IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 7, 2022, accepted February 11, 2022, date of publication February 15, 2022, date of current version February 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151641

Tasmanian Devil Optimization: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm

MOHAMMAD DEHGHANI[1], **ŠTĚPÁN HUBÁLOVSKÝ**[2], **AND PAVEL TROJOVSKÝ**[1]
[1]Department of Mathematics, Faculty of Science, University of Hradec Králové, 50003 Hradec Králové, Czech Republic
[2]Department of Applied Cybernetics, Faculty of Science, University of Hradec Králové, 50003 Hradec Králové, Czech Republic

Corresponding author: Pavel Trojovský (pavel.trojovsky@uhk.cz)

This work was supported by the Excellence Project of Faculty of Science, University of Hradec Králové, under Grant 2210/2022-2023.

ABSTRACT In this paper, a new bio-inspired metaheuristic algorithm called Tasmanian Devil Optimization (TDO) is designed that mimics Tasmanian devil behavior in nature. The fundamental inspiration used in TDO is simulation of the feeding behavior of the Tasmanian devil, who has two strategies: attacking live prey or feeding on carrions of dead animals. The proposed TDO is described, then its mathematical modeling is presented. TDO performance in optimization is tested on a set of twenty-three standard objective functions. Unimodal benchmark functions have analyzed the TDO exploitation capability, while high-dimensional multimodal and fixed-exploitation multimodal benchmark functions have challenged the TDO exploration capability. The optimization results indicate the high ability of the proposed TDO in exploration and exploitation and create a proper balance between these two indicators to effectively solve optimization problems. Eight well-known metaheuristic algorithms are employed to analyze the quality of the obtained results from TDO. The simulation results show that the proposed TDO, with its strong performance, has a higher capability than the eight competitor algorithms and is much more competitive. For further analysis, TDO is tested in optimizing four engineering design problems. Implementation results show that TDO has an effective performance in solving real-world applications.

INDEX TERMS Bio-inspired, exploitation, exploration, feeding, optimization, optimization algorithm, Tasmanian devil.

I. INTRODUCTION

Optimization is the process of determining the best solution among several candidate solutions for a problem with respect to its constraints. Advances in science and technology have led to the emergence of new and complex optimization problems as well as more details in existing optimization problems [1]. Many of these problems have features such as unknown search space, discrete search space, non-derivative objective functions, high dimensions, and non-convexity. This has led to the failure of traditional methods and mathematical analysis to effectively solve real-world optimization problems [2]. Hence, researchers and scientists tend to introduce new optimization solving methods called metaheuristic algorithms that can solve optimization problems without the need for derivative information, based on random searcher agents and using random operators. Metaheuristic algorithms have become more popular than classical methods because of their advantages such as easy comprehension, easy implementation, good performance, ability to avoid local optimization, and application to optimization problems in various sciences [3].

Two important features in the search and solution finding process in metaheuristic algorithms are exploration and exploitation. Exploration is the concept of global search in the problem-solving space to analyze different areas of the search space and not get caught in the optimal local areas, while exploitation is the concept of locally search in the neighborhood of the obtained solution to find a better solution [4]. Metaheuristic algorithms to have acceptable performance in solving optimization problems, must have a good balance between exploration and exploitation [5]. Metaheuristic algorithms are able to identify the optimal area based on the exploration phase and then converge

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

towards the global optimal solution based on the exploitation phase.

Due to the random nature of the search process in metaheuristic algorithms, the solution they provide may not be the same as the global optimal. For this reason, the solution obtained from metaheuristic algorithms is called quasi-optimal solution [6]. The performance of metaheuristic algorithms is different due to their randomness and different nature in the search process. Thus, metaheuristic algorithms may offer different solutions in solving a given optimization problem [7]. This has led to the design of numerous metaheuristic algorithms by researchers to provide better quasi-optimal solutions to optimization problems.

Metaheuristic algorithms are stochastic optimization problem-solving methods that have been inspired by various natural phenomena, the laws of physics, biological sciences, the rules of the game, and other evolutionary phenomena. These algorithms first generate candidate solutions according to the constraints of the problem. Then, they improve these candidate solutions based on the algorithm update steps in an iterative-based process. The main difference between metaheuristic algorithms is in the same process of improving candidate solutions during algorithm iterations. Metaheuristic algorithms in a general category based on main inspiration can be divided into four groups: evolutionary-based, swarm-based, physics-based, and game-based algorithms.

Applying the biological sciences alongside the theory of natural selection and Darwin's theory of evolution have inspired the design of evolutionary-based algorithms. Genetic Algorithm (GA) [8] and Differential Evolution (DE) [9] can be considered as the most famous evolutionary algorithms. In the design of GA and DE, the random operators of selection, crossover, and mutation play a key role in updating the algorithm population. The mechanism of the human immune system in the face of disease has been a fundamental inspiration in the development of the Artificial Immune System (AIS) algorithm [10].

The natural behaviors of various species of animals, birds, aquatic animals, and other living things have paved the way for the development of swarm-based algorithms. Particle Swarm Optimization (PSO) [11] and Ant Colony Optimization (ACO) [12] are among the most familiar and widely used swarm-based algorithms. PSO has employed the natural behavior of swarm movement of birds or fish. ACO has modeled the natural behavior of ants in identifying the shortest path. The animals' strategy in hunting their prey in nature represents an optimization process. Simulations of these natural behaviors have been employed in the design of metaheuristics such as Whale Optimization Algorithm (WOA [13], Marine Predators Algorithm (MPA) [14], and Grey Wolf Optimization (GWO) [15]. Search behaviors of animals with access to food sources have led to the introduction of metaheuristics such as Artificial Bee Colony (ABC) [16] and Tunicate Swarm Algorithm (TSA) [17]. Some other swarm-based algorithms are Red Fox Optimization Algorithm (RFOA) [18], Raccoon Optimization Algorithm (ROA) [19], Crow Search Algorithm (CSA) [20], Teaching-Learning Based Optimization (TLBO) [21], and Grasshopper Optimization Algorithm (GOA) [22].

Modeling of various laws in physics and physical phenomena has been considered in the introduction of physics-based algorithms. Simulated Annealing (SA) algorithm is one of the most prominent physical algorithms that is inspired in metal melting operations by the process of melting and cooling materials [23]. The simulation of Newton's laws of motion with the use of physical forces has been effective in the development of optimizers. Gravitational Search Algorithm (GSA) [24] using gravitational force, Spring Search Algorithm (SSA) [25] using elastic force, and Momentum Search Algorithm (MSA) using momentum have been designed. Physical phenomena are the source of inspiration in the design of metaheuristics, such as Water Cycle Algorithm (WCA) [26] according to the water cycle phenomenon, Small-World Optimization Algorithm (SWOA) [27] according to the mechanism of small-world phenomenon, and Black Hole (BH) [28] according to observable fact of black hole phenomena. Some other physics-based algorithms are Nuclear Reaction Optimization (NRO) [29], Multi-Verse Optimizer (MVO) [30], Artificial Chemical Reaction Optimization Algorithm (ACROA) [31], Optics Inspired Optimization (OIO) [32], Equilibrium Optimizer (EO) [33], Atom Search Optimization (ASO) [34], and Electromagnetic Field Optimization (EFO) [35].

Simulation of rules and behavior of players in different games has led to the design of game-based algorithms. Football Game Based Optimization (FGBO) [36] and Volleyball Premier Ligue (VPL) [37] algorithms are game-based metaheuristics developed based on the simulation of club competitions during a sports season. The behavior of players in collecting points and winning based on the throwing mechanism is modeled on the design of Ring Toss Game Based Optimizer (RTGBO) [38] and Darts Game Optimizer (DGO [39].

The major research question in all studies of metaheuristic algorithms is whether, given the various algorithms that have been developed, there is still a need to introduce new algorithms. The No Free Lunch (NFL) theorem [40] answers this question that the strong performance of an algorithm in solving a set of optimization problems provides no guarantee of optimal performance in other problems. Therefore, the superiority of a particular algorithm in solving all optimization problems is hypothesis rejected. The NFL theorem provides a research path for scientists to design new metaheuristic algorithms to solve optimization problems more effectively. The NFL theorem motivated the authors of this paper to come up with a new metaheuristic algorithm to effectively solve optimization problems.

What is evident from all studies of literature review and its obtained best knowledge is that Tasmanian devil behavior simulation has not been employed in the design of any metaheuristic algorithm. However, the natural behavior of the Tasmanian devil during feeding represents an optimization process in achieving the main purpose of this animal, i.e., food source. This research gap prompted the authors to design a new optimizer by simulating the Tasmanian devil feeding strategy, which is discussed in the next section.

This paper introduces a new optimization algorithm called Tasmanian Devil Optimization (TDO) that can be applied to solve various science optimization problems. The scientific contribution of this research can be expressed as follows:

1. The novelty of this paper is in the design of the new TDO optimizer based on the simulation of the Tasmanian devil's natural behavior.
2. The fundamental inspiration of TDO is the Tasmanian devil feeding mechanism in two strategies of live prey hunting and carnivore eating.
3. The various stages of TDO are described and mathematically modeled.
4. TDO is evaluated by solving twenty-three benchmark functions including unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types.
5. TDO is used to solve four engineering design problems to evaluate its performance in real-world problems.
6. To analyze the capability of the proposed algorithm, the optimization results obtained from TDO are compared with eight well-known algorithms.

In the following, the paper is organized in such a way that in Section 2, the proposed TDO algorithm is introduced and modeled. Simulation studies and results are presented in Section 3. The capability of TDO in optimizing engineering design problems is analyzed in Section IV. Finally, in Section 5, conclusions and several research suggestions are presented.

II. TASMANIAN DEVIL OPTIMIZATION

In this section, the proposed metaheuristic Tasmanian Devil Optimization (TDO) is introduced and its mathematical modeling is presented.

A. INSPIRATION AND BEHAVIOR OF TAMANIAN DEVIL

The Tasmanian devil is a carnivorous and marsupial wild animal belonging to the family Dasyuridae that lives in the island state of Tasmania. A photo of the Tasmanian devil is shown in Figure 1. Tasmanian devils are opportunistic animals, and although they are able to hunt prey, they feed on carrion if present [41]. Tasmanian devil has two strategies for feeding. In the first strategy, if Tasmanian devil finds a carrion, it feeds on it. In the second strategy, it hunts and feeds on prey by attacking it.

The modeling of this Tasmanian devil feeding mechanism is used in the TDO design.

B. MATHEMATICAL MODELLING

In this subsection, the process and how to simulate the natural behavior of Tasmanian demons during feeding is described to design an optimizer.

FIGURE 1. Tasmanian devil (take from Wikimedia Commons - Tasmanian Devil (33295981294)).

The optimization process is how to achieve the optimal solution for an optimization issue. The analogy of this process in the life and behaviors of the Tasmanian devil is like access to food. In fact, just as in the optimization process, the goal is to find the optimal solution, in the Tasmanian devil's nutritional process, the goal is to find the food source. Two important principles in the optimization process are exploration in the comprehensive search the problem-solving space and exploitation in approaching the optimal solution. The Tasmanian devil's search behavior in finding food sources in different spaces, in fact, indicates the exploration index in the optimization process in order to identify the optimal area of search space. On the other hand, the chasing process between the Tasmanian devil and the prey that occurs in a limited area is similar to the exploitation index in the local search with the aim of converging to the optimal solution. This means that mathematical modeling of Tasmanian devil strategies to reach the food source is prone to designing an optimizer to achieve optimal solutions to optimization problems.

