1.97E-21	4.13E-17
LEO *vs.* GA	1.01E-24	2.48E-20	1.44E-34

4.5 Sensitivity Analysis

The LEO method performs the optimization process using a random search of its population members in the problem-solving space in an iteration-based procedure. As a result, the LEO population size (N) and the total number of iterations (T) are expected to influence the optimization process of the proposed approach. In this regard, this subsection is dedicated to the analysis of the sensitivity of the LEO to the parameters N and T.

In the first analysis, the sensitivity of the LEO to the parameter N is evaluated. For this purpose, LEO is used for different values of the parameter N equal to 20, 30, 50, and 100 to handle the benchmark functions F1 to F23. The results of this simulation are published in Table 9 and the LEO convergence curves under this study are plotted in Fig. 6. What can be deduced from the LEO sensitivity analysis to the parameter N is that increasing the set values for the parameter N increases the algorithm's search power to achieve better solutions, and as a result, the objective function values decrease.

Table 9: Results of LEO sensitivity analysis to the parameter N

Objective functions	Number of population members			
	20	30	50	100
F_1	0	0	0	0
F_2	0	0	0	0
F_3	0	0	0	0
F_4	0	0	0	0
F_5	18.81451	14.96979	14.11042	5.062263
F_6	0	0	0	0
F_7	1.83E-05	1.82E-05	1.02E-05	6.54E-06
F_8	−10200.7	−12036.5	−12391.8	−12569.5
F_9	0	0	0	0
F_{10}	8.88E-16	8.88E-16	8.88E-16	8.88E-16
F_{11}	0	0	0	0
F_{12}	1.57E-32	1.57E-32	1.57E-32	1.57E-32
F_{13}	0.002243	1.35E-32	1.35E-32	1.35E-32
F_{14}	1.14691	0.998	0.998	0.998004
F_{15}	0.000307	0.000307	0.000307	0.000307
F_{16}	−1.03163	−1.03163	−1.03163	−1.03163
F_{17}	0.3978	0.3978	0.3978	0.397887
F_{18}	4.35	3	3	3
F_{19}	−3.86278	−3.86278	−3.86278	−3.86278
F_{20}	−3.32141	−3.322	−3.322	−3.322
F_{21}	−10.1532	−10.1532	−10.1532	−10.1532
F_{22}	−10.1372	−10.4029	−10.4029	−10.4029
F_{23}	−10.5364	−10.5364	−10.5364	−10.5364

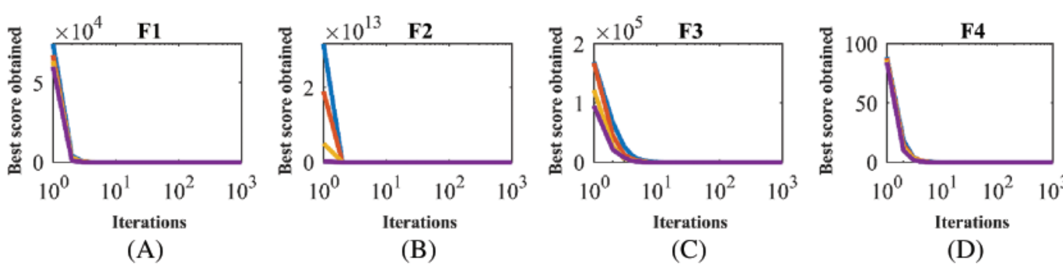

Figure 6: (Continued)

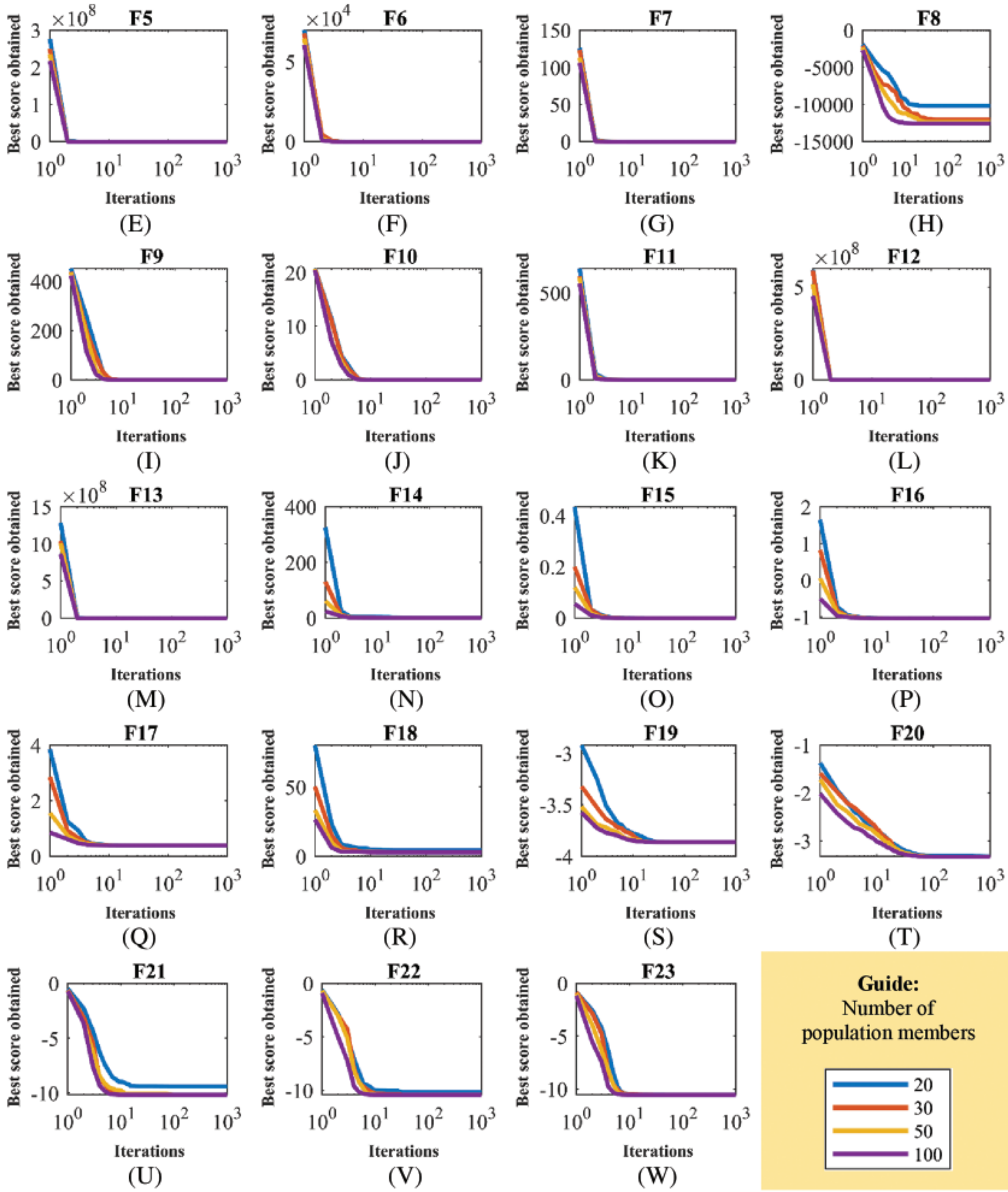

Figure 6: LEO convergence curves in the study of sensitivity analysis to the parameter N

In the analysis, the sensitivity of LEO to the T parameter is evaluated. For this purpose, LEO is utilized for different values of the T parameter equal to 200, 500, 800, and 1000 to tackle the F1 to F23 benchmark functions. The results of this simulation are released in Table 10 and the LEO convergence curves of this study are plotted in Fig. 7. Based on the simulation results obtained from the LEO sensitivity analysis to the parameter T, it is observed that by increasing the values of the parameter T, the proposed approach has converged to better results, and as a result the values of the objective function have decreased.

Table 10: Results of LEO sensitivity analysis to the parameter T

Objective functions	Maximum number of iterations			
	200	500	800	1000
F_1	2.1E-171	0	0	0
F_2	5.54E-85	5.2E-219	0	0
F_3	2.6E-108	5.8E-307	0	0
F_4	2.89E-85	4.4E-215	0	0
F_5	20.87501	18.88817	17.25557	14.96979
F_6	0	0	0	0
F_7	9.39E-05	3.92E-05	2.67E-05	1.82E-05
F_8	−10970.6	−11378.9	−11858.9	−12036.5
F_9	0	0	0	0
F_{10}	8.88E-16	8.88E-16	8.88E-16	8.88E-16
F_{11}	0	0	0	0
F_{12}	1.57E-32	1.57E-32	1.57E-32	1.57E-32
F_{13}	1.35E-32	1.35E-32	1.35E-32	1.35E-32
F_{14}	0.998	0.998	0.998	0.998
F_{15}	0.000308	0.000307	0.000307	0.000307
F_{16}	−1.03163	−1.03163	−1.03163	−1.03163
F_{17}	0.3978	0.3978	0.3978	0.3978
F_{18}	3	3	3	3
F_{19}	−3.86278	−3.86278	−3.86278	−3.86278
F_{20}	−3.32199	−3.322	−3.322	−3.322
F_{21}	−9.8983	−10.1532	−10.1532	−10.1532
F_{22}	−10.1372	−10.4029	−10.4029	−10.4029
F_{23}	−10.3518	−10.5364	−10.5364	−10.5364

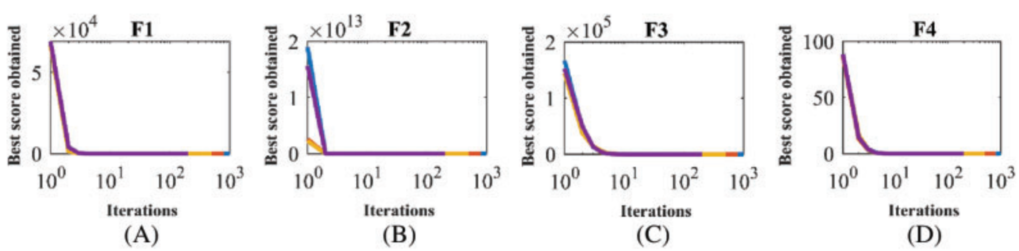

Figure 7: (Continued)

