A New Human-Inspired Metaheuristic Algorithm for Solving Optimization Problems Based on Mimicking Sewing Training

Mohammad Dehghani
University of Hradec Králové
Eva Trojovská (eva.trojovska@uhk.cz)
University of Hradec Králové
Tomáš Zuščák
University of Hradec Králové

Article

Keywords: Optimization, stochastic method, sewing, sewing training, tailor, human-based algorithm, exploration, exploitation

Posted Date: September 12th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2037953/v1

A New Human-Inspired Metaheuristic Algorithm for Solving Optimization Problems Based on Mimicking Sewing Training

Mohammad Dehghani, Eva Trojovská* and Tomáš Zuščák

Department of Mathematics, Faculty of Science, University of Hradec Králové,
500 03 Hradec Králové, Czech Republic; mohammad.dehghani@uhk.cz (M.D.);
eva.trojovska@uhk.cz (E.T.); tomas.zuscak@uhk.cz (T.Z.)

*Corresponding Author:
Eva Trojovská
University of Hradec Králové, Rokitanského 62, Hradec Králové 500 03, Czech Republic
Email address: eva.trojovska@uhk.cz

Abstract

This paper introduces a new human-based metaheuristic algorithm called Sewing Training-Based Optimization (STBO). The fundamental inspiration of STBO is the process of teaching sewing to beginner tailors. The process is described in three phases: (i) training, (ii) imitation of the instructor's skills, and (iii) practice, and is then mathematically modeled. STBO performance is evaluated on twenty-three objective functions of the types of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal. The optimization results show that STBO, with its high power of exploration and exploitation, has provided suitable solutions for benchmark functions. Also, to evaluate the quality of STBO, the results are compared with ten well-known metaheuristic algorithms. Furthermore, the simulation results show that STBO has a much more competitive performance than competitor algorithms by providing superior results. Finally, the implementation of STBO in solving four engineering design problems demonstrates the capability of the proposed approach in dealing with real-world applications.

Keywords: Optimization, stochastic method, sewing, sewing training, tailor, human-based algorithm, exploration, exploitation

Introduction

Optimization problems represent challenges with several possible solutions, one of which is the best choice. Accordingly, optimization is the process of achieving the best solution to the optimization problem. An optimization problem has three main parts: decision variables, objective function, and constraints [1]. Optimization aims to determine the values of the decision variables by considering the constraints so that the objective function is optimized [2]. Optimization problem-solving methods fall into two groups, deterministic and random approaches. Deterministic approaches deal well with linear, continuous, differentiable, and convex optimization problems. However, the disadvantage of these approaches is that their ability is lost in solving nonlinear, non-convex, non-differentiable, high-dimensional, NP-hard problems and discrete search spaces. These items, which have led to the inability of deterministic approaches, are among the features of real-world optimization problems. Stochastic algorithms, especially metaheuristic algorithms, have been introduced to meet this challenge. Metaheuristic algorithms can provide suitable solutions to optimization problems by using random search in problem-solving space and relying on random operators [3]. The critical thing about metaheuristic algorithms is that there is no guarantee that the solution obtained from these methods will be the best or global optimal. This fact has led researchers to develop numerous metaheuristic algorithms to achieve better solutions.

Metaheuristic algorithms are designed based on modeling ideas that exist in nature. Among the most famous metaheuristic algorithms, can be mentioned to Genetic Algorithm (GA) [4], Particle Swarm Optimization (PSO) [5], Ant Colony Optimization (ACO) [6], and Artificial Bee Colony (ABC) [7]. GA is based on modeling the reproductive process, PSO is developed based on modeling the swarm movement of birds and fish in nature, ACO is designed based on simulating the natural behaviors of ants, and ABC is introduced based on modeling the activities of bee colonies in search of food.

Metaheuristic algorithms must have an acceptable ability for exploration and exploitation to deliver suitable optimization performance. Exploration is the concept of global search in different parts of the problem-solving space to find the main optimal area. Exploitation means a local search around candidate solutions to find better possible solutions that may be near them. In addition to having a high quality in exploration and exploitation, balancing these two indicators is the key to the success of metaheuristic algorithms [8].

The main research question is, despite the large number of metaheuristic algorithms introduced so far, is there still a need to introduce newer methods? The answer to this question lies in the concept of the No Free Lunch (NFL) theorem [9]. According to the NFL, the good performance of an algorithm in solving a set of optimization problems does not guarantee the same performance of that algorithm in other optimization problems. This result is due to the random nature of metaheuristic algorithms in achieving the solution. The NFL states it is impossible to claim that a particular algorithm is the best optimizer for dealing with all optimization issues. As a result, the NFL theorem has encouraged researchers to design new algorithms to provide more appropriate solutions and closer to global optima for optimization problems. The NFL has also motivated the authors of this study to be able to solve optimization problems more effectively by designing a new metaheuristic algorithm.

The novelty and innovation of this paper are in designing a new algorithm called Sewing Training-Based Optimization (STBO) for optimization applications. The contributions of this article are as follows: A new human-based metaheuristic algorithm based on sewing training modeling is introduced. STBO is modeled in three phases: (i) training, (ii) imitation of the instructor's skills, and (iii) practice. STBO performance has been tested on twenty-three benchmark functions of various unimodal, high-dimensional, and fixed-dimensional multimodal types. STBO results are compared with the performance of ten well-known metaheuristic algorithms. STBO's performance in solving real-world applications is evaluated on four engineering design issues.

The rest of the paper is organized so that a literature review is presented in the section "Literature review." Next, the proposed algorithm is introduced and modeled in the section "Sewing Training-Based Optimization." Simulations and analysis of their results are presented in the section "Simulation Studies and Results." The STBO's performance in solving real-world problems is shown in the section "STBO for real-world applications." Finally, conclusions and several study proposals are provided in the section "Conclusion and future works."

Literature Review

Metaheuristic algorithms have been developed based on mathematical simulations of various natural phenomena, animal behaviors, biological sciences, physics concepts, game rules, human behaviors, and other evolution-based processes. Based on the source of inspiration used in the design, metaheuristic algorithms fall into five groups: swarm-based, evolutionary-based, physics-based, game-based, and human-based.

Swarm-based algorithms are derived from the mathematical modeling of natural swarming phenomena, the behavior of animals, birds, aquatic animals, insects, and other living organisms. For example, ant colonies can find an optimal path to supply the required food resources. Simulating this behavioral feature of ants forms the basis of ACO. Fireflies' feature of emitting flashing light and the light communication between them has been a source of inspiration in the design of the Firefly Algorithm (FA) [10]. Swarming activities such as foraging and hunting among animals are intelligence processes which are employed in the design of various algorithms such as PSO, ABC, Grey Wolf Optimizer (GWO) [11], Whale Optimization Algorithm (WOA) [12], Marine Predator Algorithm (MPA) [13], Cat and Mouse based Optimizer (CMBO) [14], Reptile Search Algorithm (RSA) [15], and Orca Predation Algorithm (OPA) [16].

Evolutionary-based algorithms are inspired by the biological sciences, the concept of natural selection, and random operators. Differential Evolution (DE) [17] and GA are two of the most significant evolutionary algorithms developed based on the mathematization of the reproductive process, concepts of Darwin's theory of evolution, and random operators of selection, mutation, and crossover.

Physics-based algorithms have been developed by simulating various laws, concepts, forces, and phenomena in physics. The physical phenomenon of the water cycle has been the main idea in designing Water Cycle Algorithm (WCA) [18]. The employment of physical forces to design metaheuristic algorithms has been successful in designing algorithms such as Gravitational Search Algorithm (GSA) [19], Spring Search

Algorithm (SSA) [20], and Momentum Search Algorithm (MSA) [21]. GSA is based on modeling the gravitational force that exists between masses at different distances from each other. SSA is inspired by the simulation of the spring tensile force and the Hook law between the weights connected by springs. MSA is developed based on the mathematization of the force of bullets' momentum that moves toward the optimal solution. Simulated Annealing (SA) [22], Flow Regime Algorithm (FRA) [23], Equilibrium Optimizer (EO) [24], and Multi-Verse Optimizer (MVO) [25] belong, e.g., among some other physics-based metaheuristic algorithms.

Game-based algorithms are formed by mathematical modeling of various game rules. Volleyball Premier League (VPL) algorithm [26] and Football Game-Based Optimization (FGBO) [27] are two game-based algorithms that are designed based on the simulation of club competitions in volleyball and football games, respectively. The players' attempt in the tug-of-war game has been the main inspiration for the Tug of War Optimization (TWO) [28] design. The skill and strategy of the players in completing the puzzle pieces have been the idea behind the Puzzle Optimization Algorithm (POA) [29] design.

