Cluster Computing (2025) 28:267
https://doi.org/10.1007/s10586-024-04931-4

Enhanced zebra optimization algorithm for reliability redundancy allocation and engineering optimization problems

Parul Punia[1] · Amit Raj[1] · Pawan Kumar[1]

Received: 30 July 2024 / Revised: 30 October 2024 / Accepted: 17 November 2024 / Published online: 25 February 2025

Abstract
In this study, an enhanced version of the recently developed Zebra Optimization Algorithm (ZOA) is introduced, which takes inspiration from the foraging and defensive behaviors of zebras. ZOA is an efficient metaheuristic algorithm that has been used to solve various complex optimization problems. However, there is potential for further enhancement in preventing premature convergence, addressing local optima stagnation, and improving solution quality. To address these issues, two key strategies are integrated into the Enhanced Zebra Optimization Algorithm (EZOA): Levy flight and lens opposition-based learning. The Levy flight strategy promotes effective exploration in the initial foraging stage, thus preventing premature convergence to sub-optimal solutions. Next, the lens opposition-based learning strategy is introduced to improve diversity and ensure convergence towards higher-quality solutions. The proposed approach is evaluated on three test cases of optimization problems, including twenty-three classical benchmark functions, a set of CEC-2019 functions, four mixed-integer reliability optimization benchmark problems, and four constrained engineering design optimization problems. The results of EZOA are compared against several well-performing metaheuristic algorithms from the literature, along with some champion algorithms of IEEE CEC competitions. Additionally, Friedman's rank, Wilcoxon signed-rank test, and Mann–Whitney U test are employed to analyze the simulation results, providing statistical validation of the robustness and significance of the proposed EZOA, which ranks first across all test cases. Furthermore, a comprehensive evaluation of time complexity, along with boxplot and convergence analysis, conducted provides insights into the effectiveness and stability of EZOA.

Keywords Metaheuristic · Optimization · Zebra optimization algorithm · Reliability optimization · Engineering design optimization · Constrained optimization

1 Introduction

The demand for highly reliable systems is continually increasing due to the growing complexity of industrial operations. These systems are designed to handle intricate tasks, making them highly sophisticated and expensive. As a result, ensuring system reliability is crucial for optimal performance. The reliability of a system or component is the probability that it is successfully performing the operation without failing, over a specified period and under defined environmental conditions. However, designing a reliable systems is challenging due to limited availability of resources and various constraints on weight, cost, and volume. System reliability can be improved when either the reliability of individual components is improved, the number of redundant components is increased, or a combination of both strategies is applied called reliability-redundancy allocation. Optimizing the reliability and redundancy of components is the most effective approach to improving system reliability. The primary objective of reliability redundancy allocation problem (RRAP) is to determine the optimal number of redundant components required to maximize overall system reliability [1]. These

✉ Pawan Kumar
drpawan@cuh.ac.in

Parul Punia
parul@cuh.ac.in

Amit Raj
amit201335@cuh.ac.in

[1] Department of Mathematics, Central University of Haryana, Mahendergarh, Haryana 123031, India

problems are mixed-integer linear programming problems subject of various constraint, where the total number of redundant components are positive integer, and the reliability of each component is a real number between zero and one. Due to the complex nature of these optimization problems, researchers have increasingly focused on efficiently finding optimal or near-optimal solutions [2]. There are various methods used for solving such problems, including traditional methods like integer programming, lagrangian multiplier, branch and bound, and dynamic programming, but they fail when the problem is computationally intense [3, 4]. Researchers effectively shifted to use meta-heuristic approaches to offer a balance between solution quality and computational efficiency, making them suitable for tackling the intricacies of complex optimization problems in various domains.

Metaheuristic algorithms are inherently stochastic and approach optimization problems as black boxes, meaning they do not rely on detailed problem specifics or constraints. Unlike traditional methods, which rely on derivatives and intricate problem details, metaheuristic algorithms operate by observing inputs and outputs. They initialize with a random population of potential solutions and iteratively update these solutions using probabilistic rules to approximate the optimal solution. This iterative process continues until a predefined stopping criterion is met. This approach allows metaheuristics to effectively address a wide range of optimization problems that are multimodal, non-convex, and non-continuous demonstrating their flexibility and robustness across various domains [5–7]. Based on the underlying source of principle, these algorithms can be broadly grouped into three classifications: Evolutionary Algorithms (EA), Swarm Intelligence-based Algorithms (SI), and Human-inspired Algorithms (HIA). EAs are inspired by the process of biological evolution, where the fittest individuals have a higher chance of surviving and reproducing, passing their traits to the next generation. So, these algorithms mimic the process of natural selection to evolve solutions over time like, genetic algorithm (GA) [8], evolutionary-programming (EP) [9], biography-based optimization (BBO) [10], differential evolution (DE) [11], etc. SI algorithms draw inspiration from collective behaviors observed in natural swarms like, cuckoo search (CS) [12], particle swarm optimization (PSO) [13], bat algorithm (BA) [14], ant colony optimization (ACO) [15], whale optimization algorithm (WOA) [16, 17], puma optimizer (PO) [18] artificial bee colony (ABC) [19], harris- hawk optimizer (HHO) [20], jellyfish search(JS) [21], salp swarm algorithm (SSA) [22], mountain gazelle timizer (MGO) [23] etc. HIA are based on human behavior or activities. Some of the HIA include, teacher-learning based optimization (TLBO) [24], mountaineering team based optimization (MTBO) [25], sine cosine algorithm (SCA) [26–28], fireworks algorithm [29], coronavirus herd immunity optimizer (CHIO) [30] etc.

The metaheuristic algorithms are widely used to solve numerous optimization problems. However, as the "No Free Lunch (NFL)" theorem suggests, there is no single metaheuristic algorithm that is universally effective for all types of optimization problems [36]. An algorithm performing well on one particular problem might perform poorly on another. Hence, the need for more efficient approaches drives researchers to develop new or improved versions of the metaheuristic algorithms. Recent advancements in optimization have introduced several highly efficient algorithms, including the African Vulture Optimization Algorithm (AVOA) [31], the Woodpecker Mating Algorithm (WMA) [32], Gorilla Troops Optimization (GTO) [33], bobcat optimization algorithm [34]. These algorithms demonstrate both high efficiency and versatility across various optimization challenges [35]. In the improved versions, the original algorithms are either hybridized with another algorithm or updated by incorporating suitable strategies or techniques to address their inherent weaknesses. Numerous researchers have focused on enhancing existing algorithms to produce higher-quality solutions from various perspectives [37]. For instance, Sharma [38] introduced an improved BOA based on bidirectional search method termed as BBOA for solving various type of CEC and engineering optimization problems.This approach allows the algorithm to explore local solutions in both directions: forward and backward, thereby improving the rate of convergence of the algorithm towards the optimal solution. Yu et al. [39] proposed an improved version of the TLBO algorithm called ITLBO, to solve engineering optimization problems. ITLBO first integrates a feedback phase to improve the exploration phase of the algorithm. To avoid premature convergence, it applies mutation crossover operation from DE. Finally, it uses a chaotic perturbation mechanism to prevent the algorithm from getting trapped in local optima. Sahoo et al. [40] proposed an improvement to moth-flame optimization algorithm called m-DMFO, by adding modified dynamic opposition learning (m-DOL) strategy to solve various benchmark optimization problems. Valian et al. [41] introduced an enhanced Cuckoo Search (CS) algorithm by replacing the fixed values of the parameters pa and a with variable values that are adjusted dynamically throughout the iterations. These parameters are crucial in regulating the algorithm's convergence rate. This improvement allows for fine-tuning, leading to significant enhancements in the algorithm's accuracy and convergence rate. Tubishat et al. [42] implemented improvements to the SSA through two approaches. Initially, opposition-based learning (OBL) is integrated during the initialization phase to boost population diversity. Later, a novel local search algorithm is introduced in the study and is integrated with SSA to enhance the algorithm's exploitation performance. Hu et al. [43] proposed a hybrid strategy to improve the seagull optimization algorithm for mechanical engineering optimization problems. The algorithm employs chaos mapping for generating diverse population. Then a nonlinear convergence parameter and inertia weight are introduced to enhance the convergence factor and balance global exploration with local exploitation. To further prevent premature convergence, an imitation crossover mutation strategy is also implemented. The improved algorithms have proven to be highly efficient in addressing the limitations of the original algorithms and effectively solving complex optimization problems. Jia and Lang [45] improved SSA by integrating crossover and Lévy flight operations into the movement patterns of the salp leaders and followers, resulting in improved convergence accuracy and stability. Parizi et al. [44] introduced the OMWA, an enhanced version of WMA designed to address classification tasks in biomedical datasets and function approximation challenges. The OMWA integrates distance-based opposition learning, self-regulatory local memory allocation, and a dynamic approach for adjusting the $H\alpha$ parameter strategies to improve the algorithm's performance by balancing exploration and exploitation effectively. These enhancements improve adaptability and accuracy in complex problem domains demonstrated through testing on benchmark and constrained engineering design problems. Kundu and Garg [46] presented a hybrid approach using the the exploration of neural network algorithm (NNA) with the exploitation of TLBO algorithm to modify the ABC algorithm for RRAPs and unconstrained optimization problems. The resulting TLNNABC algorithm features a logarithmic spiral search operator to improve the convergence rate. Futher, a probabilistic selection procedure determines whether to apply the new search operator or to retain the original search technique. Marouani [47] enhanced the inertia weight and acceleration coefficients of the PSO algorithm by employing a normal distribution for these coefficients, leading to overall performance improvements and better handling of RRAPs. Based on the improved performance of such approaches, this paper presents an enhanced version of the recently developed zebra optimization algorithm (ZOA) for improving performance across diverse optimization challenges.

ZOA is a promising newly developed metaheuristic inspired by the natural social behaviors of zebras in the wild. This algorithm mimics two primary zebra behaviors: foraging (Phase 1) and defense against predators (Phase 2). In their foraging phase, zebras follow a lead zebra, known as the pioneer zebra, to search for food. This leader guides the herd to the optimal location. While their defensive strategy against the predator is employed in response to predators. Against weaker predators, zebras employ an offensive strategy, grouping together to intimidate and drive away the threat. When facing more formidable predators, such as lions, they use a defensive strategy, executing zig-zag movements to escape. Despite its effectiveness in solving optimization problems, ZOA faces challenges such as susceptibility to local optima, reduced population diversity, and suboptimal solution quality. To address these limitations, this paper proposes an enhanced Zebra Optimization Algorithm (EZOA). This improved algorithm incorporates two key mechanisms into the original ZOA framework. First, a Lévy-flight strategy is introduced during the foraging phase to prevent stagnation and escape sub-optimal solutions. By dynamically integrating both short and long jumps through Lévy flight, EZOA navigates the solution space more effectively and explores new regions. Second, the algorithm integrates Lens Opposition-Based Learning (LOBL) to further enhance population diversity and solution quality. This combination of strategies aims to improve the overall performance of ZOA. The main highlights of the paper are given as:

1. The proposed EZOA is developed by integrating the levy flight and LOBL strategy to address the shortcomings of ZOA. With levy-flight, the algorithms identify new search regions and escape the local optima effectively. The LOBL strategy improves population diversity and enhances the overall quality of the solutions.
2. EZOA is tested on diverse optimization cases including, classical and CEC-2019 benchmark functions, four mixed-integer RRAP, and four constrained engineering optimization problems.
3. The performance of EZOA is benchmarked against high-performing metaheuristic algorithms from the literature. The simulation results showcase a significant improvement in solution quality, highlighting EZOA's ability to consistently achieve superior outcomes.
4. To evaluate the statistical significance of EZOA, Wilcoxon signed-rank, Mann–Whitney U test and Friedman's rank tests are conducted on competing algorithms. The results confirmed the superiority of the proposed algorithm in achieving optimal results compared to the other algorithms included in the study.

The rest of the paper is structured in the following manner: Section 2 provides details of the ZOA algorithm. The proposed approach and strategies incorporated to enhance the ZOA framework are discussed in detail in Sect. 3. Section 4 provides an overview of the optimization problems considered in this study. Section 5 offers a detailed discussion of the parametric settings used in the experiments. In Sect. 6, the simulation results along with comprehensive convergence, boxplot, and statistical analysis for the first test case is presented. Sections 7 and 8 focus on the simulation results for test cases 2 and 3, respectively. Finally, Sect. 9 concludes the study, summarizing the research findings and their implications.

2 Zebra optimization algorithm (ZOA)

The recently proposed zebra optimization algorithm by Trojovska et al. [48] draws inspiration from the social dynamics observed among zebras in their natural habitat. The algorithm mimics foraging behavior and defense mechanism exhibited by zebras against their predators. Zebras reside in small herds, cooperating to find food collectively and offer protection against the predators. During the foraging phase, the zebras in the herd navigate the area, guided by a Pioneer zebra. This Pioneer zebra is the best zebra among the herd which leads the way, enabling fellow members to follow suit and search for food. The defense mechanism of zebra typically involves either escaping the predators in the zig-zag pattern or gathering together to terrify them. These behaviors are further elaborated upon in this section.

2.1 Initialization

The population-based ZOA employs zebras as its constituent members of the population with each member representing a potential solution to the optimization problem. The area occupied by the zebras serves as the search space for the solution, where location of each zebra within this space determines the values of the decision variables. The initial position of the zebra within the search space is generated randomly. The population matrix of ZOA is given in Eq. (1).

$$P = \begin{bmatrix} P_1 \\ \vdots \\ P_i \\ \vdots \\ P_n \end{bmatrix}_{n\times m} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,j} & \cdots & p_{1,m} \\ \vdots & & \vdots & & \vdots \\ p_{i,1} & \cdots & p_{i,j} & \cdots & p_{i,m} \\ \vdots & & \vdots & & \vdots \\ p_{n,1} & \cdots & p_{n,j} & \cdots & p_{n,m} \end{bmatrix}_{n\times m} \tag{1}$$

where P represents the population of zebras, n is the number of members in the population, and m is the number of decision variables. P_i is the i^{th} member of the population, and $p_{i,j}$ is the value of i^{th} zebra for j^{th} decision variable.

The objective function value associated with the zebra population is expressed as in Eq. (2).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_n \end{bmatrix}_{n\times 1} = \begin{bmatrix} F(P_1) \\ \vdots \\ F(P_i) \\ \vdots \\ F(P_n) \end{bmatrix}_{n\times 1} \tag{2}$$

where F denotes the objective function value of the population, and F_i is the objective function value of the i^{th} member (zebra) of the population. The optimal solution for the problem is determined based on the objective function value of each member of the population. In the case of minimization problems, the candidate with the lowest objective function value is the optimal solution, while in case of maximization problems, the candidate with the highest objective function value is considered optimal. As the positions of the zebras, and consequently the objective function values, are updated in each iteration, the best candidate solution is also continually updated throughout the iterations.

2.2 Foraging behavior (phase 1)

In this phase, population members are updated based on simulations of zebra foraging behavior. The primary diet of zebras consists primarily of grasses and sedges but they are adaptable and may consume buds, fruits, bark, roots, and leaves when preferred foods are scarce. Their ability to forage across diverse positions enables them to mitigate issues of food scarcity in particular areas. Within the population, the best member is referred as the pioneer zebra (Pz) which guides other members within the search space towards it. The mathematical expression corresponding to the foraging behavior is given by the following equations.

$$p_{i,j}^{Phase1}(t+1) = p_{i,j}(t) + r_d \cdot (Pz_j - I \cdot p_{i,j}(t)) \tag{3}$$

$$P_i(t+1) = \begin{cases} P_i^{Phase1}(t), & F_i^{Phase1} < F_i, \\ P_i(t), & else \end{cases} \tag{4}$$

where P_i^{Phase1} represents the updated i^{th} zebra after phase 1 and $p_{i,j}^{Phase1}$ is the j^{th} dimension value. Pz_j is the position of pioneer zebra in the j^{th} dimension. F_i^{Phase1} is the fitness value of the objective function after phase 1. r_d is random number in [0, 1], $I = round(1 + rand)$, where rand is a random number within [0, 1]. Thus, I can take on values of either 1 or 2. When $I = 2$ there are more significant changes in the movement of the population.

2.3 Defense against predators (Phase 2)

During this phase, the positions of the members of the population are updated by simulating the defensive strategies of zebras against the predator attacks. The most fatal predators for zebras are lions. While hyenas, cheetahs, canines and crocodile are among the predators that threaten the zebras. Based on the kind of predator, zebras can employ the following two strategies:

1. Exploitation: When faced with a lion's attack, zebras employ the escape strategy. In the escape strategy, zebras move in zigzag pattern and incorporate random sideways turns to evade the lion's attack and flee within the vicinity of the search space. This strategy is modeled using the mode S_E in Eq. (5).

$$S_E : p_{i,j}^{Phase2}(t+1) = p_{i,j}(t) + R \cdot (2r_d - 1) \cdot (1 - \left(\frac{t}{T}\right)) \cdot p_{i,j}(t), pb \leq 0.5 \tag{5}$$

where $p_{i,j}^{Phase2}$ is the j^{th} dimension value. t and T are the current and maximum number of iterations, R is a constant equal to 0.01. pb is any number between [0, 1] which is generated randomly and determines the probability of choosing either of the two strategies.

2. Exploration: During attack by other predators, zebras employ offensive strategy. In the offensive strategy, when other predators attack a zebra, the remaining herd gathers around the attacked zebra to intimidate and confuse the predator by forming a defensive structure. This strategy is modeled using the mode S_O in Eq. (6).

$$S_O : p_{i,j}^{Phase2}(t+1) = p_{i,j}(t) + r_d \cdot (Az_j - I \cdot p_{i,j}(t)), pb > 0.5 \tag{6}$$

where Az_j is the position of attacked zebra in the j^{th} dimension. r_d is a random number between 0 and 1.

The positions of zebras are updated and the new location is accepted if it improves the objective function value. This updating process is represented using Eq. (7).

$$P_i(t+1) = \begin{cases} P_i^{Phase2}(t), & F_i^{Phase2} < F_i, \\ P_i(t), & else \end{cases} \tag{7}$$

where P_i^{Phase2} represents the updated i^{th} zebra after phase 2. F_i^{Phase2} is the fitness value of the objective function after phase 2.

3 Proposed approach: enhanced zebra optimization algorithm

This section presents the description of levy flight and lens opposition-based learning strategy (LOBL) in Sections 3.1 and 3.2 respectively. The detail of the proposed Enhanced Zebra Optimization Algorithm is given under Sect. 3.3 and its computational complexity is discussed in Sect. 3.4.

3.1 Levy flight

A levy flight, as described by Houssein [49], involves a pattern of movement characterized by linear motion in varying random directions. It combines short steps with occasional long-distance jumps that follow a heavy-tailed distribution. The process of generating random steps in Levy flights involves selection of a direction randomly to ensure equal probability of movement across all possible directions and creation of step lengths using Levy distributions. This distribution is defined by a power-law relationship $Levy(step) \sim |step|^{-1-\beta}$ with infinite variance, where β ranges from 0 to 2 and $step$ is the step length. To overcome the challenges in directly obtaining levy distribution, Mantegna's algorithm is employed to simulate stable distributions. This approach generates random step lengths, that replicate the characteristic behaviors observed in Levy flights effectively. The step lengths are determined using the following formula:

$$step = \frac{\mu}{|v|^{1/\beta}} \tag{8}$$

$$\mu \sim N(0, \sigma_\mu^2); v \sim N(0, \sigma_v^2) \tag{9}$$

here β governs the behavior of the step lengths, mimicking the heavy-tailed distribution characteristic of Levy flights and is fixed at value $\frac{3}{2}$. Both μ and v follows normal stochastic distributions. The values of σ_μ and σ_v are determined as follows:

$$\sigma_\mu = \left[\frac{\Gamma(1+\beta) \times \sin\left(\pi \times \frac{\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}}; \sigma_v = 1 \tag{10}$$

here Γ represents the standard gamma function. The levy flight trajectory in three dimensional plane representing both small and long steps is illustrated in the Fig. 1.

3.2 Lens opposition-based learning strategy

The conventional opposition based learning (OBL) [50] strategy aims to widen the search range by simultaneously generating and evaluating the current solution in the opposite direction and selecting the best as a new solution for the next iteration. The lens opposition-based learning strategy incorporates OBL and optical principle of convex lens imaging. The theory of convex lens imaging suggests that if an object is positioned farther than twice the focal length (*f*), a reverse inverted image of that object is generated on the opposite side of the lens.

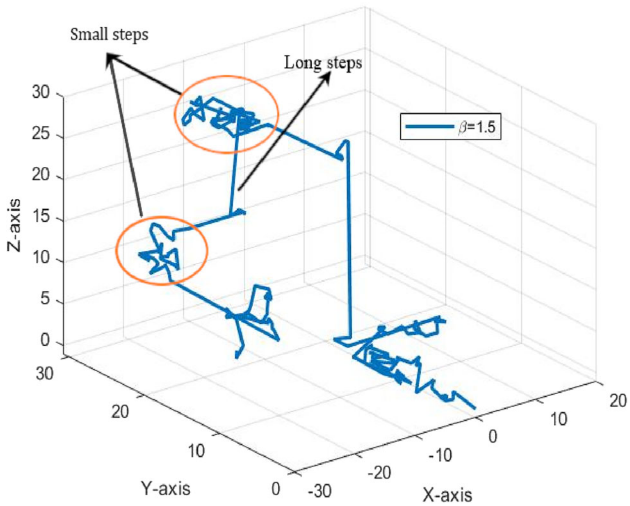

Fig. 1 Levy flight trajectory in 3-D plane

In a one-dimensional space, the coordinate axis $[L_B, U_B]$ signifies the search range, where M being the middle point of the interval, and the y-axis denoting the convex lens. Consider an object T with height H_t positioned along the coordinated axis, such that its distance from M exceeds $2f$. The projection of the object on the x-axis is PX. According to the convex lens-imaging principle, the lens refracts light to produce an reverse inverted image (T^*) of the object with height and projection on the x-axis as (H_t^*, PX^*). The visual illustration of the process is presented in Fig. 2.

The mathematical representation of the principle of convex lens imaging is given as:

$$\frac{(L_B + U_B)/2 - PX}{PX^* - (L_B + U_B)/2} = \frac{H_t}{H_t^*} \tag{11}$$

On substituting $H_t/H_t^* = s$ into Eq. (11), the resulting image (T^*) is obtained as:

$$PX^* = \frac{L_B + U_B}{2} + \frac{L_B + U_B}{2s} - \frac{PX}{s} \tag{12}$$

In general, Eq. (12) can be expanded into multidimensional space to address complex problems as:

$$PX_j^* = \frac{L_{Bj} + U_{Bj}}{2} + \frac{L_{Bj} + U_{Bj}}{2s} - \frac{PX_j}{s} \tag{13}$$

where L_{Bj} and U_{Bj} are the lower and upper bounds, and PX_j^* is the inverse solution in the j^{th} dimension. The distinction between traditional OBL and LOBL is attributed to the scale factor s. OBL serves as a special case of LOBL, where the scale factor s is set to 1. Varying the value of parameter s broadens the search range and enables the acquisition of dynamic reverse solutions making LOBL better suited for achieving the optimal solution.

After obtaining the reverse solution through LOBL strategy, fitness values of both the inverse solution and the candidate solution is compared. The one with the better fitness value is superior and is considered for the subsequent phase. The updated solution using LOBL is given as follows:

$$PX_{new} = \begin{cases} PX^*, & F(PX^*) < F(PX), \\ PX, & else \end{cases} \tag{14}$$

where PX_{new} is the updated new solution. $F(PX^*)$ and $F(PX)$ are the fitness values of the opposite solution obtained through LOBL and corresponding original solution respectively.

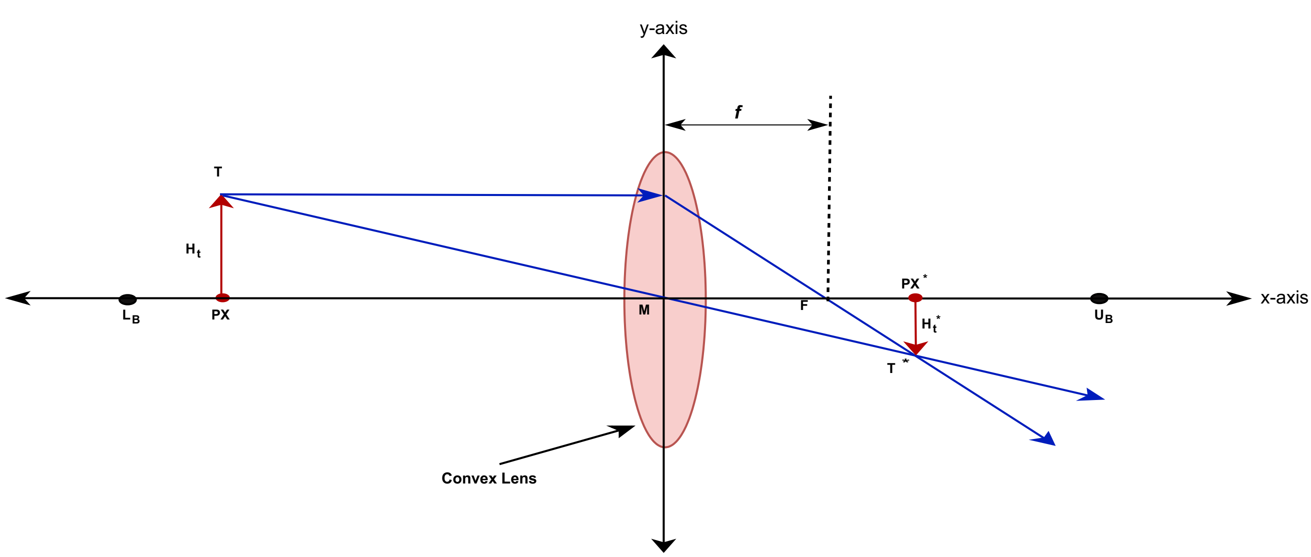

Fig. 2 Diagrman of LOBL principle

3.3 Proposed approach

Despite being efficient in handling complex and challenging optimization problems, the ZOA algorithm suffers from some inherent weaknesses, including sub-optimal stagnation and low solution quality. To overcome these issues and boost the overall performance of the algorithm, levy flight and LOBL strategies are incorporated to the original ZOA algorithm. The proposed algorithm referred to as the enhanced zebra optimization algorithm (EZOA). First, the Levy flight strategy is introduced in Phase 1 to facilitate effective exploration. This sets the stage for more targeted exploration and exploitation in the subsequent phases. Traditionally, search agents tend to converge towards the pioneer zebra, restricting exploration to local regions and leading to premature convergence to sub-optimal solutions. Levy flight introduces a combination of short and long jumps, facilitating the exploration of broader search areas in the initial stages of optimization process therby preventing premature convergence. This is mathematically given as:

$$p_{i,j}^{Phase1}(t+1) = p_{i,j}(t) + r_d \cdot (Pz_j - I \cdot p_{i,j}(t)) \cdot c \cdot Levy \tag{15}$$

here c is a constant equals to 2. Second, LOBL strategy is incorporated after the Phase 2 to improve population diversity and refine the quality of solutions through simultaneous exploration and exploitation processes. In our proposed work, the scale factor s remains fixed at 2. This strategy generates solutions that are opposite in direction but near in proximity to the current solutions. By diversifying the search space with these opposite solutions, LOBL promotes the exploration of different regions. Simultaneously, it enhances exploitation by focusing on solutions close to the current solution with the potential for higher-quality outcomes. After generating solutions through LOBL, each current solution and its inverted opposite solution are evaluated based on fitness values, and the superior solution is selected for the next iteration. This iterative refinement accelerates convergence towards superior solution regions by effectively balancing the exploration and exploitation phases, thereby improving overall solution quality. The position update of solutions through the LOBL strategy is mathematically given in equations (13) and (14).

Hence, EZOA leverages Levy flight to promote early exploration and reduce premature convergence, while LOBL improves overall solution quality by systematically exploring and exploiting diverse solution regions. The flowchart and psedudo-code of the proposed approach are given in Fig. 3 and Algorithm 1, respectively.