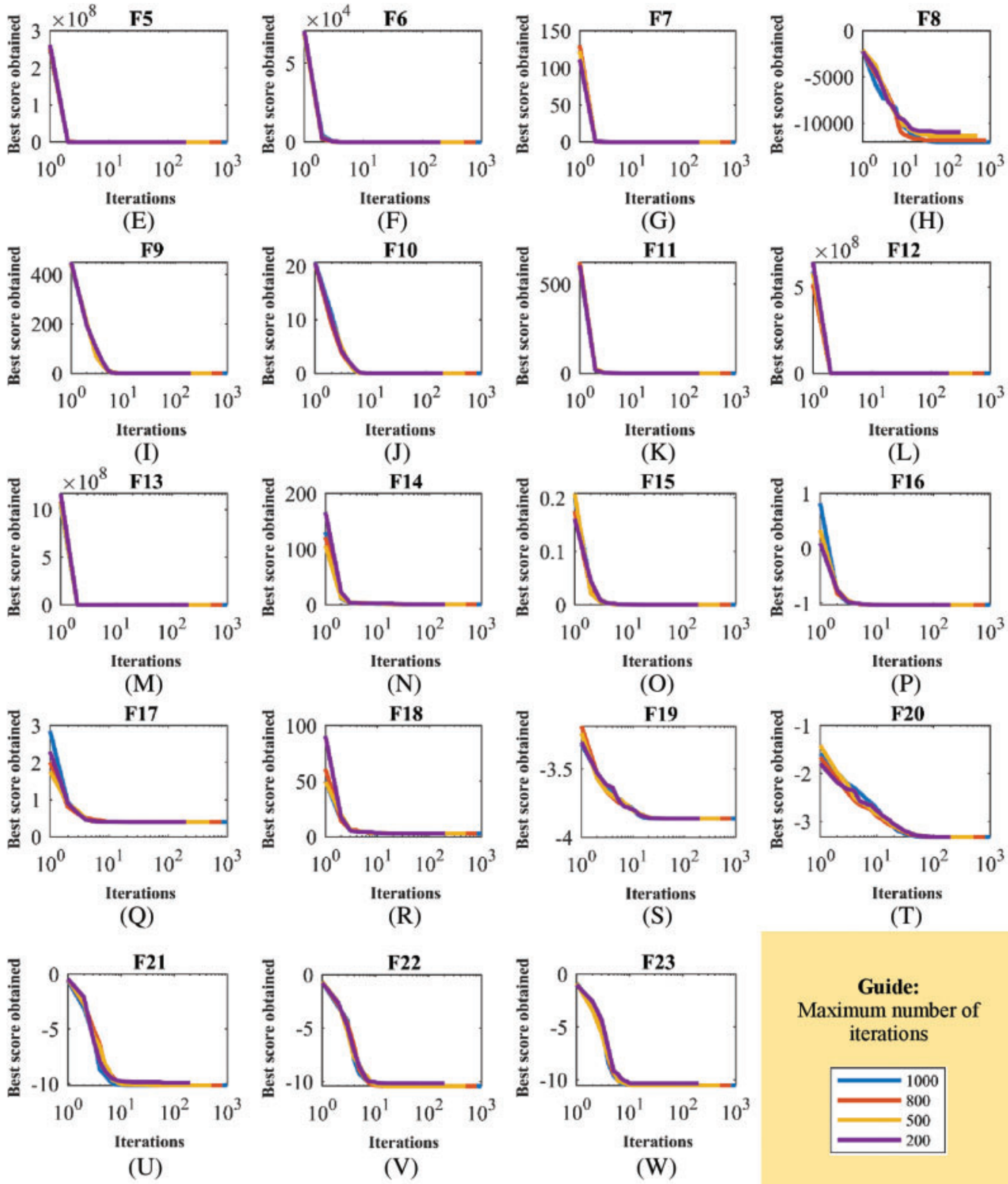

Figure 7: LEO convergence curves in the study of sensitivity analysis to the parameter T

5 Discussion

Metaheuristic algorithms provide an optimization process based on a random search in the problem-solving space. The optimization operation will be successful when, first, the problem-solving space is scanned well at the global level, and second, it is scanned around the solutions discovered at the local level.

Metaheuristic algorithms based on local search, which indicates the exploitation ability of an algorithm, scan around existing solutions to achieve a better possible solution. Exploitation gives this capability to the metaheuristic algorithm to be able to converge towards the global optimal. The exploitation power of a metaheuristic algorithm in local search is well measured in unimodal problems. These types of issues have only one optimal solution, and the goal of optimizing them is to get as close as possible to the global optimal based on the power of exploitation. The results obtained from LEO on the unimodal functions of F1 to F7 indicate the exploitation capability of the proposed method in converging towards global optimal. This ability is especially evident in handling the functions F1, F2, F3, F4, F5, and F6, as LEO is converged precisely to the global optimum. Therefore, the simulation finding of unimodal functions is the high exploitation capability of LEO in local search.

Metaheuristic algorithms based on global search, which indicates the exploration ability of the algorithm, scan different parts of the problem-solving space with the aim of identifying the main optimal area without getting caught up in local solutions. In fact, exploration gives this metaheuristic algorithm the ability to break out of local optimal solutions. The exploitation power of a metaheuristic algorithm in global search is well measured in multimodal problems. In addition to the main solution, these types of problem have several local solutions, and the purpose of optimizing them is to identify the area related to the main optimal solution based on the power of exploration. The results obtained from the use of LEO in the multimodal functions of F8 to F13 indicate the exploration ability of the proposed method in identifying the main optimal region and not getting caught in local solutions. This ability is especially evident in the handling of the functions F9 and F11, as LEO has been able to both discover the local optimal region well and converge precisely to the global optimal of these functions. Thus, the finding that simulates multimodal functions is the high exploration capability of LEO in the global search.

Although exploration and exploitation capabilities are crucial to the performance of metaheuristic algorithms, a more successful algorithm can balance these two capabilities during the optimization process. Creating this balance will lead to: first, the algorithm being able to discover the main optimal region based on exploration, and second, to converge towards the global optimal based on exploitation. The ability of a metaheuristic algorithm to strike a balance between exploration and exploitation is well measured in fixed-dimensional multimodal problems. The results obtained from the implementation of LEO on fixed-dimensional multimodal functions from F14 to F23 indicate the ability of the proposed method to strike a balance between exploration and exploitation. In addition, LEO showed the capability to explore the main optimal region and converge towards the global optimal. Therefore, the simulation finding of fixed-dimensional multimodal functions is a high capability of LEO in balancing exploration and exploitation.

6 Evaluation the CEC 2017 Test Suite

In this section, the performance of the proposed LEO approach in optimization tasks is evaluated on the CEC 2017 test suite. This set has thirty standard benchmark functions, including three unimodal functions C17-F1 to C17-F3, seven multimodal functions C17-F4 to C17-F10, ten hybrid functions C17-F11 to C17-F20, and ten composition functions C17-F21 to C17-F30. Full details of the CEC 2017 test suite are explained in [59].

The proposed LEO approach and competitor algorithms are employed in handling the CEC 2017 test suite. The simulation results are reported in Table 11. The resulting boxplots of the performance of the proposed LEO and competitor algorithms in the optimization of the CEC 2017 test suite are drawn in Fig. 8. The results show that LEO is the best optimizer for C17-F1, C17-F4 to C17-F6, C17-F8, C17-F10 to C17-F21, C17-F23 to C17-F25, and C17-F28 to C17-F30 functions. In optimizing C17-F2 and C17-F26, the proposed LEO approach is the second best optimizer for these functions after PSO. The proposed LEO approach is the second best optimizer after GSA for the functions C17-F7, C17-F9, and C17-F22. The analysis of the simulation results shows that the proposed LEO approach has provided better results in most of the benchmark functions and superior performance in the optimization of the CEC 2017 test suite compared to the competitor algorithms. Also, referring to the values obtained for the "*p*-value" index from the Wilcoxon rank sum statistical test shows that the superiority of LEO against competitor algorithms is significant from a statistical point of view.

Table 11: Optimization results for CEC 2017 test suite

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
C17-F1	mean	100.00002	1.32E+10	113895.2	2.872E+09	8690927.8	9430.0349	230476.22	76686529	245.4564	3328.3076	16493242
	best	100.00001	9.16E+09	175.65384	393204569	2039380.1	4426.4772	19273.455	53825370	100.19451	703.20384	10463278
	std	9.97E−06	3.87E+09	2.25E+05	2.17E+09	1.04E+07	4.38E+03	3.12E+05	2.98E+07	1.99E+02	2.67E+03	6.26E+06
	median	100.00001	1.269E+10	1824.5957	2.867E+09	4222785	9232.082	110843.31	66917665	176.41254	3489.0256	15334932
	ET	0.3172958	1.8727244	0.8779954	0.3733181	0.2978872	0.4491082	0.3701036	1.1978118	0.8354247	0.3481572	0.3980948
	rank	1	11	5	10	7	4	6	9	2	3	8
C17-F3	mean	300	10123.783	338.60644	12744.19	3540.6629	300.0554	4659.2923	763.2455	8808.339	300	24812.862
	best	300	6884.7583	300.16743	9428.5035	1237.6601	300.01055	2582.6579	586.05927	4051.73	300	18042.268
	std	1.54E−10	4.88E+03	4.40E+01	2.98E+03	2.60E+03	5.76E−02	2.71E+03	1.76E+02	4.56E+03	2.44E−12	1.02E+04
	median	300	8109.7558	329.75584	13075.684	2913.1907	300.03781	3796.1123	769.47312	8108.2321	300	20617.655
	ET	0.2993556	1.8736643	0.8212701	0.3644942	0.2867802	0.4102439	0.3556395	1.2062873	0.8082796	0.3183468	0.4028248
	rank	2	9	4	10	6	3	7	5	8	1	11
C17-F4	mean	400	1089.4244	403.84745	637.73782	423.53626	404.77441	417.73193	412.94333	406.64192	407.30478	416.29336
	best	400	681.4032	400.05471	408.20739	408.00923	403.92719	407.25852	409.57264	406.52228	401.28055	412.93487
	std	6.944E−09	5.66E+02	2.722759	3.19E+02	1.80E+01	9.76E−01	1.49E+01	5.02E+00	0.1756922	4.26E+00	5.30E+00
	median	400	876.77377	404.72508	526.19315	422.4949	404.64257	412.30938	410.93454	406.57131	408.30336	414.07128
	ET	0.3043906	1.8715125	0.828421	0.3301639	0.2831765	0.4272641	0.3519701	1.1745109	0.7738175	0.312928	0.3816822
	rank	1	11	2	10	9	3	8	6	4	5	7
C17-F5	mean	509.4521	570.97852	520.39673	555.36156	557.14932	516.91702	514.92238	539.19304	548.00649	539.22352	532.43573
	best	507.95967	560.71035	511.9395	525.77926	530.35745	510.94663	508.56508	531.29651	536.81332	523.57033	527.03431
	std	1.28E+00	1.31E+01	5.66E+00	3.13E+01	2.88E+01	5.14E+00	4.69E+00	7.62E+00	1.08E+01	2.19E+01	4.61E+00
	median	509.45211	566.65115	522.88405	552.43704	551.16319	516.91766	515.72158	537.9668	546.76286	530.84366	532.23219
	ET	0.3023736	1.8887621	0.8763099	0.3562381	0.291399	0.4501801	0.3750951	1.2508191	0.8026902	0.3304977	0.4101859
	rank	1	11	4	9	10	3	2	6	8	7	5
C17-F6	mean	600.00055	649.12886	600.43167	628.19803	632.02369	601.00316	601.3355	605.4803	624.93122	603.25733	607.67247
	best	600.00022	642.83648	600.03648	613.46525	616.78611	600.48046	600.11559	604.02203	613.54637	601.12871	604.66041
	std	3.03E−04	5.31E+00	6.69E−01	1.58E+01	1.49E+01	7.54E−01	2.28E+00	1.30E+00	1.05E+01	2.28E+00	3.34E+00
	median	600.00051	649.00457	600.12907	625.7324	631.1373	600.70446	600.23533	605.37232	623.64711	602.96419	607.60822
	ET	0.3619768	1.9709199	0.9639959	0.409375	0.3515911	0.5038948	0.4302542	1.4059159	0.8681172	0.4037424	0.4727251
	rank	1	11	2	9	10	3	4	6	8	5	7
C17-F7	mean	721.53846	806.79054	725.52924	792.43089	792.82487	727.82661	740.81091	758.78859	717.50492	746.01076	736.89383
	best	718.98051	796.93456	714.18747	768.65568	766.28166	723.10768	731.66864	755.46415	714.398	730.73613	728.78023
	std	1.97E+00	7.11E+00	1.19E+01	2.37E+01	2.14E+01	6.05E+00	7.39E+00	4.64E+00	3.12E+00	2.10E+01	5.42E+00
	median	721.69834	808.92042	722.87693	789.0756	796.38176	725.76904	741.21091	757.01329	716.90536	738.54308	739.42632
	ET	0.3230003	1.9364146	0.8928027	0.3756039	0.3177452	0.4635729	0.3910292	1.2818439	0.7851255	0.3475454	0.4267207
	rank	2	11	3	9	10	4	6	8	1	7	5

