Heliyon 10 (2024) e36663

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Enhanced Prairie Dog Optimization with Differential Evolution for solving engineering design problems and network intrusion detection system

Mohammad Alshinwan [a,*], Osama A. Khashan [b,**], Mohammed Khader [a], Omar Tarawneh [c], Ahmed Shdefat [d], Nour Mostafa [d], Diaa Salama AbdElminaam [e,f,***]

[a] *Faculty of Information Technology, Applied Science Private University, Amman 11931, Jordan*
[b] *Research and Innovation Centers, Rabdan Academy, Abu Dhabi P.O. Box 114646, United Arab Emirates*
[c] *Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, 11953, Jordan*
[d] *College of Engineering and Technology, American University of the Middle East, Egaila, 54200, Kuwait*
[e] *MEU Research Unit, Middle East University, Amman 11831, Jordan*
[f] *Jadara Research Center, Jadara University, Irbid, 21110, Jordan*

ARTICLE INFO

Keywords:
Prairie dog algorithm
Differential Evolution algorithm
Engineering problems
Real-world problems
Optimization problems

ABSTRACT

This paper introduces a novel hybrid optimization algorithm, PDO-DE, which integrates the Prairie Dog Optimization (PDO) algorithm with the Differential Evolution (DE) strategy. This research aims to develop an algorithm that efficiently addresses complex optimization problems in engineering design and network intrusion detection systems. Our method enhances the PDO's search capabilities by incorporating the DE's principal mechanisms of mutation and crossover, facilitating improved solution exploration and exploitation. We evaluate the effectiveness of the PDO-DE algorithm through rigorous testing on 23 classical benchmark functions, five engineering design problems, and a network intrusion detection system (NIDS). The results indicate that PDO-DE outperforms several state-of-the-art optimization algorithms regarding convergence speed and accuracy, demonstrating its robustness and adaptability across different problem domains. The PDO-DE algorithm's potential applications extend to engineering challenges and cybersecurity issues, where efficient and reliable solutions are critical; for example, the NIDS results show significant results in detection rate, false alarm, and accuracy with 98.1%, 2.4%, and 96%, respectively. The innovative integration of PDO and DE contributes significantly to stochastic optimization and swarm intelligence, offering a promising new tool for tackling diverse optimization problems. In conclusion, the PDO-DE algorithm represents a significant scientific advancement in hybrid optimization techniques, providing a more effective approach for solving real-world problems that require high precision and optimal resource utilization.

* Corresponding author at: Faculty of Information Technology, Applied Science Private University, Amman 11931, Jordan.
** Corresponding author at: Research and Innovation Centers, Rabdan Academy, Abu Dhabi 114646, United Arab Emirates.
*** Corresponding author at: Jadara Research Center, Jadara University, Irbid, 21110, Jordan.
E-mail addresses: m_shinwan@asu.edu.jo (M. Alshinwan), okhashan@ra.ac.ae (O.A. Khashan), m_khader@asu.edu.jo (M. Khader), o.husain@aau.edu.jo (O. Tarawneh), ahmed.shdefat@aum.edu.kw (A. Shdefat), nour.moustafa@aum.edu.kw (N. Mostafa), diaa.salama@miuegypt.edu.eg (D.S. AbdElminaam).

https://doi.org/10.1016/j.heliyon.2024.e36663
Received 1 May 2024; Received in revised form 16 August 2024; Accepted 20 August 2024

Available online 23 August 2024

1. Introduction

Optimization permeates every aspect of human culture. Human beings are naturally inclined to enhance and refine existing entities to address practical challenges and adapt to the present surroundings effectively [1]. Global optimization aims to identify the most optimal solution from all potential solutions for a given problem. Optimization methods can be classified into two main categories: deterministic and stochastic. While deterministic algorithms have reached a high level of development in mathematical theory, their ability to optimize and demonstrate efficiency is limited when faced with discontinuous and non-differentiable functions [2].

In some cases, these algorithms are unable to address such issues. Nevertheless, most engineering optimization problems involving numerous local optimal values exhibit characteristics of discontinuity non-differentiability and even pose challenges in mathematical model representation [3,4]. Due to these circumstances, many scholars have redirected their attention towards stochastic optimization techniques. One important characteristic is the incorporation of randomization, which allows for the potential to escape from local optima. Therefore, it is crucial to employ stochastic optimization methods to achieve the most optimal solutions for global optimization issues [5,6].

The swarm intelligence algorithm is a stochastic optimization technique that draws inspiration from the diverse behaviors exhibited by biological communities in nature. It offers a novel approach to solving optimization issues [7,8]. In the natural world, numerous species exhibit remarkable swarm intelligence activities, which involve a combination of cooperation and competition among individuals [9]. These behaviors compensate for the limitations of individual foraging and help in evading predation [10]. For instance, the act of wolves preying on other animals [11], the act of birds coming together and moving from one place to another [12], and the way bees and ants interact socially [13]. Swarm intelligence optimization algorithms can be implemented by examining the possible behaviors of individuals in a population and employing mathematical modeling to construct the operational mechanism of the population system. This includes analyzing the cooperation and competition among individuals within the population and the interaction between the population and the external environment [14,15].

Over the past few decades, individuals have been developing diverse approaches to address intricate optimization challenges. Meta-heuristic algorithms (MAs) are particularly notable among these methods and offer an effective solution [16,17]. Due to its divergence from conventional optimization methods, the meta-heuristic algorithm operates independently of gradient information and can effectively evade local optima [18]. In broad terms, meta-heuristic optimization algorithms can be classified into four main categories: algorithms that rely on human behavior, algorithms that are based on evolutionary principles, algorithms that utilize swarm intelligence, and algorithms that draw inspiration from physics or chemistry [19].

These MAs include distinguishable characteristics and are widely employed in various computer science domains, intrusion detection systems (IDS), engineering optimization design, routing planning, text clustering, image segmentation and classification, feature selection, and fault diagnosis. The “No Free Lunch (NFL) theorem” demonstrates that no algorithm can solve all optimization issues universally [20]. Hence, it is crucial to enhance existing algorithms. Various scholars employ diverse methodologies to enhance preexisting algorithms.

Engineering design problems encompass the difficulties that occur when designing and developing a project or manufacturing operation. These difficulties often involve optimizing resources such as time, money, materials, and staff to create a product that meets customer expectations while ensuring efficiency and cost-effectiveness [21,22]. Ordinary design problems encountered in engineering include designing and producing a product, looking for the most effective method to create and produce a product that meets customer needs and market expectations while reducing expenses and optimizing productivity. Supply chain management encompasses coordinating and controlling all essential product elements to guarantee their availability at the designated time and place while ensuring they are provided at a suitable cost. Production planning and control entails optimizing manufacturing activities to meet client demand and maintaining adequate inventory levels to fulfill customer requirements. Office layout: Establishing a workspace that is both ergonomic and secure enhances employee productivity while minimizing the likelihood of accidents or injuries [23,24]. Quality assurance entails ensuring that things conform to the standards set by the client and are free from any faults. Maintenance and reliability encompass the implementation of procedures to ensure the optimal operation of equipment and machinery while minimizing downtime. Cost optimization entails minimizing costs related to producing goods or delivering a service while upholding superior standards and fulfilling consumer expectations [25,26].

These challenges are intricate, requiring a profound understanding of principles and issues in engineering design and the capacity to analyze and address intricate challenges. Engineering design issues employ optimization techniques and advanced modeling methods, such as simulation and mathematical analysis, to devise and enhance production processes and innovate goods [27].

PDO-DE, which stands for Prairie Dog Optimization (PDO) [28] with Differential Evolution Algorithm (DE), is an emerging optimization technique designed to address industrial engineering design challenges. The PDO-DE method synergistically integrates PDO and DE to discover optimal solutions efficiently. In DE learning, solutions generated are indicative of the collective population. PDO, an optimization technique grounded in population dynamics, leverages the action of prairie dogs to identify the most efficient solution. Prairie dogs exhibit collective behavior, moving in a coordinated manner resembling a swarm, and effectively communicate with one another to identify optimal solutions [29,30]. Combining PDO and DE methods makes it possible to include the best and worst solutions. This approach leads to a quicker convergence towards the optimal solution and decreases the likelihood of being trapped in a local minimum. Additional investigation is required to evaluate the efficacy and suitability of this approach in addressing industrial engineering design issues, but preliminary studies have indicated its potential for industrial engineers.

This paper presents the Prairie Differential Optimization (PDO) method as a novel approach to address many intricate optimization problems. This work is primarily driven by improving optimization algorithms' performance by tackling the enduring problem of striking a balance between exploration and exploitation. Algorithms that thoroughly search the solution space and take advantage of well-established solutions are necessary for effective optimization to improve outcomes. But striking this balance is still quite difficult, especially for algorithms like the PDO algorithm, which has trouble completing tasks on time and frequently gets stuck in local optima, especially when working on high-dimensional issues.

The main goal of this research is to create a more sophisticated PDO algorithm that can function reliably in various engineering optimization scenarios and overcome these drawbacks. We suggest incorporating the Differential Evolution (DE) approach into the PDO framework. The DE approach's robust exploration capabilities bolster the PDO algorithm's exploitation strengths. By merging these two techniques, the modified PDO algorithm seeks to accomplish a more successful balance between exploration and exploitation.

This research paper examines the updated PDO algorithm, showcasing its better performance via a battery of exacting tests and contrasting it with other cutting-edge optimization methods. The outcomes demonstrate that the integrated DE approach greatly enhances the PDO algorithm's capacity to traverse complicated solution spaces, resulting in quicker convergence and more precise solutions.

In addition, this study presents an enhanced PDO method that tackles the crucial problem of striking a balance between exploration and exploitation, significantly advancing the optimization field. In addition to improving the PDO algorithm's overall performance, the suggested integration of the DE approach provides a workable solution for successfully resolving high-dimensional optimization issues. This development increases the value and effect of the improved PDO algorithm by providing new opportunities for its application to various engineering and real-world optimization challenges.

The main contribution of this work is illustrated as follows:

- We provide a novel strategy called PDO-DE. This approach is motivated by the design principles of the PDO and DE algorithms.
- DE enhances the efficacy of the PDO in diversifying the primary population and its capability to escape from local optima.
- Enhance PDO's global and local search capabilities to increase convergence accuracy.
- The PDO-DE algorithm is demonstrated to significantly improve problem-solving effectiveness in two complex domains: engineering design problems and network intrusion detection systems. Our results show superior precision and convergence rate performance compared to existing algorithms.
- The PDO-DE algorithm's performance is demonstrated using a set of twenty-three widely recognized benchmark and CEC2019 functions.
- Five engineering design problems are utilized to verify the performance of the PDO-DE.
- Implementing the PDO-DE to enhance the performance of network intrusion detection systems.
- Extensive benchmarking using 23 classical functions and real-world applications validate the effectiveness of the PDO-DE algorithm. This rigorous testing demonstrates the algorithm's adaptability and efficiency across diverse optimization problems, setting a new benchmark for future research.
- By illustrating the broad applicability of PDO-DE across different domains, this work extends the scope of hybrid optimization methods in practical applications, particularly in areas requiring rapid and accurate convergence to optimal solutions.

The rest of this work is presented as follows. Section 2 outlines the methodology of the suggested approach. Section 3 presents the experiments and results that are conducted. Section 4 presents the outcomes of implementing the suggested approach to real-world engineering. Section 5 presents the final findings and outlines potential areas for future research.

2. The proposed methodology

The subsequent section outlines the suggested technique's step-by-step procedures. The suggested approach incorporates the operators from both the Prairie Dog Optimization (PDO) algorithm and the Differential Evolution Algorithm.

2.1. Prairie Dog Optimization algorithm

Recently, Ezugwu et al. proposed a new algorithm called the Prairie Dog Optimization (PDO) algorithm that draws inspiration from the natural behavior of prairie dogs in their natural habitat. This technique determines the most advantageous solution for a specific optimization problem. Prairie dog activity involves the animals coming out of their burrows, relocating, and returning. The PDO method utilizes potential solutions to identify the optimal answer for a specified problem. The candidate solutions experience iterative updates and evolve to ascertain the most optimal alternative [28].

The colony consists of l prairie dogs (PDOs) that belong to s coteries. These professional development programs exist and function collectively inside their respective social circles. Therefore, the location of the ith PDO within a specific group can be denoted by a vector. Equation (1) concisely describes the matrix representation that shows the positions of all coteries (COTs) in the colony.

$$COT = \begin{bmatrix} COT1,1 & COT1,2 & \dots & \dots & \dots & COT1,d \\ COT2,1 & COT2,2 & \cdots & \dots & \cdots & COT2,d \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ COTs,1 & COTs,2 & \vdots & \vdots & \vdots & COTs,d \end{bmatrix} \tag{1}$$

The symbol $CT_{i,j}$ denotes the jth area of the ith coterie in the colony. Equation (2) illustrates the spatial distribution of all the PDOs inside a coterie.

$$PD = \begin{bmatrix} PD1,1 & PD1,2 & \ldots & \ldots & \cdots & PD1,d \\ PD2,1 & PD2,2 & \cdots & \cdots & \cdots & PD2,d \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ PDm,1 & PDm,2 & \vdots & \vdots & \vdots & PDm,d \end{bmatrix} \tag{2}$$

The symbol PD_i; j denotes the jth proportions of the ith prairie dog inside a group, where l and s are different numbers. The assignment of coterie and prairie dog places is accomplished using a uniform-distribution, as illustrated in Equations (3) and (4).

$$CTi, j = U(0,1) \times (Uj - Lj) + LBj \tag{3}$$

$$PDi, j = U(0,1) \times (Uj - Lj) + Lj \tag{4}$$

The Uj and Lj represent the maximum and minimum values of the jth dimension in an optimization problem (Lower and Uber bond). The upper bound, denoted as Uj, is computed by dividing Uj by s. Similarly, the lower bound, denoted as Lj, is determined by dividing Lj by s. The symbol $U(0,1)$ denotes a random variable that is uniformly distributed between 0 and 1. PDO algorithm adjusts its approach by alternating between exploration and exploitation based on four conditions. The entire number of repetitions (rep) is divided into four parts, with the initial two sections allocated for exploration and the remaining two dedicated to exploitation. The exploration is divided into two techniques based on the constraints $rep < Maxrep/4$ and $Maxrep/4 < rep < Maxrep/2$. The exploitation is divided into two methods, each governed by certain conditions: $Maxrep/2 < rep < 3Maxrep/4$ and $3Maxrep/4 < rep < Maxrep$.

2.1.1. PDO exploration stage

An assessment is conducted to determine the quality of the available food sources, and the optimal choice is selected for gathering. The development of recent burrows is contingent upon the caliber of the chosen food supply. Equation (5) represents the process of updating placements during the algorithm's exploration stage.

$$PDOi+1, j+1 = \text{ GPDOBest } j, j - eC \text{ PDOBest } i, j \times p - CPDOi, j \times \text{Levy } (L) \text{ A rep } < \frac{\text{Maxrep}}{4} \tag{5}$$

The second technique entails assessing the caliber of earlier encountered food origins and appraising the proficiency in digging. Subsequent burrows are subsequently created utilizing this excavating capability, which diminishes as the number of repetitions rises, restricting the number of tunnels that may be formed. Equation (6) denotes the process of adjusting the placements to construct burrows.

$$\text{PDOi } +1, j+1 = PDOGBest\, j, j \times rPDO \times DS \times \text{Levy}(L)\, \Lambda \frac{\text{Maxrep}}{4} \leq \text{ rep } \leq \frac{\text{Maxrep}}{2} \tag{6}$$

The current most suitable solution is denoted as GPDOBesti; j, and its effectiveness is assessed using $eCPDOBesti$; j, as shown in Equation (7). The food source warning is denoted as q and has a constant frequency of 0.1 kHz. The variable rPDO indicates the random solution's position, while the combined impact of all PDO in the colony is denoted as $CPDOi$; j, as mentioned in Equation (8). The digging resilience of the clique, referred to as D_S, relies on the food source's quality and is selected randomly using Equation (9). The Levy distribution, denoted as $Levy(L)$, is employed to optimize the investigation of the issue space with greater efficiency.

$$e \text{ CPDOBesti}, j = \text{ GPDOBesti}, j \times \Delta + \frac{PDOi, j \times \text{mean}(PDOn, s)}{G \text{ PDOBest } \times (Uj - Lj) + \Delta} \tag{7}$$

$$CPDOi, j = \frac{G \text{ PDOBesti }, j - rPDOi, j}{GPDOBesti, j + \Delta} \tag{8}$$

$$D_S = 1.5 \times r \times \left(1 - \frac{\text{rep}}{\text{Maxrep}}\right)^2 \frac{\text{rep}}{\text{Maxrep}} \tag{9}$$

The factor r introduces randomness to facilitate exploration, alternating between -1 and 1 based on whether the current repetition is odd or even. D considers any variations among the prairie dogs, although the implementation assumes they are all identical. rep represents the current iteration, while $Maxrep$ represents the highest number of repetitions permitted.