3.4 Computational complexity

The computational complexity of the EZOA algorithm depends on three main components: the number of agents in the population n, the dimensionality of the decision variables m, and the total number of iterations T. Initialization involves generating an initial population of n agents across m dimensions, resulting in a complexity of $O(n \times m)$. The position update strategy, executed over T iterations, consists of two phases: foraging behavior and defense against predators. Both the phases involve

Algorithm 1 EZOA pseudo-code

Require: Population size (n), Maximum iterations (T)
Ensure: Optimal solution
Set $s = 2$
Initialize the position of zebras randomly
Evaluate the objective function value of each zebra
while $t < T$ **do**
  Update pioneer zebra (Pz)
  Set $\sigma_\nu = 1$ & compute σ_μ using equation (10)
  Compute $Levy(\lambda)$ using equation (8)
  for $i = 1$ to N **do**
    Evaluate position of i^{th} zebra in the Phase 1, using equation (15)
    Update i^{th} zebra using equation (4)
    if $pb \leq 0.5$ **then**
      Evaluate position of i^{th} zebra in the Phase 2: S_E mode, using equation (5)
    else
      Evaluate position of i^{th} zebra in the Phase 2: S_O mode, using equation (6)
    end if
    Apply LOBL strategy using equation (13)
    Evaluate fitness value of each solution PX^* and update it using equation (14)
  end for
  Save the current best solution
end while
return The optimal solution obtained by EZOA

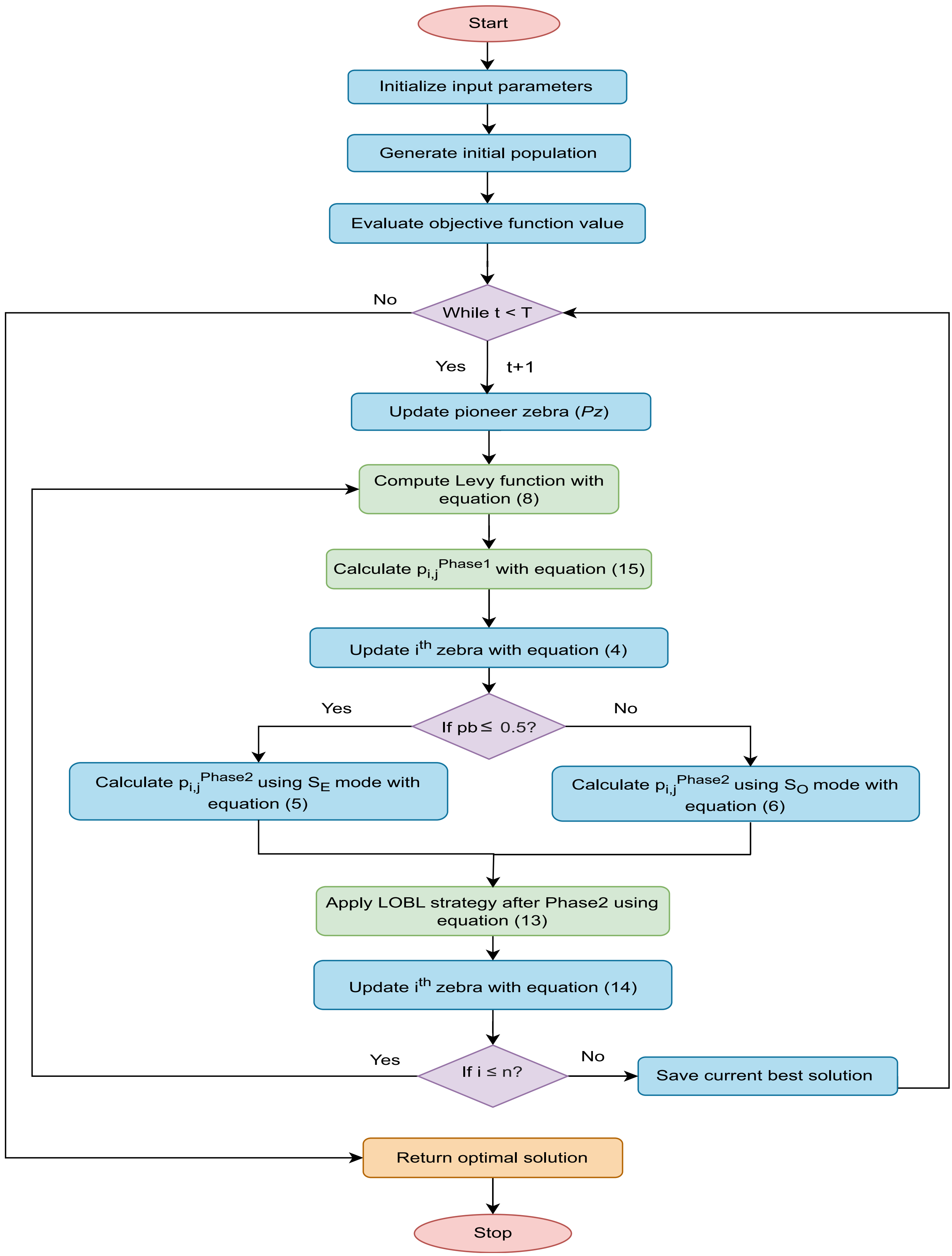

Fig. 3 Flowchart of proposed EZOA algorithm

Table 1 Details of 23 classical benchmark functions

Function	Description	Dim	Bounds	Optima	Type
Sphere	$F1(x) = \sum_{i=1}^{n} x_i^2$	[10, 30, 50, 100]	[− 100 100]	0	UN
Schwefel 2.22	$F2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[10, 30, 50, 100]	[− 10 10]	0	UN
Schwefel 1.2	$F3(x) = \sum_{i=1}^{n} \left(\sum_{j}^{i} x_j \right)^2$	[10, 30, 50, 100]	[− 100, 100]	0	UN
Schwefel 2.21	$F4(x) = \max_i(\lvert x_i \rvert, 1 \le i \le n)$	[10, 30, 50, 100]	[− 100, 100]	0	UN
Rosenbrock	$F5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[10, 30, 50, 100]	[− 30, 30]	0	UN
Step	$F6(x) = \sum_{i=1}^{n} ([x_i + 0.5^2])$	[10, 30, 50, 100]	[− 100, 100]	0	UN
Quartic	$F7(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{rand}[0.1)$	[10, 30, 50, 100]	[− 1.28, 1.28]	0	UN
Alpine	$F8(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	[10, 30, 50, 100]	[−10, 10]	0	MM
Rastrigin	$F9(x) = \sum_{i=1} n\left[x_i^{<} spanclass =' convertEndash' > 2 - 10 \cos(2\pi x_i) + 10\right]$	[10, 30, 50, 100]	[− 5.12, 5.12]	0	MM
Ackley	$F10(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) -$ $\exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + \mathrm{e}$	[10, 30, 50, 100]	[− 32, 32]	0	MM
Griewank	$F11(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[10, 30, 50, 100]	[− 600, 600]	0	MM
Salomon	$F12(x) == 1 - cos(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$	[10, 30, 50, 100]	[−100, 100]	0	MM
Schaffer Function 1	$F13(x) = 0.5 + \frac{(\sin^2(x^2 - y^2) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$	[10, 30, 50, 100]	[−15, 15]	0	MM
Shekel's Foxholes	$F14(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6})^{-1}$	2	[−65.536, 65.536]	0.998	FM
Kowalik	$F15(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[− 5, 5]	0.0003	FM
Six-hump Camel	$F16(x) = 4x_1^{<} spanclass =' convertEndash' > 2 - 2.1 x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[− 5, 5]	− 1.0316	FM
Branin	$F17(x) = (x_2 - \frac{5.1}{4\pi^2}.x_1^2 + \frac{5}{\pi}.x_1 - 6)^2 + 10 + 10\left[(1 - \frac{1}{8.\pi})\right].\cos x_1$	2	[−5,5]	0.398	FM
Goldstein-Price	$F18(x) = [1 + (x_1 + x_2 + 1)^2 (< spanclass =' convertEndash' > 19 - 14 x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (< spanclass =' convertEndash' > 18 - 32 x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2,2]	3	FM
Hartman 3	$F19(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[0, 1]	− 3.86	FM
Hartman 6	$F20(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	− 3.32	FM
Shekel 5	$F21(x) = -\sum_{i=1}^{5} (x - a_i)(x - a_i)^T + 6C_i^{-1}$	4	[0, 10]	− 10.1532	FM
Shekel 7	$F22(x) = -\sum_{i=1}^{7} (x - a_i)(x - a_i)^T + 6C_i^{-1}$	4	[0, 10]	− 10.4029	FM
Shekel 10	$F23(x) = -\sum_{i=1}^{10} (x - a_i)(x - a_i)^T + 6C_i^{-1}$	4	[0, 10]	− 10.5364	FM

Table 2 Description of CEC-2019 functions

Sr.No.	Benchmark function	Dim	Bounds	Optima
C-01	Storn's chebyshev polynomial fitting problem	9	[− 8192, 8192]	1
C-02	Invert hilbert matrix problem	16	[− 16384, 16384]	1
C-03	Lennard-jones minimum energy cluster	18	[− 4, 4]	1
C-04	Rastrigin function	10	[− 100, 100]	1
C-05	Grienwank function	10	[− 100, 100]	1
C-06	Weiersrass function	10	[− 100, 100]	1
C-07	Modified schwefel function	10	[− 100, 100]	1
C-08	Expanded schaffer F6 function	10	[− 100, 100]	1
C-09	Happy cat function	10	[− 100, 100]	1
C-10	Ackley function	10	[− 100, 100]	1

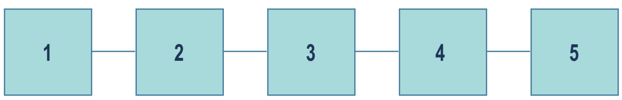

Fig. 4 Configurational diagram of series system

evaluating the fitness function for each agent, resulting in complexity $O(2 \times n \times m \times T)$. The LOBL strategy results in an additional complexity of $O(n \times m \times T)$ due to solution sorting operations. Therefore, the overall computational complexity of EZOA is $O(n \times m \times (1 + 3T))$.

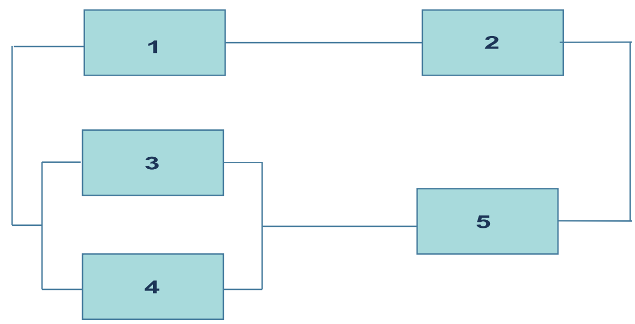

Fig. 5 Configurational diagram of series-parallel system

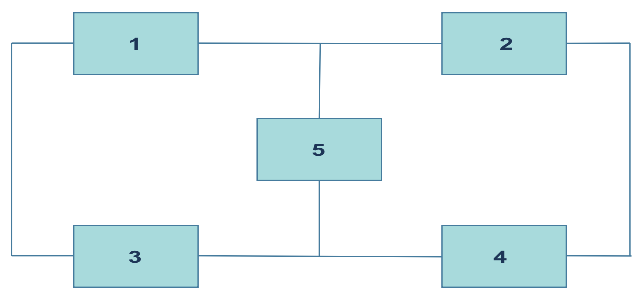

Fig. 6 Configurational diagram of complex system

4 Description of optimzation problems

To evaluate the performance of proposed EZOA algorithm, it is subjected to testing across three categories of optimization problems. The first category includes two set of benchmark functions, namely classical benchmark and CEC-2019 function. The second includes four mixed-integer linear programming RRAPs. The third includes four constrained engineering design problems.

4.1 Test case 1: benchmark functions

The classical benchmark functions are categorized into three classes: Unimodal (UN), Multional with varying dimension (MM), and Multimodal with fixed dimensions (FM) based on the number of local optima and dimensional variability [51]. Due to these characteristics, they are ideal for examining the exploration and exploitation capabilities of metaheuristic algorithms. The CEC-2019 test suite serves as a benchmark for evaluating algorithms designed for large-scale optimization challenges and exhibit diverse complexities [52]. These functions features shifts and rotation and have their global optima centered around point 1. The details and characteristics of 23 classical benchmark functions and CEC-2019 functions are presented in Tables 1 and 2, respectively.

4.2 Test case 2: reliability-redundancy allocation problems (RRAP)

Reliability is a crucial aspect in ensuring consistent operation of systems (or products) across various domains. As the demand for highly reliable and efficient systems is continously increasing in today's world, the need for reliability analysis is also becoming increasingly important. Reliability optimization effectively analyses the systems and determines the good-quality reliable ones that efficiently perform their required function over their intended time period. The general experession of reliability optimization problems is given as:

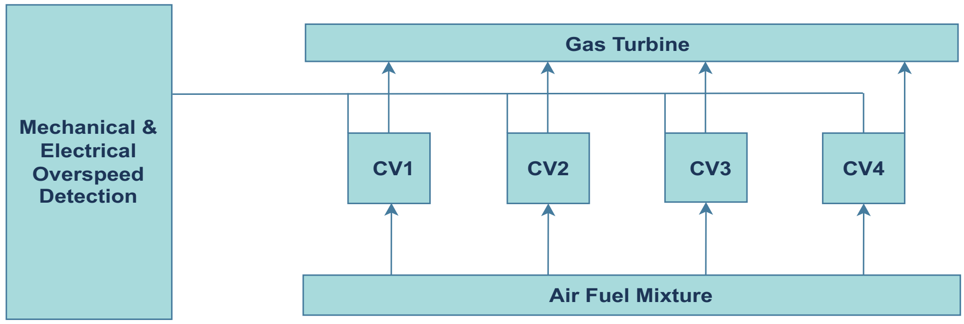

Fig. 7 Configurational diagram of gas turbine overspeed protection system

Maximize: $R_{sys}(r_i, n_i)$
Subject to: $g_c(r_i, n_i) \leq \alpha$
$0 \leq r_i \leq 1, n_i \in \mathbb{Z}^+, i = 1, 2, ..., m$

here $R_{sys}(r_i, n_i)$ is the overall reliability of the whole system. The system comprises total m subsystems, where n_i and r_i represent the number of components and the reliability of each component in i^{th} subsystem, respectively. The reliability of a substem 'i' is calculated as $R_i(n_i) = 1 - q_i^{n_i}$, where $q_i = 1 - r_i$ is the probability of component failure in subsystem 'i'. g_c denotes the set of constraints.

This section outlines four RRAP analyzed in the study to assess the performance of EZOA in optimizing these challenges. The four RRAP include series, series-parallel, complex-bridge, and gas turbine overspeed protection systems. The objective of these benchmark problems is to enhance system reliability while adhering to diverse non-linear constraints, taking into account both the reliability and redundancy of the components. In each of these cases, a few assumptions have been taken into account, as outlined below.

4.3 Assumptions

1. All redundancies are active and are not subject to repair.
2. A failure in any subsystem component will not cause the entire system to fail.
3. The system and its components can exist in only two possible states: functional or failed.

4.3.1 Constraints handling in optimization algorithms

For solving constrained optimization problems effective management of constraints is required as these problems often involve both equality and inequality constraints, making it challenging to identify feasible solutions. To address this, metaheuristic algorithms frequently employ penalty functions. The penalty functions adds a penalty to the objective function when constraints are violated, effectively transforming a constrained problem into an unconstrained one. In an exterior penalty function approach, solutions that do not meet the constraints are considered, but the objective function is adjusted with a penalty to reflect the constraint violations. It adjusts the objective function to include these penalties, aiming to find solutions that are both feasible and optimal. The penalty function method to maximize the constrained optimization problem is given as:

$$F(r, n) = f(r_i, n_i) - \lambda \sum_{c=1}^{l} max(0, g_c(r_i, n_i) - \gamma_c) \tag{16}$$

where the objective function f subjected to constraints g_c is to be maximized. l being the total numbers of constraints and λ is the penalty coefficient.

4.3.2 RRAP-1: series system

The series system is structured with five subsystems in a series configuration, as illustrated in Fig. 4. This is a mixed-integer programming problem with three non-linear constraints on the weight, volume, and cost of each component within the subsystem [53, 54]. The desciption of the system is mathematically given as:

$$\text{Maximize: } R_{\text{sys}} = \prod_{i=1}^{5} R_i(n_i) \tag{17}$$

$$g_1(r, n) = \sum_{i=1}^{5} v_i n_i^2 - V \leq 0 \tag{18}$$

$$g_2(r, n) = \sum_{i=1}^{5} \alpha_i \left(\frac{-1000}{ln(r_i)} \right)^{\beta_i} [n_i + e^{(0.25 n_i)}] - C \leq 0 \tag{19}$$

$$g_3(r, n) = \sum_{i=1}^{5} w_i n_i e^{(0.25 n_i)} - W \leq 0 \tag{20}$$

$$0 \leq r_i \leq 1, 0 \leq n_i \leq 1, n_i \in \mathbb{Z}^+, i = 1, 2, ..., 5 \tag{21}$$

where $c_i = \alpha_i \left(\frac{-1000}{ln(r_i)} \right)^{\beta_i}$. The reliability of i^{th} subsystem is given by $R_i(n_i) = 1 - q_i^{n_i}$ with q_i being the probability of

Table 3 Parameter setting of the algorithms

Algorithm	Parameter	Value
	Population Size (n)	30
	Number of Iterations	1000
	Runs	30
WOA	l	[− 1, 1]
	a	DecreasingLlinearly from 2 to 0
	Modular Modality (c)	0.1
BOA	Power Exponent (a)	0.01
	Switch Probability (P)	0.6
HHO	E_0	(− 1, 1)
	P_{max}	4
TSA	P_{min}	1
	$c_i, i = 1, 2, 3$	Random Numbers in [0,1]
RSO	R	[1, 5]
	C	[0, 2]
POA	R	0.2
ZOA	Weight Parameter (I)	[1, 2]
	Constant Parameter (R)	0.01
	pb	[0, 1]

Table 4 Data for RRAP1 and RRAP3

i	$10^5\alpha_i$	w_i	v_i	β_i	W	V	C
1	2.330	7	1	1.5	200	110	175
2	1.450	8	2	1.5			
3	0.541	8	3	1.5			
4	8.050	6	4	1.5			
5	1.950	9	2	1.5			

Table 5 Data for RRAP2

i	$10^5\alpha_i$	w_i	v_i	β_i	W	V	C
1	2.500	3.5	2	1.5	100	180	175
2	1.450	4.0	4	1.5			
3	0.541	4.0	5	1.5			
4	0.541	3.5	8	1.5			
5	2.100	3.5	4	1.5			

component failure. w_i, v_i, and c_i are weight, volume, and cost of component in i^{th} subsystem. W, V, and C represents the upper bounds on weight, volume, and cost of the system. $g_1(r,n)$, $g_2(r,n)$, and $g_3(r,n)$ are the constraints on volume, cost, and weight. The parameters for series system is given in Table 4.

Table 6 Data for RRAP4

i	$10^5\alpha_i$	w_i	v_i	β_i	W	V	C
1	1.0	6	1	1.5	500	250	400
2	2.3	6	2	1.5			
3	0.3	8	3	1.5			
4	2.3	7	2	1.5			

4.3.3 RRAP-2: series-parallel system

The configurational diagram of series-parallel system is presented in Fig. 5. The system also consistes of five subsystems in series-parallel arrangement [55–57]. The mathematical formulation of the problem is given below:

$$\text{Maximize: } R_{sys} = 1 - (1 - R_1R_2)[1 - (1 - (1 - R_3)(1 - R_4))R_5] \tag{22}$$

where $0 \le r_i \le 1, 0 \le n_i \le 1, n_i \in \mathbb{Z}^+, i = 1, 2, ..., 5.$ The problem has same constrainsts $g_1(r,n)$, $g_2(r,n)$, and $g_3(r,n)$ on volume, cost, and weight as those of the series system, which are given in Equations (18) to (20). The parametic values are provided in Table 5.

4.3.4 RRAP-3: bridge system

The structural design of bridge system problem is illustrated in Fig. 6 [58, 59]. The optimization problem is given in the following equation:

Table 7 Some metaheuristic algorithms considered for RRAP problems from literature

Algorithm	Description	Author	Publication Year
GA [55]	Genetic algorithm	Yokota et al.	1996
PSO/SSO [71]	Particle swarm optimization algorithm	Huang	2015
CPSO [72]	Co-evolutionary PSO	He and Wang	2007
MPSO [73]	Modified PSO	Liu and Qin	2014a
IPSO [56]	Improved PSO	Wu et al.	2011
SCA [58]	Soft computing approach	Gen and Yun	2006
SAA [74]	Simulated annealing algorithm	Kim et al.	2006
ABC1 [75]	Artificial bee colony	Yeh and Hsieh	2011
ABC2 [76]	Artificial bee colony	Garg et al.	2013
IA [61]	Immune based two-phase approach	Hsieh and You	2011
CS1 [60]	Cuckoo search (CS) algorithm	Valian and Valian	2013
CS2 [77]	CS algorithm	Garg	2015a
CS-GA [78]	Hybrid CS and GA algorithm	Kanagaraj et al.	2013
ICS [41]	Improved CS algorithm	Valian et al.	2013
GA-SRS [53]	RRAP (with cold redundancy)	Ardakan and Hamadani	2014
MICA [54]	Modified imperialist competitive algorithm	Afonso et al.	2013
NAFSA [79]	Novel artificial fish swarm algorithm	He et al.	2015
DE [80]	Differential evolution with levy flight strategy	Liu and Qin	2015
NMDE [81]	Nover modified DE	Zou et al.	2011
TS-DE [82]	Hybrid TS and DE algorithm	Liu and Qin	2014b
NGHS [59]	Novel global harmony search algorithm	Zou et al.	2010
INGHS [57]	Improved NGHS algorithm	Ouyang et al.	2015
EBBO [62]	Efficient biography-based optimization	Garg	2015b
TLNNABC [45]	Hybrid NNA and Teaching-learning-based optimization algorithm	Kundu and Garg	2022a
INNA [83]	Improved neural network algorithm	Kundu and Garg	2022b

$$\text{Maximize: } R_{\text{sys}} = R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 \tag{23}$$

$$- R_1R_2R_4R_5 - R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \tag{24}$$

$$0 \le r_i \le 1, 0 \le n_i \le 1, n_i \in \mathbb{Z}^+, i = 1, 2, ..., 5 \tag{25}$$

The constraints on volume, cost, and weight, denoted as $g_1(r,n)$, $g_2(r,n)$, and $g_3(r,n)$ respectively are identical to those of the series system, as described in Eqs. (18) to (20). The specific parameter values are outlined in Table 4.

4.3.5 RRAP-4: gas turbine overspeed protection system

Overspeed protection systems prevent engines or turbines, from exceeding their maximum safe operating speeds [60–62]. These systems employ a four-stage series configuration of control valves (CV1–CV4) designed to cease the fuel supply if an overspeed condition occurs. The mixed-integer non-linear programming problem of the RRAP-4 system has three constrainsts. The problem is mathematically described below as:

$$\text{Maximize: } R_{\text{sys}} = \prod_{i=1}^{4} R_i(n_i) \tag{26}$$

$$g_1(r,n) = \sum_{i=1}^{4} v_i n_i^2 - V \le 0 \tag{27}$$

$$g_2(r,n) = \sum_{i=1}^{4} \alpha_i \left(\frac{-1000}{ln(r_i)}\right)^{\beta_i} [n_i + e^{(0.25n_i)}] - C \le 0 \tag{28}$$

$$g_3(r,n) = \sum_{i=1}^{4} w_i n_i e^{(0.25n_i)} - W \le 0 \tag{29}$$

$$0 \le r_i \le 1, 0 \le n_i \le 1, n_i \in \mathbb{Z}^+, i = 1, 2, ..., 5 \tag{30}$$

Table 6 provides the parameter values and Fig. 7 illustrates the schematic diagram of the problem.

Table 8 Comparative analysis of variants of enhanced ZOA

Functions	Metric	EZOA	OZOA	LZOA
F1	Avg	0	0	4.60E-95
	SD	0	0	2.52E-94
F2	Avg	0	0	7.34E-53
	SD	0	0	2.58E-52
F3	Avg	0	0	1.28E-25
	SD	0	0	4.92E-25
F4	Avg	0	0	3.43E-44
	SD	0	0	1.17E-43
F5	Avg	25.1933	27.9382	25.3346
	SD	0.10025	0.77032	0.14101
F6	Avg	1.18E-07	1.74E+00	2.43E-07
	SD	1.10E-07	4.73E-01	2.38E-07
F7	Avg	1.04E-04	2.75E-05	2.82E-04
	SD	6.23E-05	1.42E-05	1.18E-04
F8	Avg	0	0	2.34E-54
	SD	0	0	8.88E-54
F9	Avg	0	0	0
	SD	0	0	0
F10	Avg	4.44E-16	4.44E-16	4.44E-16
	SD	0	0	0
F11	Avg	0	0	0
	SD	0	0	0
F12	Avg	6.02E-11	6.21E-11	2.01E-09
	SD	1.56E-10	1.86E-10	3.70E-09
F13	Avg	0	0	0
	SD	0	0	0
F14	Avg	0.998	2.6765	0.998
	SD	0	2.1487	0
F15	Avg	3.42E-04	3.87E-04	4.61E-04
	SD	4.88E-05	8.44E-05	9.11E-05
F16	Avg	− 1.0316	− 1.0316	− 1.0316
	SD	6.78E-16	9.82E-11	6.78E-16
F17	Avg	0.39789	0.39789	0.39789
	SD	0	3.87E-09	0
F18	Avg	3	4.8	3
	SD	9.00E-16	6.85E+00	9.79E-16
F19	Avg	− 3.8628	− 3.8626	− 3.8628
	SD	2.71E-15	1.53E-04	2.28E-15
F20	Avg	− 3.32E+00	− 3.30E+00	− 3.32E+00
	SD	1.37E-15	4.97E-02	2.14E-06
F21	Avg	− 10.1532	− 9.1325	− 10.1529
	SD	6.85E-15	2.09E+00	1.08E-03
F22	Avg	− 10.4029	− 10.1371	− 10.1372
	SD	1.14E-15	1.19E+00	1.89E+00
F23	Avg	− 10.5364	− 9.4548	− 10.2660
	SD	1.95E-15	1.66E+00	1.21E+00
Friedman	Mean	1.50	2.24	2.26
Ranking	Overall	1	2	3

4.4 Test case 3: constrained engineering optimization problems

The third test case includes four engineering design optimization problems: corrugated bulkhead design case, tubular column design case, spring design, and welded beam design case that are considered to examine the robustness of algorithms considered in the study. The design optimization problems comprise equality and inequality constraints, requiring the algorithm to apply efficient methods to navigate these complexities efficiently. The detailed descriptions of each problem are provided alongside their results in next section.

5 Parametrical and experimental setting

5.1 Test case 1 and 3

In test case 1, the proposed EZOA is evaluated against seven latest and highly efficient metaheuristic algorithms: ZOA, WOA, HHO, Butterfly optimization algorithm (BOA) [63], Tunicate swarm algorithm (TSA) [64], Rat swarm optimizer (RSO) [65], Pelican optimization algorithm (POA) [66]. Test case 3 evaluates the applicability and robustness of the algorithms across four constrained engineering design problems. In this case, the results of the EZOA algorithm are compared with ten well-recognized metaheuristic algorithms: ZOA, WOA, HHO, BOA,TSA, RSO, Fox optimizer (FOX) [67], Ant lion optimizer (ALO) [68], Seagull optimization algorithm (SOA) [69], and Sooty tern optimization algorithm (STOA) [70]. All the competitor algorithms are tested under identical conditions, with 50 search agents and a maximum of 1000 iterations. To ensure a fair evaluation of algorithmic performance and reduce the impact of stochastic variability on the results, each algorithm undergoes 30 runs. The parameters associated with each algorithm are taken from their respective articles and outlined in Table 3. The experimental results are analyzed using metrices like average, best, standard deviation, and rank.

5.2 Test case 2

In test case 2, four mixed-integer linear programming RRAP problems are addressed using metaheuristic algorithms. The comparison includes metaheuristic algorithms referenced from the literature as listed in Table 7. Both the EZOA and ZOA algorithms begin with an initial population size of 100. Each algorithm is executed 30 times with a maximum number of iterations fixed at 1500 to ensure consistent and reliable results. Statistical analysis of ZOA and EZOA includes evaluation metrics such as the average of the best results, best and worst values obtained, and standard deviation.

The experiments for this study are carried out on a computer with an Intel(R) Core(TM) i5-10210U CPU operating at 1.60 GHz and 2.11 GHz, and equipped with 16GB of RAM, using MATLAB R2021a.

6 Simulation results on test case 1: benchmark functions

This section presents the experimental results obtained for the proposed EZOA and the competing algorithms across the first test case.

6.1 Analysis of various strategy combinations of enhanced ZOA

In Sect. 3, two key enhancements to the zebra optimization algorithm are proposed aimed at improving its overall performance. To assess the effectiveness of these improvements, three algorithmic variants are compared: the proposed EZOA, ZOA integrated with an opposition-based learning technique (OZOA), and ZOA with a Levy flight technique (LZOA). Each variant is independently executed 30 times, with fixed population size 50 and iterations 1000 across classical benchmark functions. Dimension for the functions F1-F13, dimension is fixed to 30, and the function F14–F23 have fixed dimension.