1) INITIALIZATION

The proposed TDO is a population-based stochastic algorithm whose searcher agents are Tasmanian devils. The initial population of these agents is generated randomly based on the constraints of the problem. Population members of TDO, who are searchers of problem-solving space, suggest candidate values for problem variables based on their position in the search space. So mathematically, each member of a population is a vector with the number of elements equal to the number of problem variables. As a result, the set of TDO members can be modeled using a matrix in (1).

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

were X is the population of Tasmanian devils, X_i is the ith candidate solution while $x_{i,j}$ is its candidate value for the jth variable, N is the number of searching Tasmanian devils, and m is the number of variables of given problems.

The objective function of problem can be computed by placing each of the candidate solutions in the values of the variables of the objective function. As a result, the values obtained for the objective function are modeled using a vector in (2).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1}, \tag{2}$$

where F is the vector of values of the objective function and F_i is the value of the objective function obtained by the ith candidate solution. The analysis of the values obtained for the objective function shows the quality of the candidate solutions. The candidate solution that leads to the calculation of the best value for the objective function is considered the best member of the population. The best member of the population is updated based on new values in each iteration.

The population updating process in TDO is modeled on two Tasmanian devil feeding strategies. It is possible for any Tasmanian devil to eat carrion or feed on prey hunting. In TDO, it is assumed that the probability of choosing any of these strategies is equal to 50%. According to this concept, in each iteration of the TDO, each Tasmanian devil is updated based on only one of these two strategies.

2) STRATEGY 1: FEEDING BY EATING CARRION (EXPLORATION PHASE)

Sometimes the Tasmanian devil prefers to feed on carrion in the area instead of hunting. There are other predatory animals living around the Tasmanian Devil, which hunt large prey and are unable to eat it all. Additionally, these animals may not be able to eat sufficiently from their prey until the Tasmanian devil arrives. In these cases, the Tasmanian devil prefers to feed on these carrions. Tasmanian devil behavior in scanning the habitat area to find carrion is similar to the algorithm search process in problem-solving space. This Tasmanian devil strategy actually demonstrates the power of TDO exploration in scanning different areas of the search space to identify the original optimal area. The concepts expressed in the Tasmanian devil strategy of eating carcasses are mathematically modeled using (3) to (5). In the TDO design, for each Tasmanian devil, the position of other population members in the search space is assumed to be carrion locations. Random selection of one of these situations is simulated in (3) so that the k'th population member is selected as the target carrion for the i'th Tasmanian devil. Therefore, k must be chosen randomly from 1 to N while the opposite is i.

$$C_i = X_k, \quad i = 1, 2, \ldots, N, \; k \in \{1, 2, \ldots, N | k \neq i\}, \tag{3}$$

where C_i is the selected carrion by ith Tasmanian devil.

Based on the selected carrion, a new position is calculated for the Tasmanian devil in the search space. In the Tasmanian devil motion simulation in this strategy, if the objective function value of the carrion is better, the Tasmanian devil moves toward that carrion, otherwise it moves away from that carrion. This Tasmanian devil movement strategy is simulated in (4). In the last step of the first strategy, after calculating the new position for Tasmanian devil, this position is accepted if the value of the objective function is better in this new position otherwise, Tasmanian devil remains in its previous position. This update step is modeled in (5).

$$x_{i,j}^{new,S1} = \begin{cases} x_{i,j} + r \cdot (c_{i,j} - I \cdot x_{i,j}), & F_{C_i} < F_i; \\ x_{i,j} + r \cdot (x_{i,j} - c_{i,j}), & otherwise, \end{cases} \tag{4}$$

$$X_i = \begin{cases} X_i^{new,S1}, & F_i^{new,S1} < F_i; \\ X_i, & otherwise, \end{cases} \tag{5}$$

Here, $X_i^{new,S1}$ is the new status of the ith Tasmanian devil based on the first strategy, $x_{i,j}^{new,S1}$ is its value for the jth variable, $F_i^{new,S1}$ is its objective function value, F_{C_i} is its objective function value of selected carrion, r is a random number in interval [0, 1], and I is a random number which can be 1 or 2.

3) STRATEGY 2: FEEDING BY EATING PREY (EXPLOITATION PHASE)

The Tasmanian Devil's second feeding strategy is to hunt and eat prey. Tasmanian devil behavior during the attack has two stages. In the first stage, by scanning the area, it selects the prey and attacks it. Then, in the second stage, after approaching the prey, it chases it to stop it and start eating. The modeling of the first stage is similar to the modeling of the first strategy, i.e., the selection of the carcass. Therefore, the first stage of prey selection and attack it is modeled using (6) to (8). In the second strategy, when updating the i'th Tasmanian devil, the position of other population members is assumed as preys location. The k'th population member is randomly selected as prey, while k is a natural random number between 1 to N and opposite i. The prey selection process is simulated in (6).

$$P_i = X_k, \quad i = 1, 2, \ldots, N, \; k \in \{1, 2, \ldots, N | k \neq i\}, \tag{6}$$

Here, P_i is the selected prey by the ith Tasmanian devil.

After determining the prey position, a new position is calculated for the Tasmanian devil. In calculating this new position, if the objective function value of the selected prey is better, the Tasmanian devil moves towards it, otherwise it moves away from that position. Modeling of this process is presented in (7). The new position calculated for the Tasmanian devil replaces the previous position if it improves the value of the target function. This step of the second strategy is modeled in (8).

$$x_{i,j}^{new,S2} = \begin{cases} x_{i,j} + r \cdot (p_{i,j} - I \cdot x_{i,j}), & F_{P_i} < F_i; \\ x_{i,j} + r \cdot (x_{i,j} - p_{i,j}), & otherwise, \end{cases} \quad (7)$$

$$X_i = \begin{cases} X_i^{new,S2}, & F_i^{new,S2} < F_i; \\ X_i, & otherwise, \end{cases} \quad (8)$$

Here, $X_i^{new,S2}$ is the new status of i'th Tasmanian based on the second strategy, $x_{i,j}^{new,S2}$ is its value for the jth variable, $F_i^{new,S2}$ is its objective function value, and F_{P_i} is its objective function value of selected prey.

The main difference between this strategy and the first strategy is the second stage and the simulation of prey chasing. The chase of prey in the vicinity of the attack site is similar to the local search of the search space. This Tasmanian devil behavior actually demonstrates the TDO's ability to exploit to converge to better candidate solutions. In order to simulate this chase process, the Tasmanian devil follows the prey in the neighborhood of the attacked place. The prey chase stage is modeled by the Tasmanian devil using (9) to (11). At this stage, the Tasmanian devil position is considered the center of a neighborhood where the prey chasing process takes place. The radius of this neighborhood indicates the range that the Tasmanian devil follows the prey, which can be calculated using (9). Thus, a new position based on the chasing process in this neighborhood can be calculated for the Tasmanian devil, which is mathematically simulated in (10). The new calculated position is acceptable to the Tasmanian devil if it provides a better value for the objective function than its previous position. This position update process is simulated for the Tasmanian devil in (11).

$$R = 0.01(1 - \frac{t}{T}), \quad (9)$$

$$x_{i,j}^{new} = x_{i,j} + (2r - 1) \cdot R \cdot x_{i,j}, \quad (10)$$

$$X_i = \begin{cases} X_i^{new}, & F_i^{new} < F_i; \\ X_i, & otherwise, \end{cases} \quad (11)$$

where R is the neighborhood radius of the point of attacked location, t is the iteration counter, T is the maximum number of iterations, X_i^{new} is the new status of the ith Tasmanian devil in neighborhood of X_i, $x_{i,j}^{new}$ is its value for the jth variable, and F_i^{new} is its objective function value.

4) REPETITIONS PROCESS, FLOWCHART, AND PSEUDO-CODE OF TDO

When the update of all TDO members is completed, the first iteration of the algorithm ends. New values are calculated for the position of Tasmanian devils and the objective function. After this, the algorithm enters the next iteration and the TDO population update process continues until the end of the algorithm iterations according to equations (3) to (11). TDO updates and stores the best candidate solution during these iterations. After the algorithm is fully implemented, TDO introduces the best candidate solution as the solution to the problem. The various steps of TDO are presented in flowchart format in Figure 2 and its pseudocode in Algorithm 1.

C. COMPUTATIONAL COMPLEXITY

This section analyzes the computational complexity of TDO. The computational complexity of TDO initialization is equal to $O(N \cdot m)$ where N is the number of members of the Tasmanian devil population and m is the number of problem variables. TDO has a problem-solving process in the number of repetitive T. The process of updating population members on their way to the carcass or prey has a computational complexity equal to $O(N \cdot m \cdot T)$. The prey chasing process in the second strategy has a computational complexity equal to $O(N_{S2} \cdot m \cdot T)$ where N_{S2} is the number of Tasmanian demons who have used the second feeding strategy. Thus, the total computational complexity of TDO is equal to $O((N \cdot m) \cdot ((1 + T) + (T \cdot N_{S2})))$.

III. SIMULATION STUDIES AND DISCUSSION

In this section, simulation studies of TDO performance in optimization are presented. TDO is employed to solve twenty-three standard benchmark functions, including seven unimodal functions, six high-dimensional multimodal functions, and ten fixed-dimensional unimodal functions [42]. The information of these benchmark functions is presented in the Appendix and in Tables 16 to 18. The performance quality of TDO is compared with eight well-known metaheuristic algorithms, TSA, MPA, WOA, GWO, TLBO, GSA, PSO, and GA. The values of the control parameters of these algorithms are specified in Table 1.

Each of the competitor algorithms and the proposed TDO is used in twenty independent executions to optimize the benchmark functions, while each execution contains 1000 iterations. In presenting the simulation results, "*avg*" is the average of the best obtained candidate solutions and "*std*" is the standard deviation of these values.

A. EVALUATION OF UNIMODAL TEST FUNCTION (F1-F7)

The selected unimodal functions F1 to F7 have only one main optimal solution. This feature has made unimodal functions suitable for evaluating the exploitation ability of optimization algorithms. The optimization results of F1 to F7 functions using TDO and eight competitor algorithms are presented in Table 2. The simulation results show that TDO with high exploitation power has been able to provide the global optimal solution for F6. TDO is also the first best optimizer in solving F1, F2, F3, F4, F5, and F7. The analysis of the results of this table shows that TDO has been able to provide much more competitive results compared to the eight competitor