2.1.2. PDO exploitation stage

PDO employs exploitation strategies to conduct a comprehensive search in the potential areas discovered during the exploration phase. The two methodologies employed in this phase are represented by equations (10) and (11). PDO employs these two strategies depending on the constraints $Maxrep/2 \leq rep < 3Maxrep/4$ and $3Maxrep/4 \leq rep \leq Maxrep$, as mentioned earlier.

$$PDOi+1, j+1 = \text{ GPDOBest } i, j - eC \text{ PDOBest } i, j \times \varepsilon - CPDOi, j \times \text{ rand } \Lambda 3 \frac{\text{Maxrep}}{4} \leq \text{ rep } < 3 \frac{\text{Maxrep}}{4} \tag{10}$$

$$PDOi+1, j+1 = \text{ GPDOBest } i, j - PE \times \text{ rand } \Lambda 3 \frac{\text{Maxrep}}{4} \leq \text{ rep } < \text{ Maxrep} \tag{11}$$

In this instance, GPDOBesti; j denotes the current best solution discovered, while eCPDOBesti; j signifies the influence of the currently achieved ideal solution. According to Equation (10), ϵ denotes the food source's quality, whereas CPDOi; j indicates the collective impact of all PDOs in the colony, as stated in Equation (11). As denoted by the mathematical Equation (12), the predator impact is symbolized by the abbreviation PE, while rand refers to a randomly generated number ranging from 0 to 1.

$$PE = 1.5\left(1 - \frac{\text{rep}}{\text{Maxrep}}\right)^{2\frac{\text{rep}}{\text{Maxrep}}} \tag{12}$$

2.2. Differential Evolution algorithm

The Differential Evolution algorithm (DEA), proposed by Storn and Price, is a robust and efficient search method designed to tackle intricate continuous nonlinear functions. The conventional Differential Evolution (approach commences by initializing a population of N individuals represented by vector $\vec{X}_i, where\ \vec{X}_i = (X_{i1}, X_{i2}, X_{i3}, ..., X_{in}), i = 1, 2, 3, ..., N$, and n is the problem dimension. The DEA algorithm has incorporated three primary operators: mutation, crossover, and selection. The mutation and crossover operators are utilized to produce fresh candidate vectors. At the same time, a selection technique is implemented to determine the survival of either the offspring or the parent in the subsequent generation.

2.2.1. Mutation phase

An individual with genetic mutations is represented According to Equation (13) where $\vec{V}_i = (v_{i1}, v_{i2}, v_{i3}, ..., v_{in})$ and is created through the use of a mutation operator. Multiple mutation techniques are documented in the literature [31]. One often used operator is 'DE/best/1', which is defined as:

$$\vec{V}_i(t) = \vec{X}^*(t) + F(\vec{X}_\alpha(t) - \vec{V}_\beta(t)) \tag{13}$$

Here, t represents the current iteration, $\vec{X}^*(t)$ represents the best individual with the lowest $f(\vec{X}^*)$. Currently, α and β are two randomly selected indices from the range [1, N], where a, b, and i are all different from each other $\alpha \neq \beta \neq i \in 1, ..., N$). Additionally, F ∈ [0, 1] represents a mutation scaling factor influencing the differential variation between two individuals. The following operator is applied to all individuals once the mutation operator has been applied.

2.2.2. Crossover phase

The crossover parameter is utilized on each mutant individual and its associated target individual $\vec{X}_i$ to produce a trial vector, $\vec{U}_i = (u_{i1}, u_{i2}, u_{i3}, ..., u_{i,n})$. Exponential and binomial crossovers are frequently employed crossover strategies. The binomial crossover is expressed According to Equation (14):

$$u_{i,j}(t) = \begin{cases} v_{i,j}(t), & if\ r_j \le CR\ or\ j = R; \\ xi, j(t), & Otherwise \end{cases} \tag{14}$$

The index R represents a dimension randomly selected from the set $1, 2, ..., n$. This is done to guarantee that at least one dimension from $\vec{V}_i(t)$ is present in the trial individual $\vec{U}_i$, which is different from its target vector, $\vec{V}_i(t)$. The crossover rate (CR) is a value that ranges from 0 to 1, and $r_j \in [0, 1]$ is a random number that is uniformly distributed between 0 and 1. If the parameter values of the trial people exceed the pre-determined higher or lower bounds, we can assign them the upper or lower bound value accordingly.

2.2.3. Selection phase

A one-to-one greedy selection is used in DE to determine if the trial individual $\vec{U}_i(t)$ should be included in the target population for the next generation. This selection strategy promotes diversity compared to tournament, rank-based, and fitness-proportional selection. The one-to-one selection technique operates by determining the survival of the more fit person between the trial individual $\vec{U}_i(t)$ and its target counterpart $\vec{X}_i(t)$. The formulation for minimization issues is According to Equation (15):

$$\vec{X}_i(t+1) = \begin{cases} \vec{U}_i(t), & if\ f(\vec{U}_i(t)) \le f(\vec{X}_i(t)); \\ \vec{X}_i(t), & otherwise, \end{cases} \tag{15}$$

where f is the objective function. The aforementioned procedure is iterated until a termination requirement is achieved.

2.3. Proposed PDO-DE algorithm

This section delineates the proposed strategy's fundamental methodologies. The proposed methodology functions by utilizing two primary methods: Prairie Dog optimization (PDO) and the Differential Evolution Algorithm. Tackling engineering design difficulties efficiently requires finding the most optimal or nearly optimal solution for complex systems with several constraints and variables. Various optimization strategies have been developed to tackle these problems, including traditional methods (such as gradient-based techniques and linear programming) and MAs (such as genetic algorithms and particle swarm optimization).

The present study introduces a new optimization technique, Prairie Dog optimization (PDO) and Differential Evolution Algorithm (PDO-DE), to address engineering design challenges. This approach utilizes a combination of Prairie Dog Optimization and Differential Evolution Algorithm to discover solutions that are close to optimal efficiently.

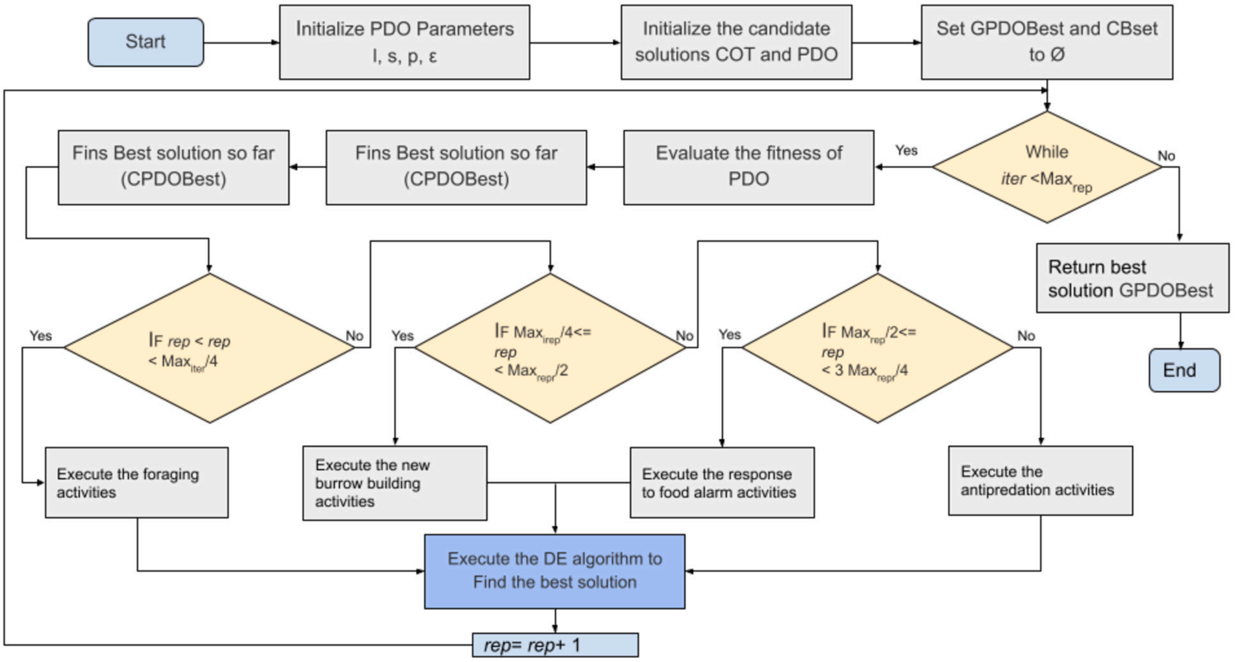

Fig. 1. Proposed PDO-DE algorithm.

Differential Evolution Algorithm is a machine learning methodology that employs the most favorable and unfavorable solutions from a population of solutions to create novel solutions. This strategy effectively addresses constraints associated with conventional optimization techniques, such as the tendency to become trapped in a local minimum, by actively investigating solutions far from the existing population. Prairie Dog Optimization is a population-based optimization technique that leverages the behavioral patterns of prairie dogs to identify the most effective solution. Prairie dogs exhibit collective behavior, moving in a coordinated manner resembling a swarm, and communicate among themselves to determine the optimal option.

The PDO-DE methodology starts by initializing a set of solutions and generating further solutions by utilizing the DE algorithm. Subsequently, the prairie dogs endeavor to choose the most favorable course of action by moving to a different location and sharing information. The technique continues until the optimal solution is achieved or a termination criterion is met. The main procedure of the proposed method is depicted in Fig. 1. This approach enhances the PDO algorithm by optimizing the balance between exploration and exploitation while searching for optimal solutions. The integration of the DE algorithm with PDO's mechanisms facilitates this balance. The DE algorithm boosts the PDO's exploratory abilities and accelerates convergence to the best solution—this synergy between PDO and DE results in improved performance of the PDO algorithm. The initialization of the HPDO algorithm involves randomly selecting the initial positions of IP agents (A) through a designated formula.

$$A_{ij} = rand \times (UpB - LoB) + LoB,\ i = 1, 2, ..., IP,\ j = 1, 2, ..., Z. \tag{16}$$

In Equation (16), Z denotes the dimension of each parameter A_i. LoB and UpB define the lower and upper bounds of the search space, respectively. A hybridization of the PDO and DE algorithms manages the update of the agents A. This hybrid operation uses a randomly generated parameter $Rand_r \in [0, 1]$ to switch between the PDO and DE mechanisms. Specifically, if $Rand_r < 0.5$, the PDO operator is employed to modify the current solution; if $Rand_r \geq 0.5$, the solution is updated using the HHO algorithm. The method is described as follows:

$$A_i(t+1) = \begin{cases} Use\ PDO\ as\ in\ Eqs.(9) - (12), & Rand_r < 0.5 \\ Apply\ DE\ as\ in\ Eq.(15), & otherwise \end{cases} \tag{17}$$

The PDO-DEO algorithm is outlined in Algorithm 1. Using the Differential Evolution Algorithm in optimization enhances the algorithm's performance by incorporating new concepts that improve its capacity to find the best solution. The Differential Evolution Algorithm provides a framework for producing diverse solutions by including contrasting elements. Furthermore, the Differential Evolution Algorithm offers a technique for producing solutions with a more evenly distributed range of values. By combining these two concepts, the optimization algorithm may methodically and effectively explore the search space and quickly approach the optimal solution, reducing the likelihood of getting stuck in suboptimal solutions.

Algorithm 1 The proposed PDO-DE Algorithm.

Initialize: Population of prairie dogs PD, differential vectors DV, and algorithm parameters.
for each generation **do**
  for each prairie dog i in PD **do**
    Evaluate the fitness of each prairie dog using $f(PD_i)$.
    Identify the global best G_{best} and local bests within the communication range.
  end for
  Prairie Dog Exploration:
  for each prairie dog i in PD **do**
    if random condition for exploration **then**
      Update position using exploration strategy:
      $PD_i^{new} = PD_i + \alpha \cdot (G_{best} - PD_i) + \beta \cdot (PD_{local} - PD_i)$
      where α and β are random weights.
    end if
  end for
  Differential Evolution Modification:
  for each prairie dog i in PD **do**
    Apply mutation operator:
    $V_i = PD_{best} + F \cdot (PD_{r1} - PD_{r2})$
    Apply crossover operator:
    $U_i[j] = \begin{cases} V_i[j] & \text{if } rand[0,1] \le CR \text{ or } j = rand\_idx \\ PD_i[j] & \text{otherwise} \end{cases}$
    where $rand\_idx$ is a randomly chosen index.
  end for
  Selection:
  for each prairie dog i in PD **do**
    Compare the fitness of the new position U_i with the current position PD_i.
    Accept the new position if it offers better fitness:
    $PD_i = \begin{cases} U_i & \text{if } f(U_i) < f(PD_i) \\ PD_i & \text{otherwise} \end{cases}$
  end for
  Check for convergence criteria or maximum iterations.
end for
Output: Return the best solution found.

2.4. Complexity analysis of PDO-DE algorithm

The PDO-DE algorithm merges Prairie Dog Optimization (PDO) with Differential Evolution (DE) to solve complex optimization problems efficiently. This section provides a detailed complexity analysis of the PDO-DE algorithm, focusing on its major computational steps across various stages.

2.4.1. Overview of algorithm stages

- **Initialization**: Complexity of $O(n)$ for generating n initial individuals.
- **Fitness Evaluation**: Each individual requires $O(f)$ operations, resulting in $O(n \times f)$ per generation.
- **Prairie Dog Exploration**: Updating positions based on local interactions, typically $O(n \times l)$.
- **Differential Evolution Modification**: Involving mutation and crossover at $O(n)$ complexity.
- **Selection**: Complexity of $O(n)$ for selecting the new generation.

2.4.2. Computational complexity

The overall computational complexity for each generation includes these steps, dominated by the evaluation step:

$$O(n \times (f + l + 2)) \tag{18}$$

Given G generations, the total complexity becomes:

$$O(G \times n \times (f + l + 2)) \tag{19}$$

This formulation indicates that the algorithm's computational load primarily depends on the number of generations G, the population size n, and the complexity of the fitness function f. The local interaction term l adds additional complexity but is typically bounded by n.

3. Experiments and results

This study will employ the Prairie Dog Optimization and Differential Evolution method (PDO-DE) to tackle global optimization difficulties. The main focus will be on 23 benchmark functions, CEC2019 functions, and five challenges in engineering. The section begins by briefly elucidating the benchmark and engineering difficulties utilized for experimentation and the experimental setup. The experimental findings are then reported comprehensively. The findings consist of the objective function values obtained by the

Table 1
Parameter values for the PDO-DE algorithm and other algorithms.

Algorithm	Parameters
HHO	ϵ_0 between -1 and 1
GOA	c = [1E-5,1]
SSA	$v_0 = 0$
WOA	$\alpha = [2\,to\,0]$
SCA	$\alpha = [0.05]$
DA	$\omega = [0.2, 0.9]$
SMA	$z = 0.03$
PDO	$\rho = 0.1, \varepsilon = 2.22E-16$

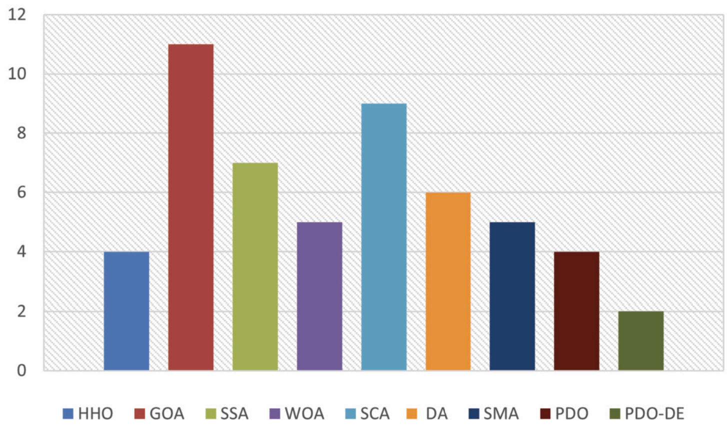

Fig. 2. Compersion of execution time.

proposed method, the number of function assessments, and the convergence curves. The performance of the proposed approach is assessed by contrasting its objective function values with those of other existing optimization methods.

Furthermore, the paper includes a comprehensive analysis of the findings, highlighting the strengths and weaknesses of the proposed approach. The debate provides vital insights into the potential applications of the suggested method and highlights areas that require more investigation. The suggested strategy is assessed utilizing comparable methodologies, which encompass the following algorithms:

- Harris hawks optimization (HHO) [12].
- Grasshopper optimization algorithm (GOA) [32].
- Salp Swarm Algorithm (SSA) [33].
- The whale optimization algorithm (WOA) [34].
- Sine Cosine Algorithm (SCA) [35].
- Dragonfly algorithm (DA) [36].
- Slime mould algorithm (SMA) [37].
- Prairie dog optimization algorithm (PDO) [28].

The experimental setup details the specific hardware and software specs and the study configuration employed. Every comparison technique was evaluated under the same conditions, using the initial parameter values, and executed for 50 iterations with 10 and 100 dimensions. The studies used Matlab 2018a on a Windows 11 PC with a Core i7 processor and 16 GB of RAM.

The worst, average, best, and standard deviation (STD) markers are employed to communicate the results of the utilized algorithms. Furthermore, when the p-value is below 0.08, the Wilcoxon rank-sum test obtains statistical evidence to ascertain if PDO-DE differs significantly from other approaches. The values for the essential parameters of the utilized algorithms are displayed in Table 1.

3.1. Qualitative analysis

This section evaluates the proposed PDO-DE approach on 23 different benchmark function challenges as described in [17]. Fig. 2 displays the conclusive outcomes of the comparison methodologies. The data presented in the figure demonstrates that the suggested technique consistently outperformed other strategies regarding execution time across all assessed tasks. The PDO-DE technique, which has been proposed, is considered the most efficient in terms of execution time.