(Continued)

Table 11 (continued)

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
C17-F8	mean	808.20841	858.60323	809.70134	849.07577	832.91949	831.03479	816.08247	833.47787	819.15294	824.87392	823.20364
	best	804.9748	855.82927	806.9649	843.52729	814.19138	819.90228	812.69591	827.68976	815.91933	812.93446	816.71073
	std	2.21E+00	3.08E+00	2.62E+00	5.12E+00	1.58E+01	2.07E+01	3.69E+00	3.95E+00	2.97E+00	1.11E+01	9.28E+00
	median	808.95463	858.21387	809.95008	848.9468	832.36631	821.12256	815.29334	834.83538	818.90419	823.87898	819.66
	ET	0.3250002	1.8945481	0.8614258	0.354857	0.3171315	0.4475804	0.3828746	1.2504869	0.7580395	0.3373446	0.4180784
	rank	1	11	2	10	8	7	3	9	4	6	5
C17-F9	mean	900	1439.2536	926.16775	1598.9883	1562.1942	900.34594	900.78655	948.97968	900	958.54298	905.0812
	best	900	1139.663	901.0761	1009.0175	1043.5495	900.00299	900.01409	928.3364	900	901.81727	901.98992
	std	2.60E−08	2.98E+02	4.44E+01	7.15E+02	6.50E+02	4.36E−01	1.32E+00	2.70E+01	0.00E+00	5.12E+01	2.16E+00
	median	900	1383.0983	905.49128	1431.7658	1345.5542	900.23299	900.18809	939.43633	900	953.61704	905.78519
	ET	0.3562098	1.9355973	0.8955252	0.3693074	0.3140052	0.4698967	0.3918213	1.2679357	0.7862795	0.3677462	0.4245071
	rank	2	9	6	11	10	3	4	7	1	8	5
C17-F10	mean	1448.2414	2472.9544	1840.8555	2330.0521	2254.8178	1612.9939	1706.457	2155.1391	2665.9463	1977.6016	1778.7483
	best	1339.1179	2202.7083	1122.4927	1523.226	1886.5293	1495.6607	1610.8705	2065.3083	2227.3131	1837.465	1530.4446
	std	1.19E+02	2.73E+02	5.13E+02	5.53E+02	4.09E+02	1.31E+02	7.72E+01	8.45E+01	3.51E+02	2.07E+02	2.21E+02
	median	1421.8198	2455.6405	2010.586	2527.3279	2189.5993	1587.4155	1712.9007	2160.193	2699.8038	1897.4982	1783.6486
	ET	0.2921257	1.9344708	0.8899964	0.3861561	0.3123344	0.4670793	0.3881009	1.3143316	0.8104788	0.3601149	0.4339666
	rank	1	10	5	9	8	2	3	7	11	6	4
C17-F11	mean	1102.4081	3988.9348	1115.9881	1288.4482	1220.2127	1126.1077	1143.8912	1151.1927	1167.7559	1138.16	5155.6612
	best	1101.035	2128.0718	1112.5021	1151.0867	1124.5424	1113.657	1134.6187	1132.2017	1132.3212	1114.3392	1359.2219
	std	1.0913706	1728.279	4.0348047	141.03395	125.71804	1.46E+01	9.882597	1.74E+01	31.540223	16.776248	3985.9942
	median	1102.4457	4053.552	1114.9148	1258.449	1175.5186	1123.2158	1141.9369	1149.9879	1166.8796	1142.2867	4377.88
	ET	0.3082884	1.9053695	0.8705133	0.3753039	0.291397	0.4437908	0.3689143	1.2571838	0.8021757	0.3487779	0.426534
	rank	1	10	2	9	8	3	5	6	7	4	11
C17-F12	mean	1209.4469	67700546	3037.4471	89572425	9353082.3	517891.28	151943.54	2295970.5	895980.25	14707.321	1692251.5
	best	1200.0549	30532622	1669.0281	330069.7	57983.924	8014.0696	41866.656	493223.58	9831.5883	1562.5217	171741.04
	std	1.80E+01	3.33E+07	1.67E+03	1.76E+08	9.76E+06	6.17E+05	2.10E+05	1.42E+06	1.19E+06	1.00E+04	2.74E+06
	median	1200.6076	67910730	2527.3454	1923878.3	8475087.7	384471.94	49316.617	2526439.2	492469.14	16361.009	400976.52
	ET	0.3032189	1.9319159	0.8627085	0.3800094	0.2951645	0.4692286	0.372782	1.2642304	0.8014802	0.3495984	0.4363488
	rank	1	10	2	11	9	5	4	8	6	3	7
C17-F13	mean	1305.6389	40894980	1343.4927	16224.82	20438.622	13523.206	11757.028	7095.1454	12468.177	5580.4752	70502.24
	best	1301.7989	115399.57	1311.2063	7208.9788	8268.3414	2358.1337	7429.7441	3996.5404	7359.7815	2151.9689	12200.378
	std	3.86E+00	5.66E+07	2.21E+01	6.98E+03	1.10E+04	1.05E+04	5.04E+03	3.39E+03	3.79E+03	3.06E+03	7.31E+04
	median	1305.1608	20451941	1352.2554	17905.32	19461.528	12026.242	10471.052	6346.504	13268.961	5291.5555	50644.828
	ET	0.3159577	1.8878708	0.9128838	0.3918017	0.3109009	0.4621428	0.4100802	1.293707	0.7846826	0.3845192	0.4292407
	rank	1	11	2	8	9	7	5	4	6	3	10
C17-F14	mean	1402.2394	5470.9716	1427.7437	4765.1136	1584.9334	1435.7655	2833.0291	1521.9175	5567.3905	7002.6586	6763.8244
	best	1400.9954	2190.0306	1401.9899	2573.429	1496.5768	1429.0875	1476.9492	1475.783	2070.8126	3681.5587	1870.5221
	std	1.8833096	2.52E+03	22.172178	1.46E+03	8.71E+01	5.06E+00	1.58E+03	3.70E+01	3060.1102	3.26E+03	5.16E+03
	median	1401.4934	5804.6809	1427.4043	5441.0149	1577.9782	1436.7638	2528.55	1525.2724	5361.7578	6433.3565	6801.1672
	ET	0.3127391	1.9024018	0.888269	0.3900152	0.309189	0.4749437	0.4114049	1.3180878	0.7786291	0.3770875	0.4499625
	rank	1	8	2	7	5	3	6	4	9	11	10
C17-F15	mean	1500.432	9246.3665	1508.1166	14722.578	5770.572	1556.754	5410.7223	1793.1079	15392.044	4459.8968	4356.8363
	best	1500.3617	4958.6671	1502.3574	4159.502	2029.7502	1535.9023	1810.0088	1692.966	6548.9487	2271.1247	1868.5744
	std	6.52E−02	5.19E+03	6.89E+00	1.09E+04	6.24E+03	1.84E+01	2.49E+03	1.48E+02	6.13E+03	1.91E+03	2.81E+03
	median	1500.4366	7766.8841	1506.3314	15203.76	2981.3282	1557.1446	6299.4192	1733.9539	17366.648	4350.8524	4234.0085
	ET	0.294	1.9022906	0.8479029	0.361073	0.2894658	0.4514907	0.3667248	1.2349542	0.8035952	0.341312	0.4035415
	rank	1	9	2	10	8	3	7	4	11	6	5
C17-F16	mean	1601.417	2049.3493	1693.6086	2139.1819	1859.9833	1875.4528	1751.4474	1702.8066	2212.7605	1872.4306	1822.0213
	best	1601.0584	2022.0876	1602.8932	1993.8675	1659.0682	1722.138	1608.1802	1640.8401	2163.3077	1720.5276	1749.2084
	std	2.69E−01	2.01E+01	1.15E+02	1.70E+02	2.08E+02	1.26E+02	1.74E+02	1.03E+02	5.96E+01	1.18E+02	4.86E+01
	median	1601.4494	2053.9084	1664.7854	2097.6784	1847.8764	1876.5461	1697.9799	1656.6107	2194.891	1897.6183	1845.052
	ET	0.3153007	1.8969569	0.8690144	0.3843161	0.3049045	0.4727657	0.3788271	1.282693	0.8012009	0.3413454	0.4216952
	rank	1	9	2	10	6	8	4	3	11	7	5

(Continued)

Table 11 (continued)