The experimental results presented in Table 8 demonstrate notable performance improvement by EZOA over both OZOA and LZOA. EZOA exhibited superior performance by attaining global minima on 17 out of 23 functions. EZOA either outperformed OZOA or attained global optima, except for function F7. Whereas EZOA outperformed LZOA on 13 functions, while both algorithms attained global optima on the remaining functions. The friedman ranking of these strategie as reported in table reveals that EZOA performed the best by achieving minimum average rank of 1.50. The rests of Wilcoxon-signed rank test further validates the effectiveness of the proposed

Table 9 Wilcoxon-signed rank test results at 0.05 significance level for the improved strategies on the benchmark test function

Competitor Algorithms	Mean value	Standard-deviation	Z-Value	p-value	Significant
EZOA vs OZOA	39.00	12.75	− 2.9025	3.74E-03	Yes
EZOA vs LZOA	52.5	15.93	− 2.9191	3.50E-03	Yes

Table 10 Simulation outcomes of the algorithms on classical benchmark functions with dimension 10

Function	Algorithm								
	Dim = 10	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
F1	Avg	0	0	8.52E-192	1.41E-94	0	5.05E-220	5.78E-14	1.13E-177
	SD	0	0	0	6.05E-94	0	0	7.91E-15	0
	Best	0	0	2.85E-219	4.19E-103	0	4.57E-245	4.39E-14	1.57E-193
	Rank	2	2	5	7	2	4	8	6
F2	Avg	0	4.32E-302	3.01E-102	1.28E-54	9.24E-279	1.57E-109	3.07E-11	8.51E-113
	SD	0	0	1.39E-101	3.76E-54	0	5.89E-109	6.48E-12	3.93E-112
	Best	0	3.38E-314	3.65E-112	2.05E-59	0	1.15E-119	1.78E-11	2.50E-121
	Rank	1	2	6	7	3	5	8	4
F3	Avg	0	0	4.48E-179	1.53E-60	0	6.33E-214	5.74E-14	8.81E-01
	SD	0	0	0	7.99E-60	0	0	6.69E-15	1.6568
	Best	0	0	2.09E-205	9.98E-74	0	1.46E-239	4.53E-14	2.71E-20
	Rank	2	2	5	6	2	4	7	8
F4	Avg	0	5.12E-271	2.38E-99	2.11E-26	7.38E-264	1.03E-106	5.13E-11	0.091993
	SD	0	0	8.28E-99	6.85E-26	0	4.20E-106	3.53E-12	0.49605
	Best	0	3.48E-279	7.71E-108	1.36E-29	0	3.61E-118	4.35E-11	1.65E-13
	Rank	1	2	5	6	3	4	7	8
F5	Avg	4.8871	6.9961	0.000456	8.5873	8.1628	5.6366	8.905	5.5571
	SD	0.088529	0.72184	0.000538	1.0162	0.21154	0.60218	0.020868	0.34489
	Best	4.7117	6.0804	9.26E-07	5.4817	7.7645	4.9456	8.8697	4.8597
	Rank	2	5	1	7	6	4	8	3
F6	Avg	7.62E-28	0.080658	5.18E-06	1.1362	0.31615	0.053304	0.64296	9.55E-06
	SD	3.06E-27	0.16377	7.02E-06	0.312	0.22315	0.10198	0.20543	7.67E-06
	Best	1.54E-32	1.12E-08	3.67E-11	0.50604	2.63E-05	1.85E-07	0.30205	2.11E-06
	Rank	1	5	2	8	6	4	7	3
F7	Avg	8.03E-05	1.74E-05	4.38E-05	0.000853	8.36E-05	6.89E-05	0.000849	0.000727
	SD	4.27E-05	1.17E-05	4.02E-05	0.000618	7.22E-05	5.42E-05	0.000267	0.000763
	Best	1.83E-05	5.55E-07	1.60E-06	0.000166	2.12E-06	7.93E-06	0.000286	6.25E-06
	Rank	4	1	2	8	5	3	7	6
F8	Avg	0	2.63E-303	4.66E-103	2.55E+00	6.51E-237	2.73E-111	1.01E+00	5.69E-01
	SD	0	0	1.95E-102	1.50E+00	0	8.79E-111	1.94E+00	1.04E+00
	Best	0	1.44E-311	6.93E-115	1.66E-01	0	9.90E-122	6.82E-13	4.53E-123
	Rank	1	2	5	8	3	4	7	6
F9	Avg	0	0.27664	0	19.9458	0	0	27.6523	0
	SD	0	1.5152	0	8.5833	0	0	15.2063	0
	Best	0	0	0	5.5896	0	0	0	0
	Rank	3	6	3	7	3	3	8	3
F10	Avg	4.44E-16	4.44E-16	4.44E-16	1.8044	5.63E-16	3.05E-15	2.02E-14	3.64E-15
	SD	0	0	0	1.7234	6.49E-16	1.60E-15	3.18E-14	2.16E-15
	Best	4.44E-16	4.44E-16	4.44E-16	4.00E-15	4.44E-16	4.44E-16	4.44E-16	4.44E-16
	Rank	2	2	2	8	4	5	7	6
F11	Avg	0	0	0	0.39153	0	0	0.004592	0.051119
	SD	0	0	0	0.17443	0	0	0.025149	0.10254
	Best	0	0	0	0.093676	0	0	0	0
	Rank	3	3	3	8	3	3	6	7
F12	Avg	8.00E-11	5.40E-10	0	3.10E-09	4.63E-09	9.35E-10	4.09E-09	5.53E-13
	SD	2.66E-10	7.65E-10	0	6.15E-09	9.92E-09	1.80E-09	8.90E-09	1.65E-12
	Best	6.44E-15	1.55E-14	0	8.18E-14	00	1.07E-13	3.92E-12	0

Table 10 (continued)

Function	Algorithm								
	Dim = 10	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
	Rank	3	4	1	6	8	5	7	2
F13	Avg	0	0	0	1.77E-03	0	0	2.08E-03	1.04E-04
	SD	0	0	0	1.58E-03	0	0	1.50E-03	5.71E-04
	Best	0	0	0	0	0	0	0	0
	Rank	3	3	3	7	3	3	8	6
Friedman	Average	2.15	3	3.31	7.17	3.92	3.92	7.31	5.23
Rank	Overall	1	2	3	4.5	4.5	8	7	6

enhancement strategies in significantly elevating the algorithm's performance, as given in Table 9.

6.2 Analysis on classical benchmark functions

To analyse the performance of proposed algorithm, it is compared against other metaheuristic optimization algorithms. Functions F1–F7 represent unimodal functions each having one global minima and no local minima. These functions assess the exploitation capability of the algorithms. The functions F8–F23 are multimodal functions where F8–F13 have varying dimensions and F14–F23 have fixed dimensions. These functions contain multiple local minima but only one global optimum, making them ideal for assessing the exploration capabilities of the algorithms. The dimensions of the multidimensional scalable functions F1–F13 are varied across 10, 30, 50, and 100 dimensions. This variation is implemented to evaluate the EZOA's efficiency in tackling high-dimensional optimization problems. By testing across multiple dimensional settings, the analysis provides insights into the scalability and robustness of the algorithm. The simulation outcomes on these functions are tabulated in Tables 10, 11, 12, 13, 14. The results also presents the friedman rank for each function and overall ranking as well.

The results for functions F1–F13 at dimension 10, as shown in Table 10, indicate that EZOA successfully achieved global minima on eight functions. HHO performace is better than EZOA on functions F5 and F12. In case of function F6, EZOA outperforms all the algorithms. The standard deviation values of EZOA is quite low for the functions suggesting its consistency in achieving better results. Also, EZOA ranks first with minimum average friedman rank of 2.15.

Table 11 provides results for functions F1–F13 at a dimension of 30, where EZOA achieved optimal solutions on eight functions and demonstrated superior performance on function F6 compared to other algorithms. Although HHO showed stronger performance on functions F5 and F12, EZOA consistently maintained the lowest standard deviation across all functions, securing an average Friedman rank of 2.19. These findings suggest that EZOA's performance remains robust as dimensionality increases.

Table 12 presents the results for functions F1–F13 with a dimension of 50. Despite the increased dimensionality, EZOA achieved optimal solutions on eight functions, specifically F1–F4, F8, F9, F11, and F13. In these tests, EZOA consistently performed on par with or better than other optimizers, demonstrating competitive results across a range of functions. However, on functions F5, F6, and F7, EZOA was outperformed by the HHO algorithm. Overall, EZOA consistently delivered high-quality solutions across the function with average friedman rank of 2.23 and demonstrated exceptional performance in most cases.

For functions F1–F13 at a dimension of 100, shown in Table 13, EZOA sustained its performance on eight of the 13 functions. Compared to other algorithms, it showed relatively lower performance on only four functions, while it excelled across the majority, consistently delivering high-quality solutions with low standard deviation values. EZOA achieved the top ranking with an average Friedman rank of 2.38, reaffirming its competitiveness in providing robust, high-performance solutions across different problem dimensions.

In the case of fixed-dimension multimodal functions (F14–F23), EZOA showcased exceptional behavior by attaining global optimal on all the functions except F15, where it still managed to give competitive result compared to the other algorithms. It ranked first with average rank of 2.3.

EZOA ranked as the top performer or matched the best in 18 out of 23 functions, in terms of average and standard

Table 11 Simulation outcomes of the algorithms on classical benchmark functions with dimension 30

Function	Algorithm								
	Dim = 30	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
F1	Avg	0	0	2.10E-194	1.39E-51	0	6.10E-214	1.05E-13	6.08E-168
	SD	0	0	0.00E+00	6.87E-51	0	0	8.30E-15	0
	Best	0	0	1.06E-213	2.32E-57	0	2.23E-231	9.02E-14	6.06E-186
	Rank	2	2	5	7	2	4	8	6
F2	Avg	0	1.12E-267	8.57E-103	2.26E-31	6.35E-279	1.24E-107	6.06E-11	6.12E-109
	SD	0	0	3.09E-102	3.60E-31	0	4.01E-107	4.88E-12	1.92E-108
	Best	0	1.10E-273	2.89E-111	2.62E-33	0	1.10E-121	4.83E-11	2.31E-117
	Rank	1	3	6	7	2	5	8	4
F3	Avg	0	0	5.74E-158	1.65E-16	0	2.95E-208	9.21E-14	11904.36
	SD	0	0	3.15E-157	8.56E-16	0	0	6.64E-15	8133.084
	Best	0	0	2.70E-195	1.03E-25	0	3.13E-239	7.74E-14	786.9545
	Rank	2	2	5	6	2	4	7	8
F4	Avg	0	6.14E-231	1.09E-97	0.000369	1.74E-230	1.72E-102	7.39E-11	34.6306
	SD	0	0	5.38E-97	0.00077	0	9.43E-102	4.79E-12	28.5649
	Best	0	1.06E-240	8.65E-106	2.54E-07	0	3.61E-119	6.50E-11	0.003603
	Rank	1	2	5	7	3	4	6	8
F5	Avg	25.1933	27.7492	0.001043	28.2623	28.5645	27.554	28.8863	26.513
	SD	0.10025	0.84911	0.00127	0.79656	0.35115	1.0869	0.023549	0.31263
	Best	25.0092	26.3401	5.51E-05	25.4887	28.1038	25.3316	28.8426	25.9671
	Rank	2	5	1	6	7	4	8	3
F6	Avg	1.18E-07	1.6556	1.56E-05	3.7742	3.3897	2.5423	4.8122	0.004539
	SD	1.10E-07	0.45175	2.62E-05	0.70084	0.4183	0.53241	0.5977	0.002426
	Best	1.90E-08	0.998	2.10E-10	2.058	2.1113	1.4995	3.3655	0.001567
	Rank	1	4	2	7	6	5	8	3
F7	Avg	0.000104	3.38E-05	3.62E-05	0.002997	0.000118	9.47E-05	0.0008	0.001011
	SD	6.23E-05	2.43E-05	4.18E-05	0.001319	0.000134	5.56E-05	0.000262	0.001157
	Best	1.28E-05	1.01E-05	6.31E-07	0.000898	3.43E-06	2.21E-05	0.000405	3.36E-06
	Rank	4	1	2	8	5	3	6	7
F8	Avg	0	2.79E-267	4.81E-101	2.43E+01	6.05E-228	4.12E-105	1.36E-13	2.12E-73
	SD	0	0	1.92E-100	4.97E+00	0	1.71E-104	2.52E-13	9.47E-73
	Best	0	1.11E-274	2.73E-113	1.45E+01	0	4.83E-120	1.12E-17	2.70E-119
	Rank	1	2	5	8	3	4	7	6
F9	Avg	0	0.96828	0	156.188	0	0	6.3383	0
	SD	0	5.3035	0	41.7686	0	0	33.8051	0
	Best	0	0	0	74.5666	0	0	0	0
	Rank	3	6	3	8	3	3	7	3
F10	Avg	4.44E-16	4.44E-16	4.44E-16	2.1608	9.18E-16	2.93E-15	3.22E-11	3.52E-15
	SD	0	0	0	1.4589	1.23E-15	1.66E-15	1.30E-11	2.23E-15
	Best	4.44E-16	4.44E-16	4.44E-16	7.55E-15	4.44E-16	4.44E-16	1.11E-11	4.44E-16
	Rank	2	2	2	8	4	5	7	6
F11	Avg	0	0	0	0.006476	0	0	5.14E-16	0.003235
	SD	0	0	0	0.009307	0	0	9.44E-16	0.01232
	Best	0	0	0	0	0	0	0	0
	Rank	3	3	3	8	3	3	6	7
F12	Avg	7.73E-11	7.19E-10	0	2.21E-09	7.19E-10	9.41E-10	4.84E-09	2.95E-12
	SD	1.24E-10	1.43E-09	0	4.24E-09	1.61E-09	1.52E-09	1.06E-08	1.40E-11
	Best	1.11E-16	6.94E-13	0	2.02E-12	0	5.00E-15	3.32E-11	0

Table 11 (continued)

Function	Algorithm								
	Dim = 30	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
	Rank	3	5	1	7	4	6	8	2
F13	Avg	0	0	0	1.98E-03	0	0	0.002397	0
	SD	0	0	0	0.001532	0	0	0.001345	0
	Best	0	0	0	0	0	0	0	0
	Rank	3.5	3.5	3.5	7	3.5	3.5	8	3.5
Friedman	Average	2.19	3.12	3.35	7.23	3.65	4.12	7.23	5.12
Rank	Overall	1	2	3	7.5	4	5	7.5	6

deviation. This demonstrates that EZOA consistently identifies solutions with less variation, underscoring its practical superiority over other algorithms. The algorithms are also evaluated based on their average runtime required to achieve optimal results. Table 15 presents the average runtime for each algorithm on classical benchmark functions, and these results are also depicted in Fig. 8. The analysis reveals that there isn't a substantial difference in runtime among the algorithms, except for function F14. However, WOA and RSO stand out for their overall efficiency, often yielding quicker results.

6.3 Analysis on CEC-2019 benchmark functions

The CEC-2019 functions provide a comprehensive testbed for evaluating the efficiency of the proposed algorithm as it includes complex unimodal, multimodal, separable, and non-separable functions. The comparison results are presented in Table 16. The performance of EZOA is at par, as it outperformed all the seven efficient metaheuristic algorithms considered in the study on 9 out of 10 functions while attaining second rank on the C-01 function. The results reveal that the proposed approach EZOA is superior to ZOA on all functions. The performace of the algorithms is in the oreder: EZOA> POA> ZOA> HHO> WOA> RSO> TSA > BOA. As evident from the lower standard deviation values, EZOA showcases consistently superior results. Next, the average runtime for each algorithm is evaluated for these functions, as shown in Table 17 and Fig. 9. The results indicate that all algorithms generally take more time for C-01 and C-06. Notably, EZOA requires more time compared to the other algorithms. However, this increased computation time is compensated by its superior performance in attaining optimal solutions, as evidenced by its first rank on nine out of ten functions. Hence, EZOA exhibits exceptional performance across all benchmark functions (C-02 to C-10) in terms of average, low standard deviation value, and best value.

6.4 Comparison analysis with some highly efficient metaheuristic optimizers

The proposed EZOA was further evaluated for its effectiveness in high-dimensional optimization tasks, comparing its performance against leading optimizers EBOwithCMAR, LSHADE, CMA-ES, and SPS_L_SHADE_EIG. The test set included classical benchmark functions F1–F23, where the functions F1–F13 were set to a dimension of 500, and the functions F14-F23 are fixed-dimensional. Each algorithm was executed 30 times, with a maximum of 50,000 function evaluations allowed per run. Table 18 presents the results, reporting the mean, standard deviation, and rank for each function. EZOA demonstrated superior performance across all high-dimensional functions (F1–F13), securing the top rank consistently. On the fixed-dimensional functions (F14–F23), EZOA achieved global optima on nine out of ten functions, with slightly reduced performance on function F15, where both EBOwithCMAR and SPS_L_SHADE_EIG performed equally well. Further, the low standard deviation value highlights its stability in reaching higher quality solutions. The average friedman ranking of 1.57 reveals its superior performance overall. The overall average Friedman ranking of 1.57 further confirms EZOA's dominant performance in handling high-dimensional optimization challenges.

6.5 Sensitivity analysis of the parameters

The algorithms involves three parameters, population size *(n)*, maximum iterations *(T)*, and scale factor (*(s)*. To

Table 12 Simulation outcomes of the algorithms on classical benchmark functions with dimension 50

Function	Algorithm								
	Dim=50	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
F1	Avg	0	0	5.58E-193	1.58E-39	0	3.52E-212	1.18E-13	1.06E-171
	SD	0	0	0	5.29E-39	0	0	7.23E-15	0
	Best	0	0	4.02E-217	1.12E-43	0	4.22E-233	1.09E-13	7.78E-187
	Rank	2	2	5	7	2	4	8	6
F2	Avg	0	1.05E-257	6.83E-99	2.79E-24	2.84E-274	3.20E-106	5.79E+21	1.24E-108
	SD	0	0	3.70E-98	5.35E-24	0	1.28E-105	2.58E+22	6.21E-108
	Best	0	6.53E-267	1.76E-112	4.67E-26	0	2.06E-117	7.11E-26	1.55E-117
	Rank	1	3	6	7	2	5	8	4
F3	Avg	0	2.04E-291	1.07E-159	0.001112	0	3.59E-206	9.97E-14	80897.96
	SD	0	0	5.88E-159	3.51E-03	0	0	5.92E-15	24785.66
	Best	0	1.29E-319	3.37E-189	6.12E-12	0	8.64E-231	8.88E-14	2.36E+04
	Rank	1.5	3	5	7	1.5	4	6	8
F4	Avg	0	4.34E-223	3.27E-97	0.71781	1.50E-75	9.86E-104	7.95E-11	52.8645
	SD	0	0	1.41E-96	5.75E-01	8.20E-75	5.40E-103	4.59E-12	33.6502
	Best	0	4.94E-230	2.66E-108	5.14E-02	0	3.54E-116	6.72E-11	8.25E-03
	Rank	1	2	4	7	5	3	6	8
F5	Avg	45.3163	48.0721	0.003226	48.3118	48.8627	48.0264	48.8864	47.0372
	SD	0.10151	0.76896	0.004124	0.78883	0.16086	0.80947	0.025919	0.44283
	Best	45.1501	46.3828	1.54E-04	46.0918	48.1809	46.112	48.8155	46.5098
	Rank	2	5	1	6	7	4	8	3
F6	Avg	4.19E-02	4.4759	1.33E-05	6.3765	6.7787	5.0441	9.6629	0.04383
	SD	9.47E-02	0.87617	1.65E-05	0.90551	0.9001	0.65347	0.67524	1.82E-02
	Best	8.33E-05	2.58E+00	6.20E-08	4.6628	4.65E+00	3.40E+00	7.1909	2.30E-02
	Rank	2	4	1	6	7	5	8	3
F7	Avg	1.17E-04	3.63E-05	4.29E-05	0.006676	0.000184	7.12E-05	0.000817	0.000855
	SD	5.18E-05	2.13E-05	2.95E-05	0.003013	1.39E-04	3.73E-05	0.000303	0.000887
	Best	3.41E-05	6.09E-06	1.53E-06	0.002717	4.49E-06	1.79E-05	0.000389	3.77E-05
	Rank	4	1	2	8	5	3	6	7
F8	Avg	0	1.62E-262	5.63E-06	53.1728	1.43E-264	3.93E-109	2.41E-16	1.27E-110
	SD	0	0	2.52E-05	9.0007	0	1.55E-108	3.24E-16	4.37E-110
	Best	0	1.52E-267	6.46E-113	40.0071	0	1.47E-117	1.99E-17	6.99E-121
	Rank	1	3	7	8	2	5	6	4
F9	Avg	0	0	0	315.1338	0	0	11.4861	0
	SD	0	0	0	45.3128	0	0	62.9122	0
	Best	0	0	0	205.8753	0	0	0	0
	Rank	3.5	3.5	3.5	8	3.5	3.5	7	3.5
F10	Avg	4.44E-16	4.44E-16	4.44E-16	1.066	1.04E-15	2.93E-15	6.48E-11	3.40E-15
	SD	0	0	0	1.5401	1.35E-15	1.66E-15	1.07E-11	2.65E-15
	Best	4.44E-16	4.44E-16	4.44E-16	1.47E-14	4.44E-16	4.44E-16	3.53E-11	4.44E-16
	Rank	2	2	2	8	4	5	7	6
F11	Avg	0	0	0	0.007052	0	0	1.69E-14	0.003304
	SD	0	0	0	0.006863	0	0	1.68E-14	1.26E-02
	Best	0	0	0	0	0	0	0	0
	Rank	3	3	3	8	3	3	6	7
F12	Avg	8.48E-11	6.30E-10	0	3.88E-09	1.76E-09	3.20E-10	3.08E-09	7.08E-13
	SD	1.55E-10	1.05E-09	0	4.50E-09	3.15E-09	5.21E-10	1.02E-08	2.82E-12
	Best	3.30E-14	4.72E-14	0	2.85E-13	0	3.50E-14	8.47E-12	0

Table 12 (continued)

Function	Algorithm								
	Dim=50	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
	Rank	3	5	1	8	6	4	7	2
F13	Avg	0	0	0	0.001876	0	0	0.002189	0.000104
	SD	0	0	0	0.001558	0	0	0.001457	0.000571
	Best	0	0	0	0	0	0	0	0
	Rank	3	3	3	7	3	3	8	6
Friedman	Average	2.23	3.03	3.34	7.15	3.92	3.96	6.84	5.19
Rank	Overall	1	2	3	8	5	4	7	6

examine the impact of these parameters, a sensitivity analysis is conducted across various test functions. This analysis explores how changes in individual parameter values influence performance, with other parameters held constant to isolate the effect of each one. Results, presented in Table 19, include average fitness values from 30 runs across two unimodal functions (F4 and F5), two varying-dimensional multimodal functions (F8 and F12), and one fixed-dimensional multimodal function (F21). While tuning the parameter n, the parametrs T and s are fixed at 1000 and 2 respectively. As the value of n increases from 20 to 80, the performnce improves but the difference in improvement from 50 to 80 in marginal. Consequently, n is set at 50 as an optimal balance between performance and computational cost. In case of tuning the parameter T, n is fixed at 50 and s at 2. It can be observed from the table that T has notable impact on performance, with significant improvements observed up to a value of 1000. The combined effect of parameter n and T on fitness value can be illustrated from 3D plot in Fig. 10.

While tuning the parameter s, n and T are set at 50 and 1000, respectively. In this case, the value of s is varied from 1 to 4. The fitness values has improved substantially as value of s is incresed fron 1 to 2 but not much improvement is observed after s=2. The variation is negligble. Consequently, the value of s is selected to be 2.

6.6 Statistical analysis

To validate the performance of the proposed EZOA algorithm on benchmark functions, three non-parametric statistical tests are conducted: Friedman's test, Wilcoxon signed-rank test, and Mann–Whitney U test. All the tests are performed at a significance level of 0.05. The statistical analysis is based on the average of the best values achieved across 30 independent runs for each function.

Friedman's test is employed to rank the performance of various algorithms and determine the statistical significance of the differences in these rankings. Lower rank of an algorithm indicates its better performance. When ties occur in algorithm's rankings, each algorithm is assigned the average of the ranks they would have occupied. The results of Friedman's test for the algorithms are shown along with their average, best, and standard deviation values in the Tables 8, 10, 11, 12, 13, 14, 16, and 18. The analysis reveal that the proposed EZOA algorithm achieved best performance among the compared algorithms, with the lowest mean ranks on classical functions acoss various dimensions and 1.15 on CEC-2019 functions, resulting in an overall rank of 1 all the cases. The radar plot in Fig. 11 displays the overall ranking of algorithms on both set of functions. The radar plot clearly shows that EZOA achieves the top rank in each case.

Next, Wilcoxon signed-rank test is conducted to compare the performance of proposed EZOA with an individual competitor algorithms. The test involves two hypotheses: the null hypothesis (H_0) and the alternative hypothesis (H_1). If the p-value is greater than 0.05, the null hypothesis is accepted, indicating no significant difference between the proposed algorithm and the competitor algorithms. Conversely, a p-value less than 0.05 led to the rejection of the null hypothesis, suggesting a significant performance difference. For all the algorithms, the p-values for the classical benchmark functions are statistically significant and highlighted in bold in Table 20. For the CEC-2019 functions, the p-values corresponding to each function are provided in Tables 21. These results strongly confirm the statistical superiority of the improved performance of the proposed EZOA over ZOA and other competitive algorithms. Hence, both the tests favor the

Table 13 Simulation outcomes of the algorithms on classical benchmark functions with dimension 100

Function	Algorithm								
	Dim=100	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
F1	Avg	0	0	8.74E-194	1.25E-27	0	2.57E-206	1.29E-13	4.44E-168
	SD	0	0	0	2.75E-27	0	0	6.36E-15	0
	Best	0	0	2.90E-220	8.62E-30	0	2.13E-226	1.18E-13	9.64E-182
	Rank	2	2	5	7	2	4	8	6
F2	Avg	0	5.36E-253	5.29E-98	9.68E-18	2.71E-278	6.76E-105	1.30E+49	1.67E-106
	SD	0	0	2.19E-97	8.80E-18	0	3.70E-104	6.33E+49	8.58E-106
	Best	0	7.47E-261	4.15E-108	4.58E-19	0	3.09E-115	1.31E+42	2.89E-117
	Rank	1	3	6	7	2	5	8	4
F3	Avg	0	3.02E-276	1.70E-135	698.6125	0	5.09E-207	1.02E-13	687282.1
	SD	0	0	9.32E-135	6.90E+02	0	0	6.43E-15	118886
	Best	0	3.67E-302	6.22E-187	6.58E+01	0	7.63E-231	9.13E-14	4.30E+05
	Rank	1.5	3	5	7	1.5	4	6	8
F4	Avg	0	9.74E-216	2.63E-96	28.8235	4.13E-11	4.09E-105	8.32E-11	73.8572
	SD	0	0	1.43E-95	1.07E+01	1.92E-10	1.76E-104	6.83E-12	26.7949
	Best	0	4.56E-223	1.52E-107	6.08E+00	0	9.56E-116	7.26E-11	1.28E-01
	Rank	1	2	4	7	5	3	6	8
F5	Avg	9.56E+01	9.83E+01	0.004674	9.83E+01	9.89E+01	9.83E+01	9.89E+01	9.72E+01
	SD	5.37E-01	3.70E-01	0.00739	5.81E-01	1.11E-01	4.28E-01	2.47E-02	5.00E-01
	Best	9.52E+01	9.76E+01	2.67E-05	9.62E+01	98.7562	9.69E+01	9.88E+01	96.3874
	Rank	2	5	1	4	7	3	6	8
F6	Avg	1.6414	13.6648	2.51E-05	13.4628	16.8212	13.0259	22.0202	0.50616
	SD	0.42755	1.4318	3.78E-05	1.2481	1.67	1.0223	0.89373	0.14469
	Best	1.0034	10.7382	5.59E-07	10.6674	13.5616	10.8818	19.9909	0.19838
	Rank	3	6	1	5	7	4	8	2
F7	Avg	1.20E-04	3.31E-05	5.15E-05	0.013899	0.000104	7.45E-05	0.000946	0.001146
	SD	7.18E-05	1.93E-05	5.28E-05	0.005848	8.77E-05	4.57E-05	0.000396	0.001116
	Best	2.47E-05	9.27E-06	1.52E-06	0.005279	2.14E-06	9.81E-06	0.000366	3.92E-05
	Rank	5	1	2	8	4	3	6	7
F8	Avg	0	6.92E-255	4.79E-103	147.933	3.94E-258	2.80E-103	2.48E-14	1.92E-108
	SD	0	0	1.45E-102	14.9349	0	1.18E-102	3.38E-14	6.70E-108
	Best	0	5.93E-262	4.19E-112	117.2983	0	2.65E-116	1.82E-16	1.81E-116
	Rank	1	3	6	8	2	5	7	4
F9	Avg	0	0	0	912.9582	0	0	0	0
	SD	0	0	0	108.1555	0	0	0	0
	Best	0	0	0	688.5266	0	0	0	0
	Rank	4	4	4	8	4	4	4	4
F10	Avg	4.44E-16	4.44E-16	4.44E-16	0.092348	6.71E-01	3.29E-15	7.57E-11	3.05E-15
	SD	0	0	0	0.50581	3.68E+00	1.45E-15	4.87E-12	2.07E-15
	Best	4.44E-16	4.44E-16	4.44E-16	3.24E-14	4.44E-16	4.44E-16	6.49E-11	4.44E-16
	Rank	2	2	2	7	8	5	6	4
F11	Avg	0	0	0	0.002028	0	0	5.73E-14	0.00354
	SD	0	0	0	0.00541	0	0	3.32E-14	1.94E-02
	Best	0	0	0	0	0	0	3.11E-15	0
	Rank	3	3	3	7	3	3	6	8
F12	Avg	9.82E-11	1.47E-09	0	3.71E-09	4.51E-09	6.05E-10	1.88E-09	5.07E-13
	SD	1.84E-10	3.11E-09	0	7.08E-09	1.15E-08	7.13E-10	2.55E-09	1.22E-12
	Best	2.99E-13	1.45E-12	0	3.44E-15	0	3.70E-14	3.77E-15	0

Table 13 (continued)

Function	Algorithm								
	Dim=100	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
	Rank	3	5	1	7	8	4	6	2
F13	Avg	0	0	0	0.001563	0	0	0.002398	0
	SD	0	0	0	0.00159	0	0	0.001345	0
	Best	0	0	0	0	0	0	0	0
	Rank	3.5	3.5	3.5	7	3.5	3.5	8	3.5
Friedman	Average	2.38	3.04	3.58	7.07	4.23	3.88	6.54	5.27
Rank	Overall	1	2	3	8	5	4	7	6

improved performance of the proposed EZOA over other competitive algorithms underscoring its effectiveness and robustness.