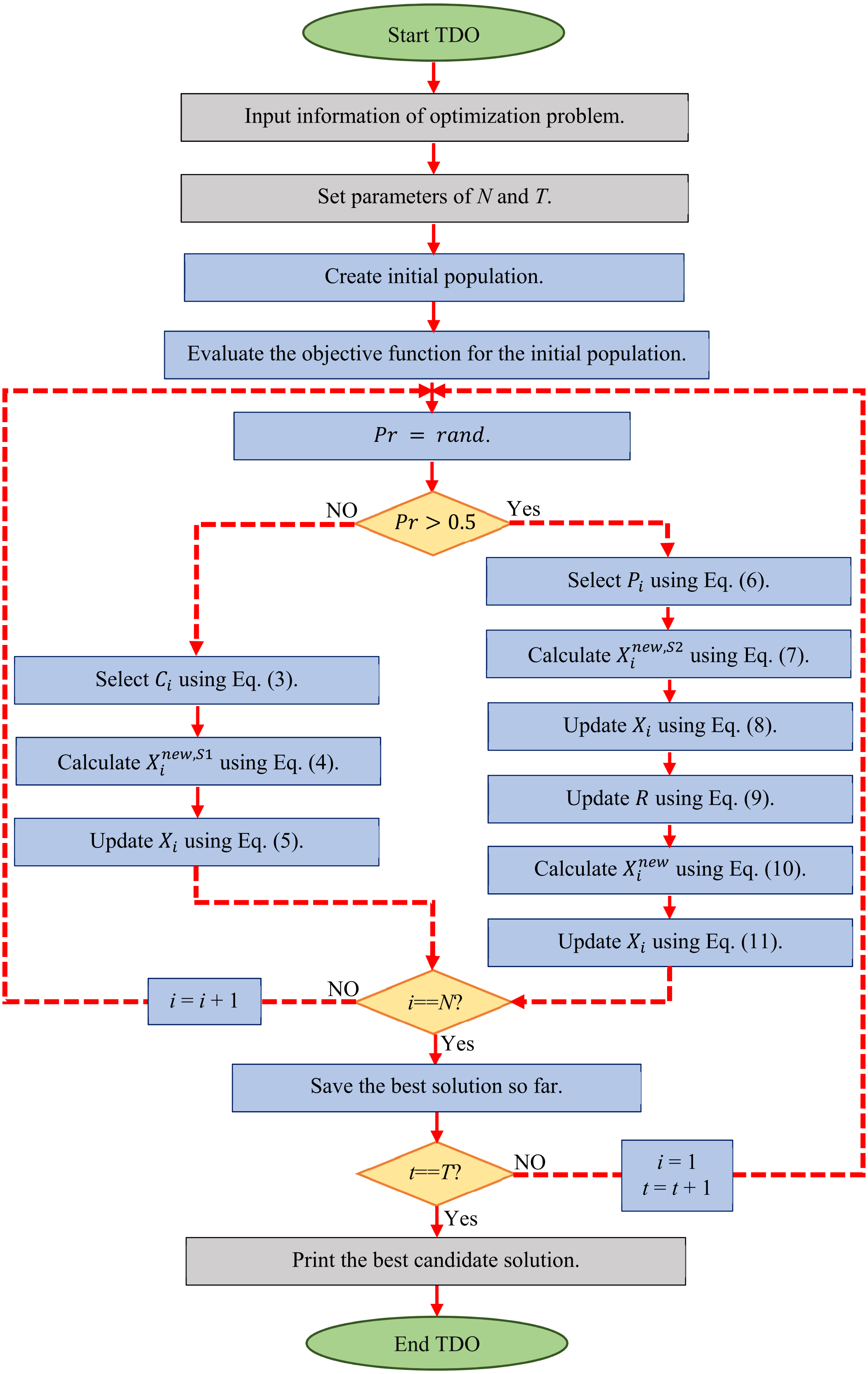

FIGURE 2. Flowchart of TDO.

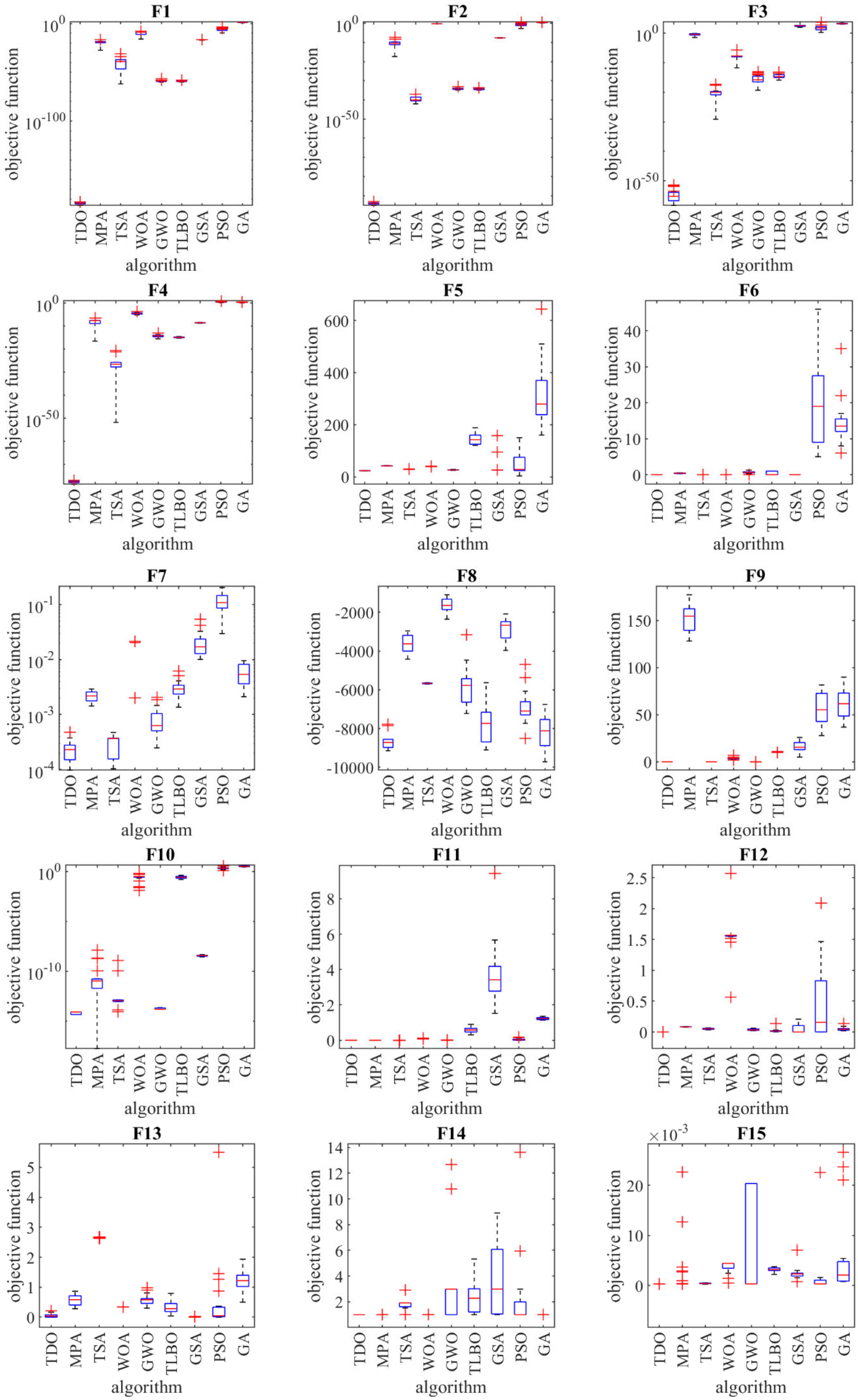

FIGURE 3. Boxplot of performance of TDO and eight competitor algorithms in solving test functions.

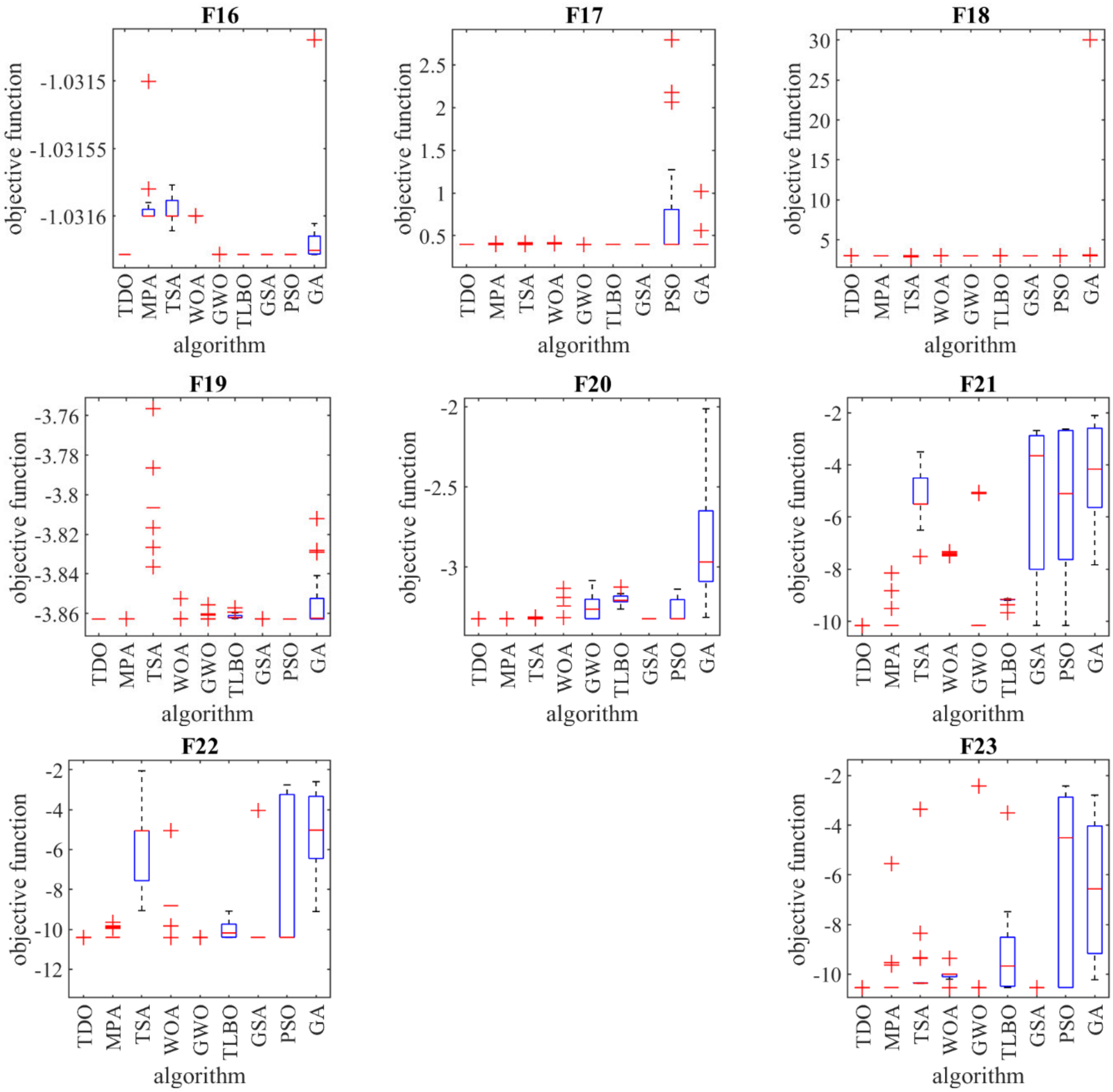

FIGURE 3. ***(Continued.)*** **Boxplot of performance of TDO and eight competitor algorithms in solving test functions.**

algorithms with high exploitation power, which shows the superiority of TDO.

B. EVALUATION OF HIGH-DIMENSIONAL MULTIMODAL TEST FUNCTION (F8-F13)

The selected high-dimensional multimodal functions have a large number of local optimal solutions. Therefore, optimization algorithms must have high exploration power in scanning the search space to find the original local optimization by passing through local optimal solutions. The implementation results of TDO and eight competitor algorithms on F8 to F13 functions are reported in Table 3. What is clear from the analysis of this table is that TDO, with its high exploration power, has provided the global optimal for F9 and F11 functions. TDO is also the best optimizer in solving F8, F10, F12, and F13. The simulation results show that TDO with high exploration ability is able to identify the main optimal area in the search space and has a superior and competitive performance compared to eight competitor algorithms.