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
C17-F17	mean	1718.215	1860.355	1740.9724	1863.2917	1865.2555	1798.9236	1770.9041	1758.3665	1966.631	1861.4076	1752.9279
	best	1706.6847	1817.3276	1732.6829	1802.7291	1824.6212	1734.4969	1743.3297	1757.5236	1759.2691	1772.4206	1749.6637
	std	7.71E+00	5.07E+01	8.21E+00	7.19E+01	4.07E+01	6.19E+01	2.84E+01	1.01E+00	1.60E+02	9.52E+01	2.46E+00
	median	1721.7195	1846.9176	1741.4033	1841.5942	1858.1224	1802.7053	1771.7776	1758.0739	1978.2256	1845.205	1753.2717
	ET	0.3658334	1.9520999	0.9804134	0.4225763	0.3617183	0.5147969	0.4358819	1.4915146	0.8211044	0.401581	0.4841767
	rank	1	7	2	9	10	6	5	4	11	8	3
C17-F18	mean	1800.7237	94816134	1833.5988	28939.221	24569.47	17408.62	23074.566	40341.67	11965.988	11312.675	9290.5265
	best	1800.0758	1566346.8	1810.6191	11253.046	3349.5232	4178.411	7853.7434	15409.717	8162.1711	3072.7269	4531.9869
	std	5.71E−01	1.83E+08	1.73E+01	1.22E+04	1.55E+04	1.63E+04	1.16E+04	1.77E+04	4.81E+03	6.63E+03	5.92E+03
	median	1800.7107	4220229.1	1837.6971	33389.233	28156.391	13637.271	24346.02	44840.006	10615.389	12384.862	7518.9237
	ET	0.3256107	1.8980687	0.8925264	0.3567493	0.3006547	0.4739268	0.3785245	1.280987	0.7779809	0.3552352	0.4123118
	rank	1	11	2	9	8	6	7	10	5	4	3
C17-F19	mean	1901.0291	679394.81	1909.263	73494.592	292988.36	2023.3414	6135.8892	2157.1287	28942.073	11413.638	20057.439
	best	1900.9488	161328.47	1902.5189	2005.9509	7901.4191	1925.8671	1933.5347	2043.9118	8744.8689	5501.8386	8219.752
	std	6.50E−02	8.04E+05	5.99E+00	1.35E+05	5.51E+05	1.68E+02	5.04E+03	1.43E+02	1.99E+04	6.29E+03	1.13E+04
	median	1901.0328	341107.87	1908.7373	8081.1832	22321.465	1946.7148	5287.4278	2113.0137	26612.935	10074.195	21059.965
	ET	0.6015283	2.1930547	1.4988764	0.6585444	0.6057556	0.7472846	0.682842	2.1939116	1.0763017	0.6634356	0.7332974
	rank	1	11	2	9	10	3	5	4	8	6	7
C17-F20	mean	2010.0534	2301.7385	2024.9164	2213.9835	2256.847	2155.3699	2054.9244	2084.6507	2329.6939	2235.7275	2052.2351
	best	2001.9904	2248.0003	2020.3081	2089.0765	2066.7782	2026.6169	2030.6709	2064.3507	2201.9677	2196.8025	2036.0513
	std	1.02E+01	5.15E+01	8.87E+00	1.49E+02	1.29E+02	9.73E+01	2.42E+01	2.56E+01	9.38E+01	3.50E+01	1.84E+01
	median	2006.6239	2298.0865	2020.5675	2170.523	2303.7862	2165.9516	2055.9022	2076.3213	2347.968	2240.2324	2047.3626
	ET	0.3549563	1.9792104	1.0031495	0.4189973	0.3629973	0.5175917	0.4502586	1.4757591	0.8379548	0.4152999	0.4851776
	rank	1	10	2	7	9	6	4	5	11	8	3
C17-F21	mean	2200	2308.2706	2290.0245	2348.6744	2337.4394	2297.0061	2292.0723	2304.9949	2361.3391	2322.6768	2308.7322
	best	2200	2245.3264	2209.0492	2336.2904	2318.0377	2201.9281	2201.4314	2205.0065	2357.5108	2309.5763	2223.4109
	std	1.221E−05	59.029993	54.252421	18.961418	14.666468	6.36E+01	60.473011	6.68E+01	5.8686381	11.367088	57.267239
	median	2200	2301.2791	2313.7008	2340.7329	2341.0693	2325.2005	2321.3294	2335.6805	2358.9014	2321.9666	2334.4727
	ET	0.3670121	1.9871627	0.9961948	0.431126	0.3589478	0.5171054	0.4407936	1.4386133	0.8538204	0.4021599	0.4880179
	rank	1	6	2	10	9	4	3	5	11	8	7
C17-F22	mean	2300.4086	2960.4831	2306.5754	2388.4242	2318.9205	2303.6665	2305.9506	2323.9978	2300.2574	2313.3484	2321.8836
	best	2300	2756.6595	2302.8892	2310.0835	2313.2141	2302.3375	2300.6213	2315.17	2300	2300.6448	2315.3845
	std	4.77E−01	2.11E+02	3.48E+00	8.99E+01	7.76E+00	9.58E−01	6.58E+00	7.23E+00	1.77E−01	2.34E+01	5.73E+00
	median	2300.3665	2917.2594	2306.5182	2386.4565	2316.1003	2303.905	2303.8566	2325.2645	2300.3161	2302.1961	2322.693
	ET	0.3999132	2.0271331	1.0408899	0.4446991	0.3889436	0.5590779	0.4744016	1.5425062	0.8605875	0.4368888	0.5218687
	rank	2	11	5	10	7	3	4	9	1	6	8
C17-F23	mean	2608.667	2697.6138	2648.389	2690.7613	2658.5666	2613.7637	2623.5065	2637.0749	2733.7617	2642.8041	2663.4253
	best	2606.5232	2673.1877	2624.4481	2672.5407	2613.963	2607.0408	2608.1065	2623.829	2724.4199	2612.6221	2652.9578
	std	2.05E+00	2.01E+01	2.07E+01	1.78E+01	3.38E+01	6.12E+00	1.60E+01	1.09E+01	9.84E+00	2.02E+01	1.44E+01
	median	2608.5333	2697.5253	2648.1834	2687.7457	2663.6208	2614.1411	2624.2502	2636.9896	2732.5358	2651.4089	2658.0759
	ET	0.4175118	1.9807793	1.0808874	0.4609724	0.431678	0.5594673	0.4829631	1.5702774	0.8909863	0.4357085	0.5259606
	rank	1	10	6	9	7	2	3	4	11	5	8
C17-F24	mean	2500.0002	2889.4365	2641.6992	2821.9182	2795.2084	2751.4754	2750.8673	2765.2163	2740.8134	2778.2175	2773.0636
	best	2500.0001	2839.3557	2500.5995	2796.5576	2751.5338	2746.0029	2737.1449	2758.0342	2500	2773.298	2765.1812
	std	4.785E−05	3.70E+01	162.78876	2.24E+01	2.95E+01	3.95E+00	2.02E+01	5.82E+00	162.61032	3.85E+00	1.03E+01
	median	2500.0002	2895.5602	2640.796	2822.5121	2806.3359	2752.3181	2743.0222	2765.5625	2802.961	2778.592	2769.4377
	ET	0.4417598	1.989241	1.098033	0.4703164	0.4488538	0.5772698	0.5009254	1.5867588	0.8846244	0.445734	0.5386319
	rank	1	11	2	10	9	5	4	6	3	8	7
C17-F25	mean	2897.7429	3383.1819	2934.6471	3128.9324	2954.6209	2898.191	2941.7004	2930.7864	2932.5136	2934.9204	2954.9786
	best	2897.7429	3352.0262	2899.7425	2943.9861	2948.9808	2897.8394	2918.7097	2909.7743	2899.585	2899.61	2952.1894
	std	3.04E−07	4.90E+01	2.33E+01	2.54E+02	5.12E+00	2.46E−01	1.54E+01	1.60E+01	2.20E+01	2.36E+01	2.69E+00
	median	2897.7429	3362.6645	2945.8363	3033.9803	2954.1309	2898.2616	2948.6451	2932.2998	2943.4085	2945.6644	2954.6376
	ET	0.418846	1.9986182	1.0641379	0.4549042	0.4141905	0.5427762	0.465997	1.5381446	0.8526837	0.4317387	0.5142647
	rank	1	11	5	10	8	2	7	3	4	6	9

(Continued)

Table 11 (continued)

		LEO	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
C17-F26	mean	2875.0004	4236.0105	3263.5095	3883.8521	3260.1768	2900.1421	2956.2263	3287.936	4125.3708	2851.8668	3024.4918
	best	2800.0013	3815.9884	2900.0298	2911.5703	2833.7931	2900.1055	2900.235	2989.7582	3570.8634	2600	2907.1933
	std	5.00E+01	4.53E+02	4.72E+02	7.64E+02	4.96E+02	3.94E−02	3.77E+01	5.87E+02	3.81E+02	1.94E+02	9.89E+01
	median	2900	4168.6271	3098.7525	3925.0346	3116.6752	2900.133	2971.5624	2997.1261	4250.1054	2894.9058	3028.4579
	ET	0.4662288	2.0127518	1.1348664	0.5017804	0.4605082	0.5896628	0.5151757	1.6685732	0.9167256	0.4923115	0.5441807
	rank	2	11	7	9	6	3	4	8	10	1	5
C17-F27	mean	3089.278	3174.6119	3111.5464	3166.2836	3173.718	3093.1151	3092.4668	3109.3677	3295.5918	3117.8607	3150.7429
	best	3088.978	3139.1529	3098.0647	3137.1017	3127.7165	3089.6471	3089.0864	3093.6744	3218.4998	3098.4758	3133.3309
	std	2.23E−01	4.12E+01	1.59E+01	3.22E+01	3.36E+01	3.08E+00	2.63E+00	2.96E+01	7.80E+01	1.89E+01	1.76E+01
	median	3089.3081	3165.578	3106.8009	3166.5004	3180.6311	3093.1165	3092.9317	3095.0128	3291.7639	3114.8985	3147.2637
	ET	0.432171	2.0431644	1.1940608	0.5065569	0.4503224	0.6019193	0.5331149	1.7209385	0.918766	0.4850874	0.5712062
	rank	1	10	5	8	9	3	2	4	11	6	7
C17-F28	mean	3025.0006	3912.0323	3307.722	3471.8196	3307.8554	3332.3379	3244.7641	3439.0869	3471.5812	3319.6545	3195.8925
	best	2800.0022	3867.8031	3100.0812	3402.2513	3187.9788	3150.0575	3177.6079	3227.0805	3421.3532	3182.82	3168.5063
	std	1.50E+02	3.69E+01	1.51E+02	9.02E+01	1.21E+02	1.22E+02	1.04E+02	2.12E+02	4.06E+01	1.03E+02	2.18E+01
	median	3100.0001	3914.5034	3343.1136	3440.3562	3315.6409	3383.7349	3201.7703	3398.7271	3472.0867	3341.8726	3200.6847
	ET	0.4020287	2.0094507	1.1278612	0.4821352	0.4198695	0.5727171	0.502574	1.6352087	0.8968212	0.4629656	0.5554748
	rank	1	11	4	10	5	7	3	8	9	6	2
C17-F29	mean	3144.724	3339.5801	3165.6945	3314.6405	3324.3104	3231.0752	3187.9911	3232.8143	3520.4547	3287.4708	3278.031
	best	3134.1607	3207.6108	3148.2017	3288.9693	3264.6542	3177.2234	3173.9741	3177.7522	3333.3629	3208.0319	3224.1938
	std	8.54E+00	1.15E+02	1.65E+01	2.43E+01	5.93E+01	5.19E+01	1.16E+01	6.03E+01	1.61E+02	5.95E+01	5.08E+01
	median	3145.445	3346.043	3163.2563	3313.3322	3317.8657	3223.867	3188.4674	3217.3081	3523.3351	3302.4084	3272.3932
	ET	0.4219881	2.0567265	1.1564944	0.4921447	0.4352592	0.589355	0.5099219	1.6727754	0.8960592	0.4754221	0.5467503
	rank	1	10	2	8	9	4	3	5	11	7	6
C17-F30	mean	3407.9782	5675648.8	5604.9984	4719850.6	3203269.4	378962.03	836407.72	31117.588	1814995.5	631248.38	2253452.4
	best	3395.1259	984371.01	3643.8625	2489911.1	28614.714	14708.39	8096.1658	21195.57	306189.65	3865.2596	228474.19
	std	1.80E+01	8.54E+06	3.63E+03	2.74E+06	2.89E+06	7.25E+05	9.54E+05	1.07E+04	2.52E+06	8.70E+05	2.00E+06
	median	3401.7333	1625841.3	3868.5572	3954248.4	3366679.2	17686.312	819283.54	29368.199	692721.36	333588.26	2288979.6
	ET	0.6317577	2.271566	1.6151179	0.7461312	0.6762314	0.81334	0.7527132	2.3858616	1.1454059	0.7137589	0.7972583
	rank	1	11	2	10	9	4	6	3	7	5	8
Sum rank		34	292	93	270	238	119	134	170	210	166	188
Mean rank		1.1724138	10.068966	3.2068966	9.3103448	8.2068966	4.1034483	4.6206897	5.862069	7.2413793	5.7241379	6.4827586
Total rank		1	11	2	10	9	3	4	6	8	5	7
p-value			1.972E−21	1.289E−19	1.972E−21	1.972E−21	3.406E−20	3.881E−21	1.972E−21	1.803E−20	7.408E−20	1.972E−21