Fig. 3 analyzes the behavior of the PDO-DE on benchmark functions F1-F13 to illustrate the function's structure, track the progress of the best solution, and evaluate its fitness achieved using the PDO-DE. Furthermore, the convergence curves of the PDO, DE, and

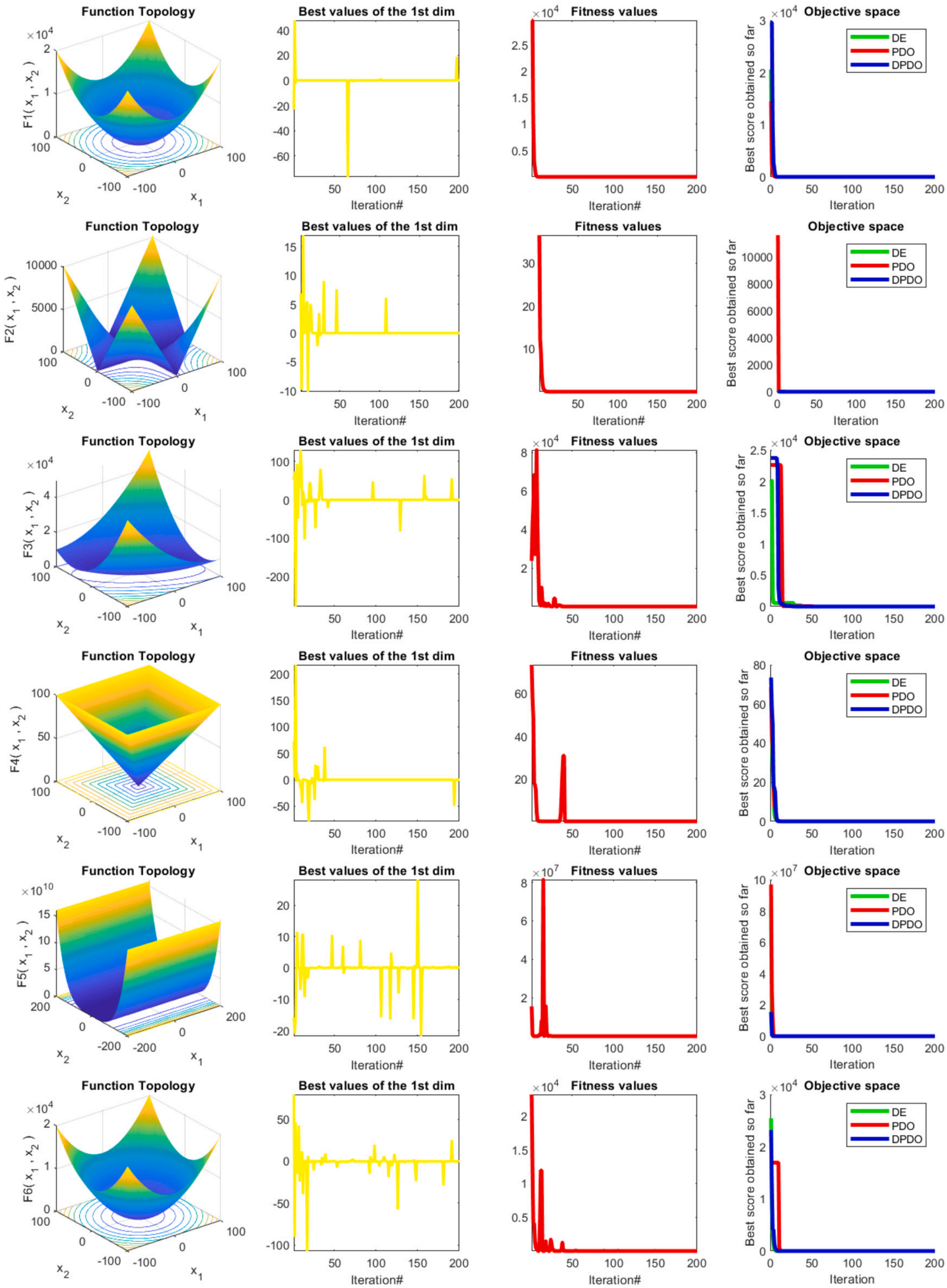

Fig. 3. Achieved qualitative experiment results for the tested 13 benchmark functions.

PDO-DE produced from the specified functions are depicted in the last column of Fig. 3. These figures demonstrate that the PDO-DE exhibits rapid convergence and high accuracy. An illustrative instance may be observed in F8, a function incorporating many modes; the PDO-DE has a faster convergence rate than the PDO and DE.

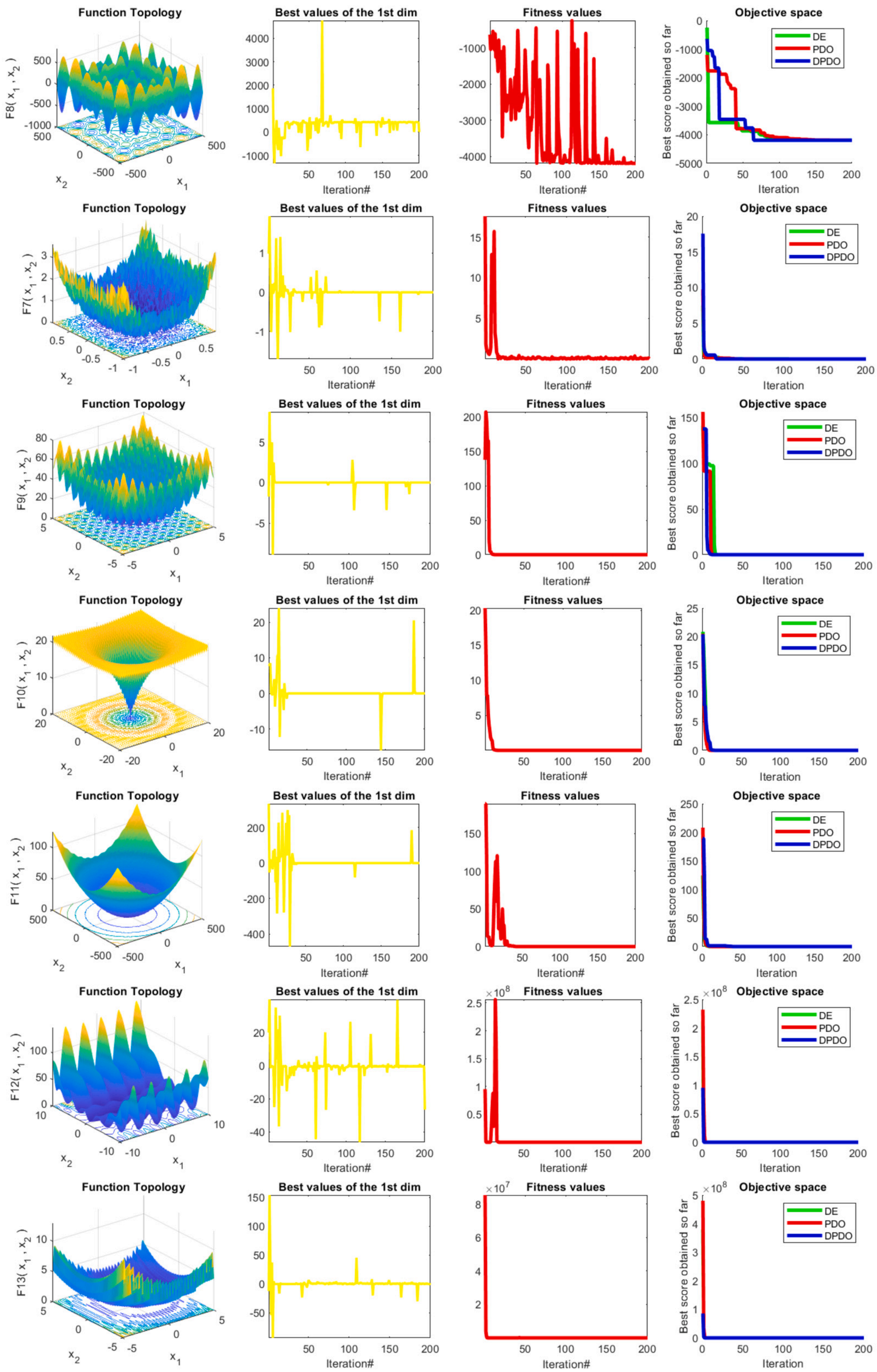

Fig. 3. (*continued*)

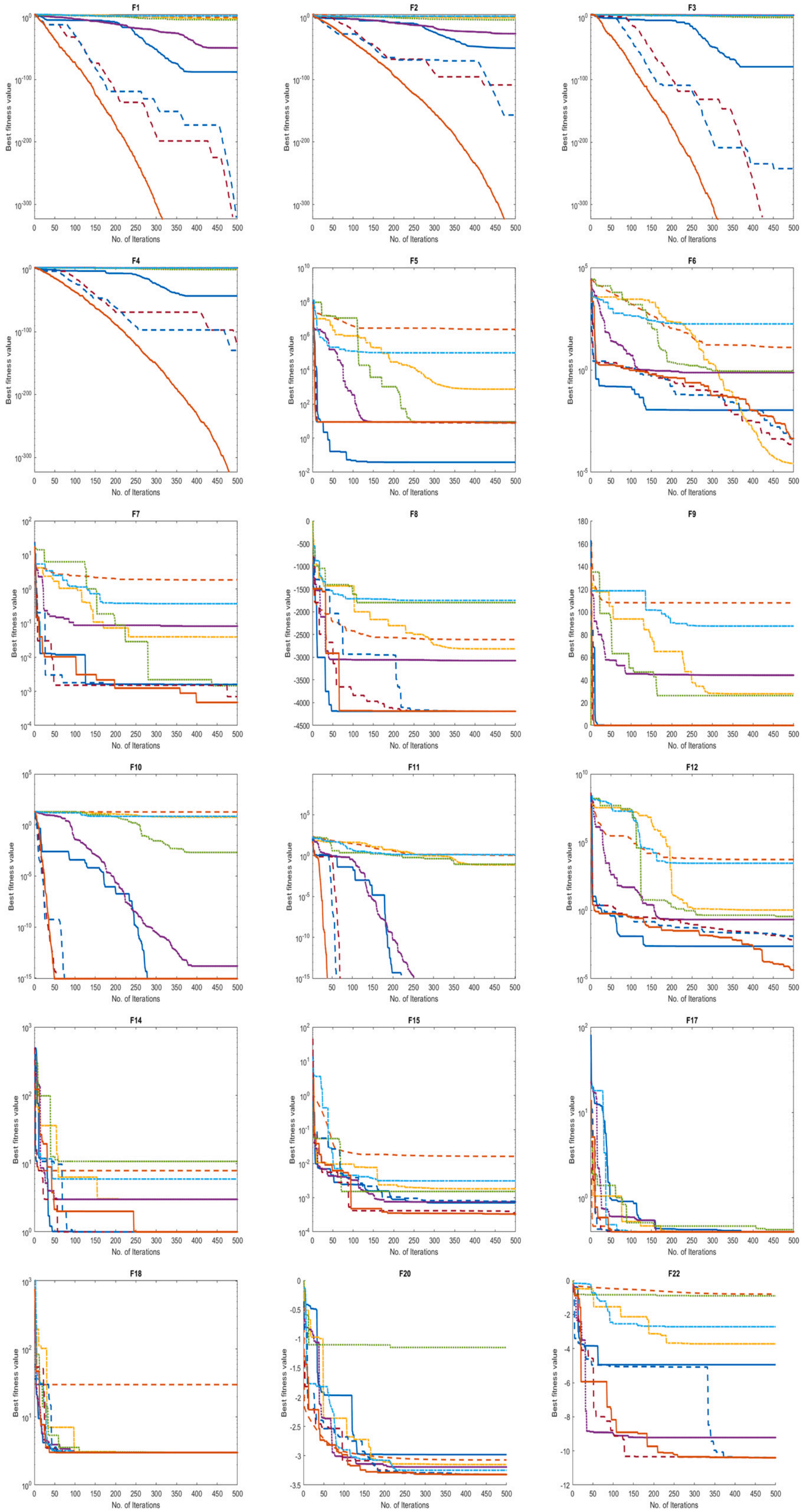

Fig. 4. A sample of qualitative analysis (Convergence behavior) of the benchmark functions based on 500 iterations.

Table 2
The results of ten benchmark functions (F1–F13) were obtained using comparative methodologies, with a dimension of 10 and 50 iterations.

Function	Measure	Comparative Algorithms								
		HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
Fun.1	Best	1.28997E-09	11971.93256	7885.284511	45.62434179	1983.798739	12811.3806	99.831123	3.63275E-52	1.95224E-49
	Average	4.19526E-10	8206.782535	5336.310064	16.38148275	490.7813722	5453.109966	47.85241519	7.26551E-53	3.90449E-50
	Worst	1.43254E-15	5757.767204	2226.687978	0.312336697	14.80068434	589.7900284	8.323702858	1.57931E-73	1.00407E-68
	STD	5.98522E-10	2331.678606	2049.129172	21.24004827	848.294567	4645.005825	42.14679608	1.62462E-52	8.73069E-50
	p-value	0.155670615	4.9113E-05	0.000394632	0.122889212	0.231875304	0.030409481	0.034777009	0.347440729	1
	h	0	1	1	0	0	1	1	0	0
Fun.2	Best	5.43778E-05	61.47531981	38.50814342	1.563159862	10.16308201	26.37318095	3.236703785	2.14632E-33	9.01812E-26
	Average	1.51381E-05	43.57550598	23.90481536	0.391425274	4.538211729	15.9159887	2.08918866	6.21973E-34	1.81024E-26
	Worst	9.65763E-09	32.67848292	9.006642887	0.021148146	1.83817754	5.592412467	1.095270907	5.98125E-40	5.84278E-36
	STD	2.35916E-05	11.30557147	10.95009294	0.662960972	3.672698506	7.474573175	0.998288818	9.37233E-34	4.02935E-26
	p-value	0.189252823	2.54495E-05	0.001222047	0.223282872	0.024559848	0.001424289	0.001582748	0.344512642	1
	h	0	1	1	0	1	1	1	0	0
Fun.3	Best	0.023413869	28737.58635	11789.13314	33757.10835	13028.15941	9870.413501	1120.21339	3.04372E-46	4.73504E-47
	Average	0.004760219	13285.44943	6956.74007	22277.8398	6854.41795	6522.64401	632.3847143	6.08743E-47	9.47076E-48
	Worst	6.84425E-11	3270.271734	1982.525295	9366.631524	3687.180954	3400.212739	286.220402	2.69565E-70	2.65509E-68
	STD	0.010428185	9476.596844	4218.795843	11150.20406	3925.058956	2704.220485	397.6768708	1.36119E-46	2.11754E-47
	p-value	0.337262571	0.013914193	0.006154552	0.002089846	0.004513396	0.000651088	0.007446292	0.428274379	1
	h	0	1	1	1	1	1	1	0	0

3.2. Simulation and experiments results of the benchmark functions

This section showcases the efficacy of PDO-DE. The implementation evaluation assesses its average, best, worst, and standard deviation (STD) values. In addition, we will do the Wilcoxon rank-sum test with a significance level of 0.08 to assess whether there is a significant difference between the proposed variant and its counterparts. Upon conducting the evaluations, we discovered that the PDO-DE demonstrates a high level of proficiency in managing most of the benchmark functions compared to comparable cutting-edge alternatives. To establish the ultimate ranking of the proposed PDO-DE, we employed the Friedman ranking test. This demonstrated the efficacy of the PDO-DE as a potent instrument capable of producing outcomes comparable to the field's top performers.

3.2.1. Scalability analysis

The behavior and convergence of the PDO-DE are compared to the HHO, GOA, SSA, WOA, SCA, DA, SMA, and PDO. Fig. 4 displays the merged curves of the nine algorithms for the 23 benchmark functions in a ten-dimensional space. The graphs provide a means to discern the rate at which the PDO-DE converges and its level of accuracy. Here, combining the two algorithms shows that the performance is significantly enhanced. The various functions demonstrate the ability of the PDO-DE to avoid suboptimal solutions. For instance, in plots F12, F14, and F19, it is evident that the algorithm consistently identifies a lower value after a specific number of rounds. Two notable examples of convergence speed are F7 and F4, in which the proposed algorithms outperform other methods by achieving the global optimum in a smaller number of iterations.

The proposed PDO-DE is compared to other swarm-inspired algorithms currently used. This scenario compares the highest and lowest fitness values, the average fitness values of the 20 separate runs, and the Standard Deviation (STD). A ranking is determined by the optimal value achieved for each algorithm. The results of the eight methods and the PDO-DE over the 23 benchmark functions in 10 dimensions are displayed in Tables 2 and 3. The values for functions F1-13 are reported in Table 2, whereas those for functions F14-F23 are listed in Table 3. According to the ranking summary shown in Table 4, the PDO-DE is ranked first in all 23 functions across ten dimensions.

The HHO is ranked second for the F1-F23, while the basic PDO is ranked third. Lastly, the ultimate position is the GOA algorithm. The p-values derived from the Wilcoxon signed-rank test provide conclusive evidence that the algorithms are statistically distinct in most cases. This fact is evident since the p-value of the majority of experiments is below 0.008. Nevertheless, in certain functions, such as F18, the PDO-DE method has surpassed other comparing algorithms.

To assess the effectiveness of the proposed PDO-DE method on complex problems, the dimensionality of the 13 benchmark functions is adjusted to 100. They employed the functions F1-F13 in this instance. The nine comparing algorithms and PDO-DE results over the 13 selected functions are presented in Table 5. Table 6 shows the final rank of tested algorithms. According to the table, the PDO-DE ranks first, followed by HHO, and WOA is in third position. Ultimately, the most unfavorable approach for this series of studies is the GOA.

Regarding Wilcoxon's rank test, the p-values indicate instances where the PDO-DE shares similarities with specific algorithms. For instance, F4 exhibits statistical resemblances to GOA, SSA, SCA, DA, SMA, and PDO. Nevertheless, it exhibits statistical dissimilarity compared to AO, HHO, and WOA. In addition, F5 shares common characteristics with all other approaches. However, in functions such as F8, the PDO-DE exhibits statistical dissimilarity compared to the other techniques. These circumstances arise as a result of the inherent character of the difficulties. Greater dimensionality presents challenges in locating the global optimum.