The Mann–Whitney *U* test is applied to the average values of classical benchmark functions to determine if there are statistically significant differences in the performance distributions between pairs of algorithms. The p-values of the test results are tabulated in Table 22. The value lower than 0.05 implies that there is statistically significant difference between the results. Also, the non-significant values are underlined in the table.

6.7 Convergence analysis

To assess the convergence behavior of the algorithms, both set of tes functions are utilized. The convergence curves are calculated by averaging the best solutions achieved across 30 independent runs for each algorithm. These curves are illustrated in Figs. 12, 13, 14, 15, 16, 17, with the EZOA indicated by a red line. Figures 12, 13, 14, 15 correspond to the performance on 13 multi-dimensional functions (F1–F13), evaluated at dimensions of 10, 30, 50, and 100. Each of these figures captures the algorithm's ability to navigate through complex landscapes as the dimensionality increases. Figure 16 presents the convergence trends for fixed-dimensional benchmark functions (F14–F23). Generally, the algorithms begin by exploring the search space extensively, characterized by significant positional changes of the search agents. As the iterations progress, these abrupt shifts diminish, allowing the algorithms to concentrate on the position of the most promising search agent, facilitating effective exploitation of the search space. Consequently, the fitness values are high during the early iterations but tend to decline gradually with iterations. The curves clearly highlights the exceptional convergence trend by EZOA on the multi-dimensional functions. Whereas, RSO, BOA, and WOA shows poor convergence trends overall. The convergence behavior by algorithms remains consistent across all dimensions. On fixed-dimensional functions, EZOA shows a strong performance, converging more rapidly toward the optimum. As shown in Fig. 17, EZOA's convergence curve for the CEC-19 benchmark functions demonstrates a sharp approach toward the solution, underscoring its effective convergence toward the optimal solution.

6.8 Boxplot analysis

To examine the distributional characteristics of the performance of the proposed EZOA on two sets of benchmark optimization problems, a boxplot analysis is conducted. Figures 18, 19, 20, 21, 22 illustrate the boxplots for specific functions, providing a view of the spread and central tendency of results across 30 independent trials. Each boxplot represents data distribution through quartiles: the whiskers denote the minimum and maximum values, while the ends of the rectangles mark the upper and lower quartiles. Figures 18, 19, 20, 21 present the boxplots for functions F5, F6, F7, F9, F10, and F12 across dimensions 10, 20, 50, and 100. The boxplots for functions F1–F4, F8, F11, and F13 are excluded from these figures as their statistical profiles under EZOA are identical to those of function F9. Since EZOA achieves the global optimal value of zero on these functions, all the key metrics mean, minimum, maximum, and standard deviation are also zero, depicting EZOA's stability in reaching optimal solutions. Figure 23 presents the boxplot for six CEC-19 test functions, namely C-01, C-02, C-05, C-07, C-08, and C-09. It can be observed from the figures that the width of EZOA's boxplot is narrower on most functions, reflecting strong consistency among results (Fig. 23).

Table 14 Simulation outcomes of the algorithms on classical benchmark functions with fixed dimensions

Function	Algorithm								
	Metric	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
F14	Avg	0.998	1.8554	0.998	6.4961	2.3156	1.0311	0.99828	1.8146
	SD	0	1.4822	8.69E-11	5.068	1.8785	0.18148	0.001008	2.4748
	Best	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
	Rank	1.5	6	1.5	8	7	4	3	5
F15	Avg	0.000342	0.001695	0.000321	0.005706	0.000832	0.000307	0.000329	0.000613
	SD	4.88E-05	0.005158	1.44E-05	0.01226	0.00031	9.38E-11	1.79E-05	0.000315
	Best	0.000307	0.000307	0.000308	0.000308	0.000423	0.000307	0.000311	0.000313
	Rank	4	7	2	8	6	1	3	5
F16	Avg	− 1.0316	− 1.0316	− 1.0316	− 1.0306	− 1.0316	− 1.0316	NaN	− 1.0316
	SD	6.78E-16	7.69E-11	3.34E-13	0.005775	4.43E-05	6.05E-16	NaN	5.67E-12
	Best	− 1.0316	− 1.0316	− 1.0316	− 1.0316	− 1.0316	− 1.0316	NaN	− 1.0316
	Rank	4	4	4	4	4	4	–	4
F17	Avg	0.39789	0.39789	0.39789	0.39789	0.40301	0.39789	0.39832	0.39789
	SD	0	1.18E-09	1.81E-08	6.43E-06	0.007007	0	0.001278	3.34E-07
	Best	0.39789	0.39789	0.39789	0.39789	0.39795	0.39789	0.39789	0.39789
	Rank	3.5	3.5	3.5	3.5	8	3.5	7	3.5
F18	Avg	3	3	3	13.8	3	3	3.0057	3
	SD	9.00E-16	2.00E-07	7.54E-10	25.169	5.73E-06	2.04E-15	0.007072	2.82E-06
	Best	3	3	3	3	3	3	3	3
	Rank	3.5	3.5	3.5	8	3.5	3.5	7	3.5
F19	Avg	− 3.8628	− 3.8628	− 3.8624	− 3.8625	− 3.6424	− 3.8628	NaN	− 3.8622
	SD	2.71E-15	2.65E-05	0.000917	0.001437	0.18162	2.53E-15	NaN	0.000993
	Best	− 3.8628	− 3.8628	− 3.8628	− 3.8628	− 3.8348	− 3.8628	NaN	− 3.8628
	Rank	2	2	5.5	4	5.5	2	–	7
F20	Avg	− 3.322	− 3.3151	− 3.1811	− 3.2364	− 1.9686	− 3.322	NaN	− 3.2564
	SD	1.37E-15	0.026308	0.084073	0.063537	0.57906	1.95E-06	NaN	0.067501
	Best	− 3.322	− 3.322	− 3.3129	− 3.3214	− 2.8635	− 3.322	NaN	− 3.322
	Rank	1.5	3	6	5	7	1.5	–	4
F21	Avg	− 10.1532	− 9.9832	− 5.5583	− 6.874	− 0.91709	− 9.9833	− 5.5161	− 9.6418
	SD	6.85E-15	0.93075	1.5359	3.3127	0.65717	0.93076	1.0174	1.5553
	Best	− 10.1532	− 10.1532	− 10.1531	− 10.1028	− 3.366	− 10.1532	− 8.5714	− 10.1532
	Rank	1	3	6	5	8	2	7	4
F22	Avg	− 10.4029	− 9.8714	− 5.2646	− 7.289	− 1.0693	− 10.2258	− 5.9417	− 9.3174
	SD	1.14E-15	1.6218	0.97037	3.5651	0.67437	0.97043	1.2566	2.517
	Best	− 10.4029	− 10.4029	− 10.4024	− 10.3615	− 3.9958	− 10.4029	− 8.4928	− 10.4029
	Rank	1	3	7	5	8	2	6	4
F23	Avg	− 10.5364	− 9.8153	− 5.4882	− 8.1841	− 1.4908	− 10.3561	− 5.8815	− 8.7333
	SD	1.95E-15	1.8698	1.3698	3.2919	0.79693	0.98735	1.022	2.8695
	Best	− 10.5364	− 10.5364	− 10.5321	− 10.517	− 4.1247	− 10.5364	− 8.4704	− 10.5364
	Rank	1	3	7	5	8	2	6	4
Friedman	Average	2.3	3.8	4.6	5.55	6.5	2.55	5.57	4.4
Rank	Overall	1	3	5	6	8	2	7	4

Table 15 Run time of algorithms on classical benchmark functions (in seconds)

Function (↓)	Algorithm(→) EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
F1	0.1622	0.0392	0.0504	0.0886	0.0334	0.0341	0.0556	0.0300
F2	0.1773	0.0569	0.0644	0.0954	0.0386	0.0449	0.0632	0.034
F3	0.6045	0.2677	0.3284	0.2022	0.1455	0.2516	0.2767	0.1402
F4	0.1730	0.0469	0.0797	0.0935	0.03705	0.0409	0.0604	0.0321
F5	0.2068	0.0656	0.1177	0.0988	0.0450	0.0569	0.0749	0.0408
F6	0.1703	0.0495	0.0909	0.0916	0.0356	0.0378	0.0554	0.0315
F7	0.3985	0.1632	0.2081	0.1509	0.0932	0.1500	0.1701	0.0899
F8	0.1825	0.0534	0.0716	0.1054	0.0437	0.0451	0.0841	0.0352
F9	0.1882	0.0629	0.1123	0.1127	0.0460	0.0520	0.0899	0.0370
F10	0.1924	0.0678	0.1169	0.1063	0.0458	0.0504	0.0797	0.0391
F11	0.2270	0.0761	0.1324	0.1092	0.0577	0.0729	0.0939	0.0482
F12	0.1782	0.0550	0.1040	0.0957	0.0390	0.0435	0.0622	0.0351
F13	0.1859	0.0464	0.0867	0.0894	0.0329	0.0334	0.0508	0.0282
F14	1.4516	0.7131	0.9142	0.3626	0.3593	0.7116	0.7481	0.3629
F15	0.1000	0.0357	0.0627	0.0284	0.0209	0.0338	0.0567	0.0232
F16	0.0905	0.0352	0.0598	0.0226	0.0194	0.0311	0.1027	0.0229
F17	0.0696	0.0245	0.0462	0.0171	0.0132	0.0217	0.0447	0.0167
F18	0.0656	0.0218	0.0441	0.0165	0.0127	0.0203	0.0428	0.0156
F17	0.1218	0.0472	0.0775	0.0320	0.0267	0.0438	0.1871	0.0293
F20	0.1315	0.0516	0.0816	0.0400	0.0296	0.0467	0.2010	0.0310
F21	0.1490	0.0575	0.0949	0.0411	0.0334	0.0601	0.1836	0.0357
F22	0.1727	0.0742	0.1115	0.0467	0.0377	0.0693	0.2156	0.0405
F23	0.2082	0.0920	0.1360	0.0563	0.0472	0.0877	0.2732	0.0498

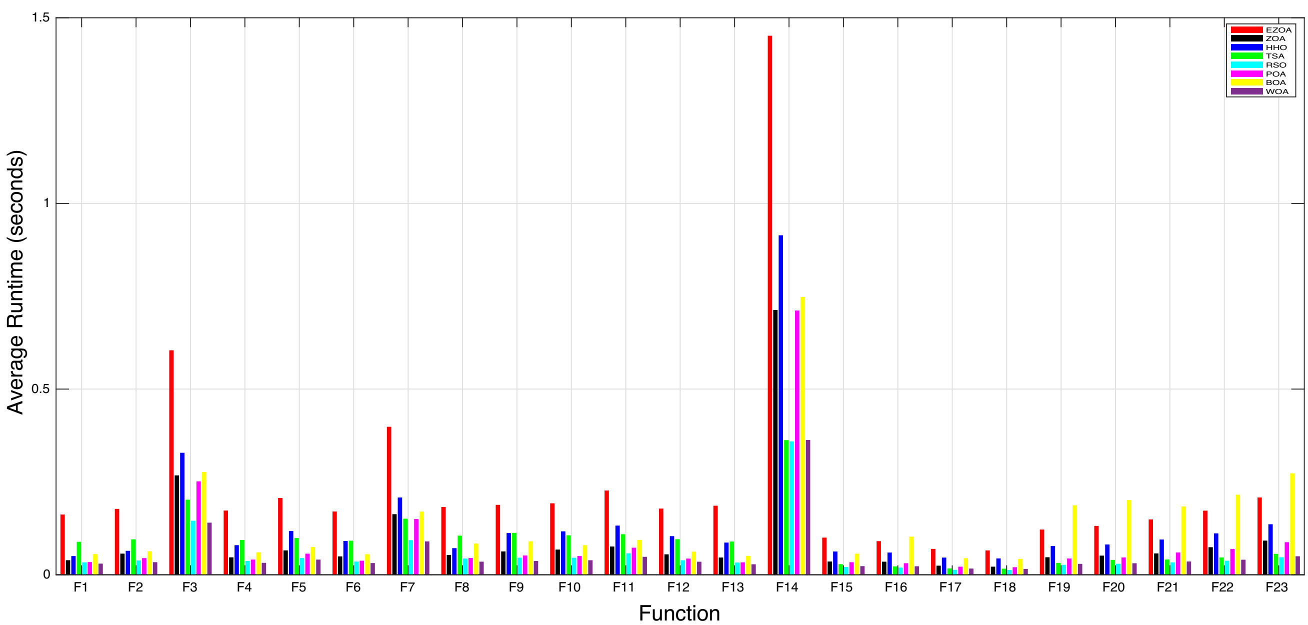

Fig. 8 Average run time by algorithms for classical benchmark functions

Table 16 Simulation outcomes of algorithms on CEC-2019 functions

Function	Algorithm								
	Metric	EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
C-01	Avg	41087.32	46941.48	4.98E+04	1.63E+08	5.56E+04	3.88E+04	4.41E+04	1.14E+10
	SD	8.72E+03	1.99E+04	3.63E+03	4.74E+08	7.05E+03	1.16E+03	2.56E+03	2.16E+10
	Best	3.95E+04	37498.81	4.47E+04	4.14E+04	4.49E+04	36831.6	4.00E+04	8.55E+05
	Rank	2	4	5	7	6	1	3	8
C-02	Avg	1.73E+01	1.74E+01	1.74E+01	1.85E+01	1.75E+01	1.73E+01	1.77E+01	1.73E+01
	SD	0	1.35E-01	6.67E-03	6.57E-01	2.69E-01	4.51E-05	1.65E-01	4.96E-03
	Best	1.73E+01	17.34289	1.73E+01	1.73E+01	1.73E+01	17.34286	1.74E+01	1.73E+01
	Rank	1	5	4	8	6	2	7	3
C-03	Avg	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01
	SD	5.47E-15	1.46E-05	3.87E-06	9.54E-04	4.82E-04	2.42E-09	3.23E-04	1.83E-08
	Best	1.27E+01	12.7024	1.27E+01	1.27E+01	1.27E+01	12.7024	1.27E+01	1.27E+01
	Rank	1.5	5	4	8	6	1.5	7	3
C-04	Avg	1.89E+01	1.36E+03	1.39E+02	3.85E+03	8.94E+03	7.09E+02	1.28E+04	2.45E+02
	SD	6.30E+00	1.64E+03	5.75E+01	2.39E+04	2.13E+03	9.57E+02	5.93E+03	9.45E+01
	Best	7.55E+00	156.9345	5.15E+01	9.58E+01	4.72E+03	27.57688	3.96E+03	1.24E+02
	Rank	1	5	2	6	7	4	8	3
C-05	Avg	1.07E+00	1.78E+00	2.60E+00	2.89E+00	3.43E+00	1.38E+00	4.39E+00	1.74E+00
	SD	2.64E-02	3.63E-01	7.31E-01	8.66E-01	4.49E-01	2.73E-01	5.11E-01	4.04E-01
	Best	1.02E+00	1.233292	1.43E+00	1.30E+00	2.71E+00	1.089516	3.20E+00	1.24E+00
	Rank	1	4	5	6	7	2	8	3
C-06	Avg	6.58E+00	7.20E+00	8.93E+00	1.05E+01	9.98E+00	7.41E+00	1.04E+01	8.98E+00
	SD	9.22E-01	6.96E-01	1.23E+00	8.98E-01	1.31E+00	9.41E-01	7.02E-01	9.40E-01
	Best	5.24E+00	5.675232	5.55E+00	8.08E+00	6.86E+00	5.185391	8.946172	7.379518
	Rank	1	2	4	8	6	3	7	5
C07	Avg	3.65E+01	1.05E+02	3.49E+02	5.62E+02	6.22E+02	1.46E+02	7.75E+02	4.57E+02
	SD	7.66E+01	7.68E+01	1.57E+02	2.55E+02	2.19E+02	1.33E+02	1.75E+02	2.19E+02
	Best	− 68.9876	− 91.186	115.6989	238.5646	171.9202	− 154.59	271.9355	157.6794
	Rank	1	2	4	6	7	3	8	5
C-08	Avg	3.84E+00	4.49E+00	5.67E+00	6.06E+00	6.20E+00	4.25E+00	6.16E+00	5.60E+00
	SD	6.22E-01	3.56E-01	3.29E-01	5.49E-01	4.04E-01	4.27E-01	2.90E-01	5.17E-01
	Best	2.428655	3.74373	4.977138	5.082087	5.423185	3.65868	5.641711	4.280811
	Rank	1	3	5	6	8	2	7	4
C-09	Avg	2.39E+00	6.25E+01	2.97E+00	6.52E+02	5.71E+02	5.04E+00	1.64E+03	4.37E+00
	SD	2.48E-02	1.89E+02	3.26E-01	9.71E+02	1.47E+02	1.07E+00	5.22E+02	6.92E-01
	Best	2.355854	2.601298	2.540137	3.182125	300.8434	2.76299	454.1116	2.856168
	Rank	1	5	2	7	6	4	8	3
C-10	Avg	1.19E+01	1.98E+01	2.01E+01	2.04E+01	2.04E+01	1.82E+01	2.05E+01	2.00E+01
	SD	1.00E+01	1.05E+00	1.19E-01	9.84E-02	1.42E-01	4.57E+00	5.31E-02	7.52E-01
	Best	1.73E-09	15.40484	19.99488	20.25508	20.13291	6.170097	20.35868	16.88247
	Rank	1	3	5	7	6	2	8	4
Friedman	Average	1.15	3.8	4	6.9	6.5	2.45	7.10	4.10
Rank	Overall	1	3	4	7	6	2	8	5

Table 17 Run time of algorithms on CEC-19 functions (in seconds)

Function (↓)	Algorithm(→) EZOA	ZOA	HHO	TSA	RSO	POA	BOA	WOA
C-01	8.5005	4.2787	5.0532	2.1391	2.1174	4.2093	4.2284	2.1522
C-02	0.0718	0.0183	0.0409	0.0328	0.0154	0.018	0.0232	0.0145
C-03	0.1687	0.0642	0.0992	0.0591	0.0389	0.0631	0.0695	0.038
C-04	0.0933	0.0336	0.0551	0.0331	0.0216	0.0329	0.0401	0.0219
C-05	0.0931	0.0330	0.0548	0.0331	0.0222	0.0317	0.0393	0.0211
C-06	4.1910	2.0821	2.5701	1.0651	1.0535	2.1007	2.1016	1.0544
C-07	0.0968	0.0354	0.0574	0.0338	0.0221	0.0333	0.0415	0.0216
C-08	0.1016	0.0371	0.0594	0.0342	0.0228	0.0366	0.0428	0.0222
C-09	0.0739	0.0233	0.0452	0.0273	0.0162	0.0233	0.0300	0.0165
C-10	0.1031	0.0382	0.0637	0.0350	0.0235	0.0385	0.0443	0.0238

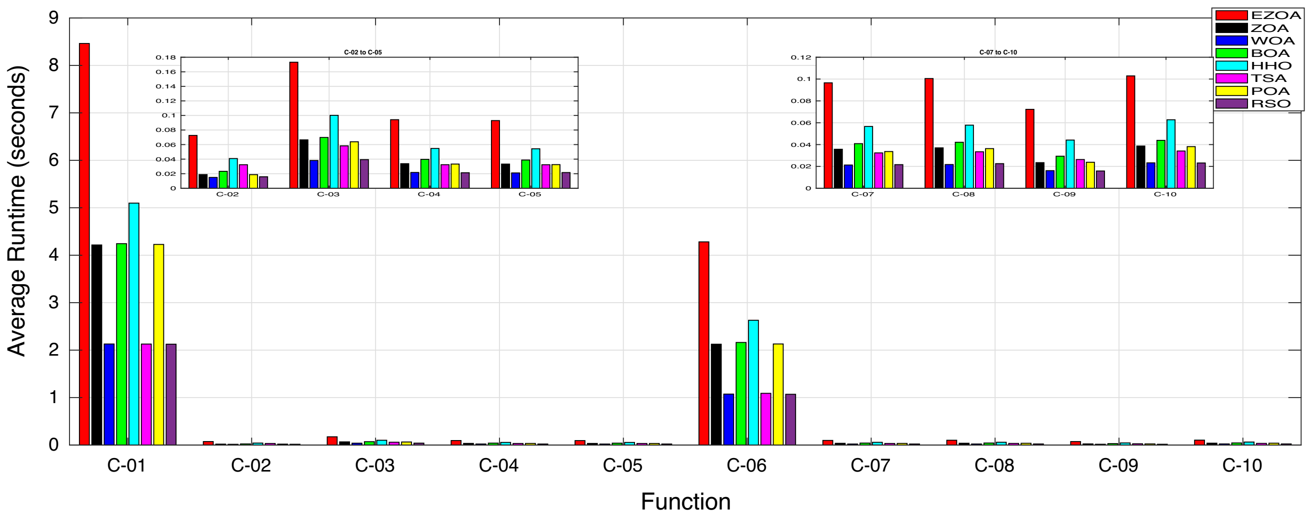

Fig. 9 Average run time for CEC-19 benchmark functions

6.9 Time complexity analysis

To design optimization algorithms that are both time-efficient and computationally feasible, thorough analysis of their time complexity is crucial. To evaluate this, the runtime complexity of the proposed EZOA algorithm, along with competitor algorithms on CEC-2019 benchmark functions, is assessed. The runtime cost is determined by first computing the time T_0, which is computed as follows:

$$\begin{aligned} &\text{for } i = 1 : r \\ &\quad v = 0.55; \\ &\quad v = v + v; \\ &\text{end} \end{aligned} \tag{31}$$

where r is 1, 00, 000. T_1 represents the computational time for function C-01 of CEC-2019 test suit for one run. T_2 represents the average computational time obtained from five independent runs of the same function. The maximum number of iterations is fixed at 500, and the algorithm parameters remain consistent throughout. The results of computational time complexity is provided in Table 23. The results indicate that the EZOA has a slight increase in computational time, yet this increase is relatively minor when considered against the significant improvements in solution quality and convergence behavior.

7 Simulation results on test case 2: RRAP problems

The simulation results of four RRAP problems (RRAP1-RRAP4) are discussed in this section. The performance of EZOA is evaluated and compared with the already existing algorithms in the literature in terms of the best solution obtained and the value of maximum possible improvement (*Imp*). The comparison results are tabulated in Tables 24,

Table 18 Comparison of proposed algorithm with some champion algorithms

Function	Algorithm					
	Metric	EZOA	EBO with CMAR	LSHADE	CMA-ES	SPS_L_SHADE_EIG
F1	Avg	2.44E-166	7.40E+04	583795.1	4.13E+04	5.41E+05
	SD	0	4.01E+03	571550.9	2.84E+03	2.91E+04
	Rank	1	3	5	2	4
F2	Avg	2.78E-84	4.60E+02	4.10E+193	1.25E+292	3.11E+150
	SD	8.48E-84	1.39E+01	61.90283	-	6.94E+150
	Rank	1	2	4	5	3
F3	Avg	1.14E-138	6.32E+06	1.34E+07	1.84E+07	7.39E+06
	SD	6.23E-138	2.69E+06	4.82E+06	4.37E+06	7.84E+05
	Rank	1	2	4	5	3
F4	Avg	5.45E-84	3.78E+01	9.55E+01	9.98E+01	9.75E+01
	SD	9.01E-84	4.56E+00	7.91E+00	2.26E-01	1.25E-01
	Rank	1	2	3	5	4
F5	Avg	498.2343	2.49E+07	1.96E+09	2.25E+07	1.34E+09
	SD	0.292706	3.08E+06	2.65E+09	3.80E+06	1.35E+08
	Rank	1	3	5	2	4
F6	Avg	1.09E+02	7.50E+04	432334.9	4.13E+04	5.61E+05
	SD	1.43E+00	4.14E+03	503413.2	3.03E+03	3.75E+04
	Rank	1	3	4	2	5
F7	Avg	0.000632	3.10E+03	20515.17	3.46E+02	1.01E+04
	SD	2.92E-04	3.03E+03	24469.78	5.25E+01	1.16E+03
	Rank	1	3	5	2	4
F8	Avg	7.15E-10	3.30E+02	38269.1	6.00E+02	9.12E+02
	SD	1.13E-09	1.38E+01	17912.23	3.10E+01	4.09E+01
	Rank	1	2	5	3	4
F9	Avg	0	4.74E+03	4776.259	5.76E+03	6.40E+03
	SD	0	7.08E+01	2277.889	7.20E+01	2.01E+02
	Rank	1	2	3	4	5
F10	Avg	4.44E-16	1.24E+01	15.7537	2.12E+01	1.94E+01
	SD	0	2.10E-01	3.678597	3.59E-02	6.81E-02
	Rank	1	2	3	5	4
F11	Avg	0	1.28E+04	4042.981	3.71E+02	4.87E+03
	SD	0	1.51E+02	3823.95	2.72E+01	4.38E+02
	Rank	1	5	3	2	4
F12	Avg	4.66E-09	7.44E-09	4.35E-09	1.31E-08	1.45E-09
	SD	9.55E-09	1.39E-08	7.94E-09	3.03E-08	2.66E-09
	Rank	3	4	2	5	1
F13	Avg	0.00E+00	3.44E-01	4.11E-12	4.99E-01	4.95E-01
	SD	0.00E+00	1.38E+01	2.25E-11	3.12E-04	6.01E-04
	Rank	1	3	2	5	4
F14	Avg	9.98E-01	9.98E-01	9.98E-01	5.07E+00	9.98E-01
	SD	1.40E-16	1.70E-16	0	3.66E+00	1.24E-16
	Rank	2.5	2.5	2.5	5	2.5
F15	Avg	4.34E-04	3.08E-04	8.14E-04	1.70E-03	3.08E-04
	SD	1.13E-04	1.34E-16	0.0003	8.11E-04	3.63E-19
	Rank	3	1.5	4	5	1.5
F16	Avg	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 1.03E+00	− 3.43E-01
	SD	1.97E-16	4.97E-16	2.28E-16	6.78E-16	4.00E-01

Table 18 (continued)