Human-based algorithms have emerged inspiring by human behaviors and interactions. The most widely used and well-known algorithm in this group is Teaching-Learning-Based Optimization (TLBO). TLBO is introduced based on the mathematization of educational interactions between teacher and students [30]. The treatment process that the doctor uses to treat patients has been a central idea in the design of the Doctor and Patients Optimization (DPO) [31]. The relationships and collaboration of team members to perform a team work and achieve the planned goal has been the source of inspiration for the Teamwork Optimization Algorithm (TOA) design [32]. Some other human-based metaheuristic algorithms are: Human Mental Search (HMS) [33], Multi-Leader Optimizer (MLO), Poor and Rich Optimization (PRO) [34], Following Optimization Algorithm (FOA) [35], and Election-Based Optimization Algorithm (EBOA) [36].

Based on the best knowledge obtained from the literature review, modeling the sewing training process has not been applied in designing any metaheuristic algorithm. However, sewing training by a training instructor to beginner tailors is an intelligent human activity that has the potential to simulate an optimizer. Therefore, a new human-based metaheuristic algorithm based on mathematical modeling of sewing training is designed in this paper to address this research gap. The design of this algorithm will be discussed in the next section.

Sewing Training-Based Optimization

This section introduces the proposed Sewing Training-Based Optimization (STBO) algorithm and presents its mathematical model.

Inspiration and main idea of STBO

The activity of teaching sewing skills by a training instructor to beginner tailors is an intelligent process. The first step for a beginner is to choose a training instructor. Selecting the training instructor is essential in improving a beginner's sewing skills. Next, the instructor teaches sewing techniques to the beginner tailor. The second step in this process is the beginner tailor's efforts to mimic the skills of the training instructor. The beginner tailor tries to bring his skills to the level of the instructor as much as possible. The third step in the sewing training process is practice. The beginner tailors try to improve their skills in sewing by practicing. The interactions between beginner tailors and training instructors indicate the high potential of the sewing training process to be considered for designing an optimizer. Mathematical modeling of these intelligent interactions is the fundamental inspiration in the design of STBO.

Mathematical Model of STBO

The proposed STBO algorithm is a population-based metaheuristic algorithm whose members are beginner tailors and training instructors. Each member of the STBO population refers to a candidate solution to the problem that represents the proposed values for the decision variables. As a result, each STBO member can be mathematically modeled with a vector and the STBO population using a matrix. The STBO population is specified by a matrix representation in Equation (1).

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

where X is the STBO population matrix, X_i is the ith STBO's member, N is the number of STBO population members, and m is the number of problem variables. At the beginning of the STBO implementation, all population members are randomly initialized using Equation (2).

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

where $x_{i,j}$ is the value of the jth variable determined by the ith STBO's member X_i, r is a random number in the interval $[0,1]$, lb_j and ub_j are the lower and upper bound of the jth problem variable, respectively.

Each STBO member represents a candidate solution to the given problem. Therefore, the problem's objective function can be evaluated based on the values specified by each candidate solution. Based on the placement of candidate solutions in the problem variables, the values calculated for the objective function can be modeled using a vector by Equation (3).

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

where F is the objective function vector and F_i is the objective function value for the ith candidate solution.

The values of the objective function are the main criterion for comparing candidate solutions with each other. The solution with the best value for the objective function is identified as the best candidate solution or the best member of the population X_{best}. Updating the algorithm's population in each iteration leads to finding new objective function values. Accordingly, in each iteration, the best candidate solution must be updated. The design of the algorithm guarantees that the best candidate solution at the end of each iteration is also the best candidate solution from all previous iterations.

The process of updating candidate solutions in STBO is perform in three phases: (i) training, (ii) imitation of the instructor skills, and (iii) practice.

Phase 1: Training (exploration)

The first phase of updating STBO members is based on simulating the process of selecting a training instructor and acquiring sewing skills by beginner tailors. For each STBO member as a beginner tailor, all other members who have a better value for the objective function are considered as training instructors for that member. The set of all candidate members as the group of possible training instructors for each STBO member $X_i,\ i=1,2,\dots,N$, is defined using the following identity

$$CSI_i=\left\{X_k \mid F_k<F_i\,, k\in\{1,2,\dots,N\}\right\}\cup\{X_{best}\}, \tag{4}$$

where CSI_i is the set of all possible candidate training instructors for the ith STBO member. In the case $X_i=X_{best}$ the only possible candidate training instructor is X_{best} itself, i. e., $CSI_i=\{X_{best}\}$. Then, for each $i\in\{1,2,\dots,N\}$, a member from the set CSI_i is randomly selected as the training instructor of the ith member of STBO, and it is denoted as SI_i. This selected instructor SI_i teaches the ith STBO member to sewing skills. Guiding members of the population under the guidance of instructors allows the STBO population to scan different areas of the search space to identify the main optimal area. This STBO update phase demonstrates the proposed approach's exploration ability in global search. At first, a new position for each population member is generated using Equation (5) to update population members based on this phase of the STBO.

$$x_{i,j}^{P1}=x_{i,j}+r_{i,j}\cdot\left(SI_{i,j}-I_{i,j}\cdot x_{i,j}\right), \tag{5}$$

where $x_{i,j}^{P1}$ is its dth dimension, F_i^{P1} is its objective function value, $I_{i,j}$ are numbers that are selected randomly from the set $\{1,2\}$, and $r_{i,j}$ are random numbers from the interval $[0,1]$.

Then, if this new position improves the objective function value, it replaces that population member's previous position. This update condition is modeled using Equation (6).

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

where X_i^{P1} is the new position of the ith STBO member based on the first phase of STBO.

Phase 2: Imitation of the instructor skills (exploration)

The second phase of updating STBO members is based on simulating beginner tailors trying to mimic the skills of instructors. In the design of STBO, it is assumed that the beginner tailor tries to bring his sewing skills to the level of the instructor as much as possible. Given that each STBO member is a vector of the dimension m and each component represents a decision variable thus, in this phase of STBO, it is assumed that each decision variable represents a sewing skill. Each STBO member imitates m_s skills of the chosen instructor, $1 \le m_s \le m$. This process moves the population of the algorithm to different areas in the search space, which indicates the STBO exploration ability. The set of variables that each STBO member imitates (i.e., the set of skills of the training instructor) is specified in Equation (7).

$$SD_i = \{d_1, d_2, \dots, d_{m_s}\}, \tag{7}$$

where SD_i is an $m_s -$ combination of the set $\{1,2, \dots, m\}$, which represents the set of the indexes of decision variables (i.e., skills) identified to imitate by the ith member from the instructor and $m_s = \left\lfloor 1 + \frac{t}{2T} m \right\rfloor$ is the number of skills selected to mimic, t is the iteration counter, and T is the total number of iterations.

The new position for each STBO member is calculated based on the simulation of imitating these instructor skills, using the following identity

$$x_{i,j}^{P2} = \begin{cases} SI_{i,j}, & j \in SD_i; \\ x_{i,j}, & else, \end{cases} \tag{8}$$

where X_i^{P2} is the newly generated position for the ith STBO member based on the second phase of STBO, $x_{i,j}^{P2}$ is the dth dimension of X_i^{P2}. This new position replaces the previous position of the corresponding member if it improves the value of the objective function

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} < F_i; \\ X_i, & else, \end{cases} \tag{9}$$

where F_i^{P2} is the objective function value of X_i^{P2}.

Phase 3: Practice (exploitation)

The third phase of updating STBO members is based on simulating beginner tailoring practices to improve sewing skills. In fact, in this phase of STBO design, local search is performed around candidate solutions with the goal of finding the best possible solutions near these candidate solutions. This phase of the STBO represents the exploitation capability of the proposed algorithm in local search. In order to mathematically model this STBO phase (with a correction to stay the all newly computed population members in the given search space), a new position around each member of the STBO is first generated using Equation (10).

$$x_{i,j}^{P3} = \begin{cases} lb_j, & x_{i,j}^* < lb_j; \\ x_{i,j}^*, & x_{i,j}^* \in [lb_j, ub_j]; \\ ub_j, & x_{i,j}^* > ub_j, \end{cases} \tag{10}$$

where $x_{i,j}^* = x_{i,j} + (lb_j + r_{i,j} \cdot (ub_j - lb_j)/t)$ and $r_{i,j}$ is a random number from the interval $[0,1]$. Then, if the value of the objective function improves, it replaces the previous position of the STBO member according to Equation (11).

$$X_i = \begin{cases} X_i^{P3}, & F_i^{P3} < F_i; \\ X_i, & else, \end{cases} \tag{11}$$

where X_i^{P3} is the new generated position for the ith STBO member based on second phase of STBO, $x_{i,j}^{P3}$ is its dth dimension, and F_i^{P3} is its objective function value.