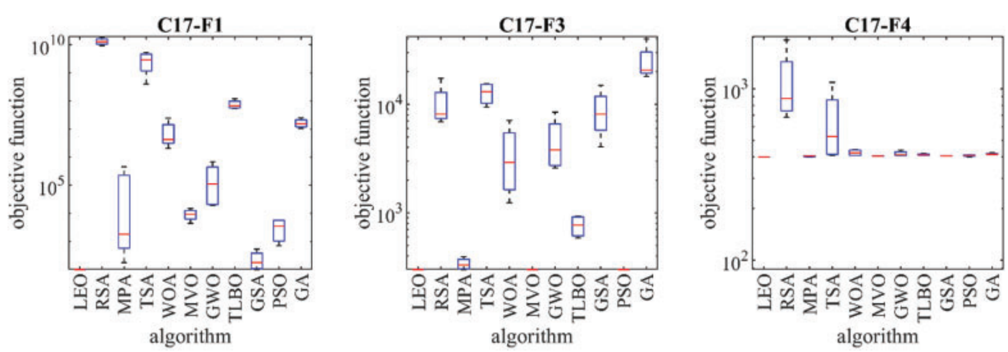

Figure 8: (Continued)

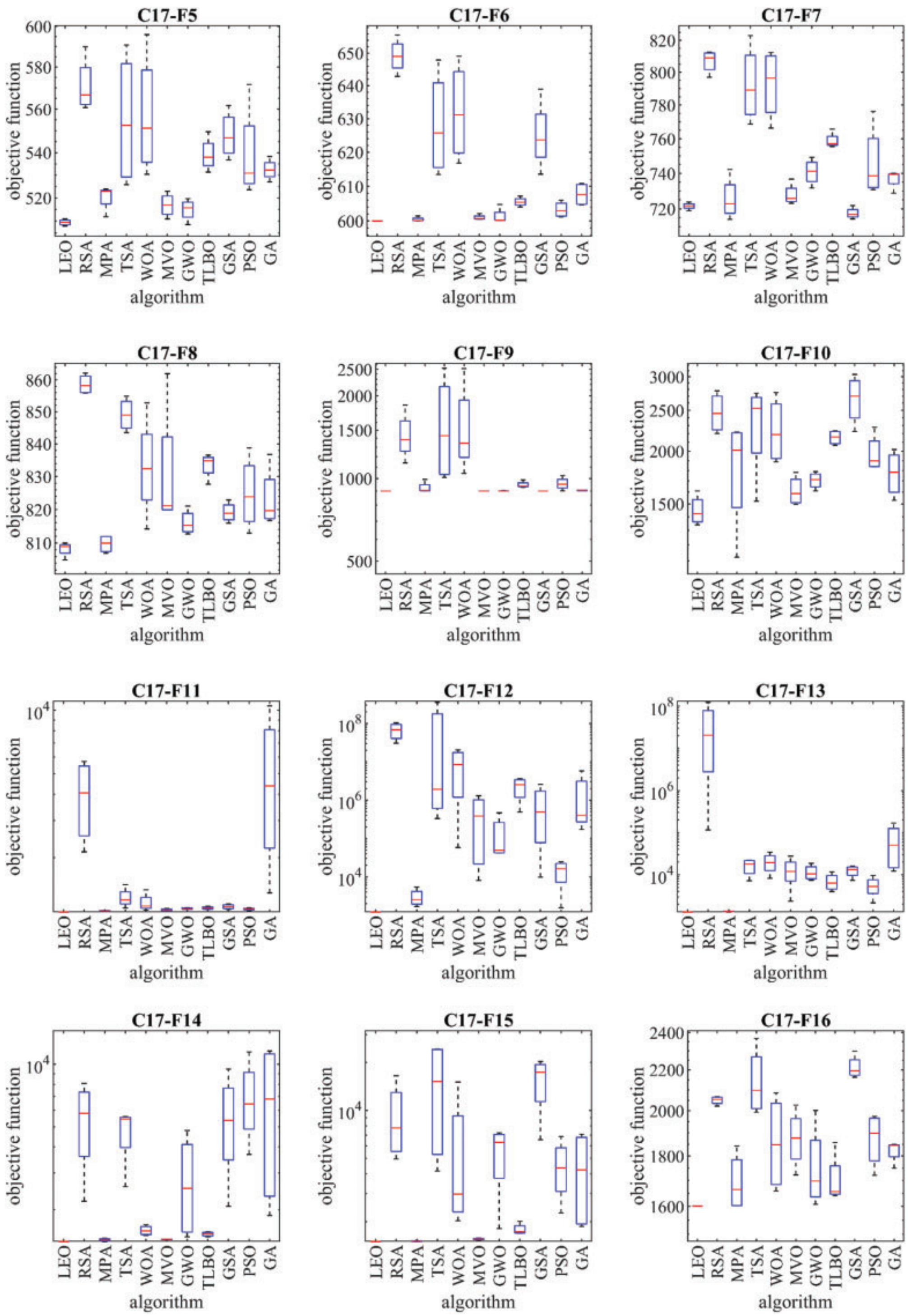

Figure 8: (Continued)

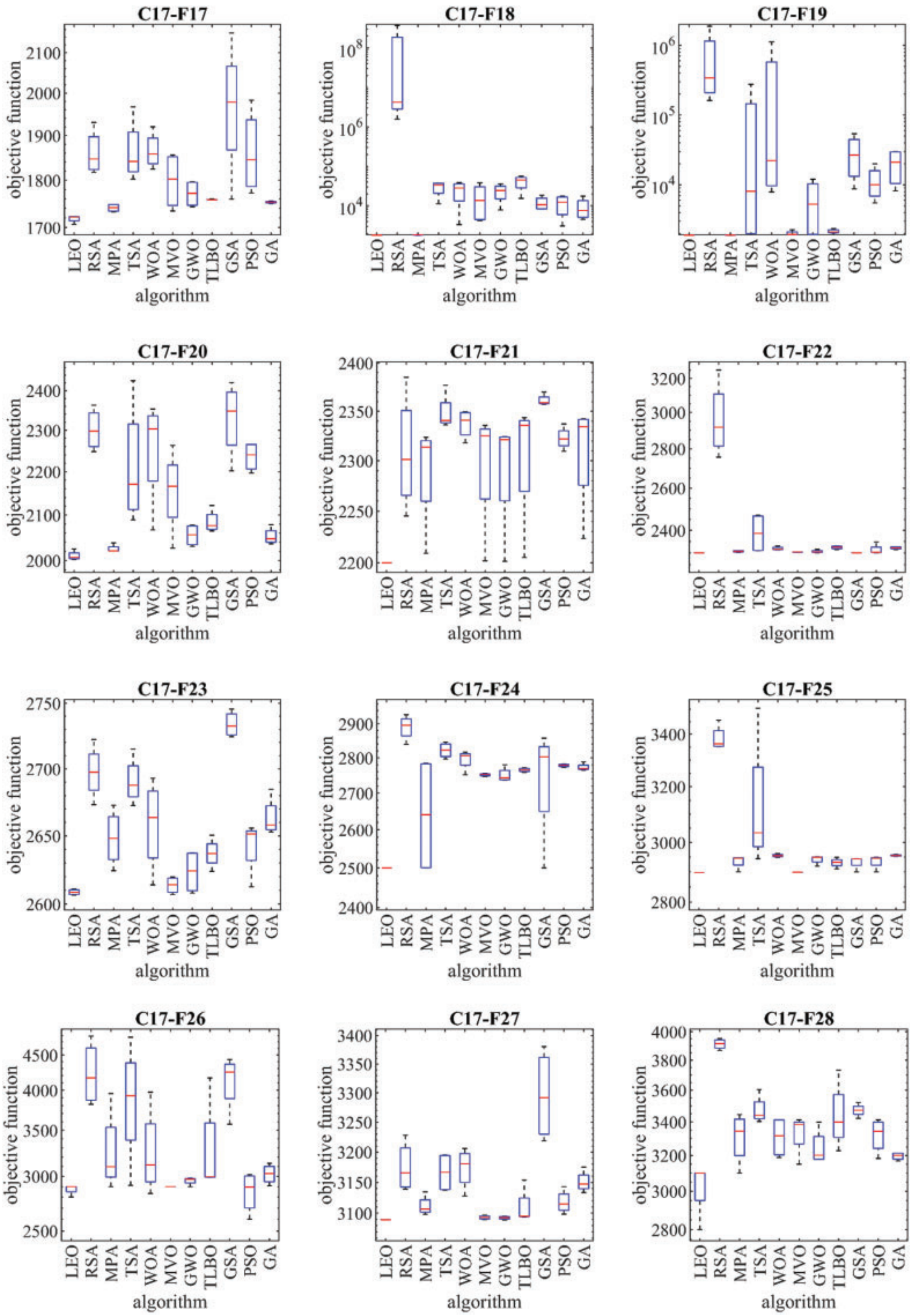

Figure 8: (Continued)

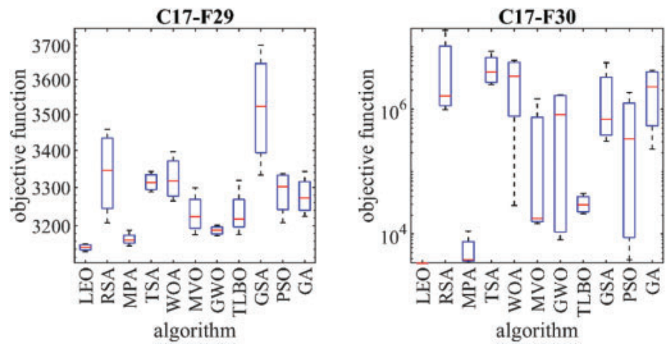

Figure 8: Boxplot diagrams of LEO and competitor algorithms performances on the CEC 2017 test suite

7 LEO for Real-World Applications

In this section, LEO's ability to solve real-world optimization applications is evaluated.

7.1 Tension/Compression Spring Design Optimization Problem

The tension/compression spring problem is a design challenge in real-world applications, the goal of which is to minimize the weight of the tension/compression spring. The schematic of this design is shown in Fig. 9. The mathematical model of tension/compression spring design is as follows [19]:

Consider: $X = [x_1,\ x_2,\ x_3\ ] = [d,\ D,\ P]$.

Minimize: $f\ \ (x) = (x_3 + 2)\, x_2 x_1^2$.