C. EVALUATION OF FIXED-DIMENSIONAL MULTIMODAL TEST FUNCTION (F14-F23)

The selected fixed-dimensional multimodal functions have a small number of variables as well as a small number of local optimal solutions. These problems challenge the exploration ability of the optimization algorithms to discover the main optimal region of search space. The optimization results obtained from TDO and eight competitor algorithms in F14 to F23 optimization are presented in Table 4. Analysis of the results of this table shows that TDO with its high exploration power, has provided the global optimal for F14 and F17. TDO

Algorithm 1 Pseudo-Code of Proposed TDO Algorithm

Start TDO.
1. Input the optimization problem information.
2. Set the number of iterations (T) and the number of members of the population (N).
3. Initialization of the position of Tasmanian devils and evaluation of the objective function.
4. For $t = 1$:T
6. For $i = 1$:N
8. If Probability < 0.5, Probability $=$ rand
9. **Strategy 1: Feeding by eating carrion (exploration phase)**
10. Select carrion for the ith Tasmanian devil using Eq. (3).
11. Calculate new status of Tasmanian devil using Eq. (4).
12. Update the ith Tasmanian devil using (5).
13. else
14. **Strategy 2: Feeding by eating prey (exploitation phase)**
15. **Stage 1: Prey selection and attacking**
16. Select prey for the ith Tasmanian devil using (6).
17. Calculate new status of Tasmanian devil using Eq. (7).
18. Update the ith Tasmanian devil using (8).
19. **Stage 2: Prey chasing**
20. Update neighborhood radius using (9).
21. Calculate new status of the ith Tasmanian devil in neighborhood of X_i using (10).
22. Update the ith Tasmanian devil using (11).
23. end if
24. end for $i = 1$:N
25. Save the best proposed solution so far.
26. end for $t = 1$:T
27. Output: The best solution obtained by TDO for given optimization problem.

End TDO.

TABLE 1. Parameter values for the competitor algorithms.

Algorithm	Parameter	Value
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
TSA		
	P_{min} and P_{max}	1, 4
	c_1, c_2, c_3	random numbers from the interval $[0, 1]$.
WOA		
	Convergence parameter (a)	a: Linear reduction from 2 to 0.
	r is a random vector in $[0, 1]$.	
	l is a random number in $[-1,1]$.	
GWO		
	Convergence parameter (a)	a: Linear reduction from 2 to 0.
TLBO		
	T_F: teaching factor	$T_F = round\,[(1 + rand)]$
	random number	rand is a random number between $[0, 1]$.
GSA		
	Alpha, G_0, R_{norm}, R_{power}	20, 100, 2, 1
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
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)

has outperformed eight competitor algorithms in solving F15, F16, and F20. Also, the analysis of "*avg*" and "*std*" criteria, indicates the more effective performance of TDO in optimizing F18, F19, F21, F22, and F23. The simulation results of F14 to F23 functions show the superiority of TDO performance in providing optimal solutions compared to eight competitor algorithms.

The performance of TDO and eight competitor algorithms in optimizing benchmark functions is presented as a boxplot in Figure 3.

D. STATISTICAL ANALYSIS

Presentation of simulation results using "*avg*" and "*std*" criteria provides valuable information on the ability of optimization algorithms and their comparison. However, it is always possible, even with the slightest probability, that the superiority of one algorithm over another is a chance. In this regard, a statistical analysis is presented to examine whether the superiority of TDO has been significant or not against any of the competitor algorithms in this subsection. To provide statistical analysis on the performance of TDO and eight competitor algorithms, Wilcoxon rank sum test [43] has been used. In this test, a p-value is used to show the significant superiority of the corresponding algorithm over a competitor algorithm.

TABLE 2. Optimization results of TDO and competitor algorithms on unimodal test function.

		TDO	TSA	MPA	WOA	GWO	GSA	TLBO	GA	PSO
F_1	avg	2.74×10^{-187}	7.71×10^{-38}	3.27×10^{-21}	2.17×10^{-9}	1.09×10^{-58}	2.0255×10^{-17}	8.33×10^{-60}	13.2405	1.77×10^{-5}
	std	0	7.00×10^{-21}	4.61×10^{-21}	7.39×10^{-25}	5.14×10^{-74}	1.1369×10^{-32}	4.9436×10^{-76}	4.76×10^{-15}	6.43×10^{-21}
F_2	avg	7.11×10^{-96}	8.48×10^{-39}	1.57×10^{-12}	0.5462	1.29×10^{-34}	2.3702×10^{-8}	7.17×10^{-35}	2.4794	0.3411
	std	1.58×10^{-94}	5.92×10^{-41}	1.42×10^{-12}	1.73×10^{-16}	1.91×10^{-50}	5.1789×10^{-24}	6.69×10^{-50}	2.23×10^{-15}	7.44×10^{-17}
F_3	avg	5.15×10^{-59}	1.15×10^{-21}	0.0864	1.763×10^{-8}	7.40×10^{-15}	279.3439	2.75×10^{-15}	1536.8963	589.4920
	std	9.71×10^{-53}	6.70×10^{-21}	0.1444	1.03×10^{-23}	5.64×10^{-30}	1.2075×10^{-13}	2.64×10^{-31}	6.60×10^{-13}	7.11×10^{-13}
F_4	avg	2.39×10^{-79}	1.33×10^{-23}	2.60×10^{-8}	2.90×10^{-5}	1.25×10^{-14}	3.2547×10^{-9}	9.41×10^{-15}	2.0942	3.9634
	std	2.85×10^{-78}	1.15×10^{-22}	9.25×10^{-9}	1.21×10^{-20}	1.05×10^{-29}	2.0346×10^{-24}	2.11×10^{-30}	2.23×10^{-15}	1.98×10^{-16}
F_5	avg	22.8329	28.8615	46.049	41.7767	26.8607	36.10695	146.4564	310.4273	50.26245
	std	3.48×10^{-15}	4.76×10^{-3}	0.4219	2.54×10^{-14}	0	3.09×10^{-14}	1.90×10^{-14}	2.09×10^{-13}	1.58×10^{-14}
F_6	avg	0	7.10×10^{-21}	0.3980	1.60×10^{-9}	0.6423	0	0.4435	14.55	20.2500
	std	0	1.12×10^{-25}	0.1914	4.62×10^{-25}	6.20×10^{-17}	0	4.22×10^{-16}	3.17×10^{-15}	1.2564
F_7	avg	9.77×10^{-5}	3.72×10^{-4}	0.0018	0.0205	0.0008	0.0206	0.0017	5.67×10^{-3}	0.1134
	std	9.54×10^{-21}	5.09×10^{-5}	0.0010	1.55×10^{-18}	7.27×10^{-20}	2.72×10^{-18}	3.878×10^{-19}	7.75×10^{-19}	4.34×10^{-17}

TABLE 3. Optimization results of TDO and competitor algorithms on high-dimensional multimodal test function.

		TDO	TSA	MPA	WOA	GWO	GSA	TLBO	GA	PSO
F_8	avg	-8753.4765	−5740.3388	−3594.1632	−1663.9782	−5885.1172	−2849.0724	−7408.6107	−8184.4142	−6908.6558
	std	6.9870	41.5	811.32651	716.3492	467.5138	264.3516	513.5784	833.2165	625.6248
F_9	avg	0	5.70×10^{-3}	140.1238	4.2011	8.52×10^{-15}	16.2675	10.2485	62.4114	57.0613
	std	0	1.46×10^{-3}	26.3124	4.36×10^{-15}	5.64×10^{-30}	3.17×10^{-15}	5.56×10^{-15}	2.54×10^{-14}	6.35×10^{-15}
F_{10}	avg	4.44×10^{-15}	9.80×10^{-14}	9.6987×10^{-12}	0.3293	1.70×10^{-14}	3.56×10^{-9}	0.2757	3.2218	2.1546
	std	1.81×10^{-17}	4.51×10^{-12}	6.1325×10^{-12}	1.98×10^{-16}	2.75×10^{-29}	3.69×10^{-25}	2.56×10^{-15}	5.16×10^{-15}	7.94×10^{-16}
F_{11}	avg	0	1.00×10^{-7}	0	0.1189	0.0037	3.7375	0.6082	1.2302	0.0462
	std	0	7.46×10^{-7}	0	8.99×10^{-17}	1.26×10^{-18}	2.78×10^{-15}	1.98×10^{-16}	8.44×10^{-16}	3.10×10^{-18}
F_{12}	avg	3.13×10^{-11}	0.0368	0.0851	1.7414	0.0372	0.0362	0.0203	0.047	0.4806
	std	1.96×10^{-10}	1.54×10^{-2}	0.0052	8.13×10^{-12}	4.34×10^{-17}	6.20×10^{-18}	7.75×10^{-19}	4.65×10^{-18}	1.86×10^{-16}
F_{13}	avg	1.30×10^{-8}	2.9575	0.4901	0.3456	0.5763	0.002	0.3293	1.2085	0.5084
	std	2.06×10^{-16}	1.56×10^{-12}	0.1932	3.25×10^{-12}	2.48×10^{-16}	4.26×10^{-14}	2.11×10^{-16}	3.22×10^{-16}	4.96×10^{-17}

The simulation results obtained from the Wilcoxon rank sum test are presented in Table 5. What can be deduced from the analysis of the results of this test is that in cases where a p-value is less than 0.05, TDO has a significant statistical superiority over the competitor algorithm.

E. SENSITIVITY ANALYSIS

TDO is able to solve optimization problems in a repetition-based process based on search space scans by members of the population of Tasmanian devils. Thus, the number of population members of Tasmanian devils and the number of iterations of the algorithm affect the performance of the TDO. In this subsection, TDO sensitivity analysis to parameters N and T is studied.

To analyze the sensitivity to the parameter N, the proposed TDO is employed for the population size of Tasmanian devils equals to 20, 30, 50, and 100 in solving F1 to F23. The simulation results of the sensitivity analysis of TDO to the parameter N are presented in Table 5. TDO convergence curves in solving these functions and for different values of N are shown in Figure 4. What can be deduced from the analysis of the simulation results is that with the increase in the population of Tasmanian devils, the search power of TDO has improved and led to a decrease in the values of the objective functions.

To analyze the sensitivity to the parameter T, the proposed TDO algorithm for different values of T equal to 100, 500, 800, and 1000 is implemented on the benchmark functions F1 to F23. The results of TDO sensitivity analysis study under

TABLE 4. Optimization results of TDO and competitor algorithms on fixed-dimensional multimodal test function.