Repetition Process and Pseudo-Code of STBO

The first STBO iteration is completed after updating all candidate solutions based on the first to third phases. Then the update process is repeated until the last iteration of the algorithm, based on Equations (4) to (11). After the full implementation of the STBO on the given problem, the best candidate solution recorded during the algorithm iteration is introduced as the solution. Finally, STBO implementation steps are presented as pseudo-code in Algorithm 1.

Algorithm 1. Pseudo-code of STBO.

```
Start STBO.
Input the optimization problem information.
Adjust N and T.
Initialize the STBO population by (2) and create vector F of the values of the objective function by (3).
For t = 1 to T
   For i = 1 to N
   Phase 1: Training (exploration)
      Determine the set of candidate training instructors for the ith member by (4). CSI_i ← {X_k | F_k < F_i , k ∈ {1,2, … , N}} ∪ {X_best}.
      Choose the training instructor SI_i from CSI_i to teach sewing the ith STBO member.
      Calculate the new position for the ith STBO member using (5). x_{i,j}^{P1} ← x_{i,j} + r_{i,j} · (SI_{i,j} − I_{i,j} · x_{i,j})
      Update the position of the ith STBO member using (6).   X_i ← { X_i^{P1},  F_i^{P1} < F_i
                                                                      { X_i,      else
   Phase 2: Imitation of the instructor skills (exploration)
      Calculate SD_i using Equation (7).
      Calculate the new position of the ith STBO member using Equation (8). x_{i,j}^{P2} ← { SI_{i,j}, j ∈ SD;
                                                                                          { x_{i,j}, else.
      Update the position of the ith STBO member using (9). X_i ← { X_i^{P2}, F_i^{P2} < F_i;
                                                                  { X_i, else.
  Phase 3: Practice (exploitation)
      Calculate the new position for the ith STBO member using (10). x_{i,j}^{P3} ← x_{i,j} + (lb_j + r_{i,j}·(ub_j − lb_j)) / t.
      Update the position of the ith STBO member using (11). X_i ← { X_i^{P3}, F_i^{P3} < F_i;
                                                                   { X_i, else.
   end
   Update the best candidate solution.
end
Output the best candidate solution obtained by STBO.
End STBO.
```

Computational Complexity of STBO

In this subsection, the computational complexity of STBO is investigated. Since the most time-consuming step in the entire algorithm is calculating the values of the objective function, which are very complicated in most real applications, the computational complexity of STBO can be estimated based on the number of population members generated in the algorithm. STBO initialization has a computational complexity equal to $O(Nm)$, where N is the number of STBO members and m is the number of problem variables. In each STBO iteration, the candidate solution is updated in three phases. Thus, the computational complexity of the STBO update process is equal to $O(3NmT)$, where T is the number of iterations of the algorithm. As a result, the total computational complexity of STBO is equal to $O(Nm\,(1 + 3T))$.

Simulation Studies and Results

In this section, the ability of the proposed STBO algorithm in optimization applications and solution presentation is evaluated. In this regard, twenty-three objective functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types are employed to test the STBO optimization capability [37]. The performance of DTBO is compared with the performance of ten well-known metaheuristic algorithms GA, PSO, GSA, MPA, WOA, TLBO, RSA, MVO, GWO, and TSA. Each of the competing metaheuristic algorithms and STBO is used in twenty independent runs, where each run contains 1000 iterations. The implementation results of metaheuristic algorithms are reported using six statistical indicators: mean, standard deviation (std), best, worst, median, and rank. The mean of rank is considered a ranking criterion of the performance of optimization algorithms in each of the objective functions. The values of the control parameters of competitor metaheuristic algorithms are listed in Table 1.

Algorithm	**Parameter**	**Value**
RSA	Sensitive parameter	$\beta = 0.01$
	Sensitive parameter	$\alpha = 0.1$
	Evolutionary Sense (ES)	ES: randomly decreasing values between 2 and -2
MPA	Binary vector	U = 0 or 1
	Random vector	R is a vector of uniform random numbers in $[0,1]$.
	Constant number	$P = 0.5$
	Fish Aggregating Devices (FADs)	$FADs$ = 0.2
TSA	c1, c2, c3	random numbers lie in the interval [0,1].
	Pmin	1
	Pmax	4
WOA	l is a random number in [-1,1].	
	r is a random vector in $[0,1]$.	
	Convergence parameter (a)	a: Linear reduction from 2 to 0.
GWO	Convergence parameter (a)	a: Linear reduction from 2 to 0.
MVO	Wormhole existence probability (WEP)	Min(WEP) = 0.2 and Max(WEP)=1.
	Exploitation accuracy over the iterations (p)	$p = 6$.
TLBO	random number	rand is a random number from interval [0,1].
	T_F: teaching factor	T_F = round $[(1 + rand)]$
GSA	Alpha	20
	G_0	100
	Rnorm	2
	Rnorm	1
PSO	Velocity limit	10% of dimension range
	Topology	Fully connected
	Inertia weight	Linear reduction from 0.9 to 0.1
	Cognitive and social constant	$(C_1, C_2) = (2, 2)$
GA	Type	Real coded
	Mutation	Gaussian (Probability = 0.05)
	Crossover	Whole arithmetic (Probability = 0.8)
	Selection	Roulette wheel (Proportionate)

Table 1. Assigned values to the control parameters of competitor algorithms.

Evaluation on Unimodal benchmark functions

The results of optimization of unimodal functions F1 to F7 using STBO and competitor algorithms are reported in Table 2. The optimization results show that STBO provides the exact optimal solution for functions F1 to F6. For optimization of function F7, STBO is the best optimizer compared to competing algorithms. The simulation results show that STBO has outperformed competitor algorithms in handling the F1 to F7 unimodal functions and has been ranked first among the compared algorithms.

Evaluation on high dimensional multimodal benchmark functions

The results obtained using STBO and competitor algorithms in optimizing high-dimensional multimodal functions F8 to F13 are presented in Table 3. Based on the results, STBO has provided the exact optimal solution for optimizing functions F9 and F11. In solving the functions F8, F10, F12, and F13, the STBO has performed better than all competitor algorithms. Analysis of the simulation results indicates the superiority of STBO over competing algorithms in handling the high-dimensional multimodal functions of F8 to F13.

Evaluation on fixed dimensional multimodal benchmark functions

The results of the implementation of STBO and competitor algorithms on fixed-dimensional multimodal functions F14 to F23 are released in Table 4. Compared to competitor algorithms, the optimization results show that STBO is the best optimizer in optimizing benchmark functions F14, F15, and F18. In optimizing functions F16, F17, and F19 to F23, the proposed STBO, and some competitor algorithms have a similar value in the "mean" index. However, STBO provides more efficient performance in these functions by providing better values of the "std" index. Analysis of the simulation results shows that STBO performs better than competitor algorithms in solving fixed-dimensional functions F14 to F23.

The performance of STBO and competitor algorithms in optimizing F1 to F23 functions is presented as a boxplot in Figure 1.

Statistical analysis

In this subsection, statistical analysis is presented to further evaluate the performance of the STBO compared to competitor algorithms. Wilcoxon sum rank test [38] has been employed to determine whether there is a statistically significant difference between the results obtained from STBO and competing algorithms. In the Wilcoxon sum rank test, the p-value index determines the significant difference between the two data samples. The results of the Wilcoxon sum rank test on the performance of STBO and competitor algorithms are reported in Table 5. Based on these results, in cases where the p-value is calculated as less than 0.05, STBO has a statistically significant superiority over the competitor algorithm.

		GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	STBO
F_1	mean	33.35136	0.181599	1.09E-16	9.67E-74	0.160829	7.38E-59	1.5E-154	2.3E-47	8.66E-50	0	0
	best	19.78438	3.14E-05	5.38E-17	4.45E-77	0.092366	4.54E-61	2.9E-169	4.07E-50	3.59E-52	0	0
	worst	56.28509	3.274664	2.92E-16	1.07E-72	0.252881	5.46E-58	2.6E-153	1.26E-46	5.74E-49	0	0
	std	8.395292	0.729657	5.21E-17	2.84E-73	0.040725	1.36E-58	5.8E-154	4.15E-47	1.43E-49	0	0
	median	33.26899	0.007379	1.02E-16	2.37E-75	0.160607	1.43E-59	1.8E-161	3.45E-48	2.18E-50	0	0
	rank	10	9	7	3	8	4	2	6	5	1	1
F_2	mean	3.188112	1.419726	5.33E-08	1.11E-38	0.252866	7.55E-35	7.6E-104	2.32E-28	5.72E-28	0	0
	best	1.869152	0.080876	3.98E-08	1.43E-39	0.150256	7.14E-36	7.7E-113	1.68E-30	3.3E-30	0	0
	worst	4.515361	13.77933	8.14E-08	6.19E-38	0.471703	2.23E-34	5.8E-103	3.62E-27	3.98E-27	0	0
	std	0.693334	3.034488	1.02E-08	1.58E-38	0.069403	7.49E-35	1.6E-103	8.02E-28	9.79E-28	0	0
	med	3.32271	0.434611	5.3E-08	4.37E-39	0.239383	3.76E-35	6.3E-107	2.23E-29	2.02E-28	0	0
	rank	10	9	7	3	8	4	2	5	6	1	1
F_3	mean	2125.752	1094.128	480.8968	2.6E-24	12.60085	9.67E-14	22774.89	2.13E-11	8.45E-13	0	0
	best	1111.139	29.63701	218.105	3.17E-28	4.866897	2.26E-19	8159.212	8.49E-20	1.16E-19	0	0
	worst	2997.68	5273.106	804.0185	2.31E-23	20.88111	1.92E-12	37690.63	2.08E-10	7.29E-12	0	0
	std	495.7954	1618.416	149.9243	5.49E-24	4.434706	4.29E-13	8502.149	6.29E-11	1.75E-12	0	0
	median	2238.566	428.5796	458.5233	2.42E-25	13.20446	3.96E-17	21453.44	2.66E-14	3.2E-14	0	0
	rank	9	8	7	2	6	3	10	5	4	1	1
F_4	mean	3.210794	6.507623	0.993225	1.93E-30	0.611151	2.22E-14	37.28855	0.003514	2.71E-19	0	0
	best	2.113563	3.511675	1.27E-08	1.32E-31	0.284853	6.78E-16	1.331001	1.18E-05	7.45E-20	0	0
	worst	4.712766	10.14096	4.183409	5.42E-30	1.408459	1.06E-13	80.18979	0.016565	6.16E-19	0	0
	std	0.64983	1.970793	1.230111	1.65E-30	0.293951	3.3E-14	29.04785	0.004762	1.37E-19	0	0
	med	3.135406	6.130021	0.579869	1.67E-30	0.563237	7.57E-15	26.40247	0.001683	2.83E-19	0	0
	rank	8	9	7	2	6	4	10	5	3	1	1
F_5	mean	420.6	112.2916	43.52647	26.53115	308.8081	26.83281	27.0278	28.46818	23.58215	10.13576	0
	best	227.4906	30.07967	25.05162	26.03302	27.29038	25.32729	26.49484	27.13541	23.00782	1.72E-28	0
	worst	688.7775	400.1077	177.7903	28.75063	2557.854	28.5481	27.97937	29.2537	24.95884	28.99011	0
	std	122.0165	85.2441	39.12394	0.592674	622.6938	0.947923	0.328332	0.570757	0.516114	14.17161	0
	median	386.0621	84.06704	26.38317	26.35271	31.67272	26.5829	26.96864	28.6545	23.39152	5.8E-26	0
	rank	11	9	8	4	10	5	6	7	3	2	1
F_6	mean	34.0323	0.028587	1.13E-16	1.18022	0.159222	0.640991	0.119577	3.83522	2.08E-09	6.617444	0
	best	14.51884	9.98E-06	4.11E-17	0.572861	0.084059	1.14E-05	0.011797	2.823119	9.15E-10	3.624796	0
	worst	71.07024	0.324785	2.43E-16	1.754355	0.247305	1.7184	0.366934	4.774542	6.27E-09	7.498843	0
	std	15.19874	0.074692	4.95E-17	0.316997	0.046291	0.371946	0.110996	0.566697	1.13E-09	1.018988	0
	med	29.14039	0.00087	1.02E-16	1.223701	0.16068	0.748235	0.082192	3.934263	1.91E-09	7.154133	0
	rank	11	4	2	8	6	7	5	9	3	10	1
F_7	mean	0.009847	0.166943	0.067112	0.001589	0.00965	0.00083	0.001159	0.004861	0.000518	0.000103	1.24E-05
	best	0.006283	0.082498	0.019988	0.000338	0.004902	0.000262	2.8E-06	0.002251	0.000114	1.21E-05	2.33E-06
	worst	0.01905	0.289493	0.255451	0.004146	0.020283	0.002168	0.005668	0.010801	0.001654	0.000304	3.79E-05
	std	0.003084	0.052796	0.050227	0.000938	0.003826	0.00054	0.001539	0.002418	0.000365	8.26E-05	9.87E-06
	median	0.009485	0.159235	0.05478	0.001583	0.008695	0.000638	0.000632	0.004281	0.00041	9.22E-05	8.63E-06
	rank	9	11	10	6	8	4	5	7	3	2	1
Sum rank		68	59	48	28	52	31	40	44	27	18	7
Mean rank		9.714286	8.428571	6.857143	4	7.428571	4.428571	5.714286	6.285714	3.857143	2.571429	1
Total rank		11	10	8	4	9	5	6	7	3	2	1

Table 2. Evaluation results on unimodal functions.

		GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	STBO
F_8	mean	-8551.34	-6891.6	-2463.53	-5320.45	-7820.6	-6233.81	-11107.1	-5989.44	-9641.09	-5392.18	-12269.7
	best	-9693.59	-8047.43	-3421.12	-6267.63	-9486.17	-7747.76	-12569.3	-6812.45	-10493.2	-5729.76	-12569.3
	worst	-7299.22	-5394.56	-1914.24	-4471.98	-6625.33	-4899.26	-8492.23	-4918.97	-8788.53	-4359.71	-11262.6
	std	761.0887	874.6514	363.8174	473.6453	741.5584	812.3497	1587.459	638.8189	434.3437	350.8895	362.218
	median	-8611.96	-6843.87	-2484.38	-5243.46	-7737.48	-6303.59	-11928.6	-6010.79	-9651.7	-5542.27	-12395.1
	rank	4	6	11	10	5	7	2	8	3	9	1
F_9	mean	58.68141	69.57591	26.01816	0	108.683	0.464705	0	170.1913	0	0	0
	best	32.80754	32.85466	13.92943	0	75.69692	0	0	98.82944	0	0	0
	worst	79.7012	114.4199	40.79327	0	166.2595	5.02123	0	249.7671	0	0	0
	std	12.70118	20.94823	6.554449	0	24.78061	1.310176	0	44.86406	0	0	0
	med	56.30895	65.84376	25.37144	0	100.1247	0	0	169.0233	0	0	0
	rank	4	5	3	1	6	2	1	7	1	1	1
F_{10}	mean	3.659085	2.869241	8.55E-09	4.26E-15	0.940669	1.72E-14	4.09E-15	1.520055	4.26E-15	8.88E-16	8.88E-16
	best	3.045616	0.978948	6.02E-09	8.88E-16	0.087874	1.15E-14	8.88E-16	1.51E-14	8.88E-16	8.88E-16	8.88E-16
	worst	4.366778	4.121509	1.33E-08	4.44E-15	2.915234	2.22E-14	7.99E-15	3.500347	4.44E-15	8.88E-16	8.88E-16
	std	0.411662	0.78912	1.91E-09	7.94E-16	0.938497	3.53E-15	2.55E-15	1.572911	7.94E-16	0	0
	median	3.720395	3.025406	8.15E-09	4.44E-15	0.703059	1.51E-14	4.44E-15	1.270002	4.44E-15	8.88E-16	8.88E-16
	rank	9	8	5	3	6	4	2	7	3	1	1
F_{11}	mean	1.524898	0.308563	8.934676	0	0.407293	0.00097	0.003062	0.008866	0	0	0
	best	1.251305	0.012446	5.463953	0	0.259112	0	0	0	0	0	0
	worst	1.791288	4.090669	17.88423	0	0.603916	0.019408	0.06124	0.017366	0	0	0
	std	0.143664	0.899622	3.180256	0	0.07769	0.00434	0.013694	0.005776	0	0	0
	med	1.488223	0.071877	7.923337	0	0.415251	0	0	0.00982	0	0	0
	rank	7	5	8	1	6	2	3	4	1	1	1
F_{12}	mean	0.155349	1.391328	0.21731	0.081032	0.807735	0.038726	0.024389	6.409807	1.64E-10	1.408487	1.57E-32
	best	0.042239	0.000558	4.3E-19	0.040371	0.001623	0.012946	0.000813	0.333781	7.55E-11	0.940108	1.57E-32
	worst	0.345347	3.204317	1.494657	0.138141	3.001919	0.105321	0.330719	13.96328	3.53E-10	1.629701	1.57E-32
	std	0.075924	1.03366	0.392116	0.025574	0.822807	0.027369	0.072623	3.761599	8.51E-11	0.24139	2.81E-48
	median	0.1459	1.500485	0.057647	0.0787	0.440093	0.029434	0.006351	5.533201	1.38E-10	1.51496	1.57E-32
	rank	6	9	7	5	8	4	3	11	2	10	1
F_{13}	mean	2.160287	3.08976	0.015448	0.905929	0.031928	0.593907	0.227661	3.0785	0.001674	0.26	1.35E-32
	best	0.993663	0.056865	6.03E-18	0.539664	0.010281	0.274715	0.039034	1.872547	1.24E-09	1.73E-31	1.35E-32
	worst	4.186692	12.9562	0.254017	1.577393	0.13647	1.105615	0.458168	3.93204	0.010987	2.9	1.35E-32
	std	0.701865	3.228647	0.057034	0.230028	0.026955	0.225443	0.143038	0.558013	0.004015	0.80616	2.81E-48
	med	2.194029	2.146609	1.23E-17	0.886316	0.026003	0.58815	0.239423	3.052745	3.12E-09	1.46E-30	1.35E-32
	rank	9	11	3	8	4	7	5	10	2	6	1
Sum rank		39	44	37	28	35	26	16	47	12	28	6
Mean rank		6.5	7.333333	6.166667	4.666667	5.833333	4.333333	2.666667	7.833333	2	4.666667	1
Total rank		9	10	8	6	7	4	3	11	2	5	1