Subject to:

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4}\ \le\ 0,\ g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3\right)} + \frac{1}{5108 x_1^2} - 1 \le\ 0,$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le\ 0, g_4(x) = \frac{x_1 + x_2}{1.5} - 1\ \le\ 0.$$

with

$0.05 \le x_1 \le 2,\ 0.25 \le x_2 \le 1.3$ and $2 \le\ x_3 \le 15$.

Figure 9: Schematic view of the tension/compression spring problem

The results of the implementation of LEO and competitor algorithms in the optimization of tension/compression spring design are reported in Tables 12 and 13. The simulation results show that LEO has provided the optimal solution to this problem with the values of design variables equal to (0.051689, 0.356718, 11.28897) and the value of the objective function equal to 0.012665. The analysis of the results shows that LEO has provided better results compared to competitor algorithms in optimizing tension/compression spring design. The LEO convergence curve during tension/compression spring design optimization is presented in Fig. 10.

Table 12: Comparison results for the tension/compression spring design problem

Algorithm	Optimum variables			Optimum cost
	d	*D*	*P*	
LEO	0.051689	0.356718	11.28897	0.012665
RSA	0.05	0.310539	15	0.013198
MPA	0.051684	0.356585	11.29675	0.012665
TSA	0.051905	0.361628	11.0584	0.012723
WOA	0.051486	0.351849	11.58021	0.012666
MVO	0.05	0.313567	14.55586	0.012978
GWO	0.05113	0.343162	12.14438	0.012689
TLBO	0.069268	0.939835	2	0.018038
GSA	0.0576	0.505834	6.273568	0.013885
PSO	0.068994	0.933432	2	0.017773
GA	0.069326	0.939615	2	0.018064

Table 13: Statistical results for the tension/compression spring design problem

Algorithm	mean	best	std	median	ET	rank
LEO	0.012665	0.012665	1.38E−18	0.012665	14.54877	1
RSA	0.018891	0.013198	0.009397	0.01332	1.054796	8
MPA	0.012666	0.012665	6.11E−07	0.012665	3.067909	2
TSA	0.013023	0.012723	0.000339	0.012881	2.232119	4
WOA	0.014042	0.012666	0.001447	0.013513	3.335292	5
MVO	0.016781	0.012978	0.001966	0.017902	1.136586	6
GWO	0.01274	0.012689	6.52E−05	0.012723	3.502764	3
TLBO	0.018521	0.018038	0.000405	0.018374	18.83583	7
GSA	0.019435	0.013885	0.003219	0.019385	4.320089	9
PSO	4.41E + 13	0.017773	1.22E + 14	0.017773	3.289217	11
GA	5.43E + 12	0.018064	1.42E + 13	0.02413	5.853759	10

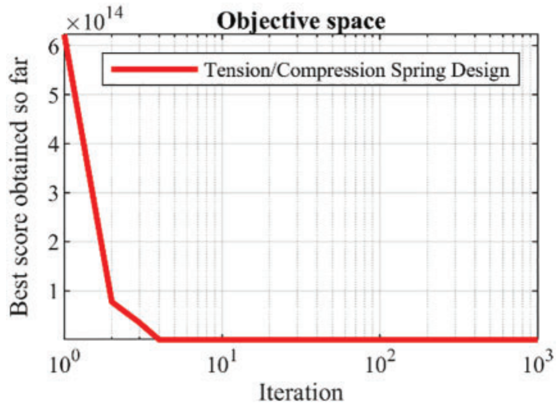

Figure 10: Convergence analysis of the LEO for the tension/compression spring design optimization problem

7.2 Welded Beam Design Optimization Problem

The welded beam design problem is an engineering issue in real-world applications to minimize the fabrication cost of the welded beam. The schematic of this design is shown in Fig. 11. The mathematical model of welded beam design is as follows [19]:

Consider: $X = [x_1,\ x_2\ , x_3\ , x_4] = [h,\ l,\ t,\ b]$.

Minimize: $f\ (x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4\ \ (14.0 + x_2)$

Subject to:

$$g_1\ (x) = \tau\ (x) - 13600\ \leq\ 0,\ g_2\ (x) = \sigma\ (x) - 30000\ \leq\ 0,$$

$$g_3\ \ (x) = x_1 - x_4 \leq\ 0,$$

$$g_4\ (x) = 0.10471 x_1^2 + 0.04811 x_3 x_4\ \ (14 + x_2) - 5.0\ \leq\ 0,$$

$$g_5\ (x) = 0.125\ -\ x_1 \leq\ 0,\ g_6\ (x) = \delta\ \ (x)\ -\ 0.25\ \leq\ 0,$$

$$g_7\ (x) = 6000\ -\ p_c\ \ (x)\ \leq\ 0,$$

where

$$\tau\ (x) = \sqrt{\tau' + (2\tau\tau')\frac{x_2}{2R} + (\tau'')^2},\ \tau' = \frac{6000}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},$$

$$M = 6000\left(14 + \frac{x_2}{2}\right),\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2\sqrt{2}\ x_1x_2\left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right),\ \sigma\ (x) = \frac{504000}{x_4x_3^2},\ \delta\ (x) = \frac{65856000}{(30\cdot 10^6)\ x_4x_3^3},$$

$$p_c\ (x) = \frac{4.013\ (30\cdot 10^6)\ x_3x_4^3}{6\cdot 196}\left(1 - \frac{x_3}{28}\sqrt{\frac{30\cdot 10^6}{4\ (12\cdot 10^6)}}\right),$$

with

$0.1 \le x_1,\ x_4 \le 2,\ 0.1 \le x_2,\ x_3 \le 10.$

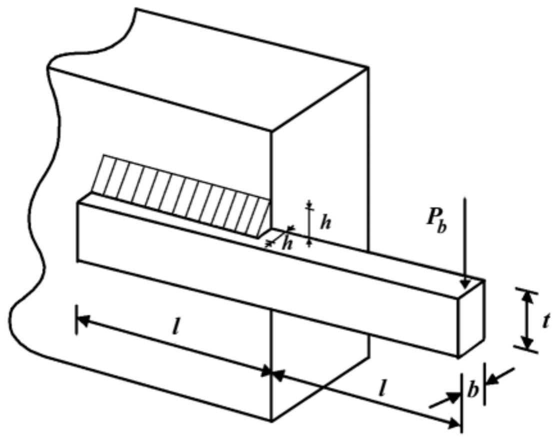

Figure 11: Schematic view of the welded beam design problem

The results of welding beam design optimization using LEO and competitor algorithms are reported in Tables 14 and 15. The simulation results show that LEO has provided the optimal solution to this problem with the values of design variables equal to (0.20573, 3.470489, 9.036624, 0.20573) and the value of the objective function equal to (1.724852). Based on the optimization results, the proposed LEO approach has provided superior performance in handling the welded beam design problem compared to competitor algorithms. The convergence curve of LEO while achieving the solution for welded beam design is plotted in Fig. 12.

Table 14: Comparison results for the welded beam design problem

Algorithm	Optimum variables				Optimum cost
	h	*l*	*t*	*b*	
LEO	0.20573	3.470489	9.036624	0.20573	1.724852
RSA	0.236392	3.235121	8.568725	0.237878	1.889845
MPA	0.20573	3.470489	9.036624	0.20573	1.724852
TSA	0.205581	3.499014	9.022691	0.20658	1.73255
WOA	0.161832	4.691246	9.365763	0.204143	1.855028
MVO	0.204638	3.487916	9.059375	0.205619	1.728592
GWO	0.205665	3.472964	9.03602	0.205822	1.725685
TLBO	0.248204	3.662307	8.684848	0.365402	2.945841
GSA	0.250196	3.715003	7.942199	0.277498	2.135258
PSO	0.330894	4.892002	7.839737	0.468597	3.930698
GA	0.329815	3.245087	6.377571	0.417296	2.597963

Table 15: Statistical results for the welded beam design problem

Algorithm	mean	best	std	median	ET	rank
LEO	1.724852	1.724852	6.83E−16	1.724852	3.363431	1
RSA	2.249565	1.889845	0.242249	2.209624	1.427522	6
MPA	1.724853	1.724852	7.34E−07	1.724852	3.909917	2
TSA	1.745368	1.73255	0.006329	1.747237	3.143402	5
WOA	2.582368	1.855028	0.761705	2.281472	4.759158	8
MVO	1.7386	1.728592	0.008201	1.737193	435.4079	4
GWO	1.727195	1.725685	0.001082	1.726991	4.935636	3
TLBO	9.02E+12	2.945841	2.67E+13	5.109645	25.56709	9
GSA	2.455682	2.135258	0.249943	2.422385	5.715263	7
PSO	1.58E+14	3.930698	2.79E+14	5.54E+13	4.841792	11
GA	5.16E+13	2.597963	1.82E+14	4.954044	8.077649	10

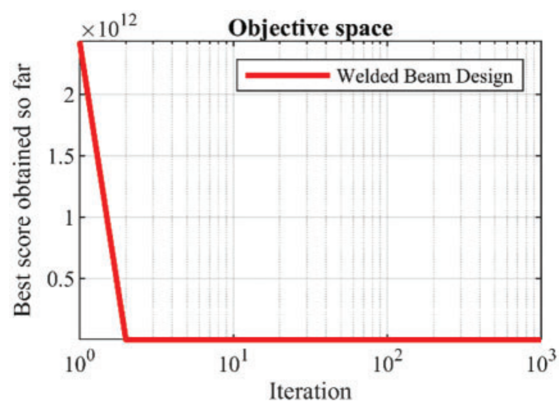

Figure 12: Convergence analysis of the proposed LEO for the welded beam design optimization problem

7.3 Speed Reducer Design Optimization Problem

The speed reducer design problem is a real-world application in engineering studies with the aim of minimizing the weight of the speed reducer. The schematic of this design is shown in Fig. 13. The mathematical model of speed reducer design is as follows [60,61]:

Consider: $X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, p, l_1, l_2, d_1, d_2]$.

Minimize: $$f(x) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right).$$

Subject to:

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,\ g_2(x) = \frac{397.5}{x_1x_2^2x_3} - 1 \le 0,$$

$$g_3(x) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0, g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0,$$

$$g_5(x) = \frac{1}{110x_6^3}\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \cdot 10^6} - 1 \le 0,$$

$$g_6(x) = \frac{1}{85x_7^3}\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \cdot 10^6} - 1 \le 0,$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0, g_8(x) = \frac{5x_2}{x_1} - 1 \le 0,$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \le 0, \; g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0,$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0,$$

with

$$2.6 \le x_1 \le 3.6, \; 0.7 \le x_2 \le 0.8, \; 17 \le x_3 \le 28, \; 7.3 \le x_4 \le 8.3,$$

$$7.8 \le x_5 \le 8.3, \; 2.9 \le x_6 \le 3.9, \; 5.0 \le x_7 \le 5.5.$$

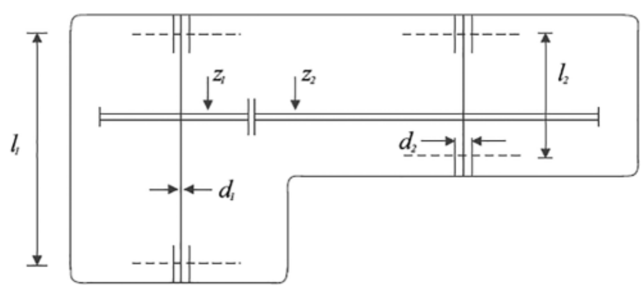

Figure 13: Schematic view of speed reducer design problem

The results of using LEO and competitor algorithms in optimizing of speed reducer design are released in Tables 16 and 17. The simulation results show that LEO has provided the optimal solution to this problem with the values of design variables equal to (3.5, 0.7, 17, 7.3, 7.8, 3.350215, 5.286683) and the value of the objective function equal to 2996.348. What is evident from the analysis of simulation results is that the proposed LEO approach has provided better results in speed reducer design and superior performance compared to competitor algorithms. The LEO convergence curve during speed reducer design optimization is shown in Fig. 14.