Table 2 (*continued*)

Function	Measure	Comparative Algorithms								
		HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
Fun.4	Best	9.49879E-07	64.41362376	50.17413726	80.99139237	66.00901277	55.37519706	14.74527955	3.15907E-21	1.85415E-27
	Average	4.21689E-07	53.36955841	38.22797809	62.67235191	37.68376843	36.09814145	8.92860099	6.31815E-22	4.15177E-28
	Worst	3.24453E-09	44.09679667	21.32962513	46.55400024	20.53089786	19.03733006	4.884795802	3.79981E-33	1.56398E-34
	STD	4.00444E-07	7.380152511	11.45096534	14.82426394	17.25288129	13.49586322	4.451242181	1.41278E-21	8.09964E-28
	p-value	0.046339898	2.14972E-07	7.16226E-05	1.28936E-05	0.001218138	0.000330345	0.002041534	0.346593592	1
	h	1	1	1	1	1	1	1	0	0
Fun.5	Best	8.918375336	44730188.96	3213097.928	136985.3253	9838643.076	5013344.415	6219.218674	8.991062471	8.992231913
	Average	4.740711238	19849170.49	2123623.847	33401.17873	2529770.734	2037389.939	2884.556686	8.937100415	8.894803122
	Worst	0.392435818	5311932.717	1262784.812	63.35960624	97968.60179	569008.5357	202.6773502	8.738921646	8.771253446
	STD	4.226842581	15451073.27	791886.6503	58360.84764	4115614.48	1805838.475	2349.539474	0.110818943	0.112524417
	p-value	0.059288707	0.02074732	0.000324658	0.236609927	0.206569484	0.035654398	0.025576193	0.565832204	1
	h	0	1	1	0	0	1	1	0	0
Fun.6	Best	0.230930855	14013.12788	7601.707043	41.67285302	2281.508134	7014.286918	242.1220508	1.63249874	1.782415675
	Average	0.119332361	6479.804196	4491.277301	15.6146427	870.3945091	3720.1229	119.9239546	1.136680823	1.571059344
	Worst	0.040121736	2389.00061	1807.026786	0.711566976	4.654498555	614.4322591	33.2407776	0.78283519	0.901687033
	STD	0.085229633	4406.640741	2723.174269	17.59542404	952.4821375	2801.001619	84.95531749	0.325677786	0.376299852
	p-value	3.03304E-05	0.011067216	0.006160182	0.112214542	0.075712741	0.017910552	0.014334949	0.086760553	1
	h	1	1	1	0	0	1	1	0	0
Fun.7	Best	0.01055495	15.42330689	2.363815093	0.765613408	0.748938596	1.791117249	0.089934661	0.018451571	0.009921997
	Average	0.007181684	9.532973139	1.030339803	0.306693208	0.485099142	0.693259753	0.052227799	0.006601459	0.005848939
	Worst	0.001050772	4.709756941	0.275812638	0.022492358	0.274248246	0.036234242	0.033921544	0.001122041	0.001398
	STD	0.00368364	5.202352889	0.846798712	0.277966312	0.172823695	0.756028517	0.023772218	0.00683874	0.003865278
	p-value	0.592026994	0.003461904	0.026854592	0.041858376	0.000259655	0.076489155	0.002594808	0.83574616	1
	h	0	1	1	1	1	0	1	0	0
Fun.8	Best	-2295.512188	-1761.986199	-1299.717717	-1915.207734	-1350.221326	-1499.985997	-1567.362794	-1982.61448	-1825.540916
	Average	-3084.313124	-1952.868062	-1868.606746	-2403.600699	-1504.585779	-1789.723975	-2104.890732	-3426.732872	-3239.416954
	Worst	-3865.90875	-2053.317165	-2733.68856	-3108.065206	-1705.43066	-2223.606729	-2679.455607	-4187.657543	-4179.432394
	STD	579.348396	120.1544767	537.0522801	599.9908775	138.6836573	288.1373349	404.3948587	1044.924367	915.6222581
	p-value	0.757100373	0.014332099	0.020272859	0.126155608	0.003042313	0.009683605	0.0350108	0.770738089	1
	h	0	1	1	0	1	1	1	0	0
Fun.9	Best	2.60797E-06	115.8158485	95.26968919	69.0215843	58.06632219	75.10691224	34.63086015	0	9.21187E-07
	Average	9.80191E-07	105.2691236	76.52610266	35.53589185	44.01162984	52.66718786	25.44370869	0	1.84237E-07
	Worst	4.43379E-12	97.80282241	63.50309523	10.6821462	29.5942212	29.76489308	11.65940646	0	0
	STD	1.34663E-06	8.37452016	11.90074753	25.90988657	10.8566309	16.59832693	9.855806657	0	4.11967E-07
	p-value	0.241864489	2.7724E-09	5.34663E-07	0.015422398	1.75788E-05	0.000102484	0.000418027	0.346593507	1
	h	0	1	1	1	1	1	1	0	0
Fun.10	Best	7.08246E-06	19.77142424	18.51302534	5.733790561	20.26629422	17.30893877	4.249701841	4.723558315	8.88178E-16
	Average	3.85376E-06	19.21215325	15.5174961	3.078996027	17.70413485	16.4167313	3.630017973	3.526854778	8.88178E-16
	Worst	8.01585E-07	18.59062547	11.53821459	0.151827402	13.40550502	15.37092417	2.970834836	2.793001723	8.88178E-16
	STD	2.84875E-06	0.531898732	3.338454594	2.287464468	3.408037088	0.922951505	0.572333603	0.843446955	0
	p-value	0.01643563	6.15808E-13	6.35862E-06	0.016818291	2.74586E-06	1.75626E-10	5.9465E-07	1.39863E-05	0
	h	1	1	1	1	1	1	1	1	0
Fun.11	Best	5.12703E-06	103.384885	86.29789546	2.481173823	39.37928692	21.91272532	2.286446089	172.1793446	4.13E-06
	Average	1.19299E-06	63.45646696	58.3459853	1.317520412	13.44380745	15.80909639	1.408677678	118.7951121	1.19299E-06
	Worst	5.61138E-10	34.451969	23.7254226	0.007324642	2.335130628	5.823029868	0.950335177	81.83212487	3.41E-10
	STD	2.21821E-06	31.20581916	28.2923524	0.996703685	15.7261065	6.256486872	0.548331464	35.9666052	2.21821E-06
	p-value	0.263513745	0.001881775	0.001729642	0.018262816	0.092310139	0.000481329	0.000431703	7.72552E-05	0.223513745
	h	0	1	1	1	0	1	1	1	0
Fun.12	Best	0.061557963	24803720.04	2975187.252	1490044.516	12582070.15	70714123.97	9.610626284	8.317016755	0.734926385
	Average	0.024444536	14118112.02	1535115.283	298570.1025	2899023.431	14459173.38	4.38278262	5.798464497	0.423059432
	Worst	0.001156394	6652617.077	244868.1401	13.93254916	10.1600239	68.22818789	1.392209899	1.059746569	0.088352714
	STD	0.022742396	7709517.612	1166210.984	666054.571	5475470.742	31450162.36	3.400393216	2.971463393	0.275406293
	p-value	0.01213972	0.003462467	0.018612696	0.34552344	0.270433882	0.334012496	0.031845679	0.003799645	1
	h	1	1	1	0	0	0	1	1	0
		1	8	6	5	7	9	3	4	2
Fun.13	Best	0.25110391	63605678.94	35751345.12	363541.1302	7553775.192	16600183.25	15.41469564	198.3423261	0.899568549
	Average	0.108785796	27494480.14	23715547.81	72720.03824	2499055.155	6721378.636	13.12903408	44.73300118	0.852974889
	Worst	0.000101863	4112570.779	201711.8416	0.573557845	159556.9819	2754809.999	10.32544087	0.402952365	0.817248493
	STD	0.106359219	28226572.24	13721166.49	162573.9337	2901352.622	5796045.328	1.838999879	85.96105367	0.042851338
	p-value	4.97818E-07	0.061045584	0.004776352	0.346506412	0.090273731	0.031961019	4.0111E-07	0.286703037	1

Table 3
The results of ten benchmark functions (F14–F23) were obtained using comparative methodologies, with a dimension of 10 and 50 iterations.

Function	Measure	Comparative Algorithms								
		HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
Fun.14	Worst	1.1719E+01	2.2901E+01	2.2901E+01	1.2671E+01	1.6443E+01	1.8304E+01	1.3619E+01	2.0153E+01	1.2671E+01
	Average	5.9822E+00	1.5652E+01	1.5590E+01	5.9009E+00	9.6506E+00	1.4139E+01	8.4176E+00	1.1087E+01	4.8415E+00
	Best	1.2456E+00	1.0763E+01	3.9683E+00	9.9804E-01	2.9821E+00	5.9288E+00	1.9920E+00	3.9683E+00	9.9807E-01
	STD	4.0348E+00	4.8709E+00	7.8680E+00	4.4563E+00	6.0587E+00	5.3213E+00	4.7700E+00	6.0000E+00	4.8999E+00
	p-value	9.7661E-01	1.0814E-02	4.3444E-02	1.0000E+00	2.9730E-01	2.9068E-02	4.1372E-01	1.5938E-01	7.2985E-01
	h	0	1	1	0	0	1	0	0	0
Fun.15	Worst	7.2916E-03	3.6538E+00	9.4642E-02	4.6014E-02	4.9513E-02	3.5854E-02	2.1157E-02	5.2012E-02	4.3638E-03
	Average	2.9623E-03	9.3275E-01	4.4021E-02	1.8504E-02	2.0559E-02	1.6282E-02	1.5317E-02	3.1430E-02	2.2102E-03
	Best	3.8597E-04	2.0958E-02	2.2525E-02	7.3080E-03	1.4603E-03	2.6888E-03	1.2736E-03	2.2901E-03	1.4122E-03
	STD	2.6074E-03	1.5597E+00	2.9278E-02	1.6481E-02	1.7685E-02	1.3698E-02	8.6086E-03	2.6401E-02	1.2182E-03
	p-value	4.3212E-02	2.3244E-01	4.9542E-01	3.8022E-01	4.6625E-01	2.8772E-01	2.3063E-01	1.0000E+00	3.8577E-02
	h	1	0	0	0	0	0	0	0	1
Fun.16	Worst	-1.0210E+00	3.8734E+02	-1.0316E+00	-2.1445E-01	-1.0170E+00	-2.1546E-01	-9.9564E-01	-1.0069E+00	-1.0310E+00
	Average	-1.0276E+00	8.0406E+01	-1.0316E+00	-8.2643E-01	-1.0256E+00	-8.6834E-01	-1.0244E+00	-1.0262E+00	-1.0314E+00
	Best	-1.0316E+00	-2.1546E-01	-1.0316E+00	-1.0313E+00	-1.0313E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00
	STD	4.3640E-03	1.7167E+02	7.7954E-06	3.4343E-01	5.8470E-03	3.6497E-01	1.6080E-02	1.0840E-02	2.5076E-04
	p-value	7.9416E-01	3.1979E-01	2.9795E-01	2.2972E-01	9.0874E-01	3.6189E-01	8.3896E-01	1.0000E+00	3.1382E-01
	h	0	0	0	0	0	0	0	0	0
Fun.17	Worst	4.4692E-01	1.1244E+01	2.7054E+00	2.8830E+00	1.2106E+00	2.7054E+00	4.2099E-01	4.1417E-01	3.9792E-01
	Average	4.1742E-01	2.6013E+00	8.6001E-01	1.1714E+00	6.2511E-01	8.6096E-01	4.0392E-01	4.0153E-01	3.9790E-01
	Best	3.9812E-01	3.9789E-01	3.9790E-01	4.0209E-01	4.0359E-01	3.9789E-01	3.9790E-01	3.9790E-01	3.9789E-01
	STD	2.4023E-02	4.8321E+00	1.0316E+00	1.0003E+00	3.3463E-01	1.0311E+00	9.6552E-03	7.0916E-03	1.6249E-05
	p-value	1.9380E-01	3.3850E-01	3.4943E-01	1.2355E-01	1.7360E-01	3.4826E-01	6.6678E-01	1.0000E+00	2.8524E-01
	h	0	0	0	0	0	0	0	0	0
Fun.18	Worst	3.6003E+00	2.3565E+03	3.7096E+00	3.7405E+01	9.0288E+00	3.0109E+00	3.0008E+01	8.6436E+01	3.0010E+00
	Average	3.1524E+00	5.8498E+02	3.1467E+00	2.2820E+01	4.3326E+00	3.0025E+00	1.4098E+01	3.5888E+01	3.0003E+00
	Best	3.0009E+00	3.0000E+00	3.0000E+00	3.2353E+00	3.0762E+00	3.0000E+00	3.1496E+00	3.0000E+00	3.0000E+00
	STD	2.5434E-01	1.0000E+03	3.1468E-01	1.4750E+01	2.6263E+00	4.7530E-03	1.4523E+01	4.5041E+01	4.0173E-04
	p-value	1.4279E-01	2.5487E-01	1.4274E-01	5.5467E-01	1.5647E-01	1.4120E-01	3.3334E-01	1.0000E+00	1.4118E-01
	h	0	0	0	0	0	0	0	0	0
Fun.19	Worst	-3.0798E+00	-6.4296E-01	-3.7898E+00	-3.5186E+00	-3.6380E+00	-3.5147E+00	-3.0842E+00	-1.0008E+00	-3.8366E+00
	Average	-3.6176E+00	-1.9943E+00	-3.8379E+00	-3.7489E+00	-3.7665E+00	-3.7206E+00	-3.7064E+00	-2.7177E+00	-3.8506E+00
	Best	-3.8051E+00	-3.1028E+00	-3.8620E+00	-3.8457E+00	-3.8424E+00	-3.8625E+00	-3.8626E+00	-3.8628E+00	-3.8603E+00
	STD	3.0633E-01	9.0107E-01	3.1344E-02	1.3186E-01	7.8732E-02	1.5799E-01	3.4781E-01	1.5673E+00	1.1202E-02
	p-value	6.7977E-01	4.1567E-03	4.2413E-01	8.0448E-01	7.1599E-01	9.3562E-01	1.0000E+00	2.0579E-01	3.8120E-01
	h	0	1	0	0	0	0	0	0	0
Fun.20	Worst	-2.0753E+00	-1.7777E-01	-2.3878E+00	-1.4342E+00	-6.5296E-01	-7.6773E-01	-2.3077E+00	-2.2505E+00	-3.0711E+00
	Average	-2.4745E+00	-8.8126E-01	-2.6962E+00	-1.7743E+00	-1.8693E+00	-2.5832E+00	-3.0332E+00	-2.9160E+00	-3.1857E+00
	Best	-2.8739E+00	-2.3590E+00	-3.1169E+00	-2.4107E+00	-2.7531E+00	-3.3102E+00	-3.3084E+00	-3.2303E+00	-3.3095E+00
	STD	3.0572E-01	8.6990E-01	3.1224E-01	3.8837E-01	9.0514E-01	1.0493E+00	4.1360E-01	3.8199E-01	1.1088E-01
	p-value	7.8327E-02	1.3751E-03	3.4838E-01	1.5688E-03	4.4390E-02	5.2396E-01	6.5379E-01	1.0000E+00	1.6787E-01
	h	0	1	0	1	1	0	0	0	0
Fun.21	Worst	-2.0889E+00	-1.9987E-01	-1.3718E+00	-1.7921E+00	-3.5136E-01	-1.3646E+00	-2.5184E+00	-2.2309E+00	-5.0514E+00
	Average	-3.4939E+00	-5.6869E-01	-3.5371E+00	-2.8124E+00	-6.0047E-01	-3.9053E+00	-6.6250E+00	-6.8756E+00	-8.6921E+00
	Best	-4.2719E+00	-1.1749E+00	-9.9026E+00	-4.1255E+00	-8.7131E-01	-8.0340E+00	-9.7152E+00	-1.0152E+01	-1.0141E+01
	STD	9.8064E-01	3.6706E-01	3.5871E+00	9.4735E-01	2.2715E-01	2.6788E+00	3.7257E+00	4.1579E+00	2.0783E+00
	p-value	9.7946E-04	2.5703E-05	2.3908E-02	4.2594E-04	2.4691E-05	1.3456E-02	3.1020E-01	4.0770E-01	1.0000E+00
	h	1	1	1	1	1	1	0	0	0
Fun.22	Worst	-3.0082E+00	-2.6133E-01	-8.9916E-01	-9.2803E-01	-5.1607E-01	-1.8009E+00	-1.5675E+00	-2.4415E+00	-5.1022E+00
	Average	-4.1112E+00	-6.6870E-01	-2.4891E+00	-4.4755E+00	-6.8023E-01	-4.0724E+00	-7.9722E+00	-4.8132E+00	-8.0327E+00
	Best	-4.9090E+00	-1.3529E+00	-4.5931E+00	-9.1062E+00	-1.0204E+00	-8.1938E+00	-1.0062E+01	-9.6436E+00	-1.0335E+01
	STD	7.5261E-01	4.3117E-01	1.4439E+00	3.4132E+00	2.3082E-01	2.9590E+00	3.6284E+00	2.8541E+00	2.3852E+00
	p-value	8.0094E-03	1.3874E-04	2.1512E-03	9.2497E-02	1.2957E-04	4.8161E-02	9.7592E-01	8.8969E-02	1.0000E+00
	h	1	1	1	0	1	1	0	0	0
Fun.23	Worst	-3.3579E+00	-3.6921E-01	-2.0723E+00	-6.5146E-01	-5.5426E-01	-2.0594E+00	-9.2807E-01	-2.3957E+00	-5.1031E+00
	Average	-4.3874E+00	-6.1584E-01	-2.4152E+00	-1.8320E+00	-1.0302E+00	-3.0600E+00	-4.2707E+00	-4.8528E+00	-7.1683E+00
	Best	-5.0393E+00	-8.3605E-01	-2.7864E+00	-4.5749E+00	-1.6175E+00	-5.0958E+00	-1.0482E+01	-1.0489E+01	-1.0309E+01
	STD	6.5103E-01	1.8871E-01	3.1747E-01	1.5791E+00	4.1332E-01	1.1755E+00	3.6664E+00	3.6255E+00	2.8245E+00
	p-value	6.4243E-02	8.4710E-04	5.7110E-03	6.1531E-03	1.3416E-03	1.7000E-02	1.9910E-01	2.9256E-01	1.0000E+00
	h	0	1	1	1	1	1	0	0	0

Table 4
The final rank of all benchmark functions (F1–F23) were obtained using comparative methodologies, with a dimension of 10 and 50 iterations.