Table 3. Evaluation results on high-dimensional multimodal functions.

		GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	STBO
F_{14}	mean	0.998102	3.212919	3.564456	0.998007	0.998004	4.221422	3.88582	6.56528	0.998004	4.469522	0.998004
	best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	1.002309	0.998004
	worst	0.998721	10.76318	8.840836	0.998034	0.998004	12.67051	10.76318	12.67051	0.998004	10.76318	0.998004
	std	0.000213	2.88548	2.189673	6.96E-06	7.3E-12	4.652866	4.146898	4.89171	1.82E-10	3.084637	0
	median	0.998004	2.487068	2.806896	0.998004	0.998004	1.990054	1.495017	4.948548	0.998004	2.982105	0.998004
	rank	5	6	7	4	2	9	8	11	3	10	1
F_{15}	mean	0.01273	0.001638	0.002156	0.003375	0.007576	0.001357	0.000693	0.008509	0.000307	0.001305	0.000307
	best	0.000767	0.000307	0.000923	0.000309	0.000336	0.000307	0.000308	0.000308	0.000307	0.000727	0.000307
	worst	0.026092	0.020363	0.00352	0.020364	0.020363	0.020363	0.002252	0.020942	0.000307	0.002601	0.000307
	std	0.010579	0.004442	0.000483	0.007325	0.009629	0.004478	0.000523	0.010056	2.92E-19	0.000536	2.99E-19
	med	0.012594	0.000307	0.002076	0.00032	0.000755	0.000308	0.000475	0.000779	0.000307	0.001149	0.000307
	rank	11	6	7	8	9	5	3	10	2	4	1
F_{16}	mean	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.02846	-1.03163	-1.02844	-1.03163
	best	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03162	-1.03163
	worst	-1.03161	-1.03163	-1.03163	-1.03162	-1.03163	-1.03163	-1.03163	-1	-1.03163	-1	-1.03163
	std	4.37E-06	1.14E-16	1.25E-16	2.49E-06	4.97E-08	3.27E-09	5.93E-11	0.009735	2.1E-10	0.007256	7.75E-16
	median	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03067	-1.03163
	rank	6	1	1	7	5	4	2	8	3	9	1
F_{17}	mean	0.524411	0.539013	0.397887	0.422207	0.397887	0.397888	0.397888	0.397906	0.397887	0.638307	0.397887
	best	0.397887	0.397887	0.397887	0.39789	0.397887	0.397887	0.397887	0.397888	0.397887	0.397901	0.397887

	worst	2.791186	2.791184	0.397887	0.882291	0.397888	0.397894	0.397891	0.397971	0.397887	5.040108	0.397887
	std	0.534343	0.538701	0	0.108293	6.52E-08	1.63E-06	8.4E-07	2.48E-05	2.71E-09	1.036187	0
	med	0.397892	0.397887	0.397887	0.397972	0.397887	0.397888	0.397887	0.397894	0.397887	0.402551	0.397887
	rank	8	9	1	7	3	5	4	6	2	10	1
F_{18}	mean	5.729191	3	3	3.000001	3	3.000011	3.000005	15.15005	3	6.169642	3
	best	3.000044	3	3	3	3	3	3	3	3	3	3
	worst	30.53809	3	3	3.000003	3.000001	3.000033	3.000032	84.00069	3	39.23578	3
	std	8.39291	2.76E-15	2.92E-15	8.96E-07	2.54E-07	9.56E-06	8.89E-06	29.67426	5.39E-14	9.865771	1.81E-16
	median	3.001628	3	3	3	3	3.000009	3.000001	3.000008	3	3.000086	3
	rank	9	2	3	6	5	8	7	11	4	10	1
F_{19}	mean	-3.86228	-3.86278	-3.86278	-3.86203	-3.86278	-3.86096	-3.8602	-3.86273	-3.86278	-3.80846	-3.86278
	best	-3.86278	-3.86278	-3.86278	-3.8627	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	-3.85487	-3.86278
	worst	-3.85745	-3.86278	-3.86278	-3.8548	-3.86278	-3.8549	-3.85378	-3.86256	-3.86278	-3.68429	-3.86278
	std	0.001431	2.09E-15	1.87E-15	0.001716	7.41E-08	0.002948	0.002953	5.09E-05	2.28E-15	0.04789	2.78E-15
	med	-3.86277	-3.86278	-3.86278	-3.86249	-3.86278	-3.86246	-3.86139	-3.86275	-3.86278	-3.82667	-3.86278
	rank	4	1	1	5	2	6	7	3	1	8	1
F_{20}	mean	-3.19552	-3.29822	-3.322	-3.23822	-3.2446	-3.23965	-3.2753	-3.23237	-3.322	-2.63831	-3.322
	best	-3.3214	-3.322	-3.322	-3.31043	-3.32199	-3.32199	-3.32198	-3.32137	-3.322	-3.15625	-3.322
	worst	-2.99692	-3.2031	-3.322	-3.08169	-3.20259	-3.02064	-3.10782	-2.84	-3.322	-1.30322	-3.322
	std	0.093531	0.048793	3.95E-16	0.065246	0.058264	0.095719	0.074262	0.146905	4.2E-16	0.417228	2.49E-16
	median	-3.18946	-3.322	-3.322	-3.1998	-3.20302	-3.26252	-3.32127	-3.31998	-3.322	-2.73165	-3.322
	rank	8	2	1	6	4	5	3	7	1	9	1
F_{21}	mean	-5.89083	-5.77879	-6.17737	-5.84595	-8.51163	-9.64743	-8.36636	-6.52198	-10.1532	-5.0552	-10.1532
	best	-9.0381	-10.1532	-10.1532	-9.44872	-10.1532	-10.1531	-10.1529	-10.138	-10.1532	-5.0552	-10.1532
	worst	-2.34247	-2.63047	-2.63047	-3.80037	-2.63047	-5.10034	-5.05374	-2.60298	-10.1532	-5.0552	-10.1532
	std	2.512564	3.703566	3.699126	1.769566	2.623726	1.55506	2.493026	3.337297	1.95E-15	2.48E-07	3.65E-15
	med	-6.83679	-2.68286	-3.51696	-5.02319	-10.1531	-10.1527	-10.1469	-5.04462	-10.1532	-5.0552	-10.1532
	rank	7	9	6	8	3	2	4	5	1	10	1
F_{22}	mean	-7.21825	-6.31807	-10.4029	-8.09591	-9.6056	-10.4024	-8.0395	-7.53629	-10.4029	-5.08767	-10.4029
	best	-10.1952	-10.4029	-10.4029	-9.92173	-10.4029	-10.4028	-10.4028	-10.3998	-10.4029	-5.08767	-10.4029
	worst	-2.62184	-1.83759	-10.4029	-4.21215	-5.08765	-10.4018	-2.76572	-1.82822	-10.4029	-5.08767	-10.4029
	std	2.472441	3.837031	2.97E-15	1.699295	1.947221	0.00028	3.034341	3.483042	3.65E-15	5.8E-07	2.88E-15
	median	-7.89012	-4.40599	-10.4029	-8.81648	-10.4029	-10.4025	-10.3974	-10.1566	-10.4029	-5.08767	-10.4029
	rank	8	9	2	5	4	3	6	7	1	10	1
F_{23}	mean	-5.78525	-5.62285	-10.5364	-8.30576	-9.99556	-9.72457	-8.77684	-5.46265	-10.5364	-5.12847	-10.5364
	best	-10.417	-10.5364	-10.5364	-9.98248	-10.5364	-10.5364	-10.5364	-10.4691	-10.5364	-5.12848	-10.5364
	worst	-2.38428	-2.42173	-10.5364	-4.06348	-5.12846	-2.42172	-2.42169	-1.67573	-10.5364	-5.12847	-10.5364
	std	2.966829	3.755817	1.73E-15	1.491856	1.664511	2.497522	3.196353	3.753624	2.51E-15	1.29E-06	6.93E-16
	med	-6.05259	-3.35328	-10.5364	-8.76487	-10.5364	-10.536	-10.5338	-2.84687	-10.5364	-5.12847	-10.5364
	rank	7	8	1	6	3	4	5	9	2	10	1
Sum rank		73	53	30	62	40	51	49	77	20	90	10
Mean rank		7.3	5.3	3	6.2	4	5.1	4.9	7.7	2	9	1
Total rank		9	7	3	8	4	6	5	10	2	11	1