Table 16: Comparison results for the speed reducer design problem

Algorithm	Optimum variables							Optimum cost
	b	m	p	l_1	l_2	d_1	d_2	
LEO	3.5	0.7	17	7.3	7.8	3.350215	5.286683	2996.348
RSA	3.54876	0.700327	17	7.3	7.8	3.354293	5.287113	3018.311
MPA	3.5	0.7	17	7.3	7.8	3.350215	5.286683	2996.348
TSA	3.517395	0.7	17	7.3	7.8	3.367129	5.287918	3008.294
WOA	3.533787	0.7	17	7.486762	7.88665	3.350644	5.286713	3013.296
MVO	3.502135	0.7	17	7.541675	8.049195	3.358798	5.28743	3007.468
GWO	3.501526	0.7	17.0011	7.387738	7.804499	3.350782	5.28704	2998.382
TLBO	3.592592	0.713244	19.94265	7.668475	8.297272	3.841732	5.297945	3844.25
GSA	3.569804	0.700714	17.58068	7.319542	8.192896	3.479603	5.33575	3205.71
PSO	3.555742	0.703886	21.66698	7.837641	8.111648	3.40429	5.431289	4085.292
GA	3.564938	0.710523	22.56662	7.917575	7.973648	3.426693	5.308706	4269.365

Table 17: Statistical results for the speed reducer design problem

Algorithm	mean	best	std	median	ET	rank
LEO	2996.348	2996.348	8.97E−13	2996.348	3.29166	1
RSA	3259.403	3018.311	80.46219	3258.554	1.513922	7
MPA	2997.327	2996.348	2.86026	2996.364	3.601877	2
TSA	3037.677	3008.294	14.77695	3039.229	2.595855	5
WOA	3110.617	3013.296	86.29035	3093.958	3.980838	6
MVO	3031.118	3007.468	11.34626	3032.71	4.210757	4
GWO	3004.897	2998.382	4.296095	3005.444	4.027501	3
TLBO	6.47E + 13	3844.25	6.06E + 13	4.48E + 13	20.44886	10
GSA	3512.39	3205.71	248.8387	3443.385	5.315226	8
PSO	2.03E + 14	4085.292	3.26E + 14	6.86E + 13	3.850949	11
GA	5.58E + 13	4269.365	6.6E + 13	2.54E + 13	6.526017	9

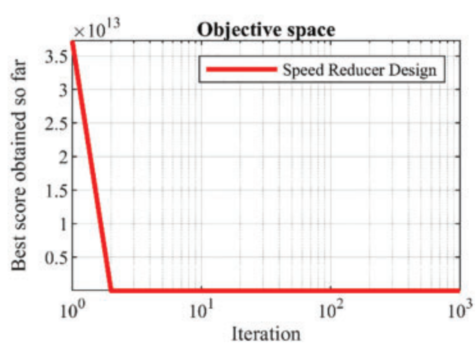

Figure 14: Convergence analysis of the LEO for the speed reducer design optimization problem

7.4 Pressure Vessel Design Optimization Problem

The pressure vessel design problem is an optimization challenge in real-world applications to minimize the design cost. The schematic of this design is shown in Fig. 15. The mathematical model of pressure vessel design is as follows [62]:

Consider: $X = [x_1,\ x_2,\ x_3,\ x_4] = [T_s,\ T_h,\ R,\ L]$.

Minimize: $f\ (x) = 0.6224x_1x_3x_4 + 1.778x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$.

Subject to:

$$g_1\ (x) = -x_1 + 0.0193x_3\ \le\ 0,\ \ g_2\ (x) = -x_2 + 0.00954x_3 \le\ 0,$$

$$g_3\ (x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le\ 0,\ g_4\ (x) = x_4 - 240\ \le\ 0,$$

with

$$0 \le x_1, x_2 \le 100\ ,\ 10 \le x_3, x_4 \le 200.$$

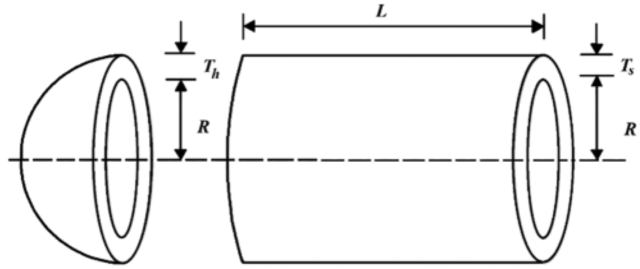

Figure 15: Schematic view of the pressure vessel design problem

The results obtained from the implementation of LEO and competitor algorithms on the pressure vessel design problem are reported in Tables 18 and 19. The simulation results show that LEO has the optimal solution to this problem with the values of design variables equal to (0.778027, 0.384579, 40.31228, 200) and the value of the objective function equal to 5882.901. The simulation results indicate that LEO performs better in pressure vessel design optimization than competitor algorithms. LEO convergence curve during pressure vessel design optimization is drawn in Fig. 16.

Table 18: Comparison results for the pressure vessel design problem

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
LEO	0.778027	0.384579	40.31228	200	5882.901
RSA	0.86013	0.593171	43.04395	168.5946	6865.859
MPA	0.778027	0.384579	40.31228	200	5882.901
TSA	0.781524	0.393985	40.37834	200	5946.404
WOA	0.888888	0.468339	45.65375	136.9744	6253.655

(Continued)

Table 18 (continued)

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
MVO	0.779395	0.387108	40.3343	199.8642	5900.834
GWO	0.778602	0.385041	40.31623	199.9834	5888.694
TLBO	1.243025	2.27283	43.75321	193.3142	16567.11
GSA	1.135128	0.561094	58.81491	112.4656	10086.81
PSO	1.404574	1.147427	71.4385	43.86409	16221.83
GA	1.441419	0.823568	59.47861	129.9565	15421.79

Table 19: Statistical results for the pressure vessel design problem

Algorithm	mean	best	std	median	ET	rank
LEO	5882.901	5882.901	1.87E-12	5882.901	3.66951	1
RSA	11252.55	6865.859	3227.668	10925.66	1.168503	7
MPA	5883.013	5882.901	0.426155	5882.901	2.989616	2
TSA	6313.121	5946.404	523.8611	6022.13	2.169944	4
WOA	8554.521	6253.655	2130.801	8233.353	3.369625	6
MVO	6740.322	5900.834	408.5954	6851.96	3.450834	5
GWO	6227.057	5888.694	516.3003	5930.408	3.286289	3
TLBO	34608.64	16567.11	14549.03	31967.28	17.56472	10
GSA	22843.39	10086.81	8114.234	22923.1	4.310473	8
PSO	42178.43	16221.83	12122.96	42055.87	3.356884	11
GA	34057.02	15421.79	12025.37	32102.06	5.450805	9

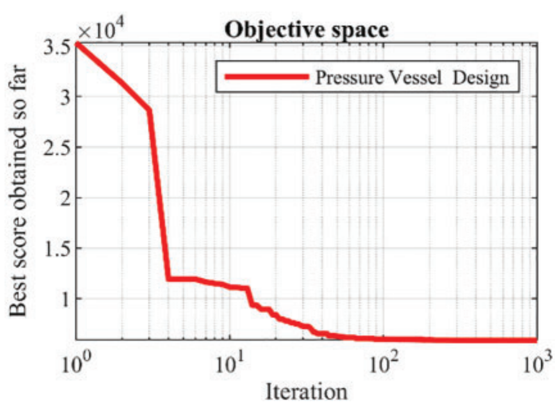

Figure 16: Convergence analysis of the LEO for the pressure vessel design optimization problem

7.5 The Effectiveness of the LEO in Solving Real-Time Applications

Real-Time Applications (RTAs) are applications that operate in specific time frames that users sense as current or immediate. Typically, RTAs are employed to process streaming data. Without ingesting and storing the data in a back-end database, real-time software should be able to sense, analyze and act on streaming data as it enters the system. RTAs usually use event-driven architecture to handle streaming data [63] asynchronously. RTAs include clustering applications, Internet of Things (IoT) applications, systems that control scientific experiments, medical imaging systems, industrial control systems, and certain monitoring systems.

Metaheuristic algorithms, including the proposed LEO approach, are effective tools for managing real-time applications. In many RTAs, a combination of metaheuristic algorithms and neural networks have been employed to optimize the performance of real-time systems. The proposed LEO approach has applications in various fields of RTAs, including sensor networks, medical applications, IoT systems, military applications, electric vehicle control, fuel injection system control, robotics applications, clustering, etc.

8 Conclusions and Future Research

This paper introduced a new human-based metaheuristic algorithm called Language Education Optimization (LEO), which has applications in optimization tasks. The fundamental inspiration behind LEO design is the process of teaching a foreign language in language schools where the teacher teaches skills to students. According to exploration and exploitation abilities, LEO was mathematically modeled in three phases (i) teacher selection, (ii) students learning from each other, and (iii) individual practice. The performance of LEO in optimization applications was tested on fifty-two benchmark functions of unimodal, multimodal, fixed-dimensional multimodal types and the CEC 2017 test suite. The optimization results showed that LEO, with its high power of exploration and exploitation, and its ability to balance exploration and exploitation, has a compelling performance in solving optimization problems. Ten well-known metaheuristic algorithms were employed to compare the results of the LEO implementation. The analysis of the simulation results showed that LEO has an effective performance in handling optimization tasks and providing solutions, and is far superior and more competitive than the competitor algorithms. The implementation results of the proposed LEO approach on four engineering design problems showed the high ability of LEO in optimizing real-world applications.

The most special advantage of the proposed LEO approach is that it does not have any control parameters and therefore does not need a parameter adjustment process. The high ability in exploration and exploitation and balancing them during the search process is another advantage of the proposed LEO. However, LEO also has limitations and disadvantages. First, as with all metaheuristic algorithms, there is no claim that LEO is the best optimizer for all optimization problems. The second disadvantage of LEO is that there is always a possibility that newer algorithms will be designed that perform better in solving optimization problems compared to the proposed approach. The third disadvantage of LEO is that, similar to other stochastic approaches. It does not provide any guarantee to provide the global optima for all optimization problems.

Following the design of LEO, several research tasks are activated for future work, the most important of which is the design of binary and multimodal versions of LEO. Employing LEO in optimization tasks in various sciences, real-time applications, and implementing LEO in real-world applications are other research suggestions of this study.

Acknowledgement: The authors thank Dušan Bednařík from the University of Hradec Kralove for our fruitful and informative discussions.