Function	Algorithm					
	Metric	EZOA	EBO with CMAR	LSHADE	CMA-ES	SPS_L_SHADE_EIG
	Rank	2	2	2	4	5
F17	Avg	0.39789	3.98E-01	0.39789	3.98E-01	3.98E-01
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Rank	2	4.5	2	4.5	2
F18	Avg	3	3	3	3	3
	SD	1.46E-15	1.86E-15	6.68E-16	1.33E-15	1.22E-15
	Rank	3	3	3	3	3
F19	Avg	− 3.86E+00	− 3.86E+00	− 3.86E+0	− 3.86E+00	− 3.59E+00
	SD	2.26E-15	2.58E-15	2.28E-15	2.71E-15	2.60E-01
	Rank	2	2	2	4	5
F20	Avg	− 3.32E+00	− 3.32E+00	− 3.30E+00	− 3.25E+00	− 2.08E+00
	SD	6.12E-08	1.36E-15	0.043556	5.99E-02	3.04E-01
	Rank	1.5	1.5	3	4	5
F21	Avg	− 1.02E+01	− 1.02E+01	− 9.9E+00	− 8.96E+00	− 9.49E-01
	SD	2.23E-08	5.69E-15	1.129757	2.73E+00	6.14E-01
	Rank	1.5	1.5	3	4	5
F22	Avg	− 1.04E+01	− 1.04E+01	− 1.04E+01	− 9.64E+00	− 7.73E-01
	SD	1.23E-08	5.71E-16	1.10E-05	2.33E+00	3.34E-01
	Rank	2	2	2	4	5
F23	Avg	− 1.05E+01	− 1.05E+01	− 1.05E+01	− 9.89E+00	− 1.05E+00
	SD	1.11E-06	1.90E-15	1.82E-15	1.99E+00	4.49E-01
	Rank	1.5	3	1.5	4	5
Friedman	Average	1.57	2.59	3.17	3.85	3.83
Rank	Overall	1	2	3	4	5

Table 19 Sensitivity analysis of control paramters

Parameter	Functions				
	F4	F5	F8	F12	F21
Population size (n)					
20	0.00E+00	25.3671	0.00E+00	5.71E-08	− 9.9838
30	0.00E+00	25.2733	0.00E+00	1.29E-08	− 10.1532
50	0.00E+00	25.1933	0.00E+00	7.79E-10	− 10.1532
80	0.00E+00	25.1162	0.00E+00	3.52E-10	− 10.1532
Maximum iterations (T)					
200	1.3571E-66	2.73E+01	1.57E-69	8.55E-03	− 10.1532
500	9.177E-172	2.64E+01	2.01E-175	1.47E-05	− 10.1532
1000	0.00E+00	2.19E+01	0.00E+00	5.82E-11	− 10.1532
1500	0.00E+00	2.16E+01	0.00E+00	1.38E-12	− 10.1532
Scale factor (s)					
1	2.81E-45	8.89E-55	25.3527	8.95E-10	− 10.1532
2	0.00E+00	0.00E+00	25.1956	6.55E-11	− 10.1532
3	0.00E+00	0.00E+00	25.1887	4.28E-11	− 10.1532
4	0.00E+00	0.00E+00	25.2299	4.15E-11	− 10.1532

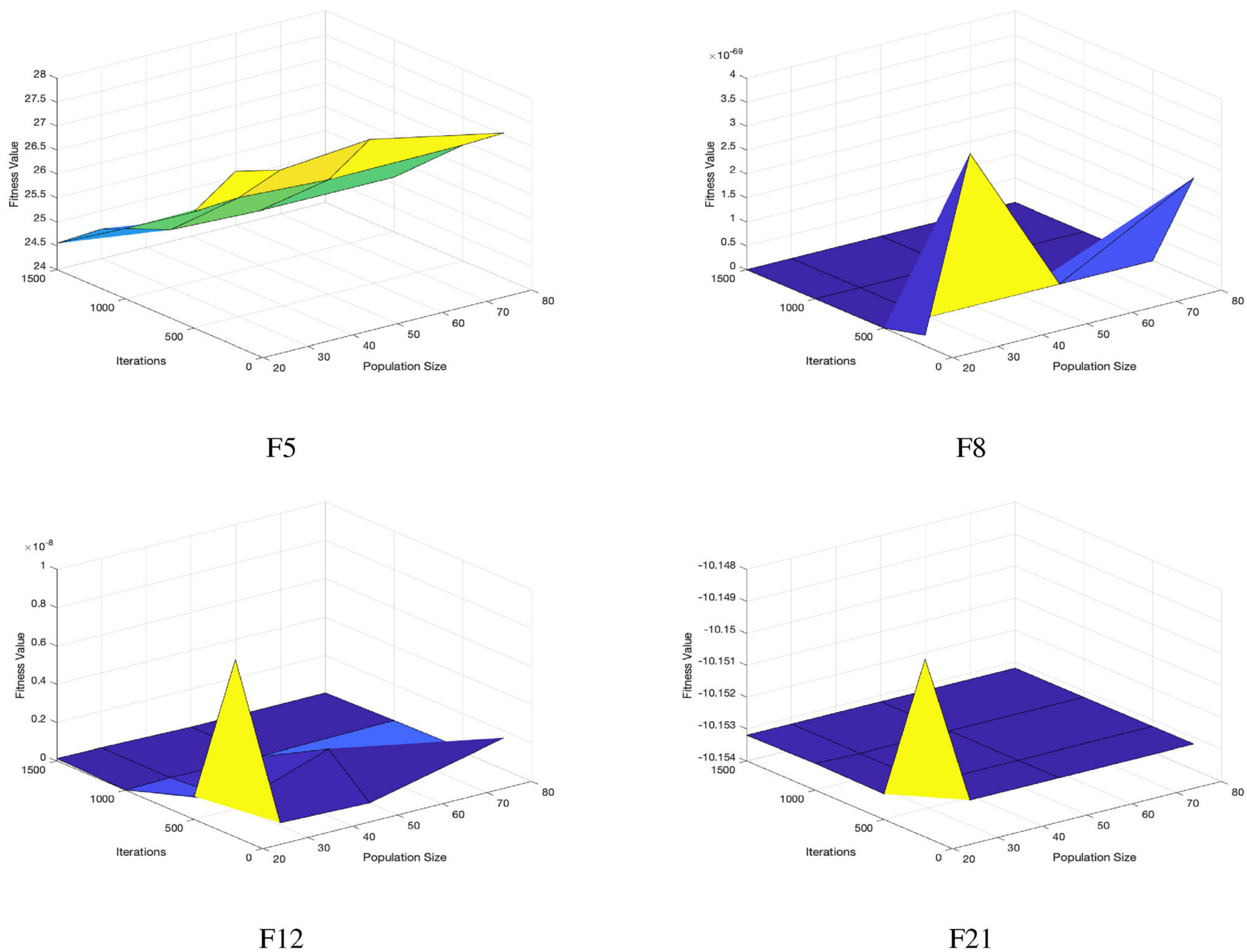

Fig. 10 Sensitivity towards parameter population size and iterations

25, 26, and 27. The maximum possible improvement index is calculated by using the following mathematical equation:

$$Imp = \frac{R_{sys}(EZOA) - R_{sys}(other)}{1 - R_{sys}(others)} \tag{32}$$

here $R_{sys}(EZOA)$ represents the best solution obtained by EZOA and $R_{sys}(others)$ is the best result obtained by the other competitive algorithm used. The greater the value of *Imp*, the greater is the improvement in the proposed EZOA. Further, a statistical analysis of EZOA and ZOA is conducted on all four RRAPs, considering the metrics: average, standard deviation, best, and worst values, which is provided in Table 28.

In the RRAP1 problem (series system), EZOA outperformed 12 algorithms GA, PSO, ABC1, SAA, MICA, NAFSA, IPSO, CS2, SSO, GA-SRS, SCA, and ZOA in achieving the maximum overall system's reliability (best optimal solution) of 0.9316822786221. The results for the series system are provided in Table 24 and the optimal value is highlighted in bold. The *Imp* of EZOA over these algorithms are 0.3242%, 38.7265%, 0.00041%, 0.46517%, 0.004228%, 0.0000155%, 0.003335%, 0.000253%, 0.263203%, 3.6662%, 0.003335%, 000.44% respectively. The optimal redundant component for the RRAP1 by EZOA is (3, 2, 2, 3, 3). The table also reports the values of slack variables, which indicate the unused resources for each constraint.

The result for RRAP2 problem (series-parallel system) are tabulated in Table 25. The best optimal result obtained by EZOA for this case is 0.999986337729189 and the optimal value is highlighted in bold. EZOA demonstrated superior result among 17 algorithms from literature, including GA, PSO, CPSO, MPSO, IPSO, ABC1, ABC2, IABC, SAA, DE, NAFSA, CS1, CS2, CS-GA, ICS, INNA, and ZOA with significant *Imp* value 56.2810%, 90.3482%, 41.5143%, 41.4923%, 39.7873%, 41.4916%, 41.4916%, 32.2657%, 41.4892%, 41.4915%, 41.4942%, 41.4917%, 41.4922%, 41.4892%, 41.4917%, 12.2930%, 15.4303%, respectively. The configuration of optimal redundant components for the RRAP2 problem by EZOA is (3, 2, 2,

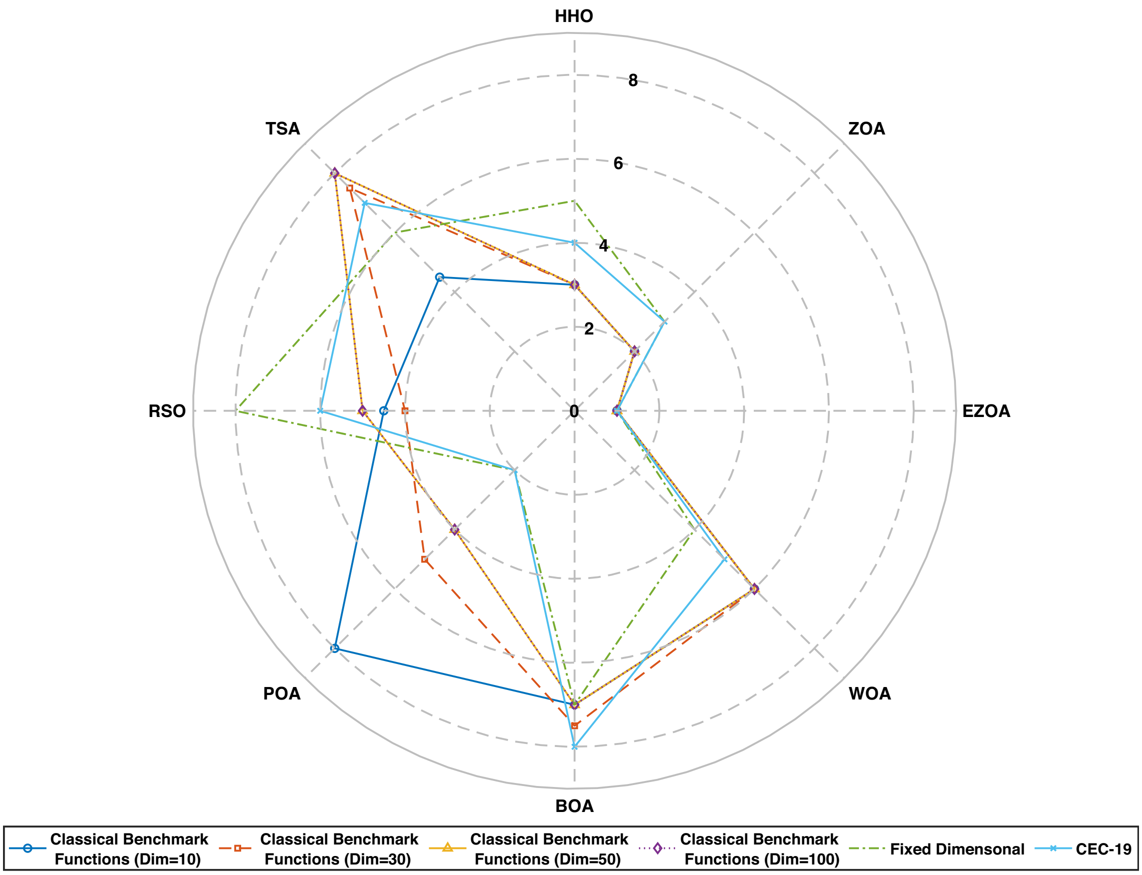

Fig. 11 Radar plot of Friedman's rank on classical benchmark and CEC-2019 test functions

Table 20 Results of Wilcoxon-signed rank test at significance level 0.05 on classical benchmark functions

Algorithms	Mean value	Standard deviation	Z-value	p-value
EZOA vs. ZOA	52.50	15.93	− 2.98	2.86E-03
EZOA vs. HHO	60.00	17.61	− 2.61	9.06E-03
EZOA vs. TSA	126.50	30.80	− 4.11	4.01E-05
EZOA vs. RSO	68.00	19.34	− 3.52	4.37E-04
EZOA vs. POA	60.00	17.61	− 2.44	1.46E-02
EZOA vs. WOA	82.50	22.96	− 3.52	4.55E-04
EZOA vs. BOA	105.00	26.79	− 3.58	3.38E-04

2, 4). Further, it can also be observed from the table that the performance of algorithms ABC1, ABC2, DE, CS1, and ICS is quite competetive with each other.

For the RRAP3 problem (complex-bridge system), the best optimal solution as highlighted in bold is attained by EZOA with the value 0.999889636879556 clearly surpassing 14 competitor alogithms namely, GA, PSO, IPSO, ABC1, ABC2, IA, SAA, MICA, NAFSA, NGHS, INGHS, CS2, EBBO, and ZOA with *Imp* of 8.67%, 66.4141%, 0.00623%, 0.0153%, 0.00097%, 0.26%, 1.78%, 0.00623%,0.00623%, 0.0334%, 0.00435%, 0.00451%, 0.000435%, and 5.71% respectively. The optimal component redundancy by the EZOA algorithm is (3, 3, 3, 3, 1) as shown in Table 26. In this case, the best optimal results obtained by IPSO, INGHS, SAA, and EBBO are nearly same.

The results for RRAP4 problem (gas turbine overspeed protection system) as given in Table 27, reveal that EZOA efficiently demonstrated superior performance by obtaioning the best optimal result of 0.999954674655967. Based on the *Imp* index, the improvement made by EZOA over the 13 algorithms namely, PSO, IPSO, IA, SAA, MICA, NMDE, INGHS, DE, TS-DE, CS2, EBBO, INNA, and ZOA are 52.4193%, 0.0103%, 0.000234%, 17.5903%, 0.00365%, 0.0103%, 0.00145%, 90.001%, 0.000011%, 0.0014%, 0.00145%, 0.0000573%, and 0.00715%,

Table 21 p-values by Wilcoxon-signed rank test on CEC-2019 test function

Function	EZOA vs						
	ZOA	HHO	TSA	RSO	POA	WOA	BOA
C-01	8.59E-02	1.00E-04	1.00E-04	1.00E-04	1.00E-04	1.00E-04	5.00E-04
C-02	9.60E-04	1.40E-04	8.00E-05	3.00E-04	3.38E-03	6.40E-04	8.00E-05
C-03	1.00E-04	6.40E-04	6.40E-04	6.40E-04	1.00E-04	5.13E-03	6.40E-04
C-04	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05
C-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05
C-06	3.66E-02	1.00E-04	8.00E-05	1.20E-04	6.34E-03	8.00E-05	8.00E-05
C-07	1.68E-02	1.00E-04	8.00E-05	8.00E-05	9.06E-03	8.00E-05	8.00E-05
C-08	1.94E-03	8.00E-05	8.00E-05	8.00E-05	3.00E-02	1.00E-04	8.00E-05
C-09	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05	8.00E-05
C-10	3.62E-03	6.00E-04	8.00E-05	8.00E-05	1.68E-02	8.00E-05	8.00E-05
Z-value	− 2.80	− 2.80	− 2.80	− 2.80	− 1.60	− 2.80	− 2.80
p-value	5.10E-03	5.10E-03	5.10E-03	5.10E-03	1.10E-01	5.10E-03	5.10E-03

respectively. The optimal redundancy configuration of the component by EZOA is (5, 6, 4, 5). The *Imp* values might seem low but even a small improvement in reliability is significant for a system's performance.

The statistical results for ZOA and EZOA are summarized in Table 28. The EZOA algorithm demonstrates superior performance by achieving better average results across all four problems. The lower standard deviation value of EZOA compared to ZOA suggests that the consistency of EZOA in achieving optimal results is higher. Also, EZOA consistently achieves better worst-case optimal solutions than ZOA across all four cases. Overall, the performance of EZOA dominates ZOA in solving RRAP problems and observed by their rankings in bold. In all these cases, the changes in reliability values achieved by various competitor algorithms may seem minor, but these improvements are highly beneficial for the system's efficiency and hence reported with precision up to 15 decimal places. The convergence curves of the proposed EZOA and ZOA for all four RRAPs are depicted in Fig. 24. The curves reveal that EZOA exhibits a more rapid and stable convergence towards the optimal solution compared to ZOA. After analyzing all the simulation results for the four RRAP problems, it is evident that the proposed EZOA algorithm is efficient in consistently delivering superior optimal results compared to existing algorithms in the literature.

8 Simulation results on constrained engineering design optimization problems

In this section, the effectiveness of the EZOA algorithm is examined through its application to four benchmark engineering design problems (P1–P4). A comparative performance analysis is carried out against the most recent metaheuristic algorithms WOA, BOA, HHO, TSA, FOX, RSO, ALO, SOA, STOA, ZOA. The detailed discussion of the obtained results is presented below.

8.1 P1: corrugated bulkhead design case (BD)

Corrugated bulkhead designs are utilized in chemical tankers to efficiently clean cargo tanks during loading operations. The objective of bulkhead design problem is to minimize the weight of wave-trough bulkhead for chemical tankers subjected to six specific constraints as depicted in Fig. 25. The four design parameters are: width (Z_1), depth (Z_2), length (Z_3), and plate thickness (Z_4), represented as $Z = (Z_1, Z_2, Z_3, Z_4)$. The mathematical formulation of the optimization problem aiming to optimize these variables for a lightweight bulkhead design is as follows:

$$\begin{aligned} \text{Consider: } Z = &\ [Z_1,\ Z_2,\ Z_3,\ Z_4] = [b,\ h,\ l,\ t] \\ \text{Min: } F(Z) = &\ \frac{5.885 Z_4 (Z_1 + Z_3)}{Z_1 + \sqrt{Z_3^2 - Z_2^2}} \\ \text{Subject to: } G_1(Z) = &\ Z_4 Z_2 \left(0.4 Z_1 + \frac{1}{6}\right) \\ &- 8.94\left(Z_1 + \sqrt{Z_3^2 - Z_2^2}\right) \geq 0 \end{aligned} \tag{33}$$

Table 22 p-value of Mann–Whitney U test

Function	Algorithm						
	ZOA	HHO	TSA	RSO	POA	BOA	WOA
Dim = 10							
F1	N/A	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F2	1.21E-12	1.21E-12	1.21E-12	2.79E-03	1.21E-12	1.21E-12	1.21E-12
F3	N/A	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F4	1.21E-12	1.21E-12	1.21E-12	0.81E-05	1.21E-12	1.21E-12	1.21E-12
F5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.70E-11	3.02E-11	6.12E-10
F6	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F7	1.17E-09	2.25E-04	3.34E-11	<u>6.74E-01</u>	1.76E-01	3.02E-11	1.36E-07
F8	8.01E-09	8.01E-09	8.01E-09	8.06E-02	8.01E-09	8.01E-09	8.01E-09
F9	<u>3.33E-01</u>	N/A	1.21E-12	N/A	N/A	1.66E-11	N/A
F10	N/A	N/A	1.01E-12	<u>3.33E-01</u>	5.36E-09	4.13E-12	2.74E-09
F11	N/A	N/A	1.21E-12	N/A	N/A	<u>3.33E-01</u>	2.79E-03
F12	1.75E-05	1.21E-12	3.26E-07	4.45E-04	6.55E-04	2.20E-07	1.98E-08
F13	N/A	N/A	2.21E-06	N/A	N/A	1.31E-07	<u>3.34E-01</u>
Dim=30							
F1	N/A	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F2	1.21E-12	1.21E-12	1.21E-12	2.79E-03	1.21E-12	1.21E-12	1.21E-12
F3	N/A	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F4	1.21E-12	1.21E-12	1.21E-12	2.79E-03	1.21E-12	1.21E-12	1.21E-12
F5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.69E-11	3.02E-11	3.02E-11
F6	3.02E-11	1.87E-07	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F7	2.49E-06	6.74E-06	3.02E-11	<u>4.03E-01</u>	<u>5.01E-01</u>	3.02E-11	4.21E-04
F8	8.01E-09	8.01E-09	8.01E-09	9.58E-03	8.01E-09	8.01E-09	8.01E-09
F9	<u>3.33E-01</u>	N/A	1.21E-12	N/A	N/A	2.16E-02	N/A
F10	N/A	N/A	1.16E-12	4.18E-02	1.83E-08	1.21E-12	9.16E-09
F11	N/A	N/A	2.93E-05	N/A	N/A	2.93E-05	<u>1.61E-01</u>
F12	2.53E-04	1.21E-12	9.51E-06	3.02E-02	1.58E-04	5.97E-09	3.46E-08
F13	N/A	N/A	3.45E-07	N/A	N/A	5.85E-09	N/A
Dim=50							
F1	N/A	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F2	1.21E-12	1.21E-12	1.21E-12	<u>3.33E-01</u>	1.21E-12	1.21E-12	1.21E-12
F3	1.21E-12	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F4	1.21E-12	1.21E-12	1.21E-12	5.58E-03	1.21E-12	1.21E-12	1.21E-12
F5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F6	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	9.51E-06
F7	2.92E-09	9.83E-08	3.02E-11	7.24E-02	2.24E-04	3.02E-11	5.19E-07
F8	8.01E-09	8.01E-09	8.01E-09	1.98E-02	8.01E-09	8.01E-09	8.01E-09
F9	N/A	N/A	1.21E-12	N/A	N/A	0.041926	N/A
F10	N/A	N/A	8.89E-13	2.14E-02	1.83E-08	1.21E-12	2.76E-07
F11	N/A	N/A	2.21E-06	N/A	N/A	1.66E-11	0.1608
F12	2.32E-02	1.21E-12	1.47E-07	8.63E-03	5.19E-02	1.11E-06	8.05E-10
F13	N/A	N/A	8.87E-07	N/A	N/A	4.79E-08	<u>3.33E-01</u>
Dim=100							
F1	N/A	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F2	1.21E-12	1.21E-12	1.21E-12	4.19E-02	1.21E-12	1.21E-12	1.21E-12
F3	1.21E-12	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12	1.21E-12
F4	1.21E-12	1.21E-12	1.21E-12	4.79E-08	1.21E-12	1.21E-12	1.21E-12
F5	1.21E-12	1.21E-12	1.21E-12	6.25E-10	1.21E-12	1.21E-12	1.21E-12

Table 22 continued

Function	Algorithm						
	ZOA	HHO	TSA	RSO	POA	BOA	WOA
F6	5.49E-11	3.02E-11	1.61E-10	3.02E-11	8.99E-11	3.02E-11	4.62E-10
F7	1.20E-08	5.27E-05	3.02E-11	2.01E-01	4.86E-03	3.34E-11	1.60E-07
F8	8.01E-09	8.01E-09	8.01E-09	8.06E-02	8.01E-09	8.01E-09	8.01E-09
F9	N/A	N/A	1.21E-12	N/A	N/A	N/A	N/A
F10	N/A	N/A	1.12E-12	4.19E-02	3.78E-10	1.21E-12	7.36E-08
F11	N/A	N/A	1.44E-04	N/A	N/A	1.21E-12	3.34E-01
F12	1.12E-04	1.21E-12	1.75E-05	3.91E-02	1.33E-04	3.81E-07	7.36E-11
F13	N/A	N/A	5.38E-06	N/A	N/A	5.85E-09	N/A
Fixed dimension							
F14	1.65E-11	1.21E-12	1.21E-12	1.21E-12	3.37E-01	1.21E-12	1.21E-12
F15	6.05E-07	4.38E-01	5.11E-01	7.39E-11	3.02E-11	3.87E-01	1.11E-06
F16	1.46E-11	1.18E-12	1.21E-12	1.21E-12	2.85E-04	1.21E-12	1.21E-12
F17	4.79E-08	1.21E-12	1.21E-12	1.21E-12	N/A	1.21E-12	1.21E-12
F18	5.42E-08	5.22E-12	5.22E-12	5.22E-12	2.23E-01	5.22E-12	5.22E-12
F19	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.32E-04	2.71E-14	1.21E-12
F20	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12	4.29E-14	1.72E-12
F21	7.57E-12	7.57E-12	7.57E-12	7.57E-12	9.92E-11	7.57E-12	7.57E-12
F22	1.34E-11	1.34E-11	1.34E-11	1.34E-11	1.73E-05	1.34E-11	1.34E-11
F23	3.16E-12	3.16E-12	3.16E-12	3.16E-12	9.75E-07	3.16E-12	3.16E-12

Underlined value are greater than 0.05

$$G_2(Z) = Z_4 Z_2^2\left(0.2Z_1 + \frac{1}{12}\right) - 2.2\left(8.94\left(Z_1 + \sqrt{Z_3^2 - Z_2^2}\right)\right)^{4/3} \geq 0 \tag{34}$$

$$G_3(Z) = Z_4 - 0.0156Z_1 - 0.15 \geq 0 \tag{35}$$

$$G_4(Z) = Z_4 - 0.0156Z_3 - 0.15 \geq 0 \tag{36}$$

$$G_5(Z) = Z_4 - 1.15 \geq 0 \tag{37}$$

$$G_6(Z) = Z_3 - Z_2 \geq 0 \tag{38}$$

$$\text{with bounds } 0 \leq Z_1, Z_2, Z_3 \leq 100 \quad \text{and} \quad 0 \leq Z_4 \leq 5 \tag{39}$$

The comparison results of the algorithms corresponding to their best outcomes are presented in Table 29. The results indicate that the EZOA algorithm achieves an optimum value of $F(Z) = 6.8429$ with design parameters $(Z_1, Z_2, Z_3, Z_4 = 57.6923, 34.1476, 57.6923, 1.05)$. From the statistical outcomes presented in Table 30, it is evident that EZOA attained the best mean value of 6.8429, followed by ZOA, ALO, TSA, and HHO outperforming all ten algorithms considered in the study. The lowest standard deviation value of EZOA algorithm also revelas the robust nature of the algorithm.