Table 4. Evaluation results on fixed-dimensional multimodal functions.

Compared Algorithms	Test function type		
	Unimodal	High-Multimodal	Fixed-Multimodal
STBO vs. RSA	1.01E-24	1.97E-21	0.001816
STBO vs. MPA	1.01E-24	1.97E-21	3.29E-11
STBO vs. TSA	1.01E-24	1.97E-21	0.000299
STBO vs. WOA	1.01E-24	1.04E-14	7.98E-21
STBO vs. MVO	1.01E-24	1.97E-21	4.09E-13
STBO vs. GWO	1.01E-24	7.8E-16	5.01E-07
STBO vs. TLBO	2.44E-24	9.08E-09	0.358845
STBO vs. GSA	1.01E-24	1.31E-20	1.44E-34
STBO vs. PSO	1.01E-24	1.04E-14	6.4E-10
STBO vs. GA	3.64E-11	1.63E-11	1.78E-12

Table 5. Wilcoxon sum rank test results.

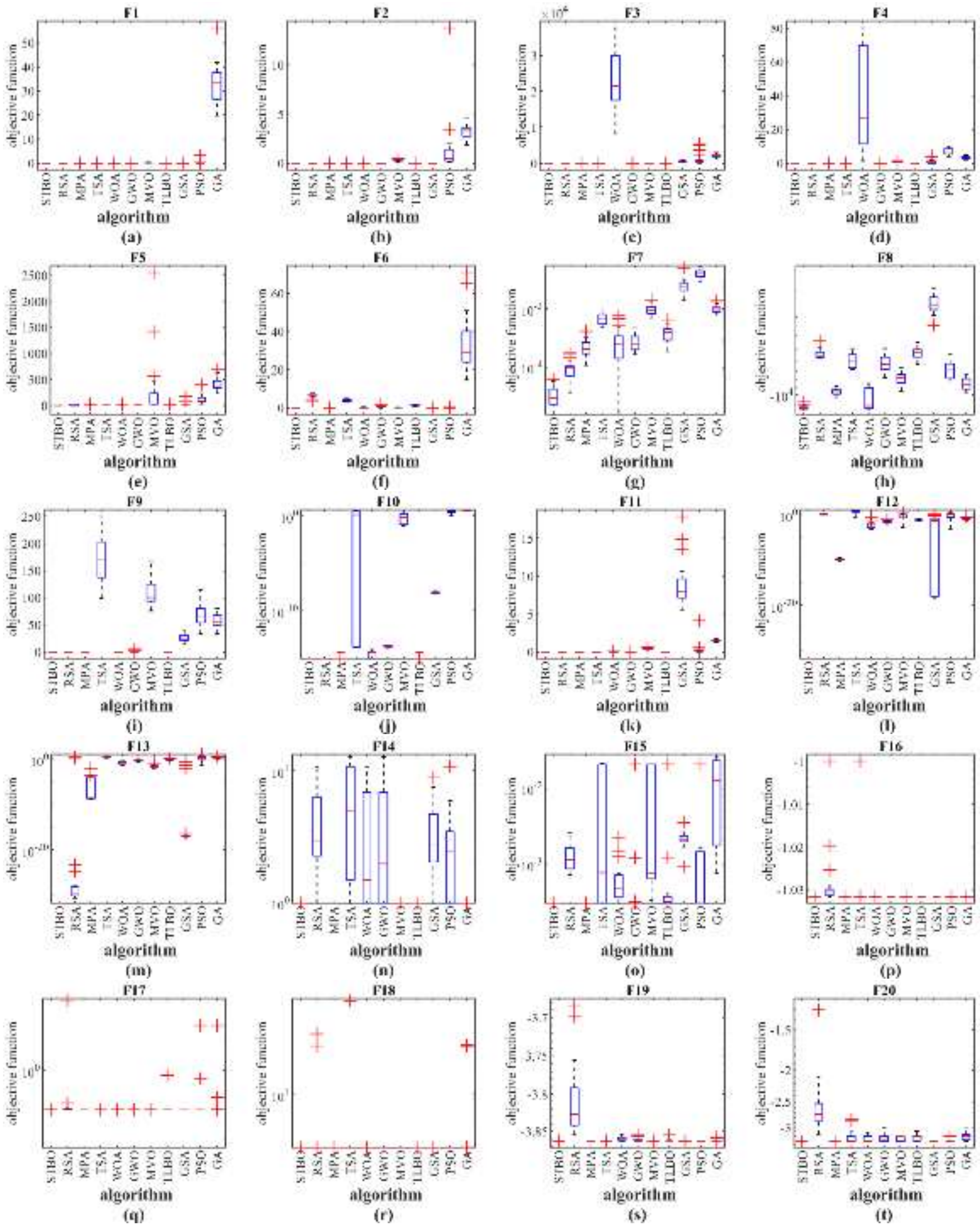


Figure 1. Boxplot of performance of STBO and competitor algorithms in solving F1 to F23.

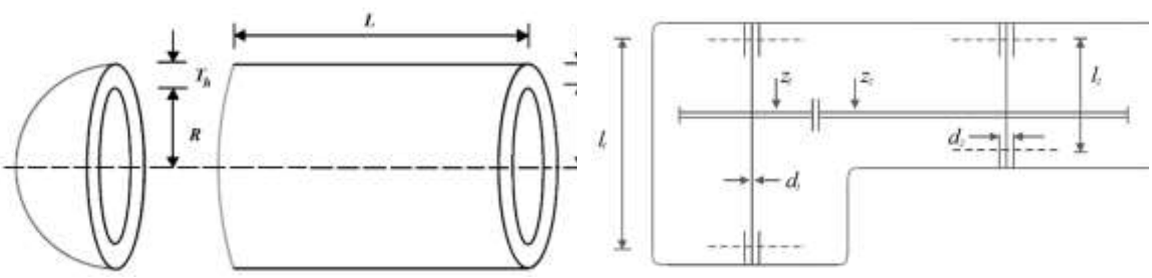
STBO for real-world applications

STBO's ability to optimize optimization problems in real-world applications is evaluated in this section. To this end, STBO and competitor algorithms have been implemented on four engineering optimization challenges. These engineering challenges are: pressure vessel design (PVD) [39], speed reducer design (SRD) [40], welded beam design (WBD) [12], and tension/compression spring design (TCSD) [12]. Schematics of these problems are presented in Figure 2.