Function	Comparative Algorithms								
	HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
Fun.1	3	9	7	4	6	8	5	1	2
Fun.2	3	9	8	4	6	7	5	1	2
Fun.3	3	8	7	9	6	5	4	2	1
Fun.4	3	8	7	9	6	5	4	2	1
Fun.5	1	9	7	5	8	6	4	3	2
Fun.6	1	9	8	4	6	7	5	2	3
Fun.7	3	9	8	5	6	7	4	2	1
Fun.8	3	6	7	4	9	8	5	1	2
Fun.9	3	9	8	5	6	7	4	1	2
Fun.10	2	9	6	3	8	7	5	4	1
Fun.11	2	8	7	3	5	6	4	9	1
Fun.12	1	9	8	5	6	7	3	4	2
Fun.14	3	9	8	2	5	7	4	6	1
Fun.15	2	9	8	5	6	4	3	7	1
Fun.16	3	9	1	8	5	7	6	4	2
Fun.17	4	9	6	8	5	7	3	2	1
Fun.18	7	9	2	4	3	5	6	8	1
Fun.19	7	9	2	4	3	5	6	8	1
Fun.20	6	9	4	8	7	5	3	2	1
Fun.21	6	9	5	7	8	4	2	3	1
Fun.22	5	9	7	4	8	6	2	3	1
Fun.23	3	9	6	7	8	5	4	2	1
Sum	74	192	137	117	136	135	91	77	31
Mean	3.217391304	8.347826087	5.956521739	5.086956522	5.913043478	5.869565217	3.956521739	3.347826087	1.347826087
Rank	2	9	8	5	7	6	4	3	1

Table 5
The results of ten benchmark functions (F1–F13) were obtained using comparative methodologies, with a dimension of 100 and 50 iterations.

Function	Measure	Comparative Algorithms								
		HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
Fun.1	Worst	1.76931E-07	154774.0192	150570.4384	2123.1015	82917.77991	113800.2758	54435.00866	44381.34975	2.09496E-49
	Average	3.65542E-08	137000.1645	134484.2116	435.0713096	50266.35878	97977.12398	38731.99932	40312.94172	4.18991E-50
	Best	3.2267E-15	107256.7989	96133.01906	2.259843439	5713.731376	86584.06291	7217.703643	38114.60316	1.54712E-70
	STD	7.85093E-08	18796.45177	21825.5547	943.715732	28367.63991	10621.11906	18952.24666	2646.650501	9.36892E-50
	p-value	0.328257881	2.02195E-07	7.43245E-07	0.332757019	0.004164026	3.19576E-08	0.001826343	6.03405E-10	1
	h	0	1	1	0	1	1	1	1	0
Fun.2	Worst	0.000106133	2.08606E+46	3.06307E+26	9.425490246	156.188196	459.149022	169.4851854	4.00096E+17	7.70236E-29
	Average	3.37409E-05	4.17213E+45	6.12614E+25	3.097594371	70.93708374	356.1665865	143.9531926	8.62595E+16	1.54067E-29
	Best	6.17085E-10	2.69155E+28	6.7979E+13	0.072292008	31.72781844	221.8311124	126.5961232	64970000980	8.77321E-36
	STD	4.76582E-05	9.32916E+45	1.36985E+26	3.927497094	48.88160854	103.5455433	19.54740611	1.75616E+17	3.44449E-29
	p-value	0.152060385	0.346593507	0.346593197	0.115814653	0.011789253	5.78959E-05	1.86557E-07	0.30401841	1
	h	0	0	0	0	1	1	1	0	0
Fun.3	Worst	2308746.554	1319107.747	1280200.38	2798461.015	1201898.125	1685831.734	255582.492	274495.3283	1.87989E-27
	Average	780279.4934	852450.6173	862632.0386	1713706.729	854772.5188	834214.9299	202886.4419	221551.5934	3.76147E-28
	Best	0.001419473	499265.663	471517.0919	874615.1747	527882.046	530366.7932	169029.6623	161413.8048	6.73824E-65
	STD	1092863.214	348523.044	357129.8493	739132.5119	253831.9357	491437.931	35046.42053	51482.82724	8.40616E-28
	p-value	0.149043625	0.000595021	0.000645148	0.000838222	6.73465E-05	0.005268685	1.20093E-06	1.13029E-05	1
	h	0	1	1	1	1	1	1	1	0

(*continued on next page*)

3.2.2. The experimental results of CEC2019 benchmark functions

This section evaluates the proposed PDO-DE approach on ten different CEC2019 challenges. Table 7 compares the fitness function values for various tactics about the worst, mean, and best outcomes in the context of the CEC2019 concerns. Table 8 unequivocally demonstrates that the proposed strategy outperformed all comparable solutions for nearly all cases. The PDO-DE technique performs superior to HHO, GOA, SSA, WOA, SCA, DA, SMA, and PDO in addressing the issue of cec05. Similarly, it outperformed HHO, GOA, SSA, WOA, SCA, DA, SMA, and PDO in solving the problem of cec9. This information is based on the Wilcoxon signed-rank test.

Furthermore, Fig. 5 illustrates the convergence patterns of the different approaches to the CEC2019 problems. The figure illustrates that the proposed strategy outperformed alternative methods. The proposed strategy addresses the main shortcomings of the original PDO method, such as the imbalance between the search phases, by effectively eliminating local optima and premature convergence. The results demonstrate the exceptional ability of the proposed PDO-DE technique to surpass other approaches on many test problems from CEC 2019. The main goal of this study has been accomplished as the suggested approach has demonstrated more significant

Table 5 (*continued*)

Function	Measure	Comparative Algorithms								
		HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
Fun.4	Worst	0.000368775	86.25787312	89.15566619	97.98209565	98.89015396	99.1565944	84.20026226	58.82250147	1.01973E-28
	Average	8.8414E-05	80.82718833	80.30069832	91.81223897	97.71829671	84.60720885	73.39936721	52.06773265	2.04156E-29
	Best	1.18711E-07	73.37373567	72.15793992	82.77960668	96.71759347	63.33604087	58.70646464	47.22842632	6.1132E-34
	STD	0.000157756	5.666183354	6.036418068	7.382191385	0.958873652	15.85208409	10.30880823	4.970346257	4.55921E-29
	p-value	0.245518783	1.01615E-09	1.76907E-09	3.01665E-09	1.53951E-16	2.23509E-06	2.4259E-07	1.17287E-08	1
	h	0	1	1	1	1	1	1	1	0
Fun.5	Worst	108.4852901	446626104.4	420490489.3	18302201.88	1110726269	317971923.9	58638618.89	61486475.6	98.97815854
	Average	60.98672773	358148004.3	362823792.9	7111486.272	818925900.4	160161595.5	45495196.7	49511272.11	98.96355308
	Best	0.123446368	276771247.7	295814347.9	107.2278972	392426419	5782045.954	14149873.04	36093938.84	98.93986636
	STD	55.69822205	66699867.12	59086872.91	7490699.487	304233680.2	122284973.5	17827778.33	9037812.319	0.016014401
	p-value	0.1658609	2.13481E-06	7.63315E-07	0.06653059	0.000316635	0.019036947	0.000451108	1.83221E-06	1
	h	0	1	1	0	1	1	1	1	0
Fun.6	Worst	3.587911105	173350.5552	158188.3638	1737.329475	188216.8765	188060.4766	39331.6315	66914.73797	24.30480794
	Average	1.049241909	151623.3721	147369.2061	820.6973282	157464.8736	122533.0155	29123.69942	47321.00107	23.24489931
	Best	0.294051482	137794.6673	128693.3633	107.4082302	121641.3904	69821.26379	15393.7859	28805.30355	21.85613056
	STD	1.428051413	13833.16617	13008.25423	721.145495	25382.53779	43162.4808	10858.66454	13549.19814	1.117730732
	p-value	3.42495E-09	8.21148E-09	6.3242E-09	0.038549575	7.06215E-07	0.000221387	0.000326107	5.21004E-05	1
	h	1	1	1	1	1	1	1	1	0
Fun.7	Worst	0.018762252	2277.033695	816.8066801	83.70773953	1518.765094	566.1764066	67.71504833	2434.789395	0.023325755
	Average	0.010300428	1847.184447	615.1519059	22.90314801	1145.264275	344.4326626	64.85669861	2135.221915	0.013896918
	Best	0.004052908	1479.605841	436.308469	0.033165026	743.401764	80.46083387	61.58574648	1698.326756	0.005418321
	STD	0.006067214	302.6366952	177.618793	35.42926356	340.7006566	227.4828184	2.462896565	286.1567027	0.007170569
	p-value	0.416809189	7.99623E-07	5.51421E-05	0.186564687	6.8208E-05	0.009563405	7.6988E-12	1.684E-07	1
	h	0	1	1	0	1	1	1	1	0
Fun.8	Worst	-17615.57079	-6813.976316	-5738.841091	-19118.65584	-3619.849329	-5946.994434	-5640.278432	-2856.114888	-8890.802802
	Average	-23132.14928	-7624.65436	-6506.560952	-25047.99078	-4537.748324	-6752.987087	-6681.393465	-3367.987467	-23677.26208
	Best	-28851.52063	-9147.274167	-7894.436244	-30054.44693	-5523.328705	-7508.030249	-8419.82693	-3804.895933	-41853.84451
	STD	4784.536466	920.6571719	878.4780418	4991.551116	771.5230559	649.6987099	1130.86314	435.9126604	14644.16776
	p-value	0.93888097	0.040167993	0.030786533	0.847901346	0.019337716	0.032530928	0.032240484	0.014671994	1
	h	0	1	1	0	1	1	1	1	0
Fun.9	Worst	3.32024E-06	1728.913985	1302.520686	771.6592502	1195.052809	1312.347934	957.2384815	1525.721208	2.32024E-06
	Average	8.71556E-07	1556.615419	1204.597914	210.2107839	827.9450905	1162.660386	875.6272656	1417.327232	7.71556E-07
	Best	5.71845E-11	1351.205953	1144.880922	0.184007254	548.883542	901.2253916	832.9371364	1263.808071	5.21845E-11
	STD	1.39839E-06	134.4997748	64.81106813	323.6500593	242.5363904	159.7622236	54.73283559	109.8798932	1.39839E-06
	p-value	0.200919411	5.33449E-09	1.23757E-10	0.184475659	6.11184E-05	2.04627E-07	4.08341E-10	2.25922E-09	0.100919411
	h	0	1	1	0	1	1	1	1	0
Fun.10	Worst	6.24201E-05	20.92412813	20.16192008	4.985205565	20.75652656	19.96663981	16.87255484	18.37568674	8.88178E-16
	Average	1.43311E-05	20.78659556	19.83988609	1.77184925	20.60892593	19.2460516	16.1918604	17.77089404	8.88178E-16
	Best	4.36871E-07	20.62056121	19.60950815	0.124373589	20.07177384	18.54933033	15.15947194	17.37122244	8.88178E-16
	STD	2.69282E-05	0.111745011	0.268493187	1.951767877	0.300883619	0.636678128	0.639538327	0.426290659	0
	p-value	0.268151696	1.24975E-18	2.01388E-15	0.076864164	3.69445E-15	2.55405E-12	1.05196E-11	1.95827E-13	0
	h	0	1	1	0	1	1	1	1	0
Fun.11	Worst	6.61046E-08	1530.101704	1238.7058	36.23643738	1541.882634	1036.40333	467.120228	1239.294196	5.61046E-08
	Average	1.36207E-08	1290.210111	1141.038415	9.976231025	1042.514463	748.0353402	385.1672916	1133.212159	1.16207E-08
	Best	6.69464E-14	1069.604482	1067.293828	0.00156863	628.0759863	360.2758122	292.1822517	1056.502053	6.69464E-14
	STD	2.9344E-08	167.5634495	75.4824484	15.12021229	410.7760407	259.6200718	72.73670367	74.18157406	2.9344E-08
	p-value	0.329657168	1.31779E-07	6.40906E-10	0.17835664	0.000467715	0.000199849	2.37341E-06	5.89567E-10	0.329657168
	h	0	1	1	0	1	1	1	1	0
Fun.12	Worst	0.029798441	562355949.7	883487590.1	30552828.71	3204940264	582722991.1	100286114.4	30523036.75	1.254104865
	Average	0.019464771	410104734	637650523.6	10270709.74	2615417460	295340883.2	40488643.29	11482941.22	1.202472061
	Best	0.004351119	195377866.3	387405710.8	36.39857641	1758798165	120950279.3	8401825.335	3000586.304	1.154109766
	STD	0.012225088	150308005.7	196748792.4	14438228.91	627236882.5	208142591.8	35097781.95	11149837.45	0.047893386
	p-value	1.64791E-11	0.000289167	8.83147E-05	0.150355107	1.42799E-05	0.013138417	0.032640782	0.050245165	1
	h	1	1	1	0	1	1	1	0	0
Fun.13	Worst	2.057721021	2200851513	2343248299	14488414.95	4665922818	1725033079	137202005.2	95905372.27	9.999006118
	Average	0.702307366	1701750739	1453221576	2919497.236	3760545241	1221087949	103445962.5	66766252.74	9.979348889
	Best	0.006281799	1396333093	1068209570	85.10687709	2551907957	390845829.3	62891600.24	43996443.79	9.915841518
	STD	0.828865176	330931676	522180971.9	6467275.965	837474178.6	552931554.2	31985886.07	20932447.89	0.035995147
	p-value	7.00365E-09	2.96608E-06	0.000253031	0.342327358	8.23452E-06	0.001137769	8.9636E-05	9.88087E-05	1
	h	1	1	1	0	1	1	1	1	0

Table 6
The final rank of all benchmark functions (F1–F23) were obtained using comparative methodologies, with a dimension of 100 and 50 iterations.

Function	Comparative Algorithms								
	HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
Fun.1	2	9	8	3	6	7	4	5	1
Fun.2	2	9	8	3	4	6	5	7	1
Fun.3	4	6	8	9	7	5	2	3	1
Fun.4	2	6	5	8	9	7	4	3	1
Fun.5	1	7	8	3	9	6	4	5	2
Fun.6	1	8	7	3	9	6	4	5	2
Fun.7	1	8	6	3	7	5	4	9	2
Fun.8	3	4	7	1	8	5	6	9	2
Fun.9	2	9	7	3	4	6	5	8	1
Fun.10	2	9	7	3	8	6	4	5	1
Fun.11	2	9	8	3	6	5	4	7	1
Fun.12	1	7	8	3	9	6	5	4	2
Fun.13	1	8	7	3	9	6	5	4	2
Sum	24	99	94	48	95	76	56	74	19
Mean	1.846153846	7.615384615	7.230769231	3.692307692	7.307692308	5.846153846	4.307692308	5.692307692	1.461538462
Rank	2	9	7	3	8	5	4	6	1

outcomes in resolving various issues when compared to the initial method and other cutting-edge strategies. The findings refute the authors' claims, as the enhanced approach utilizes diverse search tactics to acquire more optimal solutions.

4. Real-world engineering problems

4.1. Problem 1: the multiple-disc clutch brake

The variables, restrictions, and objective functions of the multiple disc clutch brake design problem are as follows: The spatial arrangement of the five variables is depicted in Fig. 6.

Below is the mathematical model for the design challenge of a multiple-disc clutch brake:

Consider:

$$\vec{\lambda} = [\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5] = [R_i R_0 T X Y]$$

Objective function:

$$f(\lambda) = \prod (R_i^2 - R_0^2) T (Y + 1) p$$

Subject to:

$$g_1(\lambda) = L_{max} - (Y + 1)(T + \alpha) \geqslant 0$$

$$g_2(\lambda) = P_{max} - P_{RY} \geqslant 0$$

$$g_4(\lambda) = P_{\max} Z_{sRmax} - P_{RY} v_{sR} \geq 0$$

$$g_5(\lambda) = Z_{s\max} - Z_{sR} \geq 0$$

$$g_6(\lambda) = W_{\max} - W \geq 0$$

$$g_7(\lambda) = U_h - sU_s \geq 0$$

$$g_8(\lambda) = W \geq 0$$

Parameters range:

$$60 \leq \lambda_1 \leq 80,$$

$$90 \leq \lambda_2 \leq 110,$$

$$1 \leq \lambda_3 \leq 3,$$

$$600 \leq \lambda_4 \leq 1000,$$

$$2 \leq \lambda_5 \leq 9$$

Table 7
The results of CEC2019 benchmark functions were obtained using comparative methodologies, with a dimension of 10 and 50 iterations.