8.2 P2: tubular column design case (TCD)

The objective of tubular design optimization problem is to design a uniform tubular column of length (l) of 250 cm that can support a compression load (P) of 2500 kgf at minimum possible cost. The design optimization involves two key variables: the column's mean diameter (Z_1) and its thickness (Z_2). Considering the material specifications for the column, including density (ρ) yield stress (σ_y), and modulus of elasticity (E), the design must ensure that the applied stress does not exceeds the buckling stress and the yield stress (constraint G_1 and G_2). The column's mean diameter and the column thickness is constrained to lie between 2 cm and 14 cm (constraints G_3 and G_4) and 0.2 cm to 0.8 cm (constraints G_5 and G_6) respectively. The mathematical formulation of the optimization problem is gives as follows:

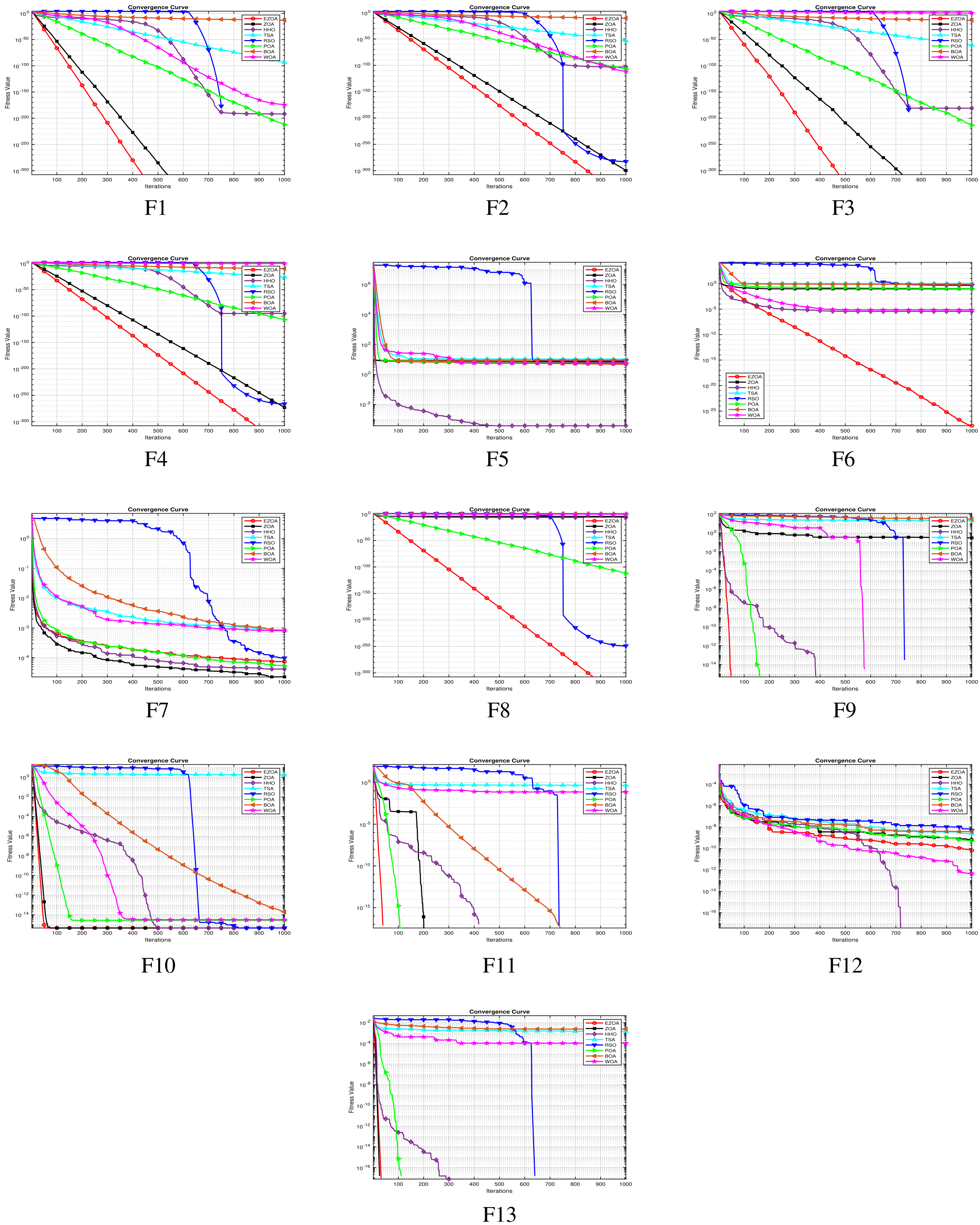

Fig. 12 Convergence plot of the algorithms on F1–F13 benchmark functions with dimension 10

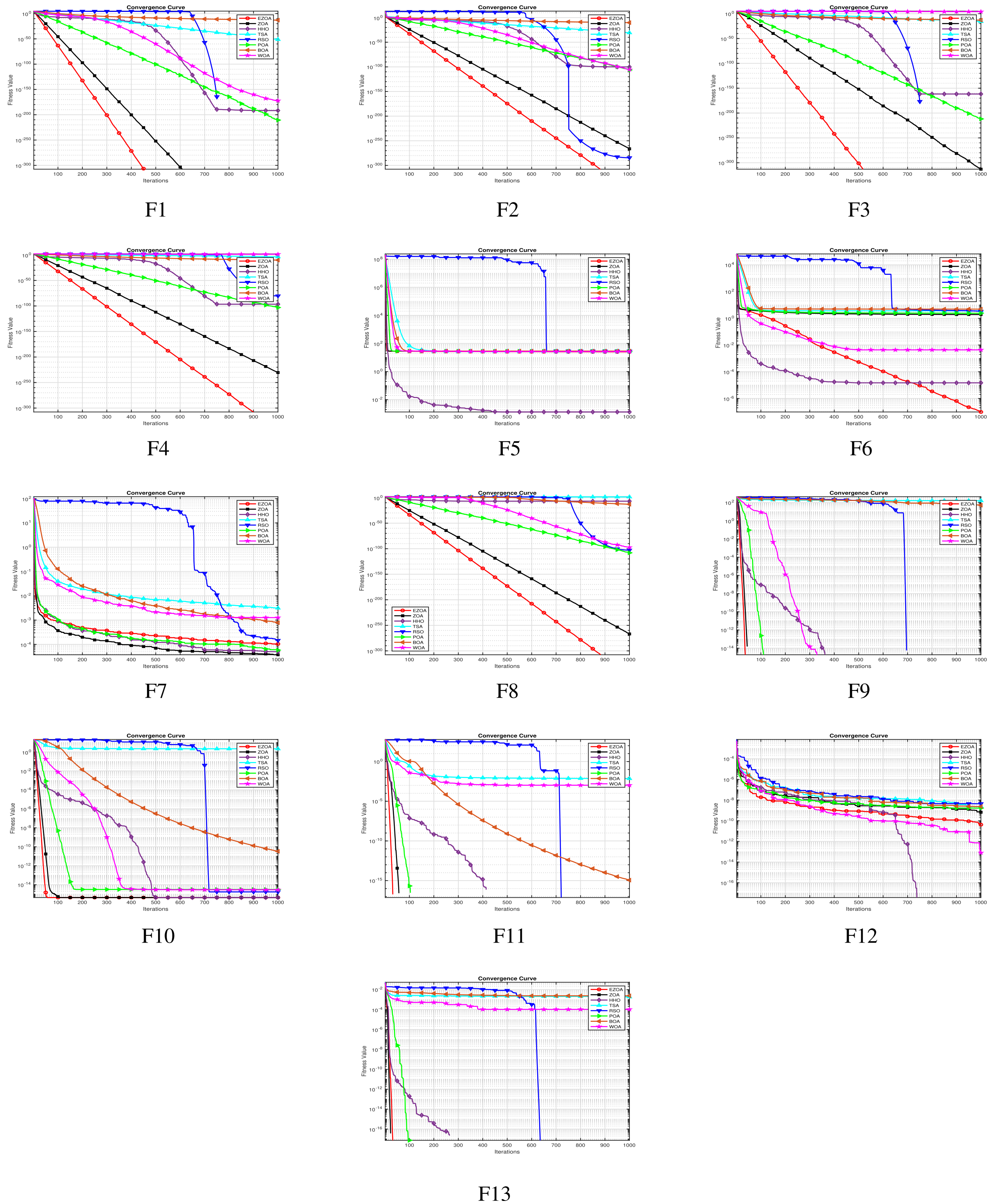

Fig. 13 Convergence plot of the algorithms on F1–F13 benchmark functions with dimension 30

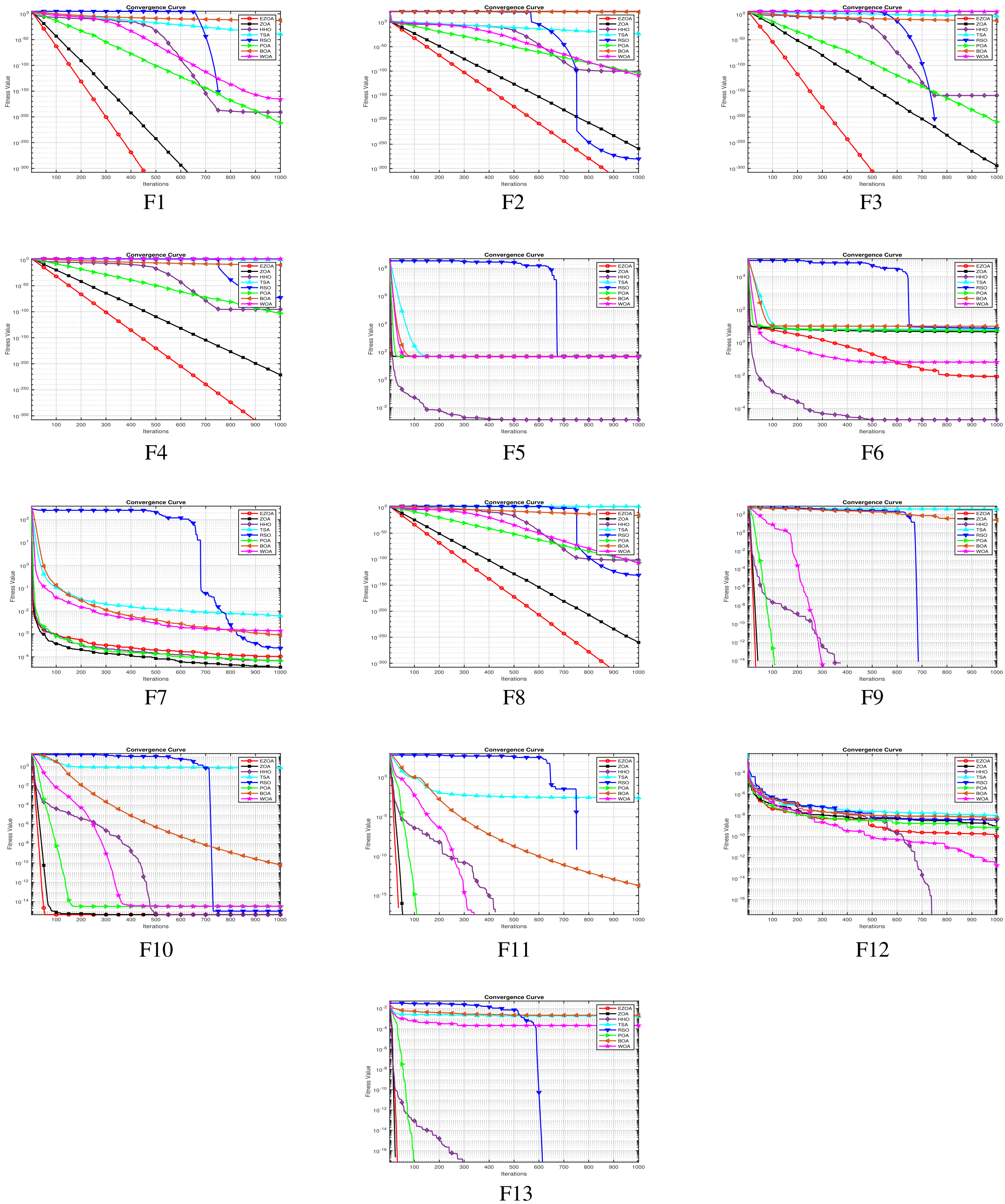

Fig. 14 Convergence plot of the algorithms on F1–F13 benchmark functions with dimension 50

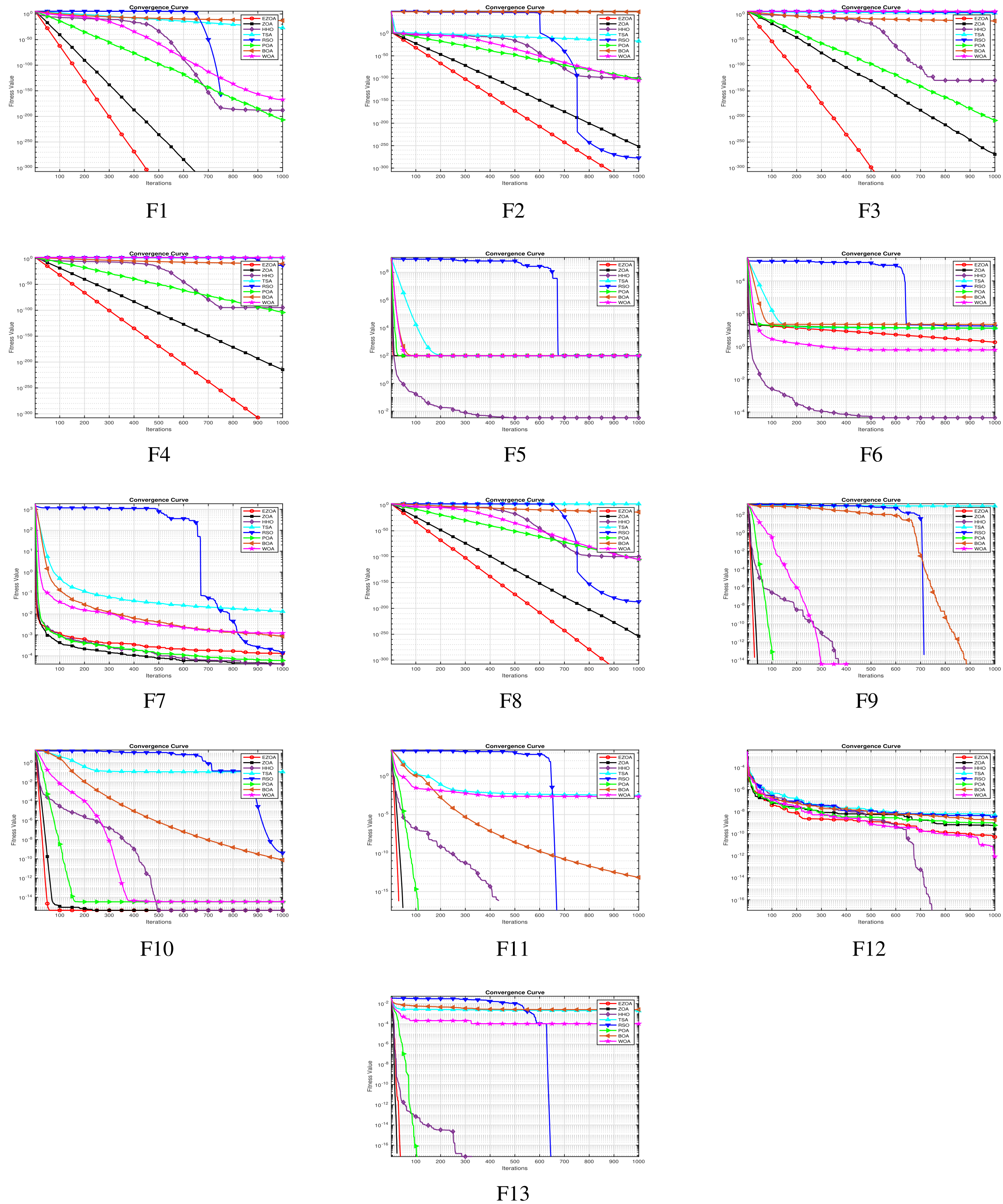

Fig. 15 Convergence plot of the algorithms on F1–F13 benchmark functions with dimension 100

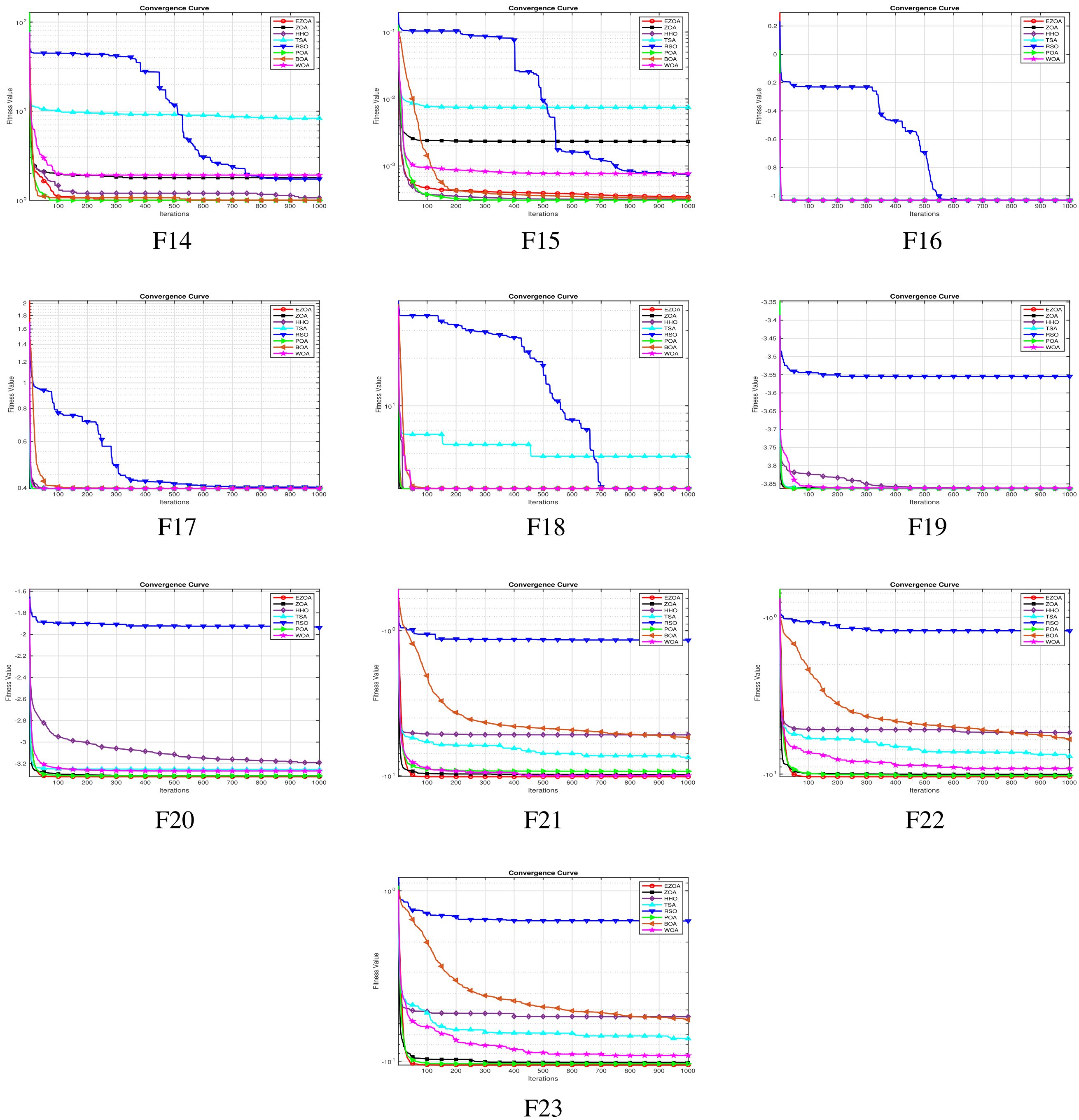

Fig. 16 Convergence plot of the algorithms on F14–F23 benchmark functions

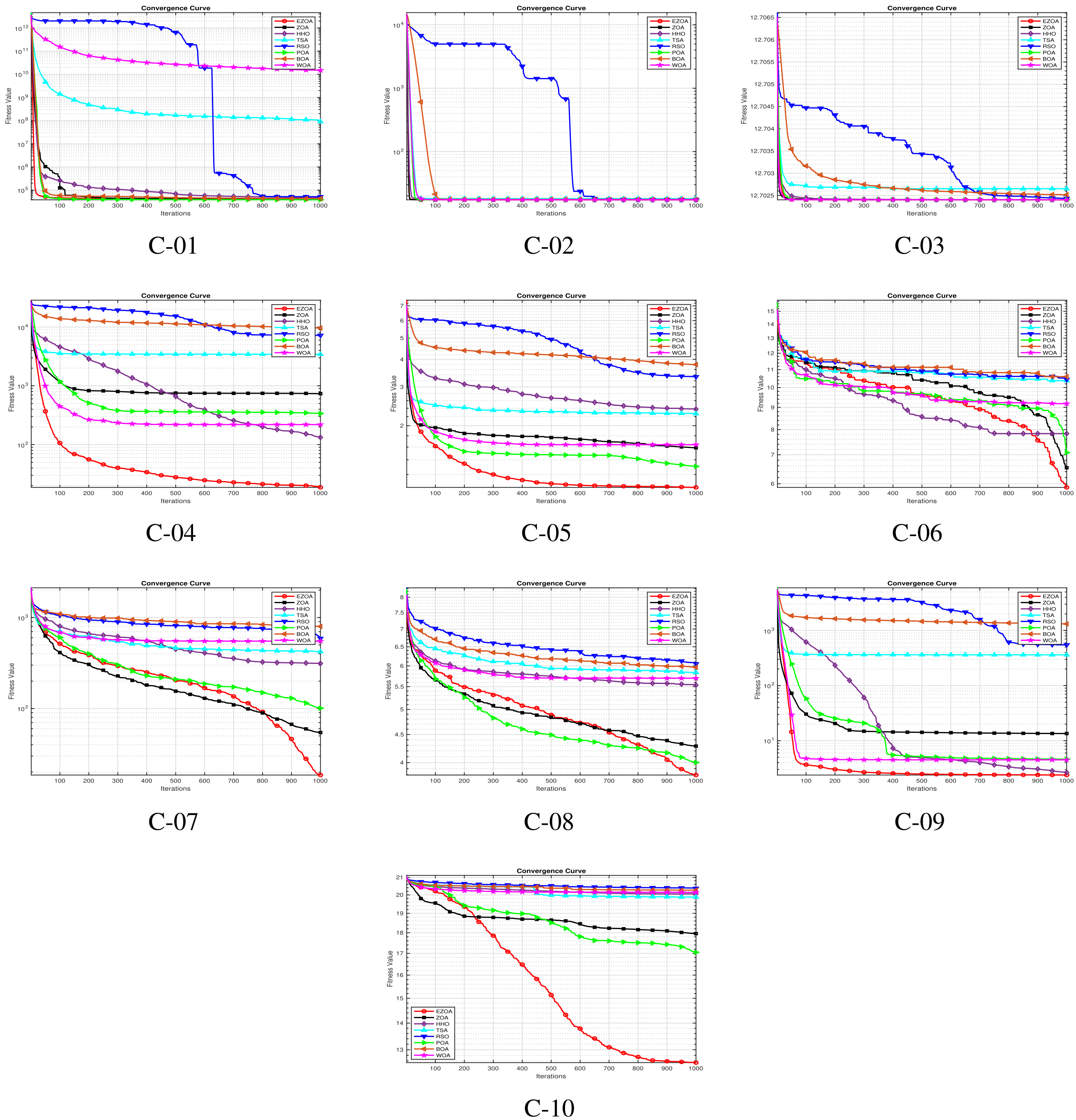

Fig. 17 Convergence plot of the algorithms on CEC-19 functions

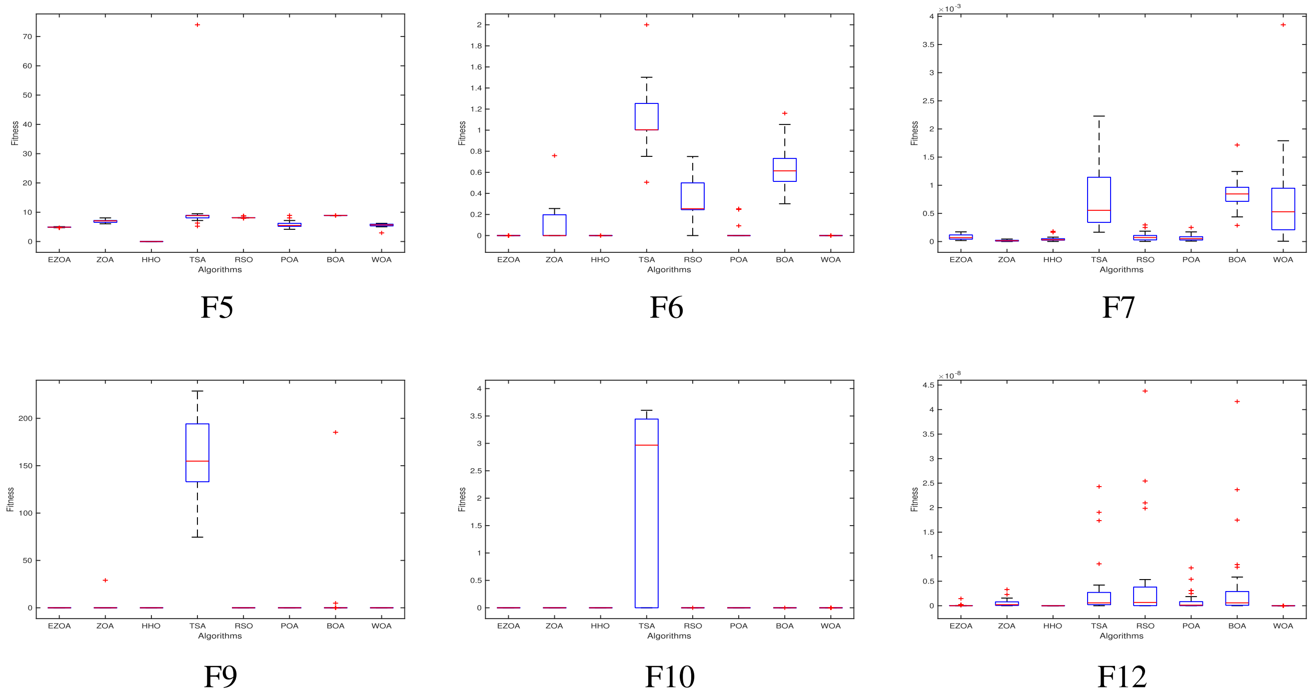

Fig. 18 Boxplot of the algorithms on some varying dimensional benchmark functions with dimension 10

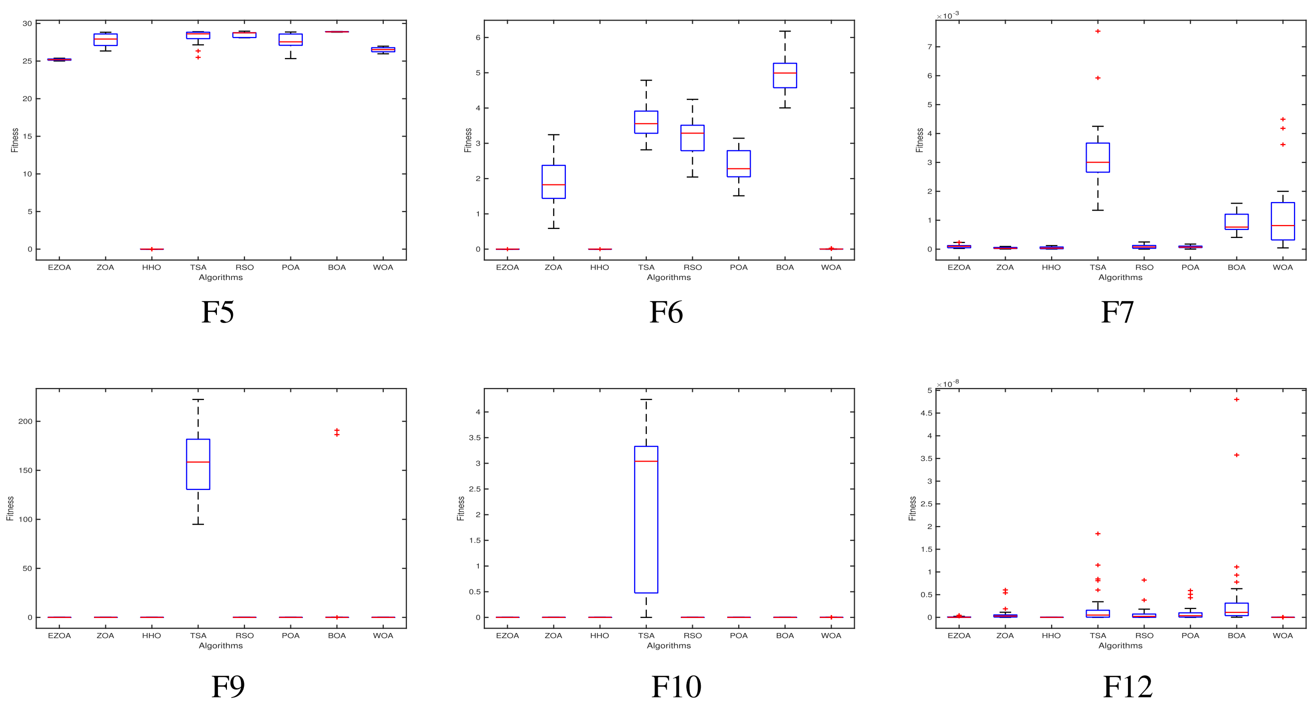

Fig. 19 Boxplot of the algorithms on some varying dimensional benchmark functions with dimension 30