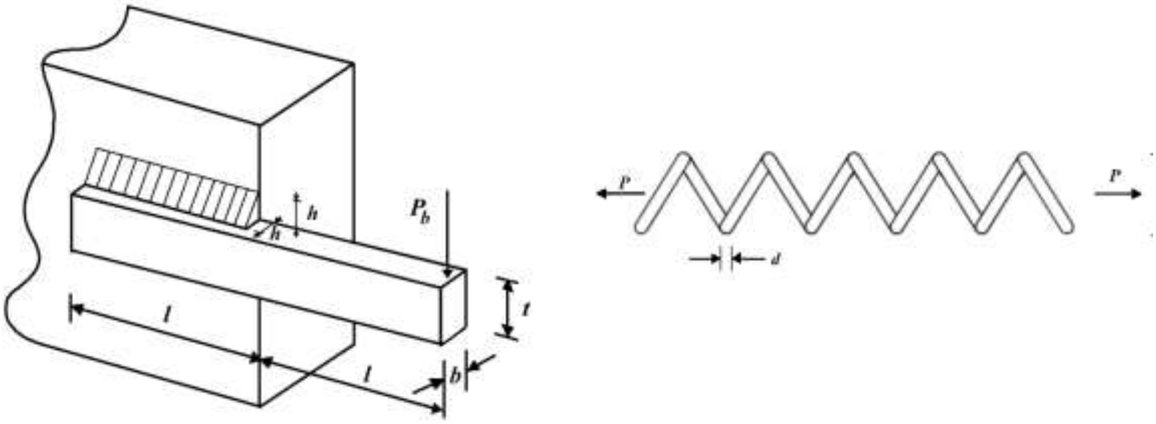
The optimization results of the four mentioned challenges are reported in Table 6. The simulation results show that STBO has superior performance to competitor algorithms in optimizing all four studied engineering challenges. What is clear from the analysis of the simulation results is that STBO has an effective capability in dealing with real-world optimization applications. The convergence curves of STBO while optimizing the mentioned optimization challenges are presented in Figure 3.

		GA	PSO	GSA	TLBO	MVO	GWO	WOA	TSA	MPA	RSA	STBO
PVD	mean	6645.562	6265.49	6842.164	6328.261	6478.841	6066.455	5892.921	5888.84	6117.763	6038.652	5888.170
	best	6581.043	5918.224	11605	6166.438	6039.984	5919.289	5917.261	5913.451	6109.88	6031.364	5884.882
	worst	8007.337	7007.411	7160.988	6513.898	7252.635	7396.34	5896.021	5893.718	6129.069	6042.513	5895.379
	std	657.679	496.2457	5791.998	126.639	327.0846	66.63439	13.91331	28.93686	38.24161	31.18698	23.71639
	median	7587.808	6114.139	6839.254	6319.815	6398.997	6417.635	5892.046	5887.624	6115.578	6036.744	5887.907
	rank	10	7	11	8	9	5	3	2	6	4	1
SRD	mean	3190.666	3174.457	3069.904	3032.78	3109.29	3009.754	3003.541	3011.73	3001.864	3000.171	3000.029
	best	3070.629	3054.173	3033.594	3005.931	3008.769	3004.291	3001.55	3004.837	2996.216	2996.171	2995.39
	worst	3317.508	3368.247	3108.816	3064.938	3215.349	3012.665	3007.795	3027.316	3007.093	3002.173	3001.627
	std	17.14086	92.69298	18.0977	13.03553	79.74166	5.845531	1.934443	10.36808	5.219098	2.015032	1.623719
	med	3202.346	3160.857	3069.595	3030.968	3109.29	3008.426	3003.087	3010.34	3000.431	2999.836	2999.061
	rank	11	10	8	7	9	5	4	6	3	2	1
WBD	mean	1.36595	2.123005	2.54876	1.820886	1.732754	2.234273	1.730198	1.728896	1.892096	1.725025	1.724605
	best	1.83841	1.876176	2.175414	1.761242	1.727502	1.822536	1.729027	1.727691	1.866157	1.727296	1.723127
	worst	2.038864	2.324247	3.008994	1.876738	1.744746	3.053648	1.730634	1.729132	2.016418	1.727726	1.726692
	std	0.139733	0.034882	0.256314	0.027592	0.004875	0.325102	0.001159	0.000287	0.00796	0.005124	0.004324
	median	1.939188	2.100775	2.499548	1.823362	1.73049	2.248652	1.730157	1.728855	1.883578	1.725997	1.72388
	rank	2	9	11	7	6	10	5	4	8	3	1
TCSD	mean	0.013192	0.014166	0.013564	0.01296	0.014599	0.014956	0.012816	0.012803	0.013898	0.0128	0.012674
	best	0.012889	0.013151	0.012987	0.012822	0.01293	0.013309	0.01279	0.012786	0.013218	0.012768	0.012652
	worst	0.015356	0.016403	0.014345	0.01312	0.018006	0.018029	0.01284	0.012834	0.01583	0.012812	0.012683
	std	0.000378	0.002092	0.000289	0.007831	0.001637	0.002293	0.004193	0.005671	0.006141	0.007417	0.001021
	med	0.013073	0.013123	0.013492	0.012965	0.014151	0.013316	0.012819	0.012806	0.013776	0.01279	0.012671
	rank	6	9	7	5	10	11	4	3	8	2	1
Sum rank		29	35	37	27	34	31	16	15	25	11	4
Mean rank		7.25	8.75	9.25	6.75	8.5	7.75	4	3.75	6.25	2.75	1
Total rank		7	10	11	6	9	8	4	3	5	2	1

Table 6. Evaluation results of four real-world applications.

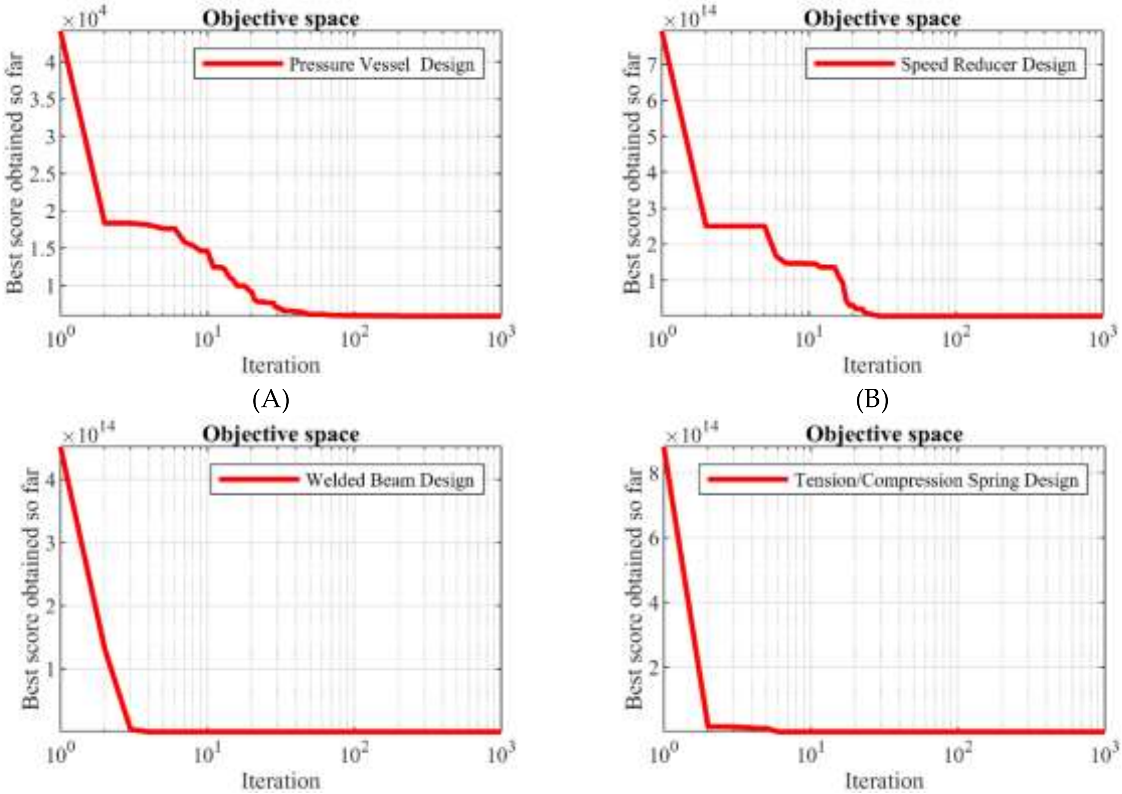


(A) (B)



(C) (D)

Figure 2. Schematics of four real-world applications: (A) PVD, (B) SRD, (C) WBD, (D) TCSD.