Function	Measure	Comparative Algorithms								
		HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
cec.1	Best	398012.1933	2.39355E+12	2.33046E+12	3.49328E+12	2.09722E+12	9.73244E+11	2.21186E+11	2.46364E+14	1038290.003
	Average	216361.2503	9.33671E+11	1.04921E+12	1.78695E+12	8.96678E+11	5.53332E+11	1.01576E+11	7.5383E+13	315292.7404
	Worst	71472.27264	66607434570	2.73955E+11	6.75504E+11	96342833129	1.82061E+11	15178373446	2.16046E+13	69800.31917
	STD	143601.2266	1.09119E+12	8.49956E+11	1.17498E+12	7.80283E+11	3.39455E+11	84324972447	9.7395E+13	407974.7526
	p-value	0.62283388	0.092064532	0.024664713	0.009350958	0.033147108	0.006542187	0.027347725	0.121752524	1
	h	0	0	1	1	1	1	1	0	0
cec.2	Best	17.87170826	20947.98818	5137.74478	516.7120372	682.3440416	4878.702554	222.7807158	33141.28489	19.4242182
	Average	17.63693367	7029.401156	3460.352154	122.2317711	267.1419636	3121.515661	94.07298316	25747.10792	18.59661544
	Worst	17.52389516	1243.814164	1363.194949	19.02775868	67.63357125	104.0409615	37.59506463	13419.43896	17.80412243
	STD	0.13641182	8288.559727	1575.301388	220.5690288	238.7836048	1953.609309	76.61196991	7474.383352	0.69852286
	p-value	0.016683005	0.095226625	0.001215996	0.324131594	0.048352205	0.007492498	0.058733975	5.75944E-05	1
	h	0	1	0	0	1	1	0	1	0
cec.3	Best	12.70396184	12.71021543	12.70499852	12.70614359	12.70539065	12.708156	12.70620309	12.70274979	12.70779236
	Average	12.70311891	12.70738739	12.70388397	12.70476926	12.70448014	12.70622774	12.70363887	12.70254071	12.704658
	Worst	12.70253134	12.70519206	12.70256479	12.70389505	12.70251337	12.70358091	12.70240991	12.70240492	12.70259269
	STD	0.000560988	0.002110914	0.001061888	0.000848709	0.001171941	0.001959259	0.001520485	0.000142521	0.002223722
	p-value	0.171846625	0.081707571	0.502377867	0.919331644	0.87820191	0.270271807	0.422161732	0.066339308	1
	h	1	0	1	0	1	0	0	1	0
cec.4	Best	30498.11692	28881.55669	20523.2423	25789.89459	14029.34989	53556.56449	4318.91472	66376.99118	5520.33296
	Average	16086.93715	21560.23279	9548.237054	15139.66085	8666.067266	19295.15644	2233.550745	23137.17896	2455.133417
	Worst	9518.971978	14679.08275	2686.563659	5547.678807	3918.175605	4701.006976	431.7962968	1796.496255	812.5439307
	STD	9158.588098	5201.787192	7489.459733	7760.153946	3807.847003	19526.40629	1541.501878	26363.25512	2136.996816
	p-value	0.011857333	6.32535E-05	0.076091468	0.007801953	0.012986047	0.091539389	0.855528246	0.118508917	1
	h	1	1	0	1	1	0	0	0	0
cec.5	Best	5.760759801	10.27091916	10.89349615	6.506994339	4.687953128	3.706052196	6.592753403	6.521697455	3.663430647
	Average	5.297391351	7.362311429	5.3593436	5.498889836	3.6958128	3.106330211	3.485847996	4.694315612	2.45328026
	Worst	4.8518076	4.554525316	2.500639095	4.461591842	2.562526576	2.290108185	2.122692142	3.279093732	2.035330172
	STD	0.373214935	2.29228472	3.390295593	0.898226155	0.814468719	0.624674567	1.916944744	1.362258663	0.682240992
	p-value	3.72586E-05	0.001779322	0.097050968	0.000310119	0.030886471	0.153075955	0.289337342	0.011036824	1
	h	1	1	0	1	1	0	0	1	0
cec.6	Best	13.7437515	11.8508348	13.16906204	13.90528382	14.9084012	13.50654852	13.26771502	14.42372344	13.40003473
	Average	12.89340943	10.90416466	11.0740273	12.75752743	14.08432164	12.67902728	12.18878039	12.66393006	12.6488664
	Worst	11.53015353	9.028735704	9.674977678	11.22003402	12.87533455	11.39681821	10.61429257	11.1018665	12.32716263
	STD	0.902070872	1.101600862	1.320779353	0.979983988	0.823248467	0.831986597	1.066602282	1.336772284	0.436123555
	p-value	0.600125816	0.01097617	0.035160189	0.82647954	0.008754945	0.944527404	0.398027849	0.981475328	1
	h	1	1	0	1	1	0	1	0	0
cec.7	Best	1257.975274	1660.815453	2309.462006	2631.712456	2063.868824	1581.90188	1720.994952	1920.210108	1417.616383
	Average	991.0068241	1452.076195	1284.050372	1608.501659	1694.122394	1367.261341	1353.841377	1476.113453	1065.457276
	Worst	684.2239313	1143.346263	541.7799926	1315.270705	1207.466091	1157.832471	831.6383878	383.5499101	550.8543627
	STD	253.7855722	217.4014833	671.4898744	573.3494406	320.5119947	154.534771	344.4424363	633.2890309	364.3061793
	p-value	0.717438435	0.075937339	0.540176345	0.111656101	0.019981514	0.126523369	0.234346563	0.244260514	1
	h	1	0	1	0	0	1	0	0	0
cec.8	Best	7.915229626	7.265116021	7.777870797	8.27653757	7.777663358	7.451385169	8.633370628	7.392306175	7.625515643
	Average	7.238490947	6.803309954	7.121703282	7.542826101	7.62291148	6.513369728	7.157368999	6.874356617	7.025140951
	Worst	6.173908402	6.203397017	6.812659575	6.76540093	7.201706082	5.724409544	6.14840599	5.960874988	6.599050947
	STD	0.760015146	0.426235865	0.396487622	0.634377965	0.239805016	0.686646831	0.969518105	0.550695095	0.399639493
	p-value	0.593691105	0.420584504	0.711304046	0.161184549	0.020894651	0.187722433	0.785136343	0.63355621	1
	h	1	0	1	0	0	1	0	0	0
cec.9	Best	3512.384464	10904.49534	4742.869585	3819.449644	5633.474516	7085.029182	2364.873964	2005.436187	1603.616167
	Average	2929.005072	6538.665859	2176.270012	2162.42379	2671.132454	3953.09258	951.3904968	928.5780334	673.0107972
	Worst	2574.209801	1066.453056	935.4077127	1068.256644	1373.107225	258.8794552	208.1843372	166.0216033	28.90712398
	STD	349.4391596	4405.809894	1542.809182	1073.211054	1765.680377	3113.528998	845.9005336	773.2471791	617.5453255
	p-value	0.000101046	0.018477619	0.0777295	0.027508152	0.043954709	0.049637117	0.568705607	0.579499441	1
	h	1	1	0	1	1	1	0	0	0
cec.10	Best	20.92244521	20.78568596	20.84720815	20.99271867	20.96534618	20.89800324	21.07166225	21.15299233	20.85246726
	Average	20.71848282	20.5992798	20.58727884	20.80290434	20.757702	20.71147253	20.931287	20.79480878	20.77457853
	Worst	20.51587032	20.24298834	20.36336228	20.64577145	20.34672421	20.45509463	20.63723051	20.36996276	20.74511882
	STD	0.146327352	0.222167238	0.178578268	0.132763083	0.239690328	0.176778088	0.17585592	0.28170892	0.044574701
	p-value	0.435958412	0.121905608	0.052441737	0.663087478	0.880823066	0.461188148	0.08951061	0.877911065	1
	h	0	0	0	0	0	0	0	0	0

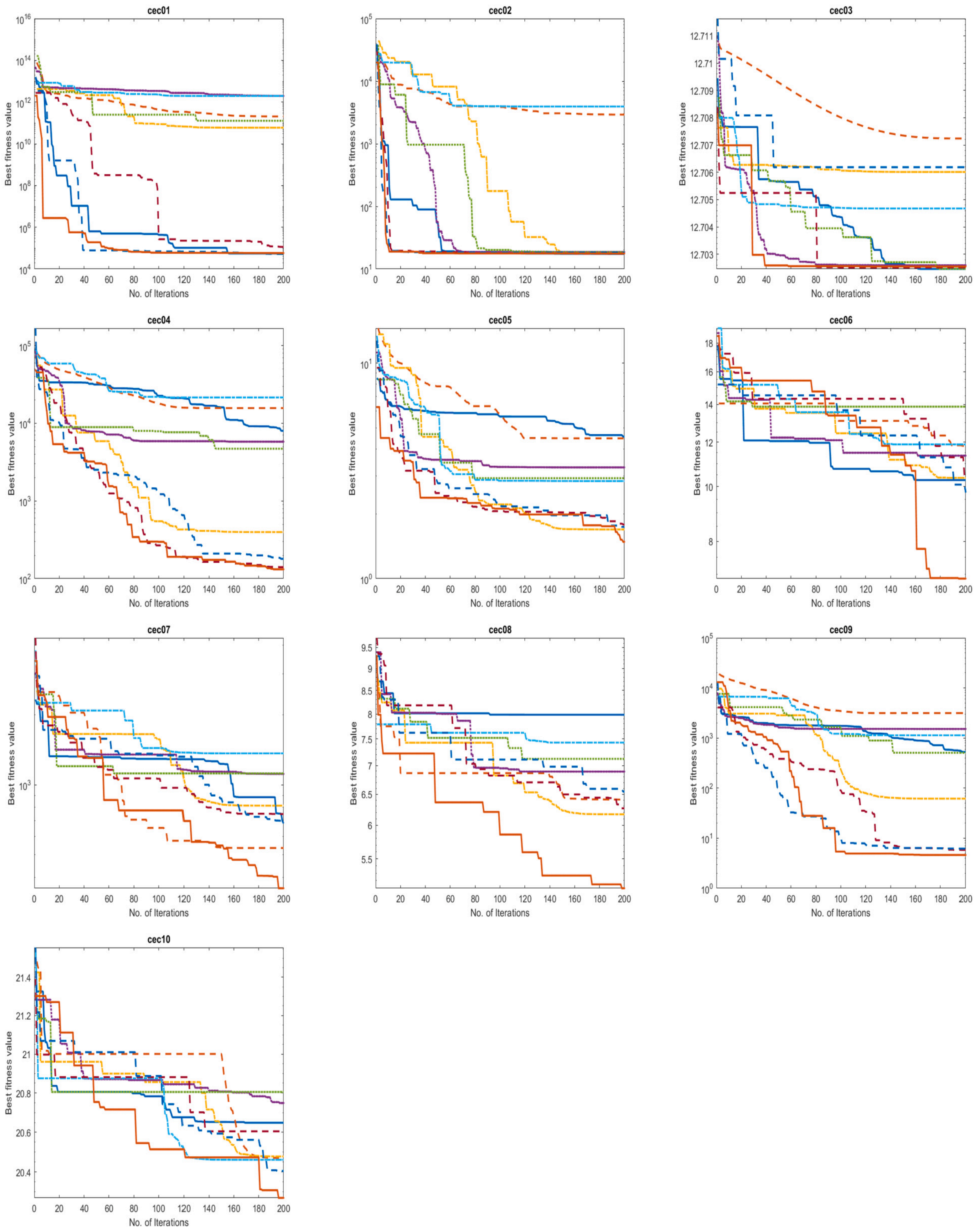

Fig. 5. Qualitative analysis (Convergence behavior) of the CEC2019 benchmark functions based on 200 iterations.

Table 8
The final rank of CEC2019 benchmark functions were obtained using comparative methodologies, with a dimension of 10 and 50 iterations.

Function	Comparative Algorithms								
	HHO	GOA	SSA	WOA	SCA	DA	SMA	PDO	PDO-DE
cec.1	1	6	7	8	5	4	3	9	2
cec.2	1	8	7	4	5	6	3	9	2
cec.3	2	9	4	7	5	8	3	1	6
cec.4	6	8	4	5	3	7	1	9	2
cec.5	6	9	7	8	4	2	3	5	1
cec.6	8	1	2	7	9	6	3	5	4
cec.7	1	6	3	8	9	5	4	7	2
cec.8	7	2	5	8	9	1	6	3	4
cec.9	7	9	5	4	6	8	3	2	1
cec.10	4	2	1	8	5	3	9	7	6
Sum	43	60	45	67	60	50	38	57	30
Mean	4.3	6	4.5	6.7	6	5	3.8	5.7	3
Rank	3	7	4	8	6	5	2	5	1

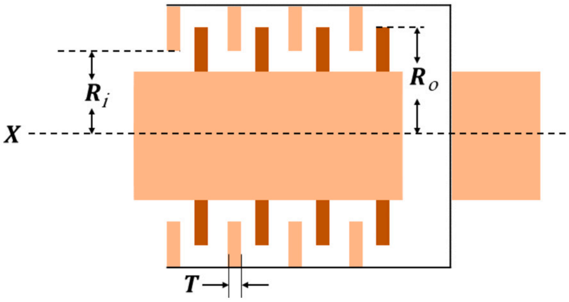

Fig. 6. A model of the multiple-disc clutch brake.

Table 9
Empirical findings pertaining to the issue of a multi-disc clutch brake.

	Best obtained results					
Algorithm	R_i	R_0	T	X	Y	Best obtained weight
HHO	70.02	90.02	1	810	3	0.314656611
GOA	70.03	90.0349	1	801.7305	2.974	0.31376
SSA	70.02	90.02	1	910	3	0.314656
WOA	70.02	90.02	1	910	3	0.314656
SCA	70.02	90.02	1	910	3	0.314656
DA	70.02	90.02	1	810	3	0.314656611
SMA	70.02	90.02	1	910	3	0.314656
PDO	70.02	90.02	1	910	3	0.314656
PDO-DE	70.02	90.02	1	600	2	0.234752458

Other Parameters:

$$U_h = \frac{2}{3}\mu XY \frac{R_0^3 - R_i^2}{R_0^2 - R_i^3}, P_{RY} = \frac{X}{\Pi\left(R_0^2 - R_i^2\right)},$$

$$Z_{RZ} = \frac{2\Pi\left(R_0^3 - R_i^3\right)}{90\left(R_0^2 - R_i^2\right)}, W = \frac{I_Y \Pi n}{30\left(U_h + U_f\right)}$$

$$\Delta R = 20 \text{ mm}, I_Y = 55\text{kgmm}^2, P_{\max} = 1 \text{ MPa}, X_{\max} = 1000 \text{ N},$$

$$W_{\max} = 15 \text{ s}, \mu = 0.5, \ s = 1.5, U_s = 40\text{Nm}, M_f = 3\text{Nm},$$

$$n = 250\text{rpm}, z_{\text{sr max}} = 10 \text{ m/s}, L_{\max} = 30 \text{ mm}$$

The optimal weights of PDO-DE and the comparison methods in this issue are detailed in Table 9, in contrast to the other six optimization methods. The optimal weight determined via PDO-DE is 0.234752458. The corresponding values for the five variables are $R_i = 70.02$, $R_0 = 90.02$, T= 1, X = 600, and Y = 2, respectively. Hence, based on our research, we may infer that PDO-DE is superior to other ways of resolving this issue.

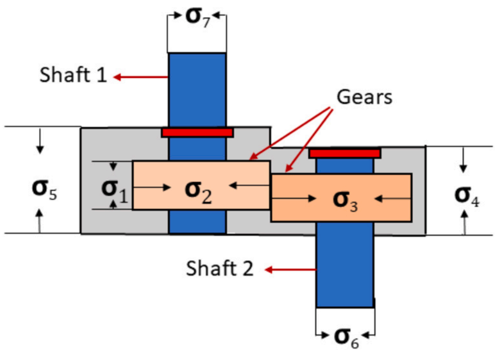

Fig. 7. A model of the speed reducer design problem.

4.2. Problem 2: speed reducer design problem

Fig. 7 presents the variable diagram. The objective of the reducer design challenge is to ascertain the minimum weight of the reducer while satisfying four design constraints. The four design constraints encompass stress in the shaft, lateral displacement of the shaft, stress on the shaft, and bending stress on the gear teeth. The variables in the reducer design problem are the following:

- The width of the tooth surface is denoted by σ_1.
- The gear module (σ_2) refers to the size of the gear teeth.
- The quantity representing the count of teeth on the pinion is denoted as σ_3.
- The distance between the bearings of the first shaft (σ_4).
- The distance between the bearings on the second shaft (σ_5).
- The first shaft's diameter is denoted as σ_6.
- The measurement of the second shaft's diameter (σ_7).

The problem can be mathematically formulated and expressed by a set of constraint functions: Consider:

$$\sigma = [\sigma_1 \sigma_2 \sigma_3 \sigma_4 \sigma_5 \sigma_6 \sigma_7]$$

Objective function:

$$f(\vec{\sigma}) = 07854 \times \sigma_1 \times \sigma_2^2 \times (3.3333 \times \sigma_3^2 + 14.9334 \times \sigma_3 - 43.0934) - 1.508 \times \sigma_1 \times (\sigma_6^2 + \sigma_7^2) + 7.4777 \times \sigma_6^3 + \sigma_7^3 + 0.7854 \times \sigma_4 \times \sigma_6^2 + \sigma_5 \times \sigma_7^2$$

Subject to:

$$g_1(\vec{\sigma}) = \frac{27}{\sigma_1 \times \sigma_2^2 \times \sigma_3} - 1 \leq 0$$

$$g_2(\vec{\sigma}) = \frac{397.5}{\sigma_1 \times \sigma_2^2 \times \sigma_3^2} - 1 \leq 0$$

$$g_3(\vec{\sigma}) = \frac{1.93 \times \sigma_4^3}{\sigma_2 \times \sigma_3 \times \sigma_6^4} - 1 \leq 0$$

$$g_4(\vec{\sigma}) = \frac{1.93 \times \sigma_5^3}{\sigma_2 \times \sigma_3 \times \sigma_7^4} - 1 \leq 0$$

$$g_5(\vec{\sigma}) = \frac{1}{110 \times \sigma_6^3} \sqrt{\left(\frac{745 \times \sigma_4}{\sigma_2 \times \sigma_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0$$

$$g_6(\vec{\sigma}) = \frac{1}{85 \times \sigma_7^3} \sqrt{\left(\frac{745 \times \sigma_5}{\sigma_2 \times \sigma_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0$$

$$g_7(\vec{\sigma}) = \frac{\sigma_2 \times \sigma_3}{40} - 1 \leq 0$$

$$g_8(\vec{\sigma}) = \frac{5 \times \sigma_2}{\sigma_1} - 1 \leq 0$$

Table 10
Empirical findings pertaining to the issue of the speed reducer.