		TDO	TSA	MPA	WOA	GWO	GSA	TLBO	GA	PSO
F_{14}	avg	0.998	1.9923	0.998	0.998	3.7408	3.5913	2.2721	0.9986	2.1735
	std	0	2.65×10^{-7}	4.27×10^{-16}	9.43×10^{-16}	6.45×10^{-15}	7.94×10^{-16}	1.98×10^{-16}	1.56×10^{-15}	7.94×10^{-16}
F_{15}	avg	0.0003	0.0004	0.003	0.0049	0.0063	0.0024	0.0033	5.39×10^{-2}	0.0535
	std	2.27×10^{-16}	9.01×10^{-4}	4.09×10^{-15}	3.49×10^{-18}	1.16×10^{-18}	2.90×10^{-19}	1.22×10^{-17}	7.07×10^{-18}	3.87×10^{-19}
F_{16}	avg	-1.03163	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	std	1.97×10^{-16}	2.65×10^{-16}	4.46×10^{-16}	9.93×10^{-16}	3.97×10^{-16}	5.95×10^{-16}	1.43×10^{-15}	7.94×10^{-16}	3.47×10^{-16}
F_{17}	avg	0.3978	0.3991	0.3979	0.4047	0.3978	0.3978	0.3978	0.4369	0.7854
	std	0	2.15×10^{-16}	9.12×10^{-15}	2.48×10^{-17}	8.68×10^{-17}	9.93×10^{-17}	7.44×10^{-17}	4.96×10^{-17}	4.96×10^{-17}
F_{18}	avg	3	3	3	3	3	3	3.0009	4.3592	3
	std	1.85×10^{-16}	2.65×10^{-15}	1.95×10^{-15}	5.69×10^{-15}	2.08×10^{-15}	6.95×10^{-16}	1.58×10^{-15}	5.95×10^{-16}	3.67×10^{-15}
F_{19}	avg	-3.86278	−3.8066	−3.8627	−3.8627	−3.8621	−3.8627	−3.8609	−3.85434	−3.8627
	std	2.15×10^{-16}	2.63×10^{-15}	4.24×10^{-15}	3.19×10^{-15}	2.48×10^{-15}	8.34×10^{-15}	7.34×10^{-15}	9.93×10^{-17}	8.93×10^{-15}
F_{20}	avg	-3.322	−3.3206	−3.3211	−3.2424	−3.2523	−3.0396	−3.2014	−2.8239	−3.2619
	std	4.20×10^{-16}	5.69×10^{-15}	1.14×10^{-11}	7.94×10^{-16}	2.18×10^{-15}	2.18×10^{-14}	1.78×10^{-15}	3.972×10^{-16}	2.97×10^{-16}
F_{21}	avg	-10.1532	−5.5021	−10.1532	−7.4016	−9.6452	−5.1486	−9.1746	−4.3040	−5.3891
	std	2.19×10^{-16}	5.46×10^{-13}	2.53×10^{-11}	2.38×10^{-11}	6.55×10^{-15}	2.97×10^{-16}	8.53×10^{-15}	1.58×10^{-15}	1.48×10^{-15}
F_{22}	avg	-10.4029	−5.0625	−10.4029	−8.8165	−10.4025	−9.0239	−10.0389	−5.1174	−7.6323
	std	3.80×10^{-16}	8.46×10^{-14}	2.81×10^{-11}	6.75×10^{-15}	1.98×10^{-15}	1.64×10^{-12}	1.52×10^{-14}	1.29×10^{-15}	1.58×10^{-15}
F_{23}	avg	-10.5364	−10.3613	−10.5364	−10.0003	−10.1302	−8.9045	−9.2905	−6.5621	−6.1648
	std	3.36×10^{-16}	7.64×10^{-12}	3.98×10^{-11}	9.13×10^{-15}	4.56×10^{-15}	7.14×10^{-14}	1.19×10^{-15}	3.87×10^{-15}	2.78×10^{-15}

TABLE 5. *p*-values obtained from Wilcoxon rank sum test.

Compared Algorithms	Functions type		
	Unimodal	High-Multimodal	Fixed-Multimodal
TDO vs. MPA	0.015625	0.0625	0.01953125
TDO vs. TSA	0.015625	0.03125	0.00390625
TDO vs. WOA	0.015625	0.03125	0.0078125
TDO vs. GWO	0.015625	0.03125	0.01171875
TDO vs. TLBO	0.015625	0.03125	0.005859375
TDO vs. GSA	0.03125	0.03125	0.01953125
TDO vs. PSO	0.015625	0.03125	0.00390625
TDO vs. GA	0.015625	0.03125	0.001953125

the changes of parameter T are reported in Table 6. The behavior of TDO convergence curves under the influence of parameter T is presented in Figure 5. What is evident from the simulation results of the sensitivity analysis is that the increase in values T has led the algorithm to converge to better solutions and reduce the values of the objective functions.

IV. TDO APPLICATION FOR ENGINEERING DESIGN PROBLEMS

The performance of TDO in real-world applications is evaluated by optimizing four engineering design optimization problems including welded beam design, pressure vessel design, speed reducer design, and tension/compression spring design.

A. WELDED BEAM DESING OPTIMIZATION PROBLEM

Welded beam design is a minimization problem which its main purpose is to reduce the fabrication cost of welded beam [13]. A schematic of this problem is shown in Figure 6. The optimum values of the design variables and the values of the objective function using TDO and eight competitor algorithms are presented in Table 8. TDO provides the best candidate solution by providing the values of the design variables equal to (0.205730, 3.470521, 9.036603, 0.205731) and the corresponding objective function value equal to 1.724901. The statistical results of the performances of TDO and eight competitor metaheuristics are presented in Table 9. The simulation results show that TDO is superior to eight competitor algorithms by providing optimal performance. The convergence curve behavior of TDO in achieving the optimal solution for the welded beam design problem is shown in Figure 7.

B. PRESSURE VESSEL DESING OPTIMIZATION PROBLEM

Pressure vessel design is a minimization problem whose main purpose is to reduce the total cost of material, welding, and forming of a cylindrical vessel [44]. A schematic of this problem is shown in Figure 8. The implementation results of TDO and eight competitor algorithms in optimizing the pressure vessel design problem are presented in Table 10.

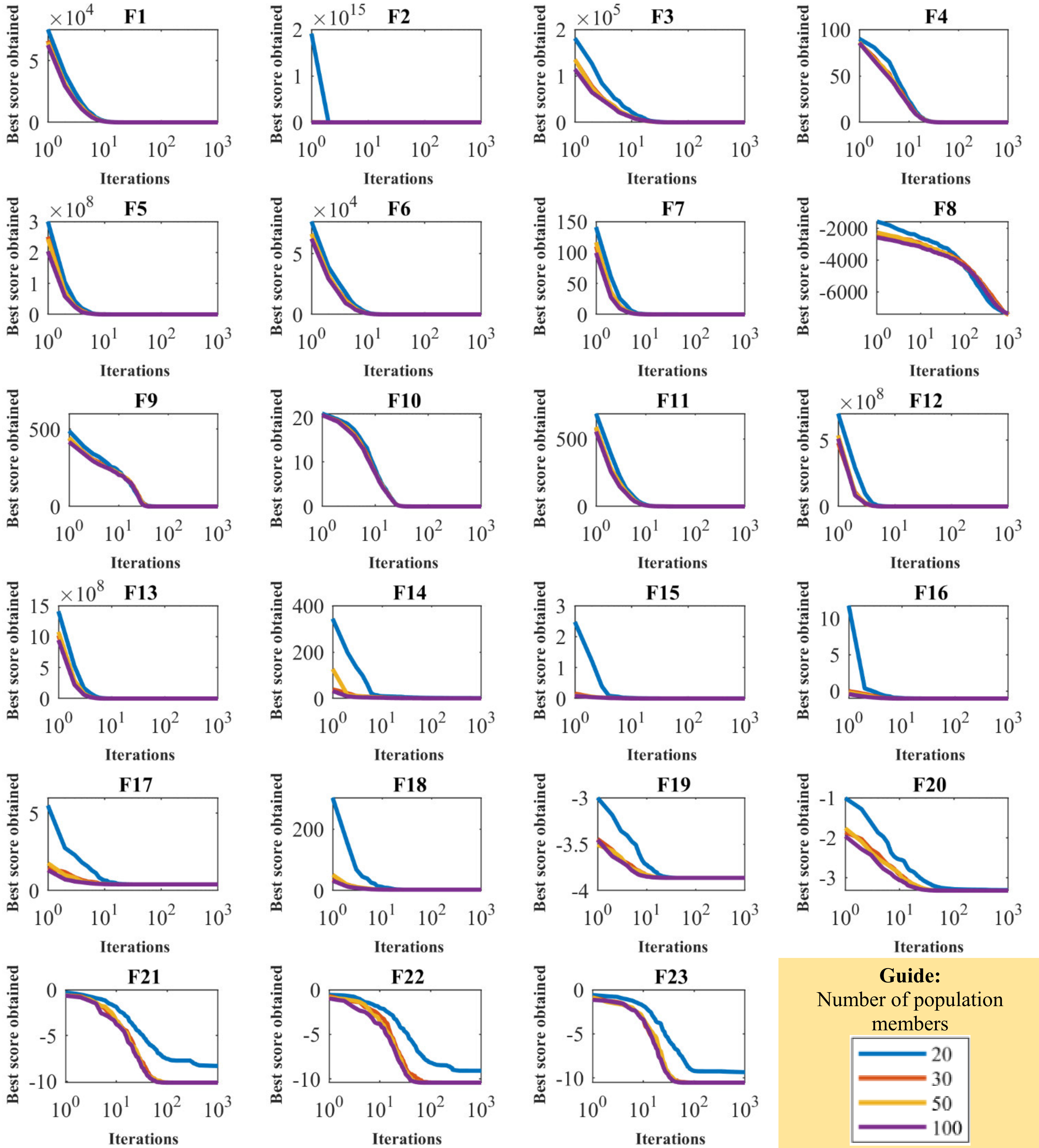

FIGURE 4. Sensitivity analysis of the TDO for the number of population members.

TDO provides the optimal 11 by providing better values for design variables equal to (0.7780535, 0.3860383, 40.31357, 199.9841) and the corresponding objective function value equal to 5887.1783. The statistical results obtained from the implementation of TDO and eight metaheuristics are presented in Table 11. The simulation results show the superiority of TDO in solving the pressure vessel design problem more effectively than eight competitor algorithms. The TDO convergence curve to optimize this problem is shown in Figure 9.

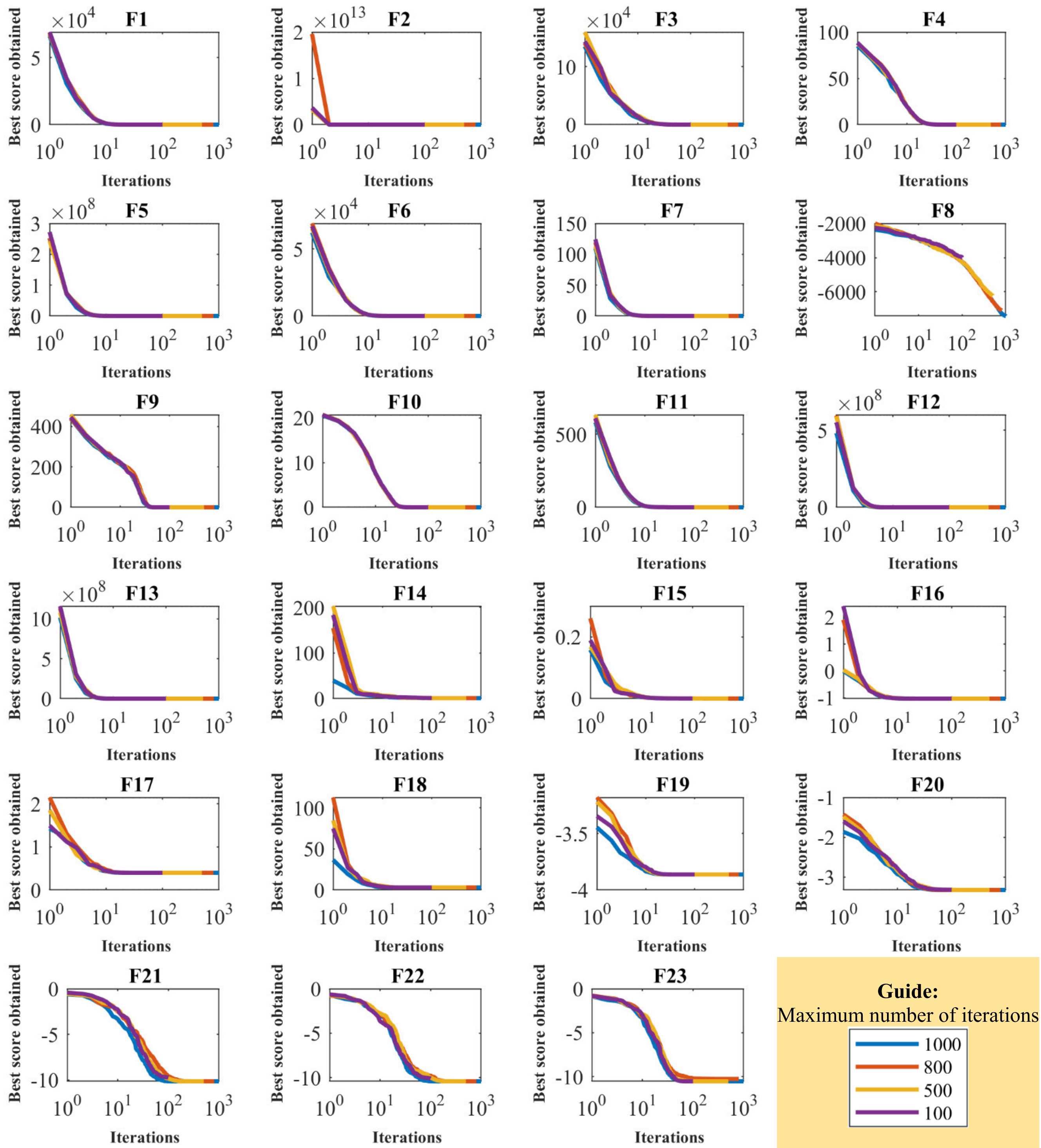

FIGURE 5. Sensitivity analysis of the TDO for the maximum number of iterations.