Funding Statement: The research was supported by the Project of Specific Research PřF UHK No. 2104/2022–2023, University of Hradec Kralove, Czech Republic.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Dhiman, G. (2021). SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. *Knowledge-Based Systems, 222,* 106926. DOI 10.1016/j.knosys.2021.106926.
2. Zeidabadi, F. A., Dehghani, A., Dehghani, M., Montazeri, Z., Hubálovský. et al. (2022). SSABA: Search step adjustment based algorithm. *Computers, Materials & Continua, 71(2),* 4237–4256. DOI 10.32604/cmc.2022.023682.
3. Mohammadi-Balani, A., Nayeri, M. D., Azar, A., Taghizadeh-Yazdi, M. (2021). Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Computers Industrial Engineering, 152,* 107050. DOI 10.1016/j.cie.2020.107050.
4. Cavazzuti, M. (2013). Deterministic optimization. In: *Optimization methods: From theory to design scientific and technological aspects in mechanics*, pp. 77–102. Berlin, Heidelberg: Springer.
5. Gonzalez, M., López-Espín, J. J., Aparicio, J., Talbi, E. G. (2022). A hyper-matheuristic approach for solving mixed integer linear optimization models in the context of data envelopment analysis. *PeerJ Computer Science, 8,* e828. DOI 10.7717/peerj-cs.828.
6. Črepinšek, M., Liu, S. H., Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR), 45(3),* 1–33. DOI 10.1145/2480741.2480752.
7. Goldberg, D. E., Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning, 3(2),* 95–99.
8. Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, pp. 1942–1948. Perth, WA, Australia.
9. Karaboga, D., Basturk, B. (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: *Foundations of fuzzy logic and soft computing. IFSA 2007. Lecture notes in Computer Science*, pp. 789–798. Berlin, Heidelberg, Springer.
10. Dorigo, M., Maniezzo, V., Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1),* 29–41.
11. Wolpert, D. H., Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1(1),* 67–82. DOI 10.1109/4235.585893.
12. Zhao, W., Wang, L., Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering, 388,* 114194. DOI 10.1016/j.cma.2021.114194.
13. Zhong, C., Li, G., Meng, Z. (2022). Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-Based Systems, 251(5),* 109215. DOI 10.1016/j.knosys.2022.109215.
14. Zamani, H., Nadimi-Shahraki, M. H., Gandomi, A. H. (2022). Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Computer Methods in Applied Mechanics and Engineering, 392,* 114616. DOI 10.1016/j.cma.2022.114616.
15. Dhiman, G., Garg, M., Nagar, A., Kumar, V., Dehghani, M. (2020). A novel algorithm for global optimization: Rat swarm optimizer. *Journal of Ambient Intelligence and Humanized Computing, 12,* 1–26.

16. Dhiman, G., Kaur, A. (2019). STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of Artificial Intelligence, 82,* 148–174. DOI 10.1016/j.engappai.2019.03.021.
17. Dhiman, G., Kumar, V. (2018). Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowledge-Based Systems, 159,* 20–50. DOI 10.1016/j.knosys.2018.06.001.
18. Jiang, Y., Wu, Q., Zhu, S., Zhang, L. (2022). Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems. *Expert Systems with Applications, 188,* 116026. DOI 10.1016/j.eswa.2021.116026.
19. Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95,* 51–67. DOI 10.1016/j.advengsoft.2016.01.008.
20. Abdollahzadeh, B., Gharehchopogh, F. S., Mirjalili, S. (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers Industrial Engineering, 158,* 107408. DOI 10.1016/j.cie.2021.107408.
21. Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A. H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications, 152,* 113377. DOI 10.1016/j.eswa.2020.113377.
22. Chopra, N., Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications, 198,* 116924. DOI 10.1016/j.eswa.2022.116924.
23. Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69,* 46–61. DOI 10.1016/j.advengsoft.2013.12.007.
24. Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., Gandomi, A. H. (2022). Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications, 191,* 116158. DOI 10.1016/j.eswa.2021.116158.
25. Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., Al-Atabany, W. (2022). Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation, 192,* 84–110. DOI 10.1016/j.matcom.2021.08.013.
26. Dhiman, G., Kumar, V. (2017). Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software, 114,* 48–70. DOI 10.1016/j.advengsoft.2017.05.014.
27. Kaur, S., Awasthi, L. K., Sangal, A. L., Dhiman, G. (2020). Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence, 90,* 103541. DOI 10.1016/j.engappai.2020.103541.
28. Storn, R., Price, K. (1997). Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11(4),* 341–359. DOI 10.1023/A:1008202821328.
29. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220(4598),* 671–680. DOI 10.1126/science.220.4598.671.
30. Pereira, J. L. J., Francisco, M. B., Diniz, C. A., Oliver, G. A., Cunha Jr, S. S. et al. (2021). Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Systems with Applications, 170,* 114522. DOI 10.1016/j.eswa.2020.114522.
31. Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems, 101,* 646–667. DOI 10.1016/j.future.2019.07.015.
32. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179(13),* 2232–2248. DOI 10.1016/j.ins.2009.03.004.
33. Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M. (2012). Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers Structures, 110,* 151–166. DOI 10.1016/j.compstruc.2012.07.010.

34. Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. (2021). Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Applied Intelligence, 51(3),* 1531–1551. DOI 10.1007/s10489-020-01893-z.
35. Dehghani, M., Montazeri, Z., Dhiman, G., Malik, O., Morales-Menendez, R. et al. (2020). A spring search algorithm applied to engineering optimization problems. *Applied Sciences, 10(18),* 6173. DOI 10.3390/app10186173.
36. Mirjalili, S., Mirjalili, S. M., Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27(2),* 495–513. DOI 10.1007/s00521-015-1870-7.
37. Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems, 191,* 105190. DOI 10.1016/j.knosys.2019.105190.
38. Dehghani, M., Samet, H. (2020). Momentum search algorithm: A new meta-heuristic optimization algorithm inspired by momentum conservation law. *SN Applied Sciences, 2(10),* 1–15. DOI 10.1007/s42452-020-03511-6.
39. Rao, R. V., Savsani, V. J., Vakharia, D. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43(3),* 303–315. DOI 10.1016/j.cad.2010.12.015.
40. Dehghani, M., Trojovský, P. (2021). Teamwork optimization algorithm: A new optimization approach for function minimization/maximization. *Sensors, 21(13),* 4567. DOI 10.3390/s21134567.
41. Trojovský, P., Dehghani, M. (2022). A new optimization algorithm based on mimicking the voting process for leader selection. *PeerJ Computer Science, 2,* cs976. DOI 10.7717/peerj-cs.976.
42. Ayyarao, T. L., RamaKrishna, N., Elavarasam, R. M., Polumahanthi, N., Rambabu, M. et al. (2022). War strategy optimization algorithm: A new effective metaheuristic algorithm for global optimization. *IEEE Access, 10,* 25073–25105. DOI 10.1109/ACCESS.2022.3153493.
43. Dehghani, M., Mardaneh, M., Malik, O. (2020). FOA: 'Following' optimization algorithm for solving power engineering optimization problems. *Journal of Operation and Automation in Power Engineering, 8(1),* 57–64.
44. Mousavirad, S. J., Ebrahimpour-Komleh, H. (2017). Human mental search: A new population-based metaheuristic optimization algorithm. *Applied Intelligence, 47(3),* 850–887. DOI 10.1007/s10489-017-0903-6.
45. Dehghani, M., Trojovská, E., Trojovský, P. (2022). A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. *Scientific Reports, 12(1),* 9924. DOI 10.1038/s41598-022-14225-7.
46. Trojovská, E., Dehghani, M. (2022). A new human-based metahurestic optimization method based on mimicking cooking training. *Scientific Reports, 12(1),* 1–24. DOI 10.1038/s41598-022-19313-2.
47. Moosavi, S. H. S., Bardsiri, V. K. (2019). Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Engineering Applications of Artificial Intelligence, 86,* 165–181. DOI 10.1016/j.engappai.2019.08.025.
48. Dehghani, M., Mardaneh, M., Guerrero, J. M., Malik, O., Kumar, V. (2020). Football game based optimization: An application to solve energy commitment problem. *International Journal of Intelligent Engineering and Systems, 13,* 514–523. DOI 10.22266/ijies.
49. Moghdani, R., Salimifard, K. (2018). Volleyball premier league algorithm. *Applied Soft Computing, 64,* 161–185. DOI 10.1016/j.asoc.2017.11.043.
50. Kaveh, A., Zolghadr, A. (2016). A novel meta-heuristic algorithm: Tug of war optimization. *International Journal of Optimization in Civil Engineering, 6(4),* 469–492.
51. Zeidabadi, F. A., Dehghani, M., Trojovský, P., Hubálovský, Š, Leiva, V. et al. (2022). Archery algorithm: A novel stochastic optimization algorithm for solving optimization problems. *Computers, Materials & Continua, 72(1),* 399–416. DOI 10.32604/cmc.2022.024736.

52. Zeidabadi, F. A., Dehghani, M. (2022). POA: Puzzle optimization algorithm. *International Journal of Intelligent Engineering and Systems, 15(1),* 273–281.
53. Dehghani, M., Montazeri, Z., Givi, H., Guerrero, J. M., Dhiman, G. (2020). Darts game optimizer: A new optimization technique based on darts game. *International Journal of Intelligent Engineering and Systems, 13,* 286–294.
54. Doumari, S. A., Givi, H., Dehghani, M., Malik, O. P. (2021). Ring toss game-based optimization algorithm for solving various optimization problems. *International Journal of Intelligent Engineering and Systems, 14(3),* 545–554.
55. Dehghani, M., Montazeri, Z., Malik, O. P. (2019). DGO: Dice game optimizer. *Gazi University Journal of Science, 32(3),* 871–882. DOI 10.35378/gujs.484643.
56. Dehghani, M., Montazeri, Z., Malik, O. P., Ehsanifar, A., Dehghani, A. (2019). OSA: Orientation search algorithm. *International Journal of Industrial Electronics, Control and Optimization, 2(2),* 99–112.
57. Yao, X., Liu, Y., Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3(2),* 82–102. DOI 10.1109/4235.771163.
58. Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1,* 80–83. DOI 10.2307/3001968.
59. Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P. et al. (2016). Evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. *Technology Report*. DOI 10.13140/RG.2.2.12568.70403.
60. Gandomi, A. H., Yang, X. S. (2011). Benchmark problems in structural optimization. In: Koziel, S., Yang, X. S. (Eds.), *Computational optimization, methods and algorithms. studies in computational intelligence*, vol. 356. Berlin, Heidelberg: Springer.
61. Mezura-Montes, E., Coello, C. A. C. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (Eds.), *MICAI 2005: Advances in artificial intelligence*, pp. 652–662, vol. 3789. Berlin, Heidelberg: Springer.
62. Kannan, B., Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design, 116(2),* 405–411. DOI 10.1115/1.2919393.
63. Rao, A. S., Radanovic, M., Liu, Y., Hu, S., Fang, Y. et al. (2022). Real-time monitoring of construction sites: Sensors, methods, and applications. *Automation in Construction, 136,* 104099. DOI 10.1016/j.autcon.2021.104099.