(C) (D)

Figure 3. Convergence curves of STBO on four real-world applications.

Conclusion and future works

This paper introduced a new optimization algorithm called Sewing Training-Based Optimization (STBO) to solve optimization problems. The interactions between the training instructor and the beginner tailors are the main inspiration in the design of STBO. The proposed STBO was modeled and designed in three phases: (i) training, (ii) imitation of the instructor's skills, and (iii) practice. The optimization ability of STBO was tested on twenty-three objective functions of unimodal and multimodal types. The optimization of unimodal functions showed the STBO exploitation ability in local search. The results of the multimodal functions indicated the STBO exploration power in the global search and the identification of the main optimal area. Ten well-known metaheuristic algorithms were employed to compare the performance of STBO in optimization. The simulation results showed that STBO has superior and competitive performance compared to some well-known metaheuristic algorithms, providing better results in most of the objective functions studied in this paper. STBO implementation on four engineering design challenges demonstrated the capability of the proposed algorithm in real-world applications.

Introducing the STBO activates several research tasks for future studies. Developing binary and multimodal versions is one of the most specific STBO research proposals. Employing STBO in various applications of optimization in science as well as in real-world applications are other suggestions for further studies.

Funding information

This work was supported by the Project of Specific Research, Faculty of Science, University of Hradec Králové, No. 2104/2022.

Author contribution statements

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

Competing interest

The authors declare that they have no competing interests.

Informed Consent

Informed consent was not required as no human or animals were involved.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Data availability

All data generated or analyzed during this study are included directly in the text of this submitted manuscript. There are no additional external files with datasets.

Additional information

The authors declare that no experiments on humans have been carried out in connection with this manuscript and therefore no human data have been generated in our research. Correspondence and requests for materials should be addressed to E.T.

References

[1] Ray, T. & Liew, K.-M. Society and civilization: an optimization algorithm based on the simulation of social behavior," *IEEE Transactions on Evolutionary Computation* **7**, 386-396 (2003).

[2] Kaidi, W., Khishe, M. & Mohammadi, M. Dynamic Levy Flight Chimp Optimization. *Knowledge-Based Systems* **235**, 107625 (2022).

[3] Sergeyev, Y. D., Kvasov, D. & Mukhametzhanov, M. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Scientific reports* **8**, 1-9 (2018).

[4] Goldberg, D. E. & Holland, J. H. Genetic Algorithms and Machine Learning. *Machine Learning* **3**, 95-99, (1988).

[5] Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proc. ICNN'95—International Conference on Neural Networks*, 1942–1948 (IEEE, 1995).

[6] Dorigo, M. & Stützle, T. *Handbook of Metaheuristics*, chap. Ant Colony Optimization: Overview and Recent Advances, 311–351 (Cham: Springer International Publishing, 2019)..

[7] Karaboga, D. & Basturk, B. 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, 789–798 (Springer, 2007).

[8] Wang, J.-S. & Li, S.-X. An improved grey wolf optimizer based on differential evolution and elimination mechanism. *Scientific reports* **9**, 1-21 (2019).

[9] Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1**, 67-82 (1997).

[10] Yang, X.-S. Firefly algorithms for multimodal optimization. In Stochastic Algorithms: Foundations and Applications. SAGA 2009, 169–178 (Springer, Berlin, Heidelberg, 2009).

[11] Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey Wolf Optimizer. *Advances in Engineering Software* **69**, 46-61 (2014).

[12] Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Advances in Engineering Software* **95**, 51-67 (2016).

[13] Faramarzi, A., Heidarinejad, M., Mirjalili, S. & Gandomi, A. H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications* **152**, 113377 (2020).

[14] Dehghani, M., Hubálovský, Š. & Trojovský, P. Cat and Mouse Based Optimizer: A New Nature-Inspired Optimization Algorithm. Sensors **21**, 5214 (2021).

[15] Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W. & Gandomi, A. H. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications* **191**, 116158 (2022).

[16] Jiang, Y. , Wu, Q., Zhu, S. & Zhang, L. Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems. *Expert Systems with Applications* **188**, 116026 (2022).

[17] Storn, R. & Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* **11**, 341-359 (1997).

[18] Eskandar, H., Sadollah, A., Bahreininejad, A. & Hamdi, M. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures* **110**, 151-166 (2012).

[19] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. GSA: a gravitational search algorithm. *Information sciences* **179**, 2232-2248 (2009).

[20] Dehghani, M., Montazeri, Z., Dhiman, G., Malik, O., Morales-Menendez, R., Ramirez-Mendoza, R. A., Dehghani, A., Guerrero, J. M. & Parra-Arroyo, L. A spring search algorithm applied to engineering optimization problems. *Applied Sciences* **10**, 6173 (2020).

[21] Dehghani, M. & Samet, H. Momentum search algorithm: A new meta-heuristic optimization algorithm inspired by momentum conservation law. *SN Applied Sciences* **2**, 1-15 (2020).

[22] Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671-680 (1983).

[23] Tahani, M. & Babayan, N. Flow Regime Algorithm (FRA): a physics-based meta-heuristics algorithm. *Knowledge and Information Systems* **60**, 1001-1038 (2019).

[24] Faramarzi, A., Heidarinejad, M., Stephens, B. & Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems* **191**, 105190 (2020).

[25] Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications* **27**, 495-513 (2016).

[26] Moghdani, R. & Salimifard, K. Volleyball premier league algorithm. *Applied Soft Computing* **64**, 161-185 (2018).

[27] Dehghani, M., Mardaneh, M., Guerrero, J. M., Malik, O. & Kumar, V. Football game based optimization: An application to solve energy commitment problem. *Int. J. Intell. Eng. Syst* **13**, 514-523 (2020).

[28] Kaveh, A. & Zolghadr, A. A novel meta-heuristic algorithm: Tug of War optimization. *Iran University of Science & Technology* **6**, 469-492 (2016).

[29] Zeidabadi, F. A. & Dehghani, M. POA: Puzzle Optimization Algorithm. *International Journal of Intelligent Engineering and Systems* **15**, 273-281 (2022).

[30] Rao, R. V., Savsani, V. J. & Vakharia, D. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design* **43**, 303-315 (2011).

[31] Dehghani, M., Mardaneh, M., Guerrero, J. M., Malik, O. P., Ramirez-Mendoza, R. A., Matas, J., Vasquez, J. C. & Parra-Arroyo, L. A new "Doctor and Patient" optimization algorithm: An application to energy commitment problem. *Applied Sciences* **10**, 5791 (2020).

[32] Dehghani, M. & Trojovský, P. Teamwork Optimization Algorithm: A New Optimization Approach for Function Minimization/Maximization. *Sensors,* vol. 21, no. 13, pp. 4567, 2021.

[33] Mousavirad, S. J. & Ebrahimpour-Komleh, H. Human mental search: a new population-based metaheuristic optimization algorithm. *Applied Intelligence* **47**, 850-887 (2017).

[34] Moosavi, S. H. S. & Bardsiri, V. K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Engineering Applications of Artificial Intelligence* **86**, 165-181 (2019).

[35] Dehghani, M., Mardaneh, M. & Malik, O. FOA:'Following'Optimization Algorithm for solving Power engineering optimization problems. *Journal of Operation and Automation in Power Engineering* **8**, 57-64 (2020).

[36] Trojovský, P. & Dehghani, M. A new optimization algorithm based on mimicking the voting process for leader selection. *PeerJ Computer Science* 8, e976 (2022).

[37] Yao, X., Liu, Y. & Lin, G. Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation* **3**, 82-102 (1999).

[38] Wilcoxon, F. Individual comparisons by ranking methods. *Biometr. Bull.* **1**, 80–83 (1945).

[39] Kannan, B. & Kramer, S. N. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design* **116**, 405-411 (1994).

[40] Mezura-Montes, E. & Coello, C.A.C. Useful infeasible solutions in engineering optimization with evolutionary algorithms. In *Advances in Artificial Intelligence (MICAI 2005). Lecture Notes in Computer Science*, 652–662 (Springer, 2005).