C. SPEED REDUCER DESING OPTIMIZATION PROBLEM

Speed reducer design is a minimization problem whose main purpose is to reduce the weight of the speed reducer [45], [46]. A schematic of this problem is shown in Figure 10. The application results of TDO and eight competitor metaheuristics in optimizing the speed reducer design problem are presented in Table 12. TDO has been able to provide the optimal solution to this problem with the values of the design variables equal to (3.5, 0.7, 17, 7.3, 7.8, 3.35021, 5.28668) and the corresponding objective function value equal to 2996.3482. The statistical results of the implementation of TDO and eight competitor metaheuristics

TABLE 6. Sensitivity analysis of the TDO for the number of population members.

Objective Function	Number of Population Members			
	20	30	50	100
F_1	3.1×10^{-191}	2.74×10^{-187}	6.4×10^{-184}	2.6×10^{-186}
F_2	5.8×10^{-100}	7.10×10^{-96}	2.19×10^{-94}	5.47×10^{-95}
F_3	9.34×10^{-63}	5.15×10^{-59}	4.24×10^{-50}	8.15×10^{-57}
F_4	$4.88\times 10\text{-}^{82}$	2.39×10^{-79}	3.65×10^{-78}	5.45×10^{-79}
F_5	26.38517	22.8323	23.78355	22.6415
F_6	0	0	0	0
F_7	0.000374	9.76×10^{-5}	7.47×10^{-5}	8.57×10^{-5}
F_8	-7380.31	-8753.4765	-8808.49	-8984.2
F_9	0	0	0	0
F_{10}	6.39×10^{-15}	4.44×10^{-15}	$5.33\times 10\text{-}^{15}$	4.44×10^{-15}
F_{11}	0	0	0	0
F_{12}	0.000722	3.1302×10^{-11}	2×10^{-10}	4.14×10^{-12}
F_{13}	1.20053	1.2988×10^{-8}	4.1856×10^{-9}	2.8×10^{-10}
F_{14}	1.635169	0.998	0.998004	0.998004
F_{15}	0.00131	0.0003	0.000307	0.000307
F_{16}	-1.03163	-1.03163	-1.03163	-1.03163
F_{17}	0.397887	0.3978	0.397887	0.397887
F_{18}	3	3	3	3
F_{19}	-3.86278	-3.86278	-3.86278	-3.86278
F_{20}	-3.30991	-3.322	-3.322	-3.322
F_{21}	-8.32308	-10.1532	-10.1532	-10.1532
F_{22}	-9.07412	-10.4029	-10.4029	-10.4029
F_{23}	-9.34421	-10.5364	-10.5364	-10.5364

TABLE 7. Sensitivity analysis of the TDO for the maximum number of iterations.

Objective Function	Maximum Number of Iterations			
	100	500	800	1000
F_1	1.74×10^{-14}	1.4×10^{-90}	1.8×10^{-147}	2.74×10^{-187}
F_2	2.46×10^{-8}	6.99×10^{-47}	3.18×10^{-76}	7.10×10^{-96}
F_3	0.036366	1.49×10^{-23}	3.09×10^{-41}	5.15×10^{-59}
F_4	1.31×10^{-6}	1.98×10^{-38}	5.34×10^{-63}	2.39×10^{-79}
F_5	28.53298	26.44308	25.16697	22.8323
F_6	0	0	0	0
F_7	0.003101	0.000585	0.00044	9.76×10^{-5}
F_8	-3994.9	-6218.77	-7075.22	-8753.4765
F_9	5.43×10^{-11}	0	0	0
F_{10}	2.49×10^{-8}	6.04×10^{-15}	6.04×10^{-15}	4.44×10^{-15}
F_{11}	8.03×10^{-14}	0	0	0
F_{12}	0.078086	6.1×10^{-5}	2.09×10^{-8}	3.1302×10^{-11}
F_{13}	1.238462	0.315307	0.117137	1.2988×10^{-8}
F_{14}	1.287459	0.998004	0.998004	0.998
F_{15}	0.000479	0.000308	0.000307	0.0003
F_{16}	-1.03163	-1.03163	-1.03163	-1.03163
F_{17}	0.397887	0.397887	0.397887	0.3978
F_{18}	3	3	3	3
F_{19}	-3.86278	-3.86278	-3.86278	-3.86278
F_{20}	-3.316	-3.322	-3.322	-3.322
F_{21}	-9.73376	-10.1532	-10.1532	-10.1532
F_{22}	-10.0821	-10.4029	-10.4029	-10.4029
F_{23}	-10.5364	-10.5364	-10.266	-10.5364

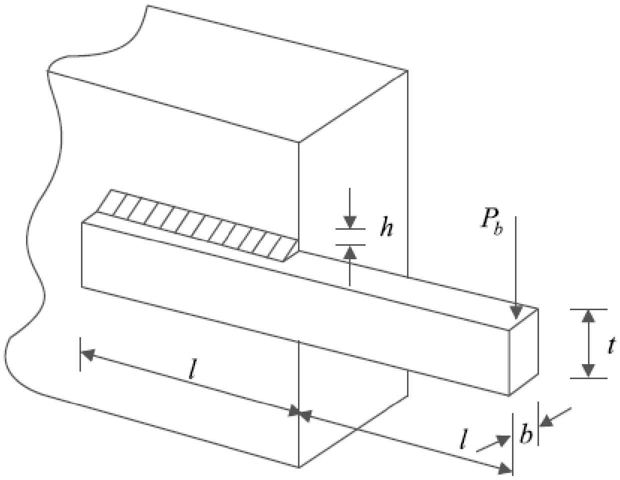

FIGURE 6. Schematic view of the welded beam design problem.

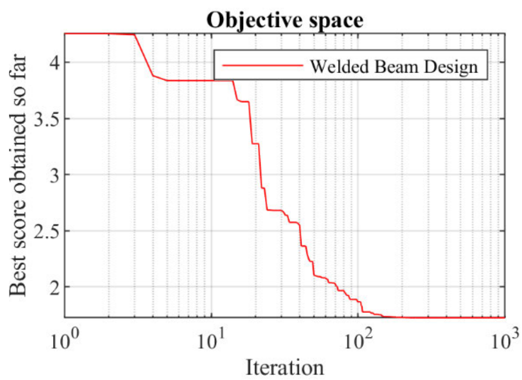

FIGURE 7. Convergence analysis of the TDO for the welded beam design optimization problem.

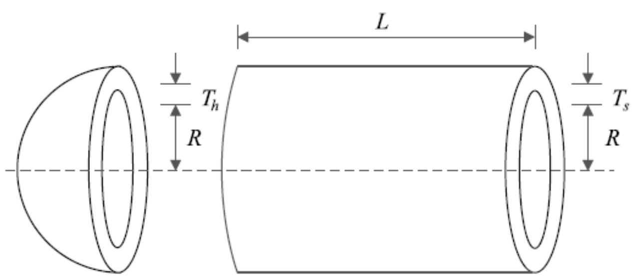

FIGURE 8. Schematic view of pressure vessel design problem.

are presented in Table 13. The simulation results show the superiority of TDO compared to eight competitor algorithms in minimizing the objective function of this problem. The TDO convergence curve during achieving the optimal solution is shown in Figure 11.

D. TENSION/COMPRESSION SPRING DESING OPTIMIZATION PROBLEM

Tension/compression spring design is a minimization problem whose main purpose is to reduce the tension/compression spring weight [13]. A schematic of this problem is shown

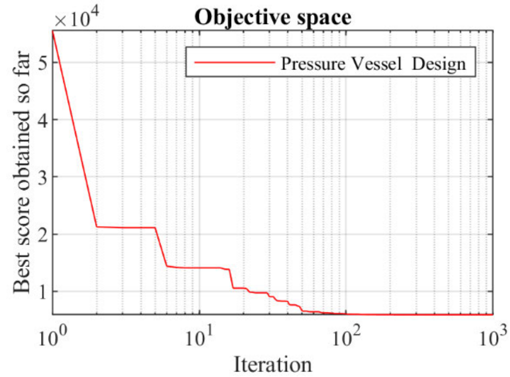

FIGURE 9. Convergence analysis of the TDO for the pressure vessel design optimization problem.

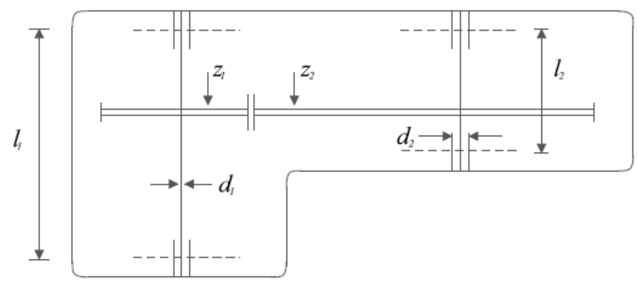

FIGURE 10. Schematic view of speed reducer design problem.

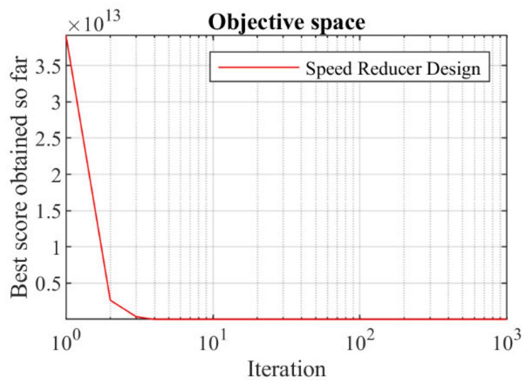

FIGURE 11. Convergence analysis of the TDO for the speed reducer design optimization problem.

FIGURE 12. Schematic view of tension/compression spring problem.

in Figure 12. The values obtained for the design variables and the objective function of this problem are presented in Table 14. TDO presents the optimal solution to the problem by providing the values of the design variables equal to (0.0518001, 0.359375, 11.1509) and the value of the objective function equal to 0.012671024. The statistical results

TABLE 8. Comparison results for the welded beam design problem.

Algorithm	Optimum Variables				Optimum Cost
	h	*l*	*t*	*b*	
TDO	0.205730	3.470521	9.036603	0.205731	1.724901
TSA	0.205563	3.474846	9.035799	0.205811	1.725661
MPA	0.205678	3.475403	9.036964	0.206229	1.726995
WOA	0.197411	3.315061	10.00000	0.201395	1.820395
GWO	0.205611	3.472103	9.040931	0.205709	1.725472
TLBO	0.204695	3.536291	9.004290	0.210025	1.759173
GSA	0.147098	5.490744	10.00000	0.217725	2.172858
PSO	0.164171	4.032541	10.00000	0.223647	1.873971
GA	0.206487	3.635872	10.00000	0.203249	1.836250

TABLE 9. Statistical results for the welded beam design problem.