Algorithm	Best obtained results σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	Lowest obtained result
HHO	3.616129	0.71	17	7.31	7.92151	3.452570	5.276749	3031.873069
GOA	3.53	0.71	17	7.31	7.670395	3.562399	5.275814	3019.583370
SSA	3.521765	0.71	17	7.31	7.91	3.45098	5.289213	3033.564
WOA	3.630134	0.71	17	8.30	7.91	3.37701	5.287719	3029.002
SCA	3.417485	0.7001	17	7.739684	8.089954	3.381415	5.287051	3015.137489
DA	3.530152	0.7004	17	7.569146	7.97833	3.385578	5.289773	3011.09
SMA	3.522765	0.71	17	7.31	7.8	3.46104	5.289213	3029.559
PDO	3.540485	0.7001	17	7.728284	8.080954	3.371511	5.289051	3010.137489
PDO-DE	3.4932	0.7	17	7.2198	7.7275	3.3711	5.2584	3006.7327

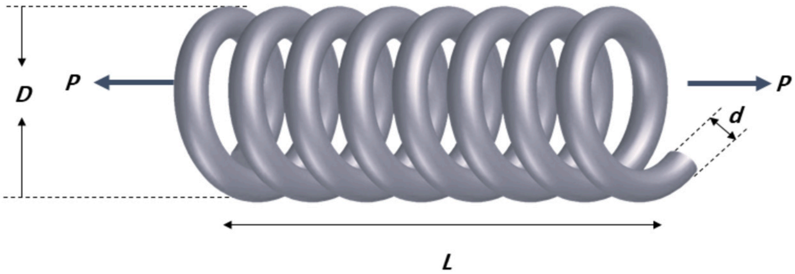

Fig. 8. A model of the Spring design problem.

$$g_9\left(\vec{\sigma}\right) = \frac{\sigma_1}{12 \times \sigma_2} - 1 \leq 0$$
$$g_{10}\left(\vec{\sigma}\right) = \frac{1.5 \times \sigma_6 + 1.9}{\sigma_4} - 1 \leq 0$$
$$g_{11}\left(\vec{\sigma}\right) = \frac{1.1 \times \sigma_7 + 1.9}{\sigma_5} - 1 \leq 0$$

Parameters range:

$$2.6 \leq \sigma_1 \leq 3.6, 0.7 \leq \sigma_2 \leq 0.8, 17 \leq \sigma_3 \leq 28, 7.3 \leq \sigma_4 \leq 8.3, 7.3 \leq \sigma_5 \leq 8.3, 2.9 \leq \sigma_6 \leq 3.9, 5\sigma_7 \leq 5.5$$

Table 10 presents the outcomes of the PDO-DE and the compared strategies. The PDO-DE approach achieved the highest rank in this table, surpassing all other ways of addressing this problem. The PDO method obtained second place, followed by DA, SCA, and GOA.

4.3. Problem 3: spring design

Fig. 8 illustrates the objective of this work, which is to decrease the mass of a spring. The minimization process has specific constraints, including shear stress, surge frequency, and minimum deflection. The variables in this problem are as follows:

- The wire diameter (d).
- Mean coil diameter (D).
- Number of active coils (N).

The mathematical model of the Spring design is expressed as follows:
Consider:

$$\vec{\lambda} = [\lambda_1, \lambda_2, \lambda_3] = [d\,D\,N],$$

Objective function:

$$f(\vec{\lambda}) = (\lambda_3 + 2)\lambda_2\lambda_1^2$$

Subject to:

$$g_1(\vec{\lambda}) = 1 - \frac{\lambda_2^3\lambda_3}{71785\lambda_1^4} \leq 0$$

Table 11
Empirical findings about the issue of the spring design.

	Best obtained results			
Algorithm	*d*	*D*	*N*	Best cost
HHO	0.06273	0.367647	11.234443	0.011786
GOA	0.062155	0.359777	12.150432	0.0117806
SSA	0.06	0.327	14.11	0.0119455
WOA	0.062646	0.3445	11.408026	0.011712
SCA	0.06299455	0.3591188	10.90842186	0.011685
DA	0.0627	0.36685	11.3119	0.011685
SMA	0.06241	0.369085	11.80320	0.011682
PDO	0.06273	0.367045	11.255420	0.011673
PDO-DE	0.060599	0.316181	10.2101150	0.011670

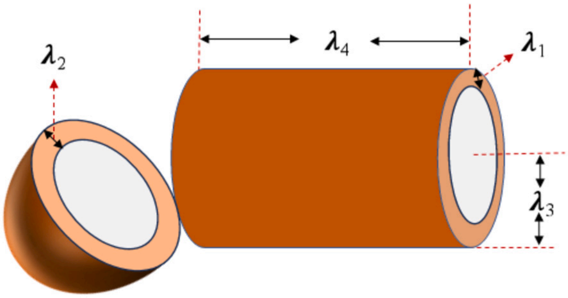

Fig. 9. A model of the Spring design problem.

$$g_2(\vec{\lambda}) = \frac{4\lambda_2^2 - \lambda_1\lambda_2}{12,566(\lambda_2\lambda_1^3 - \lambda_1^4)} + \frac{1}{5108\lambda_1^2} \le 0$$

$$g_3(\vec{\lambda}) = 1 - \frac{140.45\lambda_1}{\lambda_2^2\lambda_3} \le 0$$

$$g_4(\vec{\lambda}) = \frac{\lambda_1 + \lambda_2}{1.5} - 1 \le 0$$

Variable range:

$$0.05 \le \lambda_1 \le 2.00,\ 0.25 \le \lambda_2 \le 1.30,\ \text{and } 2.00 \le \lambda_3 \le 15.00.$$

The results of all comparison methods and the recommended PDO-DE for addressing the spring design issue are presented in Table 11. The optimal parameter values are presented in Table 11, along with the highest achieved results for all comparison algorithms. The variables with the values $\lambda = (d = 0.060599, D = 0.316181, N = 10.2101150)$ yield the optimal objective's value: $F(\lambda) = 0.011670$. This demonstrates that the suggested PDO-DE method is superior to other state-of-the-art approaches in terms of giving a more reliable solution.

4.4. Problem 4: the pressure vessel

The pressure vessel, which has hemispherical caps and a cylindrical shape (see Fig. 9), must be built with minimal expenses. The construction of the compressed air tank must adhere to the American Society of Mechanical Engineers (ASME) regulations for boilers and pressure vessels [38]. The tank operates at a pressure of 3,000 pounds per square inch (psi) and has a minimum volume of 750 cubic feet (ft3). The final price is determined by the cumulative costs of welding, material, and forming charges. The optimization factors considered the following:

- The length of the cylindrical segment.
- The inner radius.
- The thickness of the cylinder skin.
- The thickness of the spherical head.
- The inner radius.

Thickness can only be expressed as discrete numbers integer multiples of 0.0625. The mathematical formulation of this problem can be stated as:

Table 12
Empirical findings pertaining to the pressure vessel.

Algorithm	Optimal results for variables				
	λ_1	λ_2	λ_3	λ_4	Best cost
HHO	1.123	0.711	50.01	113.11 7	832.1
GOA	1.271	0.711	60.1603	69.7221 68	45.9242
SSA	1.126	0.71	59.30116 45.	67555 71	54.75
WOA	0.8211	0.4412	43.098812 176.	63745 60	61.91
SCA	0.8211	0.4412	43.091401 176.7	345 60	61.0801
DA	0.8211	0.4412	43.099551	176.6421 60	61.81431
SMA	1.124	0.6161	54.9775601	85.4643624 76	01.8355
PDO	0.8155	0.4281	42.089311	177.648589 60	61.839
PDO-DE	0.813	0.4281	42.070053	175.625672 60	61.5245

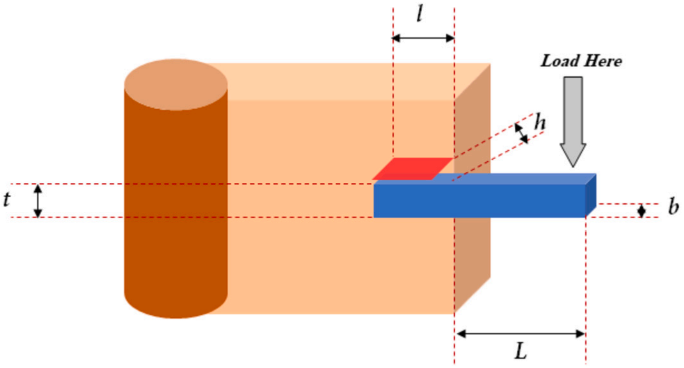

Fig. 10. A model of the Welded design problem.

Objective function:

$$f(\lambda) = 0.6224\lambda_1\lambda_3\lambda_4 + 1.7781\lambda_2\lambda_3^2 + 3.1661\lambda_1^2\lambda_4 + 19.84\lambda_1^2\lambda_3$$

Subject to:

$$g_1(\lambda) = -\lambda_1 + 0.0193\lambda_3 \leq 0$$
$$g_2(\lambda) = -\lambda_2 + 0.00954\lambda_3 \leq 0$$
$$g_3(\lambda) = -\pi\lambda_3^2\lambda_4 - \frac{4}{3}\pi\lambda_3^3 + 1296000 \leq 0$$
$$g_4(\lambda) = \lambda_4 - 240 \leq 0$$

where:

$$1 \times 0.0625 \leq \lambda_1,\ \lambda_2 \leq 99 \times 0.0625,\ 10 \leq \lambda_3 \leq 200\, and\, 10 \leq \lambda_4 \leq 240.$$

Table 12 illustrates the comparison algorithms and the PDO-DE method used to address the vessel design problem. The optimal parameter values are displayed in Table 12, along with the best results achieved by all the algorithms that were compared. Table 12 demonstrates that the proposed PDO-DE outperforms other state-of-the-art methods by offering a more reliable solution that assigns the optimal variables at $\lambda = (\lambda_1 = 0.813, \lambda_2 = 0.4281, \lambda_3 = 42.070053, \lambda_4 = 175.625672)$, resulting in the best objective value of $f(\lambda) = 61.5245$.

4.5. Problem 5: welded beam

The welded beam design issue is a widely recognized case study used to assess the efficacy of PDO-DE. The concept was initially introduced in [39] to reduce the total manufacturing cost of a welded beam by utilizing four finding variables, as depicted in Fig. 10. The variables consist of the following: The weld thickness (h). The length (l) of the joint beam. The height of the beam (t). Thickness (b).

The mathematical formulation of this problem can be stated as:

Consider

$$\vec{\chi} = [\chi_1, \chi_2, \chi_3, \chi_4] = [h, l, t, b]$$

Table 13
Empirical findings pertaining to the Welded beam.

	Optimal results for variables				
Algorithm	h	l	t	b	Best cost
HO	0.2601	6.38	8.3792	0.4634	2.6301
GOA	0.21783	3.670501	9.237724	0.40613	1.924902
SSA	0.256	6.4191	8.4895	0.4457	2.582
WOA	0.2551	6.4602	8.4895	0.4634	2.5839
SCA	0.19213	4.056899	10.201	0.402416	2.07987
DA	0.2622	6.4321	8.4885	0.4401	2.5797
SMA	0.241794	3.268992	9.189079	0.408805	1.923011
PDO	0.215187	3.727557	9.204333	0.406941	1.934857
PDO-DE	0.26406111	2.0423029	8.47022978	0.453219	1.915701

Minimize

$$f\left(\vec{\chi}\right) = 1.10471\chi_1^2\chi_2 + 0.04811\chi_3\chi_4\left(14.0 + \chi_2\right)$$

Subject to

$$g1\left(\vec{\chi}\right) = \tau\left(\chi\right) - \tau_{max} \le 0$$

$$g2\left(\vec{\chi}\right) = \lambda - \lambda_{max} \le 0$$

$$g3\left(\vec{\chi}\right) = \delta - \delta_{max} \le 0$$

$$g4\left(\vec{\chi}\right) = \chi_1 - \chi_4 \le 0 \le 0$$

$$g5\left(\vec{\chi}\right) = P - P_C\left(\vec{\chi}\right) \le 0$$

$$g6\left(\vec{\chi}\right) = 0.125 - \chi_1 \le 0$$

$$g7\left(\vec{\chi}\right) = 1.10471\chi_1^2 + 0.04811\chi_3\chi_4\left(14 + \chi_2\right) - 5 \le 0$$

Variable range

$$0.125 \le \chi_1 \le 5,\ 0.1 \le \chi_2,\ \chi_3 \le 10,\ and\ 0.1 \le \chi_4 \le 5.$$

where

$$\tau\left(\vec{\chi}\right) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{\chi_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}\chi_1\chi_2}, \tau' = \frac{MR}{J}, M = P\left(L + \frac{\chi_2}{2}\right)$$

$$R = \sqrt{\frac{\chi_1^2}{4} + \left(\frac{\chi_1 + \chi_3}{2}\right)^2}, J = 2\left\{\sqrt{2}\chi_1\chi_2\left[\frac{\chi_2^2}{4} + \left(\frac{\chi_1 + \chi_3}{2}\right)^2\right]\right\},$$

$$\chi\left(\vec{\chi}\right) = \frac{6PL}{E\chi_3^2\chi_4},\ \delta\left(\vec{\chi}\right) = \frac{6PL^3}{E\chi_3^2\chi_4},$$

$$P_C\left(\vec{\chi}\right) = \frac{4.013E\sqrt{\frac{\chi_3^2\chi_4^6}{36}}}{L^2}\left(1 - \frac{z_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$\lambda_{max} = 3000psi,\ \delta_{max} = 0.25in,\ \tau_{max} = 30,000psi.$$

$$E = 30 \times 10^6 psi,\ G = 12 \times 10^6 psi$$

$$L = 14in,\ P = 6000lb.$$

The proposed PDO-DE is implemented to address the problem of the welded beam. The PDO-DE algorithm outperforms compared algorithms and produces consistent results with the optimal variables at χ = (*h*=*0.26406111, l*=*2.0423029, t*=*8.47022978, b*=*0.453219*) and the optimal cost at $f(\chi)$=*1.915701*, the results are shown in Table 13. This demonstrates that PDO-DE can effectively address the problem of welded beam design.

4.6. Problem 6: network intrusion detection system

As internet usage continues to expand, so do its vulnerabilities, prompting the implementation of Intrusion Detection Systems (IDS) to safeguard security. IDSs serve as protective measures, identifying external intrusions, unauthorized accesses, and network

Table 14
Dataset Sizes.

Dataset Split	Number of Records
Training Set	4,898,431
Testing Set	311,029

malfunctions. By analyzing data such as port scanning and abnormal traffic patterns, IDSs can detect intrusions and alert network administrators. Intrusion detection poses a classification challenge, and identifying optimal features is crucial for classification techniques to be effective. Common classification methods include neural networks, fuzzy logic, data mining techniques, and meta-heuristics. The proposed PDO-DE is implemented to enhance the precision of detecting intrusion. It is utilized to select features with support vector machine (SVM) assets for classification.

4.6.1. Support vector machine

Support Vector Machines (SVM) are a widely used supervised machine learning model known for their effectiveness in classification tasks, including intrusion detection. They excel in linear classification and regression, offering robustness and adaptability even with limited training data. One of their key advantages is their ability to operate without assumptions about the underlying data, instead identifying hyperplanes to separate data points effectively. In Intrusion Detection Systems (IDS), where attack distributions can be imbalanced, SVMs have succeeded due to their ability to generalize well, handle multiple classes efficiently, and operate with low classification times. The SVM classifier is central to SVM-based intrusion detection systems, generating models of the target system and correlating attribute data with classification outcomes [40].

4.6.2. Dataset

The NSL-KDD dataset is used in this work to show the performance of the PDO-DE for intrusion detection. The KDD CUP 1999 dataset, derived from the DARPA 98 IDS evaluation program by Lincoln Labs, features around five million connection records from a U.S. Air Force military simulation, serving as a standard benchmark for testing intrusion detection algorithms. To overcome issues like duplicate records, skewed distributions, and redundancies in the KDD dataset, the improved NSL-KDD dataset was introduced. It includes 41 attributes categorized into four groups: time-based and host-based traffic, basic, and content attributes, which can be discrete or continuous. This dataset labels deviations from normal network behavior as attacks, categorizing them into 24 types, such as Denial of Service (DoS), Probe, user-to-root (U2R), and remote-to-local (R2L) attacks [41].

Each connection record in the dataset has 41 features, which fall into one of three categories:

- Basic Features: These are characteristics like duration, protocol type, service, and flag that are obtained from TCP/IP connections.
- Content Features: These features include counts like the quantity of unsuccessful login attempts and file creation operations obtained from the data contained in a connection.
- Traffic characteristics: This category includes the number of connections to the same host and to the same service. It is obtained from a two-second time window.

The training set consists of about 4.9 million connection records, and the testing set consists of about 311,000 connection records, as shown in Table 14. This large dataset offers a thorough foundation for testing and developing network intrusion detection methods.

4.6.3. Results and discussions on IDS

The evaluation of the PDO-DE algorithm is based on four metrics: accuracy (ACC), feature count (NoF), false alarm rate (FR), and detection rate (DR). These metrics are calculated using counts of true positives (correctly identified intrusions), false positives (normal activities incorrectly flagged as intrusions), true negatives (normal activities rightly identified), and false negatives (intrusions wrongly classified as normal). The computational methods for these metrics are detailed below:

$$FR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{20}$$

$$DR = \frac{True_{positive}}{True_{positive} + False_{Negative}} \tag{21}$$

$$NoF = Total_{Number\ of\ features} - Non\ selected_{Features} \tag{22}$$

$$ACC = \frac{True_{positive} + True_{negative}}{True_{positive} + False_{negative} + True_{negative}} \tag{23}$$

This section comprehensively evaluates several state-of-the-art algorithms, including the original PDO and DE and the proposed PDO-DE, HHO, GOA, SSA, and WOA. The study assesses the effectiveness of these algorithms in handling the Intrusion Detection System (IDS) problem compared to the suggested technique. Improvements have been made with the suggested PDO-DE to address shortcomings like premature convergence, solution diversity, and slow search speed in the initial PDO algorithm. The impact of varying the number of PDs on optimization, striking a balance between exploration and exploitation searches, is also examined in this section.