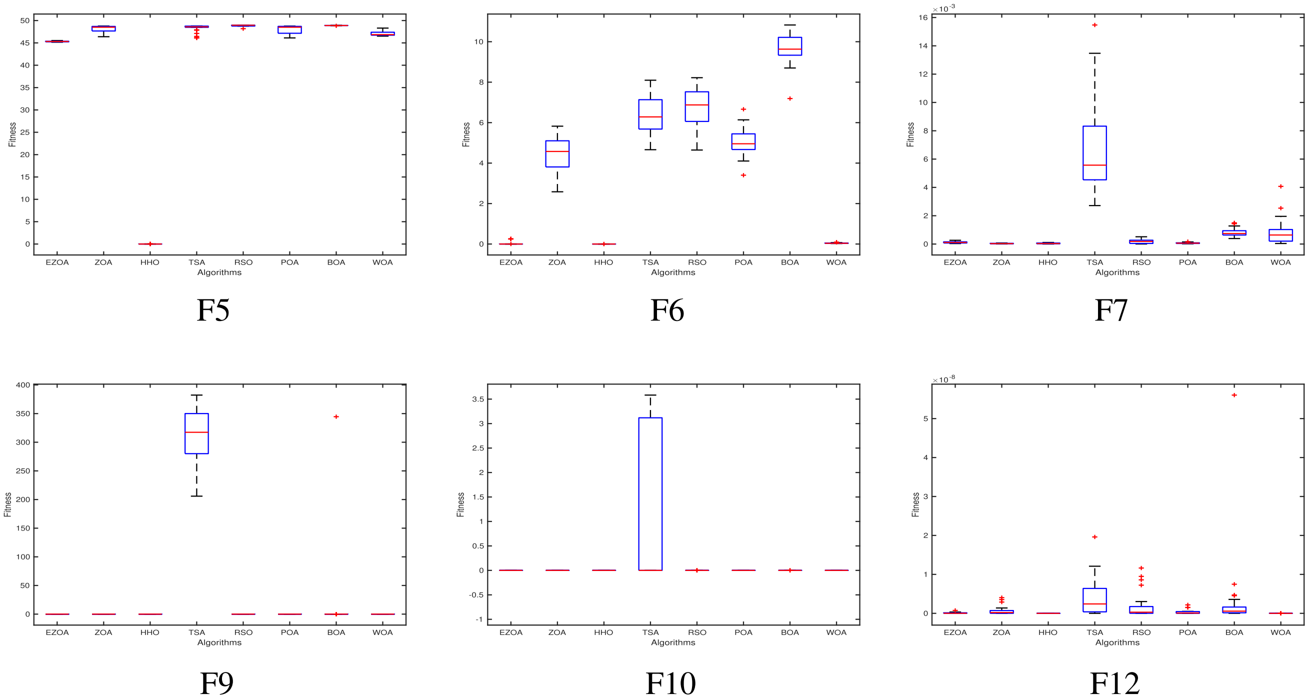

Fig. 20 Boxplot of the algorithms on some varying dimensional benchmark functions with dimension 50

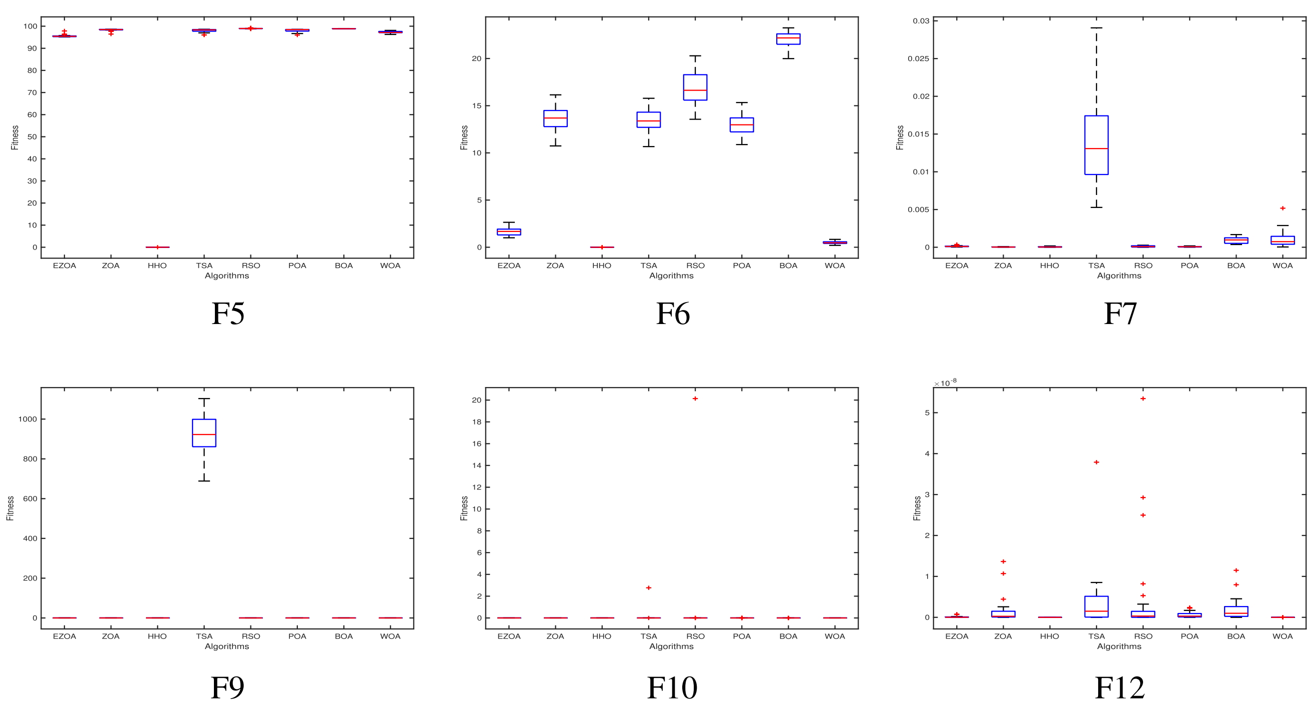

Fig. 21 Boxplot of the algorithms on some varying dimensional benchmark functions with dimension 100

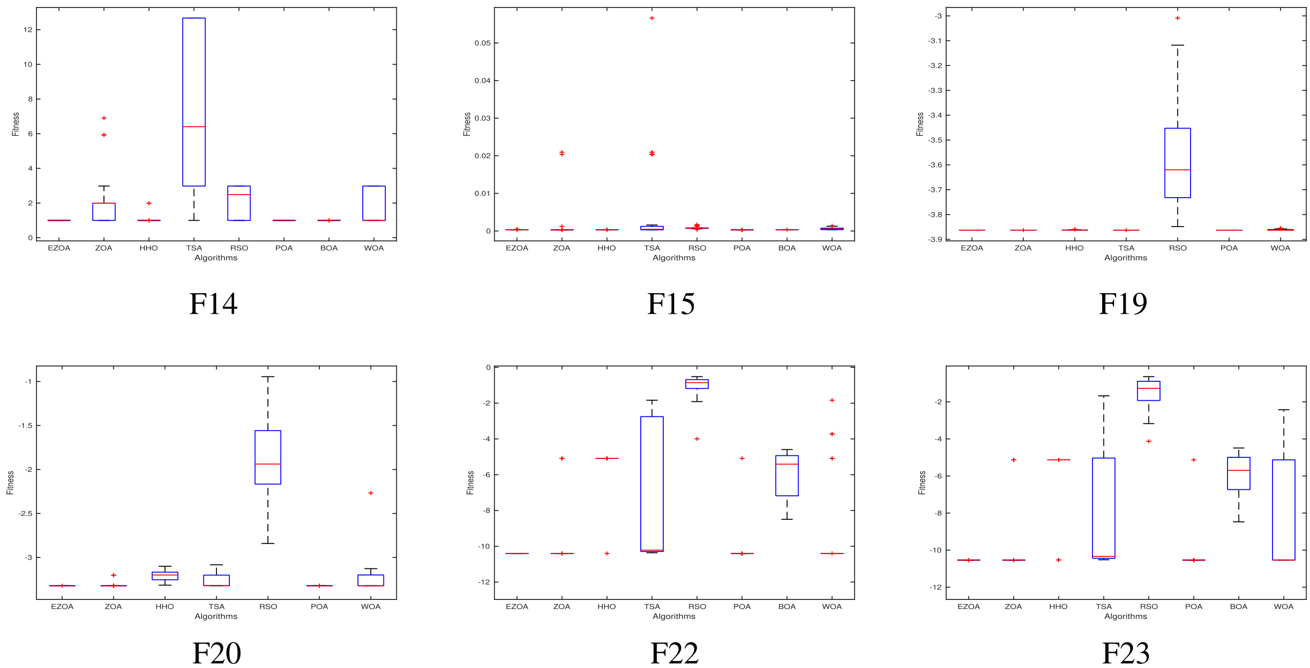

Fig. 22 Boxplot of the algorithms on some fixed dimension benchmark functions

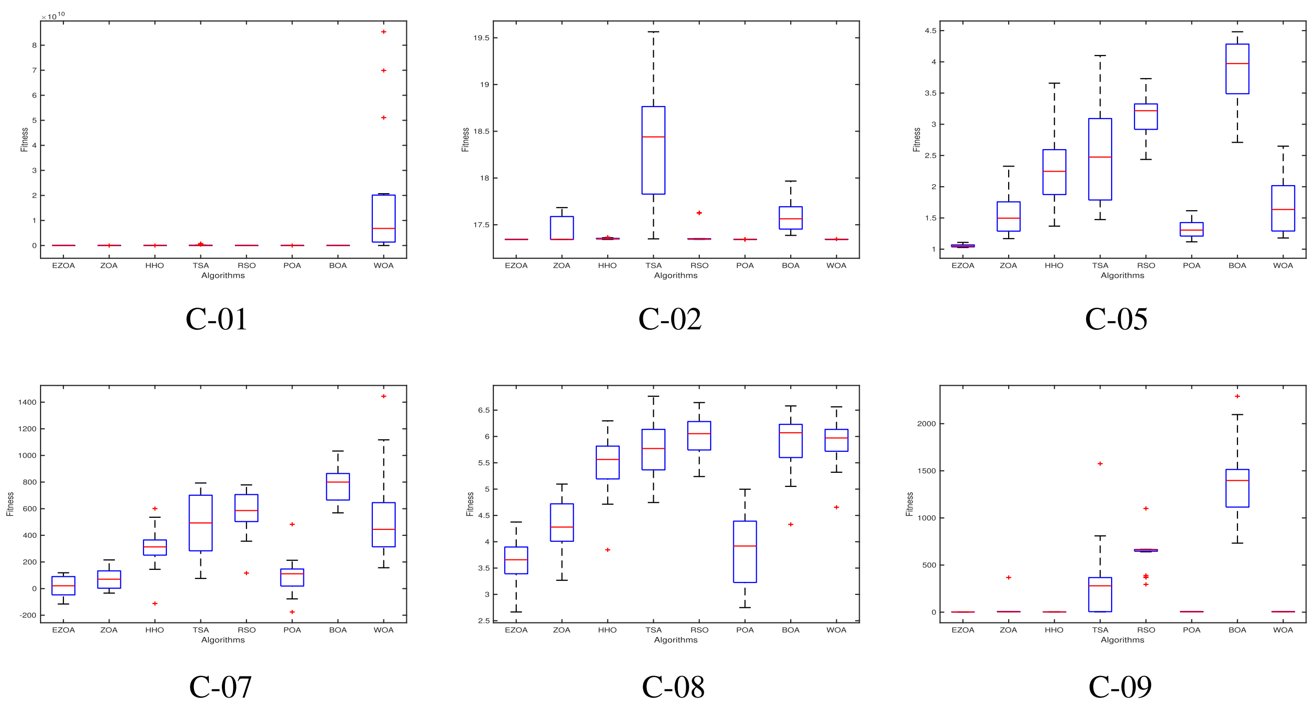

Fig. 23 Boxplot of the algorithms on some CEC-19 functions

$$
\begin{aligned}
\text{Consider: } Z &= [Z_1,\ Z_2] = [d,\ t] \\
\text{Min: } F(Z) &= 9.8Z_1Z_2 + 2Z_1 \\
\text{Subject to: } G_1(Z) &= \left(\frac{P}{\pi Z_1 Z_2 \sigma_y}\right) - 1 \le 0
\end{aligned}
\tag{40}
$$

$$
G_2(Z) = \left(\frac{8PL^2}{\pi^3 E Z_1 Z_2 (Z_1^2 + Z_2^2)}\right) - 1 \le 0 \tag{41}
$$

$$
G_3(Z) = \frac{2}{Z_1} - 1 \le 0 \tag{42}
$$

Table 23 Analysis of time complexity

Algorithm	Time (in sec)			
	T0	T1	T2	(T2 − T1)/T0
EZOA	0.024	7.038	7.0961	2.4208
ZOA	0.024	3.4177	3.4565	1.6167
HHO	0.024	4.0832	4.1181	1.4542
TSA	0.024	1.7329	1.7493	0.6833
RSO	0.024	1.7726	1.7556	0.7083
POA	0.024	3.4984	3.4651	1.3875
WOA	0.024	3.4331	3.5041	2.9583
BOA	0.024	1.7453	1.7906	1.8875

$$G_4(Z) = \frac{Z_1}{14} - 1 \leq 0 \tag{43}$$

$$G_5(Z) = \frac{0.2}{Z_2} - 1 \leq 0 \tag{44}$$

$$G_6(Z) = \frac{Z_2}{0.8} - 1 \leq 0 \tag{45}$$

$$\text{with bounds } 2 \leq Z_1 \leq 14 \quad \text{and} \quad 0.2 \leq Z_4 \leq 0.8 \tag{46}$$

The configurational diagram of the tubular design case is given in Fig. 26. Table 31 presents the best outcome obtained from 30 independent runs. The optimal performance of $F(Z) = 26.48636147$ is reported by EZOA, corresponding to the design parameters $(Z_1, Z_2 = 5.4522, 0.29163)$. The statistical results of EZOA along with other competitive algorithms, are summarized in Table 32. The mean value of EZOA is also 26.48636147 with standard deviation value $3.65E-15$, indicating consistently superior performance in achieving optimal result. Even the worst result obtained by EZOA surpassed those of all other algorithms.

8.3 P3: tension/compression spring design case (SD)

The aim of SD problem is to minimize the weight of the spring by developing an optimal design. The reduction of spring's mass is achieved by adjusting the dimensions of three design variables: the mean coil diameter (Z_1), the diameter (Z_2) of wire, and the number of active coils (Z_3). The problem involves constraints on minimum (G_1), shear stress (G_2), impact frequency (G_3), and outer diameter limit (G_4). The configurational diagram of SD is illustrated in Fig. 27. The problem is represented mathematically as:

Table 24 Comparison of optimal results for RRAP1 (Series System) with existing literature

Algorithm	R_{sys}	r_1	r_2	r_3	r_4	r_5	Slack variables			$(n_1, n_2, n_3, n_4, n_5)$	$Imp(\%)$
							g_1	g_2	g_3		
GA	0.931460	0.782391	0.866712	0.901747	0.717266	0.783795	27	5.32E-02	7.518918	(3, 2, 2, 3, 3)	3.24E-01
PSO	0.8885037	0.800593	0.7404932	0.8291438	0.6368614	0.8870428	4	16.7757	4.814615	(2, 3, 2, 4, 2)	38.72647
ABC1	0.931682	0.779399	0.871837	0.902885	0.711403	0.78780	27	2.18E-04	7.5189182	(3, 2, 2, 3, 3)	4.08E-04
SAA	0.931363	0.777143	0.867514	0.896696	0.717739	0.793889	27	0	7.518918	(3, 2, 2, 3, 3)	4.65E-01
MICA	0.93167939	0.779874	0.872057	0.903426	0.710960	0.786902	27	2.27E-05	7.5189182411	(3, 2, 2, 3, 3)	4.23E-03
NAFSA	0.931682268	0.779388413	0.871720982	0.903033391	0.711418362	0.787789288	27	6.73E-09	7.518918	(3, 2, 2, 3, 3)	1.55E-05
IPSO	0.931680	0.780037307	0.87178343	0.90240890	0.71147356	0.78738760	27	1.01E-04	7.518918	(3, 2, 2, 3, 3)	3.34E-03
CS2	0.931682106	0.779439734	0.871995212	0.902873050	0.711127088	0.787986374	27	4.43E-07	7.518918241	(3, 2, 2, 3, 3)	2.53E-04
SSO	0.93150199	0.78271484	0.87335199	0.90264893	0.71313477	0.77729797	27	1.82E-03	7.51891824	(3, 2, 2, 3, 3)	2.63E-01
GA-SRS	0.929082263568	0.76459335	0.88752892	0.91539527	0.69350544	0.77603145	27	2.27E-05	7.5189182411	(3, 2, 2, 3, 3)	3.66622
SCA	0.931680	0.779427	0.869482	0.902674	0.714038	0.786896	27	1.21E-01	7.518918	(3, 2, 2, 3, 3)	3.36E-03
ZOA	0.9316792743150	0.77794	0.7999	0.90226	0.71151	0.85969	27	1.44E-04	10.5725	(3, 3, 2, 3, 2)	4.40E-03
EZOA	**0.9316822786221**	0.77975	0.87155	0.90292	0.71159	0.78757	27	8.39E-04	7.5189	(3, 2, 2, 3, 3)	–

Table 25 Comparison of optimal results for RRAP2 (series-parallel system) with existing literature

Algorithm	R_{sys}	r_1	r_2	r_3	r_4	r_5	Slack variables			$(n_1, n_2, n_3, n_4, n_5)$	$Imp(\%)$
							g_1	g_2	g_3		
GA	0.99996875	0.838193	0.855065	0.878859	0.911402	0.850355	53	0	7.110849	(3, 3, 1, 2, 3)	56.2810
PSO	0.99985845	0.84025282	0.88865099	0.62375055	0.93984950	0.75158691	68	0.9169151	4.01770364	(4, 3, 2, 1, 2)	90.3482
CPSO	0.99997664	0.81918526	0.84366421	0.89472992	0.89537628	0.86912724	40	5.61E-04	1.609289	(2, 2, 2, 2, 4)	41.5143
MPSO	0.9999766487381	0.81958939	0.84458412	0.89534134	0.89581626	0.86852902	40	8.12E-05	1.609288966	(2, 2, 2, 2, 4)	41.4923
IPSO	0.99997731	0.8197457	0.8450080	0.8954581	0.9009032	0.8684069	40	1.46952234	1.60928897	(2, 2, 2, 2, 4)	39.7873
ABC1	0.999976649036	0.819591561	0.844951068	0.895428548	0.895522339	0.868490229	40	5.98E-04	1.609288966	(2, 2, 2, 2, 4)	41.4916
ABC2	0.999976649054	0.8197377535	0.8449911	0.895529544	0.895433687	0.8684348245	40	1.39E-10	1.609288966	(2, 2, 2, 2, 4)	41.4916
IABC	0.999979829614	0.8277437120	0.8471386118	0.856891035	0.856634348	0.8759765311	38	1.83E-05	0.705901612	(3, 3, 2, 2, 3)	32.2657
SAA	0.99997665	0.819596	0.845000	0.895514	0.895519	0.868456	40	7.00E-06	1.609289	(2, 2, 2, 2, 4)	41.4892
DE	0.9999766490660	0.81965932	0.84498074	0.89550642	0.89550643	0.86844775	40	1.96E-07	1.609288966	(2, 2, 2, 2, 4)	41.4915
NAFSA	0.999976648004	0.819787575	0.845671944	0.89486836	0.895908269	0.868295831	40	3.12E-08	1.609289	(2, 2, 2, 2, 4)	41.4942
CS1	0.999976649	0.819927087	0.845267657	0.895491554	0.895440692	0.868318775	40	1.61E-05	1.6092890	(2, 2, 2, 2, 4)	41.4917
CS2	0.999976648818	0.8194832325	0.8447830845	0.895810554	0.895220217	0.868542487	40	2.72E-10	1.609288966	(2, 2, 2, 2, 4)	41.4922
CS-GA	0.99997665	0.819660256	0.844981615	0.895519305	0.895492245	0.868447587	40	1.70E-08	1.690928897	(2, 2, 2, 2, 4)	41.4892
ICS	0.999976649	0.819927087	0.845267657	0.895491554	0.895440692	0.868318775	40	1.61E-05	1.6092890	(2, 2, 2, 2, 4)	41.4917
INNA	0.99998442283	0.84342538	0.79318760	0.89238731	0.89260221	0.86456512	20	1.56E-05	0.26819176	(2, 3, 2, 2, 4)	12.2930
ZOA	0.999983844949936	0.81582	0.84095	0.86172	0.86219	0.87731	15	4.01E-04	2.68E-01	(2, 2, 3, 2, 4)	15.4303
EZOA	**0.999986337729189**	0.77496	0.86932	0.88939	0.89121	0.86433	30	2.04E-02	1.950	(3, 2, 2, 2, 4)	–

Table 26 Comparison of the optimal result for RRAP3 (complex system) with existing literature

Algorithm	R_{sys}	r_1	r_2	r_3	r_4	r_5	Slack variables			$(n_1, n_2, n_3, n_4, n_5)$	$Imp(\%)$
							g_1	g_2	g_3		
GA	0.99987916	0.814090	0.864614	0.890291	0.701190	0.734731	18	3.76E-01	4.26477	(3, 3, 3, 3, 1)	8.6700
PSO	0.99967140	0.770616	0.901092	0.892786	0.600830	0.73451	37	16.545713	1.41180322	(3, 3, 2, 2, 3)	66.4141
IPSO	0.99988963	0.828684	0.858026	0.9136462	0.648034	0.702276	5	3.59E-06	1.560466	(3, 3, 2, 4, 1)	6.23E-03
ABC1	0.99988962	0.828087	0.857805	0.704163	0.648146	0.914240	5	25.43	1.56046628	(3, 3, 2, 4, 1)	1.53E-02
ABC2	0.999889635809	0.827970	0.857875	0.914186	0.648355	0.703575	5	3.75E-04	1.56046628	(3, 3, 2, 4, 1)	9.70E-04
IA	0.9998893505	0.816624	0.868767	0.858749	0.710279	0.753429	18	4.04E-08	4.264770	(3, 3, 3, 3, 1)	2.59E-01
SAA	0.99988764	0.868116	0.807263	0.872862	0.712667	0.751034	40	7.30E-03	1.609289	(3, 3, 3, 3, 1)	1.7772
MICA	0.99988963	0.827642	0.857478	0.914197	0.649274	0.704092	5	4.43E-05	1.560466	(3, 3, 2, 4, 1)	6.23E-03
NAFSA	0.99988963	0.828322	0.857974	0.914221	0.647757	0.703007	5	1.55E-05	1.560466	(3, 3, 2, 4, 1)	6.23E-03
NGHS	0.99988960	0.829840	0.857989	0.913339	0.646745	0.703109	5	5.94E-06	1.56046629	(3, 3, 2, 4, 1)	3.34E-02
INGHS	0.9998896364	0.827985	0.857680	0.914156	0.648481	0.704865	5	1.89E-06	1.560466	(3, 3, 2, 4, 1)	4.35E-04
CS2	0.9998896319	0.827856	0.857626	0.914753	0.648217	0.702670	5	1.07E-10	1.5604662	(3, 3, 2, 4, 1)	4.51E-03
EBBO	0.9998896364	0.828061	0.858040	0.914149	0.647969	0.704205	5	1.45E-04	1.560466	(3, 3, 2, 4, 1)	4.35E-04
ZOA	0.999882951276930	0.82624	0.82654	0.86578	0.85484	0.71835	0.61786	5 60.231	0.2409	(3, 3, 3, 3, 1)	5.71
EZOA	**0.999889636879556**	0.81638	0.86887	0.85876	0.71038	0.75336	18	1.30e-04	4.2648	(3, 3, 3, 3, 1)	–

Table 27 Comparison of the optimal result for RRAP4 (gas turbine overspeed system) with existing literature

Algorithm	R_{sys}	r_1	r_2	r_3	r_4	Slack variables			(n_1, n_2, n_3, n_4)	$Imp(\%)$
						g_1	g_2	g_3		
PSO	0.99990474	0.92952331	0.81370356	0.8866375	0.8998718	37	11.5266	11.6447	(4, 6, 5, 5)	52.4193
IPSO	0.999954670	0.90163164	0.84997020	0.9482183	0.8881289	55	9.00E-06	24.0819	(5, 5, 4, 5)	1.03E-02
IA	0.99995467455	0.90158863	0.8881938	0.94816602	0.8499698	55	1.25E-04	15.363463	(5, 5, 4, 6)	2.34E-04
SAA	0.999945	0.895644	0.885878	0.912184	0.887785	50	9.38E-01	28.8037	(5, 5, 5, 5)	17.5903
MICA	0.999954673	0.901489	0.850035	0.948130	0.888238	55	2.14E-03	24.8018	(5, 5, 4, 5)	3.65E-03
NMDE	0.999954670	0.9016148	0.8499211	0.9481414	0.8882229	55	1.06E-05	24.801883	(5, 6, 4, 5)	1.03E-02
INGHS	0.999954674	0.901556583	0.888243885	0.948111097	0.84998174	55	5.05E-05	24.801882722	(5, 6, 4, 5)	1.45E-03
DE	0.99954670	0.901615	0.849921	0.948141	0.888222	55	1.01E-05	24.8018	(5, 6, 4, 5)	90.0010
TS-DE	0.9999546746081	0.901615	0.849921	0.948141	0.888223	55	1.90E-04	24.80188272	(5, 6, 4, 5)	1.06E-05
CS2	0.999954674	0.90159808	0.888226184	0.94810186	0.84998078	55	8.82E-10	15.36346	(5, 5, 4, 6)	1.45E-03
EBBO	0.999954674	0.90156292	0.88822494	0.94815595	0.849952895	55	2.70E-05	15.363463	(5, 5, 4, 6)	1.45E-03
INNA	0.99995467463	0.905888	0.8882576	0.9481337	0.8498978	55	5.27E-09	15.363463	(5, 5, 4, 6)	5.73E-05
ZOA	0.999954671417218	0.89967	0.88531	0.91584	0.88556	50	2.72E-03	24.8037	(5, 5, 5, 5)	7.15E-03
EZOA	**0.999954674655967**	0.90153	0.8499	0.94814	0.88829	55	5.03E-04	24.8019	(5, 6, 4, 5)	–

Table 28 Statistical analysis of RRAP problems

Problem	Algorithm	Average	Standard deviation	Best	Worst	Rank
RRAP1	EZOA	0.930970863388110	2.45E-03	0.931682278622065	0.928487371752721	**1**
	ZOA	0.926472825230163	6.83E-03	0.931679274315023	0.912106195280438	2
RRAP2	EZOA	0.999984989514011	2.05E-06	0.999986337729189	0.999979768842379	**1**
	ZOA	0.999976377003113	1.10E-05	0.999983844949936	0.999923245983722	2
RRAP3	EZOA	0.999888189294089	1.29E-05	0.999889636879556	0.99865227216082	**1**
	ZOA	0.999841798880036	5.09E-05	0.999882951276930	0.999683052383139	2
RRAP4	EZOA	0.999947278188393	7.78E-06	0.999954674655967	0.999907434440473	**1**
	ZOA	0.999886231700715	7.60E-05	0.999954671417218	0.999566553523182	2

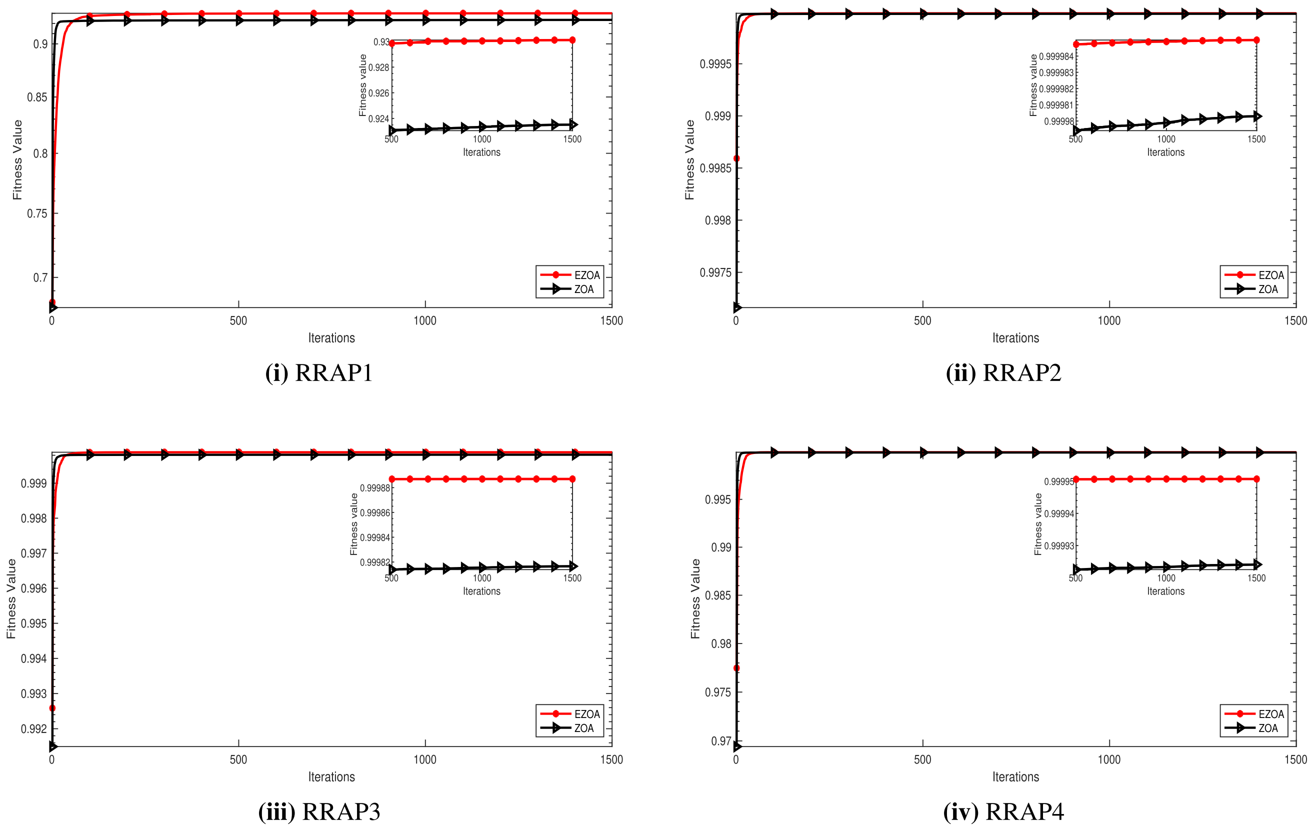

Fig. 24 Convergence curve of EZOA for RRAP problems

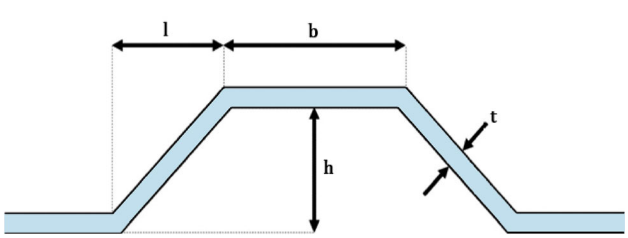

Fig. 25 Configurational diagran of corrugated bulkhead design case

Table 29 Comparison of optimal results for BD case

Algorithm	Optimum variables				Optimum cost
	y_1	y_2	y_3	y_4	
EZOA	57.6923	34.1476	57.6923	1.05	**6.8429**
ZOA	57.1224	34.1456	57.6977	1.0501	6.8436
WOA	29.1592	36.7564	57.6456	1.05	6.8881
BOA	56.9413	35.8543	53.5847	1.05926	7.082
HHO	57.2756	33.2008	51.9266	1.05	6.8748
TSA	56.5725	33.7820	55.5939	1.05386	6.8559
FOX	65.0228	41.4308	91.1410	1.57181	7.3446
RSO	19.2015	29.8209	39.9074	1.06494	7.5490
ALO	55.8542	34.1326	57.6891	1.05	6.8431
SOA	0	38.2307	58.095	1.05876	6.8586
STOA	0	37.9984	61.7404	1.11364	6.8606