Algorithm	Best	Mean	Worst	SD	Median
TDO	1.724901	1.725211	1.725395	0.0000104	1.725183
TSA	1.725661	1.725828	1.726064	0.000287	1.725787
MPA	1.726995	1.727128	1.727564	0.001157	1.727087
WOA	1.820395	2.230310	3.048231	0.324525	2.244663
GWO	1.725472	1.729680	1.741651	0.004866	1.727420
TLBO	1.759173	1.817657	1.873408	0.027543	1.820128
GSA	2.172858	2.544239	3.003657	0.255859	2.495114
PSO	1.873971	2.119240	2.320125	0.034820	2.097048
GA	1.836250	1.363527	2.035247	0.139485	1.9357485

TABLE 10. Comparison results for the pressure vessel design problem.

Algorithm	Optimum Variables				Optimum Cost
	T_s	T_h	R	L	
TDO	0.7780535	0.3860383	40.31357	199.9841	5887.1783
TSA	0.8303737	0.4162057	42.75127	169.3454	6048.7844
MPA	0.779035	0.384660	40.327793	199.65029	5889.3689
WOA	0.778961	0.384683	40.320913	200.00000	5891.3879
GWO	0.845719	0.418564	43.816270	156.38164	6011.5148
TLBO	0.817577	0.417932	41.74939	183.57270	6137.3724
GSA	1.085800	0.949614	49.345231	169.48741	11550.2976
PSO	0.752362	0.399540	40.452514	198.00268	5890.3279
GA	1.099523	0.906579	44.456397	179.65887	6550.0230

TABLE 11. Statistical results for the pressure vessel design problem.

Algorithm	Best	Mean	Worst	SD	Median
TDO	5887.1783	5890.0206	5892.1952	1.0215	5888.9142
TSA	6048.7844	6052.6241	6071.2496	2.893	6050.2282
MPA	5889.3689	5891.5247	5894.6238	13.910	5890.6497
WOA	5891.3879	6531.5032	7394.5879	534.119	6416.1138
GWO	6011.5148	6477.3050	7250.9170	327.007	6397.4805
TLBO	6137.3724	6326.7606	6512.3541	126.609	6318.3179
GSA	11550.2976	23342.2909	33226.2526	5790.625	24010.0415
PSO	5890.3279	6264.0053	7005.7500	496.128	6112.6899
GA	6550.0230	6643.9870	8005.4397	657.523	7586.0085

obtained from the optimization of the tension/compression spring design problem using TDO and eight competitor metaheuristics are presented in Table 15. The simulation results show that TDO has a superior performance compared to eight competitor algorithms in solving this problem. The convergence curve behavior of TDO in providing the optimal solution to the tension/compression spring design problem is shown in Figure 13.

TABLE 12. Comparison results for the speed reducer design problem.

Algorithm	Optimum Variables							Optimum Cost
	b	m	p	l_1	l_2	d_1	d_2	
TDO	3.5	0.7	17	7.3	7.8	3.35021	5.28668	2996.3482
TSA	3.50159	0.7	17	7.3	7.8	3.35127	5.28874	2998.5507
MPA	3.506690	0.7	17	7.380933	7.815726	3.357847	5.286768	3001.288
WOA	3.500019	0.7	17	8.3	7.8	3.352412	5.286715	3005.763
GWO	3.508502	0.7	17	7.392843	7.816034	3.358073	5.286777	3002.928
TLBO	3.508755	0.7	17	7.3	7.8	3.461020	5.289213	3030.563
GSA	3.600000	0.7	17	8.3	7.8	3.369658	5.289224	3051.120
PSO	3.510253	0.7	17	8.35	7.8	3.362201	5.287723	3067.561
GA	3.520124	0.7	17	8.37	7.8	3.366970	5.288719	3029.002

TABLE 13. Statistical results for the speed reducer design problem.

Algorithm	Best	Mean	Worst	SD	Median
TDO	2996.3482	2997.812	2999.6142	1.1642	2997.016
TSA	2998.5507	2999.640	3003.889	1.93193	2999.187
MPA	3001.288	3005.845	3008.752	5.83794	3004.519
WOA	3005.763	3105.252	3211.174	79.6381	3105.252
GWO	3002.928	3028.841	3060.958	13.0186	3027.031
TLBO	3030.563	3065.917	3104.779	18.0742	3065.609
GSA	3051.120	3170.334	3363.873	92.5726	3156.752
PSO	3067.561	3186.523	3313.199	17.1186	3198.187
GA	3029.002	3295.329	3619.465	57.0235	3288.657

TABLE 14. Comparison results for the tension/compression spring design problem.

Algorithm	Optimum Variables			Optimum Cost
	d	D	P	L
TDO	0.0518001	0.359375	11.1509	0.012671024
TSA	0.051144	0.343751	12.0955	0.012674000
MPA	0.050178	0.341541	12.07349	0.012678321
WOA	0.05000	0.310414	15.0000	0.013192580
GWO	0.05000	0.315956	14.22623	0.012816930
TLBO	0.050780	0.334779	12.72269	0.012709667
GSA	0.05000	0.317312	14.22867	0.012873881
PSO	0.05010	0.310111	14.0000	0.013036251
GA	0.05025	0.316351	15.23960	0.012776352

TABLE 15. Statistical results for the tension/compression spring design problem.

Algorithm	Best	Mean	Worst	SD	Median
TDO	0.012671024	0.012681410	0.012701561	0.00002042	0.012678251
TSA	0.012674000	0.012684106	0.012715185	0.000027	0.012687293
MPA	0.012678321	0.012697116	0.012720757	0.000041	0.012699686
WOA	0.013192580	0.014817181	0.017862507	0.002272	0.013192580
GWO	0.012816930	0.014464372	0.017839737	0.001622	0.014021237
TLBO	0.012709667	0.012839637	0.012998448	0.000078	0.012844664
GSA	0.012873881	0.013438871	0.014211731	0.000287	0.013367888
PSO	0.013036251	0.014036254	0.016251423	0.002073	0.013002365
GA	0.012776352	0.013069872	0.015214230	0.000375	0.012952142

TABLE 16. Unimodal objective functions.

Objective Function	Range	Dim	F_{min}
$F_1(X)=\sum_{i=1}^{m} x_i^2$	[−100,100]	30	0
$F_2(X)=\sum_{i=1}^{m}\lvert x_i\rvert+\prod_{i=1}^{m}\lvert x_i\rvert$	[−10,10]	30	0
$F_3(X)=\sum_{i=1}^{m}\left(\sum_{j=1}^{i} x_i\right)^2$	[−100,100]	30	0
$F_4(X)=max\{\lvert x_i\rvert\},\quad 1\le i\le m$	[−100,100]	30	0
$F_5(X)=\sum_{i=1}^{m-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2)\right]$	[−30,30]	30	0
$F_6(X)=\sum_{i=1}^{m}([x_i+0.5])^2$	[−100,100]	30	0
$F_7(X)=\sum_{i=1}^{m} ix_i^4+r,$ where r is a random real number in the range 0 to 1	[−1.28,1.28]	30	0

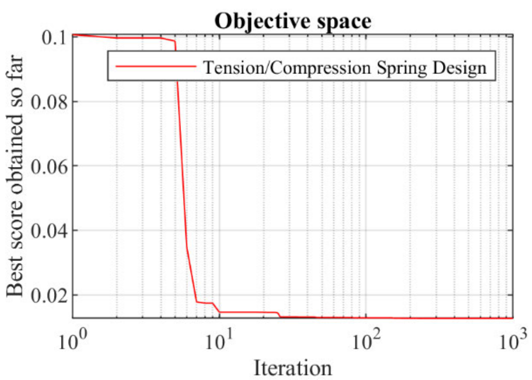

FIGURE 13. Convergence analysis of the TDO for the tension/compression spring design optimization problem.

V. CONCLUSION AND FUTURE WORKS

In this paper, a new bio-inspired metaheuristic algorithm called Tasmanian Devil Optimization (TDO) was introduced. The fundamental inspiration of TDO is the Tasmanian devil feeding behavior in nature, which has two strategies (i) eating carrion and (ii) feeding through hunting. TDO mathematical modeling was presented along with a description of its steps and strategies. The performance of TDO in solving optimization problems was tested on twenty-three objective functions of unimodal and multimodal types. The optimization results of unimodal functions showed the exploitation ability of TDO in convergence towards global optimal. The optimization results of multimodal functions showed that TDO has a high exploration ability in the scanning search space, passing local areas, and discovering the main optimal area. To analyze the quality of TDO results, its performance was compared with eight well-known algorithms, TSA, MPA, WOA, GWO, TLBO, GSA, PSO, and GA. What was concluded from the simulation results was that TDO by providing strong performance and creating the appropriate balance between exploration and exploitation, is superior than the eight competitor algorithms and provides far more competitive optimization results. TDO's performance in optimizing four design problems showed TDO's high ability to solve real-world optimization problems.

The authors provide perspectives for future studies in this paper, the main ones being the design of binary and multi-objective TDO versions. The use of TDO in solving optimization problems in various sciences and real-world problems are other suggestions that open the way for further studies.

APPENDIX A

See Tables 16–18.

APPENDIX B
WELDED BEAM DESIGN PROBLEM

$$
\begin{aligned}
&\textit{Consider } X=[x_1,x_2,x_3,x_4]=[h,l,t,b].\\
&\textit{Minimize } f(x)=1.10471x_1^2x_2+0.04811x_3x_4(14.0+x_2).\\
&\textit{Subject to}: g_1(x)=\tau(x)-13600\le 0,\\
&\qquad g_2(x)=\sigma(x)-30000\le 0,\\
&\qquad g_3(x)=x_1-x_4\le 0,\\
&\qquad g_4(x)=0.10471x_1^2+0.04811x_3x_4(14+x_2)\\
&\qquad\qquad -5.0\le 0,
\end{aligned}
$$

TABLE 17. High-dimensional multimodal objective functions.