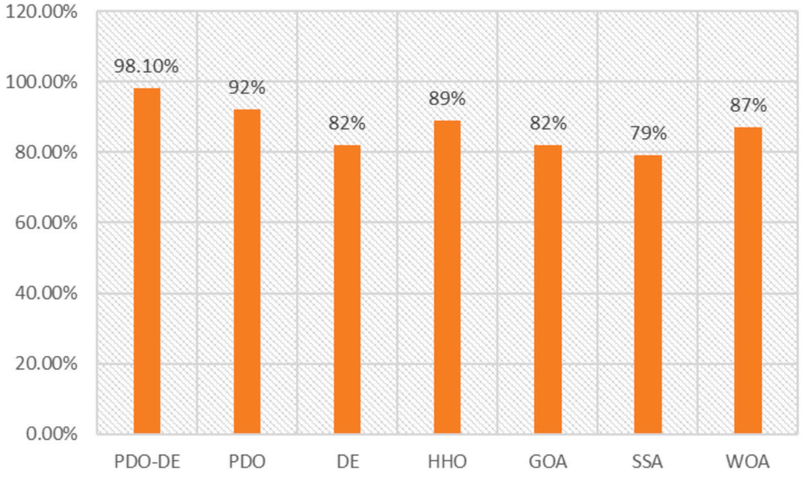

Fig. 11. The results of the Detection Rate.

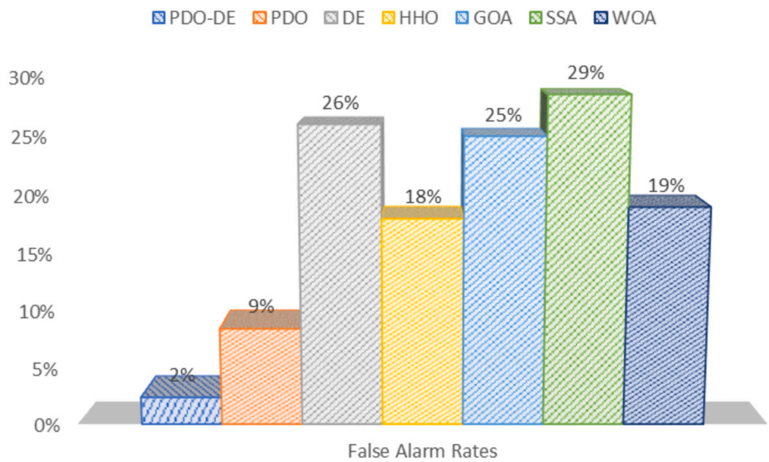

Fig. 12. The results of the False Alarm Rate.

Detection rate The PDO-DE method demonstrates superior performance in detection rates compared to the PDO and DE algorithms, achieving a detection rate of 98.1% against PDO 92% and DE 82%. This marks a significant enhancement in the network intrusion detection capabilities, with results depicted in Fig. 11. Repeated testing confirms the reliability of PDO-DE, noting an increase in system complexity and a maintained high detection rate when integrating DE features.

False alarm rate The PDO-DE method also shows a marked improvement in reducing false alarms, with a rate of only 2.4%, in contrast to higher rates observed in other methods: 26% for DE, 8.5% for PDO and between 18%, 25%, 29%, and 19% for HHO, GOA, SSA, and WOA, respectively. This advancement leads to fewer irrelevant alerts from the IDS, enhancing operational efficiency, as detailed in Fig. 12.

Accuracy The PDO-DE method's accuracy is outstanding, reaching 96%. This outperforms PDO at 92% and DE at 87% and significantly surpasses HHO, GOA, SSA, and WOA, which range from 80% to 86%. This improvement results from the algorithm's multi-objective function, which prioritizes accuracy and refines the intrusion detection system's precision. The summarized results in Fig. 13 underscore the method's precision and reliability.

Number of features The PDO-DE selects only 9% of features for effective intrusion detection, compared to 16% and 22% by PDO and SMA, respectively. This reduction optimizes the number of relevant characteristics the IDS manages, minimizing system overhead. This efficiency in feature selection underscores the robustness of PDO-DE, as presented in Fig. 14. The strategic increase in prairie dogs enhances the diversity and selection quality of the features.

The IDS results show that the PDO-DE method significantly outperforms the PDO and DE algorithms in several key areas of network intrusion detection. It achieves a higher detection rate of 98.1%, greatly reducing false alarms with a rate of only 2.4%, and improves accuracy to 96%. Additionally, it selects fewer but more effective features for intrusion detection, only 9% compared to

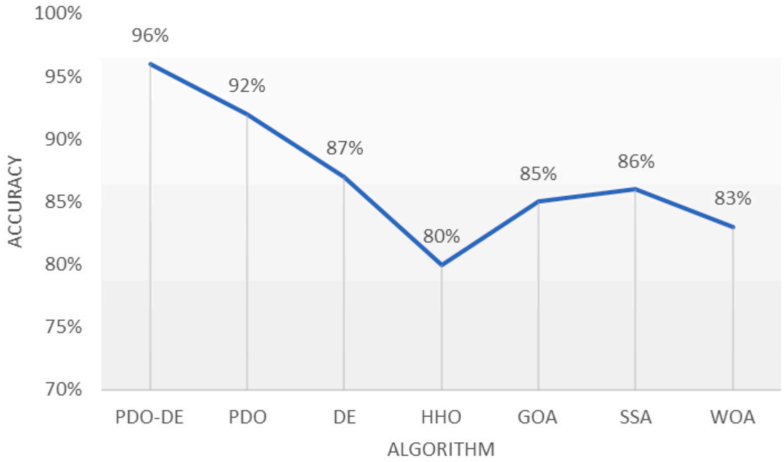

Fig. 13. The results of the Accuracy.

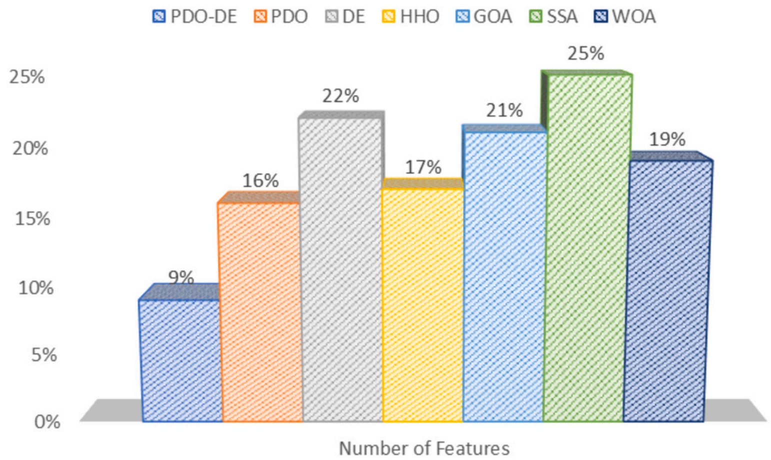

Fig. 14. The results of the Number of Features.

PDO's 16% and DE's 23%. These improvements in detection rate, false alarm rate, accuracy, and feature selection efficiency highlight the PDO-DE method's enhanced capability and operational efficiency in network security environments.

4.7. Evaluation of the suggested approach: advantages and disadvantages

The suggested method integrates two optimization techniques, PDO and DE, to enhance optimization performance and attain superior outcomes. ***Advantages:***

- DE algorithm enriches the population's diversity and prevents premature convergence by producing contrasting solutions.
- The purpose of using PDO is to expand the search area and prevent being trapped in local optima. DE aids in achieving a balance between exploration and exploitation by incorporating a stochastic element into the search procedure.
- The technique is assessed using benchmark functions and engineering design optimization challenges. It surpasses certain other cutting-edge optimization algorithms in accuracy and efficiency.

Disadvantages:

- The effectiveness of the suggested method is highly dependent on the parameter settings. If the parameters are not appropriately chosen, it can result in inferior outcomes or slow convergence.
- The method may require a substantial number of function evaluations to get the best solution, which can require a high computational cost for problems with many dimensions.
- The user's text is a bullet point. This approach may not be appropriate for specific targeted optimization issues and may exhibit suboptimal performance on benchmark functions not included in the study.

5. Conclusions

This paper introduces a new population-based metaheuristic method incorporating Prairie dog optimization (PDO) and the Differential Evolution algorithm to enhance the PDO algorithm's searchability.

The efficacy of the proposed PDO-DE is evaluated by assessing its ability to balance exploration and exploitation through experimentation with classical and CEC2019 benchmark test functions. The findings showcased the efficacy of PDO-DE in identifying the most favorable global solutions and exhibiting more consistent convergence when compared to other widely recognized optimization algorithms documented in the literature. The effectiveness of the proposed PDO-DE is determined using statistical analysis utilizing the Freidman ranking test. The statistical results further validated the resilience of the proposed PDO-DE in effectively conducting both exploration and exploitation.

In addition, the suggested PDO-DE approach is applied to address five practical engineering issues. The outcomes obtained by the PDO-DE algorithm verified its potential to provide superior (almost optimal) solutions compared to various metaheuristic algorithms, such as HHO, WOA, DA, SMA, SCA, GOA, and the basic PDO. PDO-DE demonstrated a significant ability to manage diverse restrictions in optimization situations.

The suggested PDO-DE algorithm effectively addressed single-objective continuous optimization problems. Researchers should explore the possibility of building a binary version of the algorithm. Additionally, it is possible to create a multi-objective version of the PDO. Researchers may also explore the possibility of adjusting and hybridizing the PDO-DE. Extending the PDO-DE to address various discrete or continuous challenges can be a promising effort.

In conclusion, the PDO-DE algorithm represents a significant scientific advancement in hybrid optimization techniques, providing a more effective approach for solving real-world problems that require high precision and optimal resource utilization. Our extensive experiments on 23 benchmark functions and five engineering design problems demonstrate that PDO-DE consistently achieves superior performance metrics. Notably, the algorithm shows an average accuracy improvement of 2.5% over existing state-of-the-art methods. In the context of network intrusion detection, PDO-DE achieved an overall accuracy of 96%, a detection rate of 98.1%, and a false alarm rate of 2.4%. These results underline the algorithm's robustness, reliability, and efficiency.

The significance of these improvements is profound, as they highlight PDO-DE's ability to deliver precise and reliable solutions in complex and dynamic environments. The enhanced convergence speed and reduced computational time further establish its applicability in various domains. Future work will explore the application of PDO-DE to other challenging optimization problems and investigate further enhancements to the algorithm's framework. The promising results achieved in this study suggest that PDO-DE can be a valuable tool in both engineering optimization and cybersecurity, paving the way for more advanced and practical applications in these fields.

Future research will focus on extending the application of PDO-DE to other challenging optimization problems, such as large-scale industrial optimization, bioinformatics, and financial modeling. Additionally, exploring the integration of other metaheuristic algorithms could further enhance the performance of PDO-DE. We also plan to investigate adaptive mechanisms within the algorithm to improve its adaptability to different problem landscapes dynamically. Furthermore, real-world implementations and case studies in engineering and cybersecurity will be conducted to validate the practical effectiveness and scalability of PDO-DE in diverse scenarios.

The promising results achieved in this study suggest that PDO-DE can be a valuable tool in both engineering optimization and cybersecurity, paving the way for more advanced and practical applications in these fields.

Ethical approval

Not applicable.

CRediT authorship contribution statement

Mohammad Alshinwan: Formal analysis, Data curation, Conceptualization. **Osama A. Khashan:** Supervision, Project administration, Data curation. **Mohammed Khader:** Formal analysis, Conceptualization. **Omar Tarawneh:** Formal analysis, Data curation, Conceptualization. **Ahmed Shdefat:** Resources, Investigation, Funding acquisition, Data curation. **Nour Mostafa:** Formal analysis, Conceptualization. **Diaa Salama AbdElminaam:** Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data are available upon reasonable request from the corresponding author, Diaa Salama AbdElminaam.

Acknowledgements

The authors gratefully acknowledge the Rabdan Academy for their invaluable support and resources provided throughout this research, which significantly contributed to its successful completion.

References

[1] J. Sun, Q. Zhang, E.P. Tsang, De/eda: a new evolutionary algorithm for global optimization, Inf. Sci. 169 (3–4) (2005) 249–262.
[2] J. Mockus, J. Mockus, Global optimization and the Bayesian approach, in: Bayesian Approach to Global Optimization: Theory and Applications, 1989, pp. 1–3.
[3] M.H. Nadimi-Shahraki, H. Zamani, Dmde: diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization, Expert Syst. Appl. 198 (2022) 116895.
[4] B. Morales-Castaneda, D. Zaldivar, E. Cuevas, A. Rodriguez, M.A. Navarro, Population management in metaheuristic algorithms: could less be more?, Appl. Soft Comput. 107 (2021) 107389.
[5] J.S. Arora, Introduction to Optimum Design, Elsevier, Amsterdam, The Netherlands, 2004.
[6] M.S. Daoud, M. Shehab, L. Abualigah, M. Alshinwan, M.A. Elaziz, M.K.Y. Shambour, D. Oliva, M.A. Alia, R.A. Zitar, Recent advances of chimp optimization algorithm: variants and applications, J. Bionics Eng. (2023) 1–23.
[7] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: algorithms and applications, Swarm Evol. Comput. 33 (2017) 1–17.
[8] H. Mamdouh Farghaly, T. Abd El-Hafeez, A new feature selection method based on frequent and associated itemsets for text classification, Concurr. Comput., Pract. Exp. 34 (25) (2022) 7258.
[9] S. Garnier, J. Gautrais, G. Theraulaz, The biological principles of swarm intelligence, Swarm Intell. 1 (2007) 3–31.
[10] J. Kennedy, Swarm intelligence, in: Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies, Springer, Boston, MA, USA, 2006, pp. 187–219.
[11] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[12] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[13] D. Karaboga, Artificial bee colony algorithm, Scholarpedia 5 (3) (2010) 6915.
[14] H. Ahmed, J. Glasgow, Swarm intelligence: concepts, models and applications, School of Computing, 2012, Queens University Technical Report.
[15] M. Shehab, L. Abualigah, H. Al Hamad, H. Alabool, M. Alshinwan, A.M. Khasawneh, Moth–flame optimization algorithm: variants and applications, Neural Comput. Appl. 32 (2020) 9859–9884.
[16] K. Rajwar, K. Deep, S. Das, An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges, Artif. Intell. Rev. (2023) 1–71.
[17] M. Hijjawi, M. Alshinwan, O.A. Khashan, M. Alshdaifat, W. Almanaseer, W. Alomoush, H. Garg, L. Abualigah, Accelerated arithmetic optimization algorithm by cuckoo search for solving engineering design problems, Processes 11 (5) (2023) 1380.
[18] M. Abd Elaziz, A. Dahou, L. Abualigah, L. Yu, M. Alshinwan, A.M. Khasawneh, S. Lu, Advanced metaheuristic optimization techniques in applications of deep neural networks: a review, Neural Comput. Appl. (2021) 1–21.
[19] L. Abualigah, A.H. Gandomi, M.A. Elaziz, H.A. Hamad, M. Omari, M. Alshinwan, A.M. Khasawneh, Advances in meta-heuristic optimization algorithms in big data text clustering, Electronics 10 (2) (2021) 101.
[20] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[21] A. Slowik, H. Kwasnicka, Evolutionary algorithms and their applications to engineering problems, Neural Comput. Appl. 32 (2020) 12363–12379.
[22] E. Hassan, T. Abd El-Hafeez, M.Y. Shams, Optimizing classification of diseases through language model analysis of symptoms, Sci. Rep. 14 (1) (2024) 1507.
[23] M. Aljanabi, M.H. Qutqut, M. Hijjawi, Machine learning classification techniques for heart disease prediction: a review, Int. J. Eng. Technol. 7 (4) (2018) 5373–5379.
[24] A. Nasayreh, A.S. Jaradat, H. Gharaibeh, W. Dawaghreh, R.M. Al Mamlook, Y. Alqudah, Q. Al-Na'amneh, M.S. Daoud, H. Migdady, L. Abualigah, Jordanian banknote data recognition: a cnn-based approach with attention mechanism, J. King Saud Univ, Comput. Inf. Sci. 36 (4) (2024) 102038.
[25] J.-F. Boujut, P. Laureillard, A co-operation framework for product–process integration in engineering design, Des. Stud. 23 (6) (2002) 497–513.
[26] A. Gungor, S.M. Gupta, Issues in environmentally conscious manufacturing and product recovery: a survey, Comput. Ind. Eng. 36 (4) (1999) 811–853.
[27] A. Al Tawil, A. Shaban, L. Almazaydeh, A comparative analysis of convolutional neural networks for breast cancer prediction, Int. J. Electr. Comput. Eng. 14 (3) (2024) 3406–3414.
[28] A.E. Ezugwu, J.O. Agushaka, L. Abualigah, S. Mirjalili, A.H. Gandomi, Prairie dog optimization algorithm, Neural Comput. Appl. 34 (22) (2022) 20017–20065.
[29] L. Abualigah, A. Diabat, C.-L. Thanh, S. Khatir, Opposition-based Laplacian distribution with prairie dog optimization method for industrial engineering design problems, Comput. Methods Appl. Mech. Eng. 414 (2023) 116097.
[30] A. Omar, T. Abd El-Hafeez, Quantum computing and machine learning for Arabic language sentiment classification in social media, Sci. Rep. 13 (1) (2023) 17305.
[31] Q.-K. Pan, P.N. Suganthan, L. Wang, L. Gao, R. Mallipeddi, A differential evolution algorithm with self-adapting strategy and control parameters, Comput. Oper. Res. 38 (1) (2011) 394–408.
[32] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[33] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[34] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[35] S. Mirjalili, Sca: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[36] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (2016) 1053–1073.
[37] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: a new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323.
[38] B. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, 1994.
[39] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Methods Appl. Mech. Eng. 191 (11–12) (2002) 1245–1287.
[40] S. Suthaharan, S. Suthaharan, Support vector machine, in: Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning, 2016, pp. 207–235.
[41] L. Dhanabal, S. Shantharajah, A study on nsl-kdd dataset for intrusion detection system based on classification algorithms, Int. J. Adv. Res. Comput. Commun. Eng. 4 (6) (2015) 446–452.