Objective Function	Range	Dim	F_{min}
$F_8(X)=\sum_{i=1}^{m} -x_i \sin(\sqrt{\lvert x_i\rvert})$	[−500,500]	30	−12569
$F_9(X)=\sum_{i=1}^{m} [x_i^2-10\cos(2\pi x_i)+10]$	[−5.12,5.12]	30	0
$F_{10}(X)=-20\exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m} x_i^2}\right)-\exp\left(\frac{1}{m}\sum_{i=1}^{m} cos(2\pi x_i)\right)+20+e$	[−32,32]	30	0
$F_{11}(X)=\frac{1}{4000}\sum_{i=1}^{m} x_i^2-\prod_{i=1}^{m} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[−600,600]	30	0
$F_{12}(X)=\frac{\pi}{m}\{10\sin(\pi y_1)+\sum_{i=1}^{m}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{m} u(x_i,10,100,4)$, where $y_i=1+\frac{x_i+1}{4}, u(x_i,a,i,n)=\begin{cases}k(x_i-a)^n, & x_i>a;\\ 0, & -a\le x_i\le a;\\ k(-x_i-a)^n, & x_i<-a,\end{cases}$	[−50,50]	30	0
$F_{13}(X)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{m}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2[1+\sin^2(2\pi x_m)]\}+\sum_{i=1}^{m} u(x_i,5,100,4)$, where $u(x_i,a,i,n)=\begin{cases}k(x_i-a)^n, & x_i>a;\\ 0, & -a\le x_i\le a;\\ k(-x_i-a)^n, & x_i<-a.\end{cases}$	[−50,50]	30	0

$$\begin{aligned} g_5(x) &= 0.125 - x_1 \le 0,\\ g_6(x) &= \delta(x) - 0.25 \le 0,\\ g_7(x) &= 6000 - p_c(x) \le 0. \end{aligned}$$

where

$$\begin{aligned}
\tau(x) &= \sqrt{\tau' + (2\tau\tau')\frac{x_2}{2R} + (\tau'')^2},\\
\tau' &= \frac{6000}{\sqrt{2}\,x_1x_2},\\
\tau'' &= \frac{MR}{J},\\
M &= 6000\left(14+\frac{x_2}{2}\right),\\
R &= \sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2},\\
J &= 2\left\{x_1x_2\sqrt{2}\left[\frac{x_2^2}{12}+\left(\frac{x_1+x_3}{2}\right)^2\right]\right\},\\
\sigma(x) &= \frac{504000}{x_4x_3^2}\\
\delta(x) &= \frac{65856000}{(30\cdot 10^6)\,x_4x_3^3},\\
p_c(x) &= \frac{4.013\,(30\cdot 10^6)\sqrt{\frac{x_3^2x_4^6}{36}}}{196}\left(1-\frac{x_3}{28}\sqrt{\frac{30\cdot 10^6}{4(12\cdot 10^6)}}\right).
\end{aligned}$$

With

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

APPENDIX C
PRESSURE VESSEL DESIGN PROBLEM

$$\begin{aligned}
&Consider\ X = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L].\\
&Minimize\ f(x) = 0.6224x_1x_3x_4 + 1.778x_2x_3^2\\
&\qquad + 3.1661x_1^2x_4 + 19.84x_1^2x_3.\\
&Subject\ to: g_1(x) = -x_1 + 0.0193x_3 \le 0,\\
&\qquad g_2(x) = -x_2 + 0.00954x_3 \le 0,\\
&\qquad g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0,\\
&\qquad g_4(x) = x_4 - 240 \le 0.
\end{aligned}$$

With

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

APPENDIX D
SPEED REDUCER DESIGN PROBLEM

$$\begin{aligned}
&Consider\ X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]\\
&\qquad = [b, m, p, l_1, l_2, d_1, d_2].\\
&Minimize\ f(x) = 0.7854x_1x_2^2\\
&\qquad \left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right)\\
&\qquad - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777
\end{aligned}$$

TABLE 18. Fixed-dimensional multimodal objective functions.

Objective Function	Range	*Dim*	F_{min}
$F_{14}(X) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	$[-65.53,65.53]$	2	0.998
$F_{15}(X) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	$[-5,5]$	4	0.00030
$F_{16}(X) = 4x_1^2 - 2.1 \cdot x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]$	2	−1.0316
$F_{17}(X) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	[-5,10] × [0,15]	2	0.398
$F_{18}(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-5,5]$	2	3
$F_{19}(X) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	$[0,1]$	3	−3.86
$F_{20}(X) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	$[0,1]$	6	−3.22
$F_{21}(X) = -\sum_{i=1}^{5} [(X - a_i) \cdot (X - a_i)^T + 6c_i]^{-1}$	$[0,10]$	4	−10.1532
$F_{22}(X) = -\sum_{i=1}^{7} [(X - a_i) \cdot (X - a_i)^T + 6c_i]^{-1}$	$[0,10]$	4	−10.4029
$F_{23}(X) = -\sum_{i=1}^{10} [(X - a_i) \cdot (X - a_i)^T + 6c_i]^{-1}$	$[0,10]$	4	−10.5364

$$\times \left(x_6^3 + x_7^3\right) + 0.7854(x_4x_6^2 + x_5x_7^2).$$

$$Subject\ to: g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0,$$

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

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

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

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

$$- 1 \leq 0,$$

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

$$- 1 \leq 0,$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0,$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0,$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,$$

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

With

$$2.6 \leq x_1 \leq 3.6, \quad 0.7 \leq x_2 \leq 0.8, \; 17 \leq x_3 \leq 28, \\ 7.3 \leq x_4 \leq 8.3, \quad 7.8 \leq x_5 \leq 8.3, \; 2.9 \leq x_6 \leq 3.9, \quad and \\ 5 \leq x_7 \leq 5.5.$$

APPENDIX E
TENSION/COMPRESSION SPRING DESIGN PROBLEM

$$Consider\ X = [x_1, x_2, x_3] = [d, D, P].$$
$$Minimize\ f(x) = (x_3 + 2)\, x_2 x_1^2.$$
$$Subject\ to: g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$
$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$
$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$
$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0.$$

With

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

ACKNOWLEDGMENT
The authors would like to thank Eva Trojovská for her significant help with the graphic design of the article.

REFERENCES
[1] J. O. Agushaka and A. E. Ezugwu, "Advanced arithmetic optimization algorithm for solving mechanical engineering design problems," *PLoS ONE*, vol. 16, no. 8, Aug. 2021, Art. no. e0255703.
[2] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
[3] R. G. Rakotonirainy and J. H. van Vuuren, "Improved metaheuristics for the two-dimensional strip packing problem," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106268.
[4] X. Wu, S. Zhang, W. Xiao, and Y. Yin, "The exploration/exploitation tradeoff in whale optimization algorithm," *IEEE Access*, vol. 7, pp. 125919–125928, 2019.
[5] S.-H. Liu, M. Mernik, D. Hrnčič, and M. Črepinšek, "A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova's mass transfer model," *Appl. Soft Comput.*, vol. 13, no. 9, pp. 3792–3805, Sep. 2013.
[6] F. Zhang, J. Chen, T. Mao, and Z. Wang, "Feedback interval optimization for MISO LiFi systems," *IEEE Access*, vol. 9, pp. 136811–136818, 2021.
[7] K. Fukada, M. Parizy, Y. Tomita, and N. Togawa, "A three-stage annealing method solving slot-placement problems using an ising machine," *IEEE Access*, vol. 9, pp. 134413–134426, 2021.
[8] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 95–99, 1988.
[9] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[10] S. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evol. Comput.*, vol. 8, no. 4, pp. 443–473, Dec. 2000.
[11] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.
[12] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
[13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016.
[14] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired Metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.
[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[16] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Proc. Int. Fuzzy Syst. Assoc. World Congr.*, 2007, pp. 789–798.
[17] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103541.
[18] D. Połap and M. Woźniak, "Red fox optimization algorithm," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114107.
[19] S. Z. Koohi, N. A. W. A. Hamid, M. Othman, and G. Ibragimov, "Raccoon optimization algorithm," *IEEE Access*, vol. 7, pp. 5383–5399, 2019.
[20] A. G. Hussien, M. Amin, M. Wang, G. Liang, A. Alsanad, A. Gumaei, and H. Chen, "Crow search algorithm: Theory, recent advances, and applications," *IEEE Access*, vol. 8, pp. 173548–173565, 2020.
[21] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.
[22] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Grasshopper optimization algorithm: Theory, variants, and applications," *IEEE Access*, vol. 9, pp. 50001–50024, 2021.
[23] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," *Simulated Annealing: Theory and Applications*. Dordrecht, The Netherlands: Springer, 1987, pp. 7–15.
[24] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.
[25] M. Dehghani, Z. Montazeri, A. Dehghani, and A. Seifi, "Spring search algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law," in *Proc. IEEE 4th Int. Conf. Knowl.-Based Eng. Innov. (KBEI)*, Dec. 2017, pp. 0210–0214.
[26] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm—A novel Metaheuristic optimization method for solving constrained engineering optimization problems," *Comput. Struct.*, vols. 110–111, pp. 151–166, Nov. 2012.
[27] H. Du, X. Wu, and J. Zhuang, "Small-world optimization algorithm for function optimization," in *Proc. Int. Conf. Natural Comput.*, 2006, pp. 264–273.
[28] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2012.
[29] Z. Wei, C. Huang, X. Wang, T. Han, and Y. Li, "Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization," *IEEE Access*, vol. 7, pp. 66084–66109, 2019.
[30] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016.
[31] B. Alatas, "ACROA: Artificial chemical reaction optimization algorithm for global optimization," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13170–13180, 2011.
[32] A. H. Kashan, "A new metaheuristic for optimization: Optics inspired optimization (OIO)," *Comput. Oper. Res.*, vol. 55, pp. 99–125, Mar. 2015.
[33] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.
[34] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowl.-Based Syst.*, vol. 163, pp. 283–304, Jan. 2019.
[35] H. Abedinpourshotorban, S. M. Shamsuddin, Z. Beheshti, and D. N. A. Jawawi, "Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm," *Swarm Evol. Comput.*, vol. 26, pp. 8–22, Feb. 2016.
[36] M. Dehghani, M. Mardaneh, J. Guerrero, O. Malik, and V. Kumar, "Football game based optimization: An application to solve energy commitment problem," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 514–523, Oct. 2020.
[37] R. Moghdani and K. Salimifard, "Volleyball premier league algorithm," *Appl. Soft. Comput.*, vol. 64, pp. 161–185, Mar. 2018.
[38] S. Doumari, H. Givi, M. Dehghani, and O. Malik, "Ring toss game-based optimization algorithm for solving various optimization problems," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 3, pp. 545–554, Jun. 2021.

[39] M. Dehghani, Z. Montazeri, H. Givi, J. Guerrero, and G. Dhiman, "Darts game optimizer: A new optimization technique based on darts game," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 286–294, Oct. 2020.

[40] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[41] O. L. Buchmann and E. R. Guiler, "Behaviour and ecology of the Tasmanian devil, Sarcophilus harrisii," in *The Biology of Marsupials*. London, U.K.: Macmillan, 1977, pp. 155–168.

[42] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[43] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. New York, NY, USA: Springer-Verlag, 1992, pp. 196–202.

[44] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *J. Mech. Des.*, vol. 116, no. 2, pp. 405–411, 2008.

[45] A. H. Gandomi and X.-S. Yang, "Benchmark problems in structural optimization," in *Computational Optimization, Methods and Algorithms*. Berlin, Germany: Springer, 2011, pp. 259–281.

[46] E. Mezura-Montes and C. A. C. Coello, "Useful infeasible solutions in engineering optimization with evolutionary algorithms," in *Proc. Mex. Int. Conf. Artif. Intell.*, 2005, pp. 652–662.

MOHAMMAD DEHGHANI received the B.S. degree in electrical engineering from the Shahid Bahonar University of Kerman, Iran, in 2012, the M.S. degree in electrical engineering from Shiraz University, Shiraz, Iran, in 2016, and the Ph.D. degree from the Shiraz University of Technology, Shiraz, in 2020. His current research interests include optimization, metaheuristic algorithms, power systems, and energy commitment.

ŠTĚPÁN HUBÁLOVSKÝ received the M.Sc. and Ph.D. degrees from the Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, in 1995 and 1998, respectively. In 2012, he became an Associate Professor at the Faculty of Informatics and Management, University of Hradec Králové, Czech Republic. He is currently the Vice Dean at the Faculty of Science, University of Hradec Králové. His research interests include technical cybernetics, computer simulation and optimization, and big data processing.

PAVEL TROJOVSKÝ received the M.Sc. degree in teaching of mathematics, physics, and computer science from the University of Hradec Králové, Hradec Králové, Czech Republic, in 1989, and the Ph.D. degree in general questions of mathematics and computer science from the Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, in 2001. In 2011, he became an Associate Professor of system engineering and informatics with the University of Pardubice, Pardubice, Czech Republic. He is currently an Associate Professor and the Vice Dean for Creative Activities at the Faculty of Science, University of Hradec Králové. His research interests include number theory and its applications in cryptography, applied mathematics, computer simulation and optimization, and big data processing.