Table 30 Statistical comparison outcomes of algorithms for BD case

Algorithm	Mean	Best	Worst	Standard deviation
EZOA	**6.8429**	**6.8429**	**6.842958010**	**1.37E-11**
ZOA	6.8793	6.8436	7.087668362	7.31E-02
WOA	7.4695	6.8881	9.015888699	7.00E-01
BOA	7.5994	7.0820	8.510571467	3.59E-01
HHO	7.0644	6.8748	7.964090674	2.61E-01
TSA	6.9242	6.8559	7.359054208	1.26E-01
FOX	9.5095	7.3446	10.96985167	1.07E+00
RSO	10.9565	7.5490	18.01109000	2.99E+00
ALO	6.8917	6.8431	7.148598755	6.98E-02
SOA	7.6237	6.8586	8.294008301	6.98E-02
STOA	7.5983	6.8606	8.315121533	7.02E-01

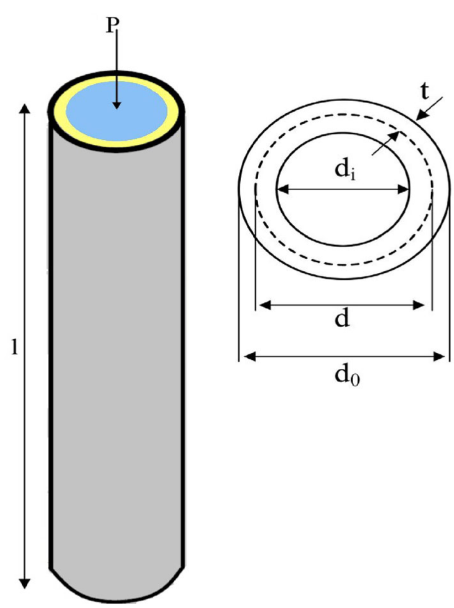

Fig. 26 Configurational diagran of tubular column design case

Table 31 Comparison of optimal results for TCD case

Algorithm	Optimum variables		Optimum cost
	y_1	y_2	
EZOA	5.4522	0.29163	**26.48636147**
ZOA	5.4524	0.29162	26.48636510
WOA	5.3653	0.30591	26.48893571
BOA	5.6681	0.28161	26.50760320
HHO	5.4540	0.29153	26.48751483
TSA	5.4538	0.29167	26.48798450
FOX	6.2646	0.25381	26.48637385
RSO	7.0588	0.22692	26.76573023
ALO	5.4522	0.29163	26.48636180
SOA	5.4536	0.29158	26.48886049
STOA	5.4509	0.29257	26.4902648

Bold value indicate the optimal results achieved by the algorithms

Table 32 Statistical comparison outcomes of algorithms for TCD case

Algorithm	Mean	Best	Worst	Standard deviation
EZOA	**26.48636147**	**26.48636147**	**26.48636147**	**3.65E-15**
ZOA	26.48672246	26.48636510	26.48748855	3.28E-04
WOA	26.79142425	26.48893571	27.45925039	3.02E-01
BOA	26.72432208	26.50760320	27.07467119	1.42E-01
HHO	26.52487532	26.48751483	26.59884171	2.87E-02
TSA	26.50134387	26.48798450	26.51816100	7.65E-03
FOX	26.64679417	26.48637385	28.11144597	4.31E-01
RSO	28.40947842	26.76573023	30.45130070	1.19E+00
ALO	26.48636236	26.48636180	26.48636328	4.37E-07
SOA	26.51274935	26.48886049	26.53909978	1.15E-02
STOA	26.51378867	26.49026480	26.53489020	1.25E-02

Bold values indicate the optimal results achieved by the algorithms

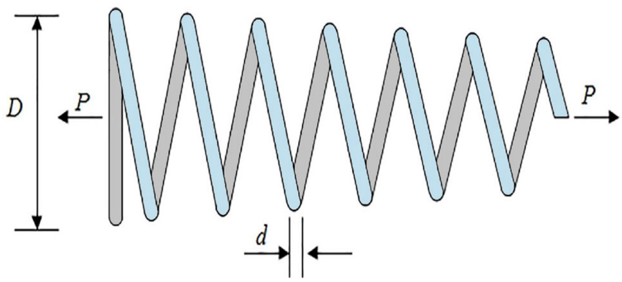

Fig. 27 Configurational diagran of spring design case

Table 33 Comparison of optimal results for tension/compression spring design case

Algorithm	Optimum variables			Optimum cost
	Y_1	Y_2	Y_3	
EZOA	0.05	0.317261	14.0514	**0.012666**
ZOA	0.053406	0.39942	9.1652	0.012719
WOA	0.059855	0.58686	4.5587	0.012774
BOA	0.059282	0.55889	5.1799	0.012859
HHO	0.057954	0.52702	5.5322	0.012929
TSA	0.053482	0.4007	9.2106	0.012715
FOX	0.0512726	0.346712	11.9267	0.012676
RSO	0.053335	0.39661	9.3535	0.012809
ALO	0.055251	0.4486	7.4095	0.012694
SOA	0.0501227	0.319421	13.9702	0.012690
STOA	0.0505138	0.328677	13.2586	0.012733

Bold values indicate the optimal results achieved by the algorithms

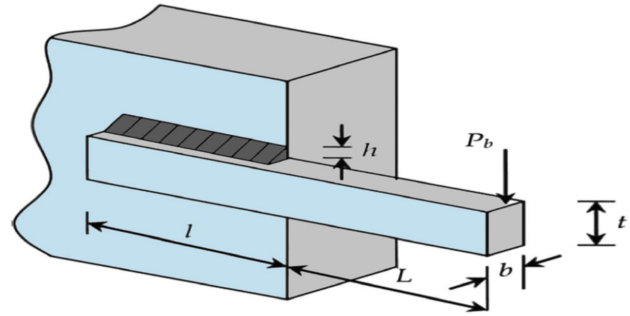

Fig. 28 Configuration diagram of welded beam design case

$$
\begin{aligned}
&\text{Consider } Z = [Z_1,\ Z_2,\ Z_3] = [d,\ D,\ N]\\
&\text{Minimize } F(Z) = (Z_3 + 2)Z_2Z_1^2,\\
&\text{Subject to : } G_1(Z) = 1 - \frac{Z_2^3Z_3}{71785Z_1^4} \leq 0,\\
&G_2(Z) = \frac{4Z_2^2 - Z_1Z_2}{12566(Z_2Z_1^3 - Z_1^4)} + \frac{1}{5108Z_1^2} \leq 0,\\
&G_3(Z) = 1 - \frac{140.45Z_1}{Z_2^2Z_3} \leq 0,\\
&G_4(Z) = \frac{Z_1 + Z_2}{1.5} - 1 \leq 0,\\
&\text{with bounds: } 0.05 \leq Z_1 \leq 2.00,\\
&0.25 \leq Z_2 \leq 1.30,\\
&2.00 \leq Z_3 \leq 15.0.
\end{aligned} \tag{47}
$$

Table 34 Statistical comparison outcomes of algorithms for SD case

Algorithm	Mean	Best	Worst	Standard deviation
EZOA	**0.012701**	**0.012666**	**0.012779768**	**3.09E-05**
ZOA	0.013380	0.012719	0.015513549	7.02E-04
WOA	0.014166	0.012774	0.016035845	9.57E-04
BOA	0.014595	0.012859	0.019224249	1.66E-03
HHO	0.013637	0.012929	0.015667116	5.93E-04
TSA	0.012890	0.012715	0.014027554	2.87E-04
FOX	0.012816	0.012676	0.013529018	2.29E-04
RSO	0.014892	0.012809	0.018384	1.71E-03
ALO	0.013317	0.012694	0.017764731	1.31E-03
SOA	0.012776	0.012690	0.012855007	3.45E-05
STOA	0.012815	0.012733	0.013427574	1.52E-04

Bold values indicate the optimal results achieved by the algorithms

Table 33 shows the optimal results of the proposed EZOA algorithm and competing algorithms for the SD case. The EZOA algorithm produces an optimal value of $F(Z) = 0.012666$ with decision variables $(Z_1, Z_2, Z_3) = (0.05, 0.317261, 14.0514)$. Table 34 summarizes the statistical results of the case. The mean value obtained across various runs by EZOA is 0.012701, which is least among the algorithms tested. Additionally, the standard deviation is the lowest at $3.09E-05$ for EZOA, indicating its significant superiority in delivering consistent results.

8.4 P4: welded beam design case (WBD)

The welded beam design (WBD) problem represents a fundamental challenge in structural engineering, focusing on minimizing the fabrication costs of a welded beam as depicted in Fig. 28. The design process involves four key design variables: width of the weld (Z_1), length of connected part (Z_2), beam height (Z_3), and beam thickness (Z_4). The constraints that must be adhered during the optimization process includes bending stress (θ), end deflection (δ), shear stresss (τ), and buckling load (P_b). The mathematical depiction of WBD case is given as follows:

$$
\begin{aligned}
&\text{Consider } Z = [Z_1,\ Z_2,\ Z_3,\ Z_4] = [h,\ l,\ t,\ b]\\
&\text{Min} \quad F(Z) = 1.10471\ Z_1^2 Z_2 + 0.04811\ Z_3 Z_4 (14 + Z_2)\\
&\text{Subject to:} \quad G_1(Z) = \tau(Z) - \tau_{\max} \le 0,\\
&G_2(Z) = \sigma(Z) - \sigma_{\max} \le 0,\\
&G_3(Z) = \delta(Z) - \delta_{\max} \le 0,\\
&G_4(Z) = Z_1 - Z_4 \le 0,\\
&G_5(Z) = P - P_b(Z) \le 0,\\
&G_6(Z) = 0.125 - Z_1 \le 0,\\
&G_7(Z) = 0.10471\ Z_1^2 + 0.04811\\
&Z_3 Z_4 (14.0 + Z_2) - 5.0 \le 0\\
&\text{with bounds} \quad 0.1 \le Z_1, Z_4 \le 2, 0.1 \le Z_2, Z_3 \le 10\\
&\text{where} \quad \tau(Z) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{Z^2}{2R} + (\tau'')^2},\\
&\tau' = \frac{P}{\sqrt{2} Z_1 Z_2}, \quad \tau'' = \frac{MR}{J}\\
&R = \sqrt{\frac{Z_2^2}{4} + \frac{(Z_1 + Z_3)^2}{4}},\\
&J = 2\left\{\sqrt{2} Z_1 Z_2 \left[\frac{Z^2}{4} + \frac{(Z_1 + Z_3)^2}{4}\right]\right\},\\
&M = P\left(L + \frac{Z_2}{2}\right),\\
&\sigma(Z) = \frac{6PL}{Z^4 Z_2^3},\\
&\delta(Z) = \frac{6PL^3}{E Z_2^3 Z_4},\\
&P_b(Z) = \frac{4.013E\sqrt{\frac{Z^2 Z_6^4}{36}}}{L^2}\left(1 - \frac{Z^3}{2L}\sqrt{\frac{E}{4G}}\right),\\
&P = 6000\ \text{lb}, \quad G = 12 \times 10^6\ \text{psi}\\
&\tau_{\max} = 13,600\ \text{psi}, \quad \sigma_{\max} = 30,000\ \text{psi},\\
&\delta_{\max} = 0.25\ \text{in.}\\
&E = 30 \times 10^{16}\ \text{psi}, L = 14\ \text{in.}
\end{aligned}
\tag{48}
$$

Table 35 Comparison of optimal results for WBD case

Algorithm	Optimum variables				Optimum weight
	y_1	y_2	y_3	y_4	
EZOA	0.20573	3.4705	9.0366	0.20573	**1.7248524609**
ZOA	0.17945	4.1422	9.0369	0.20573	1.724936886
WOA	0.57001	1.273	7.0053	0.60104	1.801150099
BOA	0.2233	3.7538	8.4669	0.2403	1.944563244
HHO	0.16095	4.8273	9.4812	0.20361	1.799196772
TSA	0.20564	3.4795	9.0505	0.20686	1.732190842
FOX	0.14623	5.4623	9.0286	0.20612	1.742945542
RSO	0.41808	1.3214	5.6834	0.59669	1.760812832
ALO	0.1605	4.8043	9.0366	0.20573	1.728008020
SOA	0.20484	3.4288	9.2055	0.20512	1.732481946
STOA	0.20562	3.4839	9.0459	0.20592	1.729569821

Bold value indicate the optimal results achieved by the algorithms

Table 36 Statistical comparison outcomes of algorithms for WBD case

Algorithm	Mean	Best	Worst	Standard deviation
EZOA	**1.7254**	**1.7248524609**	**1.7304205527**	**1.12E-03**
ZOA	1.7488	1.724936886	1.98751714	5.65E-02
WOA	2.346	1.801150099	3.550720704	5.70E-01
BOA	2.9947	1.944563244	4.465489821	6.18E-01
HHO	1.9171	1.799196772	2.084039028	8.38E-02
TSA	1.7448	1.732190842	1.755176669	6.45E-03
FOX	1.9852	1.742945542	2.261450298	1.45E-01
RSO	3.4008	1.760812832	6.922618114	1.31E+00
ALO	1.7770	1.72800802	1.897772411	5.58E-02
SOA	1.7406	1.732481946	1.761440011	7.55E-03
STOA	1.7426	1.729569821	1.765044221	9.82E-03

Bold values indicate the optimal results achieved by the algorithms

The results for the WBD problem are detailed in Table 35, highlighting the impressive performance of the EZOA algorithm in achieving optimal solutions with variable settings $(Z_1, Z_2, Z_3) = (0.20573, 3.4705, 9.0366, 0.20573)$. This configuration resulted in objective function value of 1.7248524609. In addition, the statistical analysis presented in Table 36 shows the superior performance of EZOA across all four metrics compared to other algorithms. In particular, EZOA reaches an average minimum cost of 1.7254 with a low standard deviation of 1.12E-03. These findings highlight the effectiveness of EZOA in providing quality solutions.

9 Conclusion

The study introduces EZOA, an enhanced version of the zebra optimization algorithm designed for complex optimization problems. Initially, the Levy flight strategy is integrated into ZOA's foraging behavior, allowing for a broader initial search of the solution space and mitigating the risk of premature convergence to sub-optimal solutions. In the later phase, LOBL strategy is applied to further improve the algorithm's performance by diversifying search directions. This approach helps to avoid local optima, thereby enhancing solution quality and achieving more effective convergence.

To assess EZOA's performance, firstly it is tested on classical benchmark functions on dimensions 10, 30, 50, and 100 and CEC-2019 functions. The simulation results are compared against seven recently developed meta-heuristic algorithms using metrics such as average performance, best results, and standard deviation. The execution time required by these algorithms to achieve optimal results is also obtained. To verify the superiority of proposed algorithim, it is then tested against some efficient champion algorithms from IEEE CEC competitions. To ensure the reliability of the simulation results, non-parametric statistical tests-specifically, Friedman's test, the Wilcoxon signed-rank test, and the Mann–Whitney U test-are conducted on the competing algorithms. Friedman's test reveals the superior performance of the proposed approach, ranking it first overall in each case, while the Wilcoxon signed-rank and Mann–Whitney tests confirm the statistical significance of EZOA's improvements. Additionally, convergence and boxplot analyses demonstrate the consitency in attaining the superior results. Further, the application of EZOA is evaluated on two additional test cases: benchmark mixed-integer RRAPs and engineering design optimization problems. The results obtained by EZOA for RRAPs showcases superior performance in all four cases, compared with those from several existing algorithms in the literature. Next, EZOA is also analyzed on four benchmark engineering problems and results are compared to ten efficient algorithm. The better statistical results obtained by EZOA among these algorithms indicated its robust nature. Hence, the thorough analysis of the results reveals that our proposed EZOA excels in handling various complex optimization problems.

In future work, EZOA could be further refined by incorporating innovative techniques aimed at effectively addressing multi-objective optimization problems. Exploring the application of EZOA in real-world contexts, such as resource allocation and supply chain management, is a promising area for future research. Further, hybridizing EZOA with other optimization techniques could enhance its capabilities and performance in diverse scenarios.

Author contributions Parul Punia contributed to developing the methodology, drafting the manuscript, and compiling the results. Amit Raj contributed to preparing the figures and tables included in the manuscript. Pawan Kumar contributed to data analysis, reviewing and editing of the manuscript, and data analysis. All authors reviewed and approved the final version of the manuscript.

Funding The second author sincerely thanks the Council of Scientific & Industrial Research (CSIR) for awarding the Junior Research Fellowship (JRF) under file number 09/1152(0024)/2020-EMR-1 and for their support, which made this research possible.

Data availibility This article includes the complete set of data generated during the study.

Declarations

Conflict of interest The authors have no competing interests to declare.

Ethical approval Not Applicable.

Consent for publication Not Applicable.

References

1. Marouani, H., Al-Mutiri, O.: Optimization of reliability-redundancy allocation problems: a review of the evolutionary algorithms. Comput. Mater. Contin. **71**(1), 537–571 (2022)
2. Liu, G.S.: A combination method for reliability-redundancy optimization. Eng. Optim. **38**(04), 485–499 (2006)
3. Kuo, W., Prasad, V.R.: An annotated overview of system-reliability optimization. IEEE Trans. Reliab. **49**(2), 176–187 (2000)
4. Tillman, F.A., Hwang, C.L., Kuo, W.: Optimization techniques for system reliability with redundancy—a review. IEEE Trans. Reliab. **26**(3), 148–155 (1977)
5. Punia, P., Raj, A., Kumar, P.: An enhanced beluga whale optimization algorithm for engineering optimization problems. J. Syst. Sci. Syst. Eng. (2024). https://doi.org/10.1007/s11518-024-5608-x

6. Gong, J., Karimzadeh Parizi, M.: GWMA: the parallel implementation of woodpecker mating algorithm on the GPU. J. Chinese Inst. Eng. **45**(6), 556–568 (2022)
7. Raj, A., Punia, P., Kumar, P.: A novel hybrid pelican-particle swarm optimization algorithm (HPPSO) for global optimization problem. Int. J. Syst. Assur. Eng. Manage. (2024). https://doi.org/10.1007/s13198-024-02386-9
8. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1998). https://doi.org/10.7551/mitpress/3927.001.0001
9. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. **3**(2), 82–102 (1999)
10. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008)
11. Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution-an updated survey. Swarm Evol. Comput. **27**, 1–30 (2016)
12. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. **29**, 17–35 (2016)
13. Zhang, Y., Wang, S., Ji, G.: A comprehensive survey on particle swarm optimization algorithm and its applications. Math. Probl. Eng. **2015**(1), 931256 (2015)
14. Yang, X.S., Hossein Gandomi, A.: Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**(5), 464–483 (2012)
15. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Comput. Intell. Mag. **1**(4), 28–39 (2006)
16. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Software **95**, 51–67 (2016)
17. Zhang, J., Li, H., Parizi, M.K.: HWMWOA: a hybrid WMA-WOA algorithm with adaptive cauchy mutation for global optimization and data classification. Int. J. Inf. Technol. Decis. Mak. **22**(04), 1195–1252 (2023)
18. Abdollahzadeh, B., Khodadadi, N., Barshandeh, S., Trojovský, P., Gharehchopogh, F.S., El-kenawy, E.S.M., Mirjalili, S.: Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning. Cluster Comput. **27**(4), 1–49 (2024)
19. Ghambari, S., Rahati, A.: An improved artificial bee colony algorithm and its application to reliability optimization problems. Appl. Soft Comput. **62**, 736–767 (2018)
20. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: algorithm and applications. Future Gener. Comput. Syst. **97**, 849–872 (2019)
21. Khare, A., Kakandikar, G.M., Kulkarni, O.K.: An insight review on jellyfish optimization algorithm and its application in engineering. Rev. Comput. Stud. **9**(1), 31–40 (2022)
22. Chen, D., Liu, J., Yao, C., Zhang, Z., Du, X.: Multi-strategy improved salp swarm algorithm and its application in reliability optimization. Math. Biosci. Eng. **19**(5), 5269–5292 (2022)
23. Abdollahzadeh, B., Gharehchopogh, F.S., Khodadadi, N., Mirjalili, S.: Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv. Eng. Software. **174**, 103282 (2022)
24. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. **43**(3), 303–315 (2011)
25. Faridmehr, I., Nehdi, M.L., Davoudkhani, I.F., Poolad, A.: Mountaineering team-based optimization: a novel human-based metaheuristic algorithm. Mathematics **11**(5), 1273 (2023)
26. Zhong, M., Wen, J., Ma, J., Cui, H., Zhang, Q., Parizi, M.K.: A hierarchical multi-leadership sine cosine algorithm to dissolving global optimization and data classification: the COVID-19 case study. Comput. Biol. Med. **164**(10), 107212 (2023)
27. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. **96**, 120–133 (2016)
28. Parizi, M.K., Keynia, F., Bardsiri, A.K.: HSCWMA: a new hybrid SCA-WMA algorithm for solving optimization problems. Int. J. Inf. Technol. Decis. Making. **20**(02), 775–808 (2021)
29. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Advances in Swarm Intelligence: First International Conference. ICSI, pp. 355–364. Springer, Berlin Heidelberg (2010)
30. Al-Betar, M.A., Alyasseri, Z.A.A., Awadallah, M.A., Abu Doush, I.: Coronavirus herd immunity optimizer (CHIO). Neural Comput. Appl. **33**, 5011–5042 (2021)
31. Abdollahzadeh, B., Gharehchopogh, F.S., Mirjalili, S.: African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput. Ind. Eng. **158**, 107408 (2021)
32. Karimzadeh Parizi, M., Keynia, F., Khatibi Bardsiri, A.: Woodpecker mating algorithm (WMA): a nature-inspired algorithm for solving optimization problems. Int. J. Nonlinear Anal. Appl. **11**(1), 137–157 (2020)
33. Abdollahzadeh, B., Soleimanian Gharehchopogh, F., Mirjalili, S.: Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int. J. Intell. Syst. **36**(10), 5887–5958 (2021)
34. Benmamoun, Z., Khlie, K., Bektemyssova, G., Dehghani, M., Gherabi, Y.: Bobcat optimization algorithm: an effective bio-inspired metaheuristic algorithm for solving supply chain optimization problems. Sci. Rep. **14**(1), 20099 (2024)
35. Karimzadeh Parizi, M., Keynia, F., Khatibi Bardsiri, A.: Woodpecker mating algorithm for optimal economic load dispatch in a power system with conventional generators. Int. J. Indus. Electron. Control Opt. **4**(2), 221–234 (2021)
36. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
37. Punia, P., Raj, A., Kumar, P.: ILO: an improved Lemur optimizer for global optimization. Arab. J. Sci. Eng. **49**, 13075–13098 (2024)
38. Sharma, T.K.: Enhanced butterfly optimization algorithm for reliability optimization problems. J. Ambient Intell. Hum. Comput. **12**(7), 7595–7619 (2021)
39. Yu, K., Wang, X., Wang, Z.: An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. J. Intell. Manuf. **27**, 831–843 (2016)
40. Sahoo, S.K., Saha, A.K., Nama, S., Masdari, M.: An improved moth flame optimization algorithm based on modified dynamic opposite learning strategy. Artif. Intell. Rev. **56**(4), 2811–2869 (2023)
41. Valian, E., Tavakoli, S., Mohanna, S., Haghi, A.: Improved cuckoo search for reliability optimization problems. Comput. Ind. Eng. **64**(1), 459–468 (2013)
42. Tubishat, M., Idris, N., Shuib, L., Abushariah, M.A., Mirjalili, S.: Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Syst. Appl. **145**, 113122 (2020)
43. Hu, G., Wang, J., Li, Y., Yang, M., Zheng, J.: An enhanced hybrid seagull optimization algorithm with its application in engineering optimization. Eng. Comput. **39**(2), 1653–1696 (2023)
44. Karimzadeh Parizi, M., Keynia, F.: OWMA: An improved self-regulatory woodpecker mating algorithm using opposition-based learning and allocation of local memory for solving optimization problems. J. Intell. Fuzzy Syst. **40**(1), 919–946 (2021)
45. Kundu, T., Garg, H.: A hybrid TLNNABC algorithm for reliability optimization and engineering design problems. Eng. Comput. **38**(6), 5251–5295 (2022)

46. Jia, H., Lang, C.: Salp swarm algorithm with crossover scheme and Lévy flight for global optimization. J. Intell. Fuzzy Syst. **40**(5), 9277–9288 (2021)
47. Marouani, H.: Optimization for the redundancy allocation problem of reliability using an improved particle swarm optimization algorithm. J. Optim. **2021**(1), 6385713 (2021)
48. Trojovská, E., Dehghani, M., Trojovský, P.: Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm. IEEE Access **10**, 49445–49473 (2022)
49. Houssein, E.H., Saad, M.R., Hashim, F.A., Shaban, H., Hassaballah, M.: Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng. Appl. Artif. Intell. **94**, 103731 (2020)
50. Tizhoosh, H. R.: Opposition-based learning: a new scheme for machine intelligence. In International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), vol. 1, pp. 695-701 IEEE (2005)
51. Wong, W. K., Ming, C. I.: A review on metaheuristic algorithms: recent trends, benchmarking and applications. In 2019 7th International Conference on Smart Computing & Communications (ICSCC), pp. 1-5. IEEE (2019)
52. Kumar, A., Wu, G., Ali, M.Z., Mallipeddi, R., Suganthan, P.N., Das, S.: A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evol. Comput. **56**, 100693 (2020)
53. Ardakan, M.A., Hamadani, A.Z.: Reliability-redundancy allocation problem with cold-standby redundancy strategy. Simul. Pract. Theory **42**, 107–118 (2014)
54. Ardakan, L.D., Mariani, V.C., dos Santos Coelho, L.: Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization. Expert Syst. Appl. **40**(9), 3794–3802 (2013)
55. Yokota, T., Gen, M., Li, Y.X.: Genetic algorithm for non-linear mixed integer programming problems and its applications. Comput. Ind. Eng. **30**(4), 905–917 (1996)
56. Wu, P., Gao, L., Zou, D., Li, S.: An improved particle swarm optimization algorithm for reliability problems. ISA Trans. **50**(1), 71–81 (2011)
57. Ouyang, H.B., Gao, L.Q., Li, S., Kong, X.Y.: Improved novel global harmony search with a new relaxation method for reliability optimization problems. Inf. Sci. **305**, 14–55 (2015)
58. Gen, M., Yun, Y.: Soft computing approach for reliability optimization: state-of-the-art survey. Reliab. Eng. Syst. Saf. **91**(9), 1008–1026 (2006)
59. Zou, D., Gao, L., Wu, J., Li, S., Li, Y.: A novel global harmony search algorithm for reliability problems. Comput. Ind. Eng. **58**(2), 307–316 (2010)
60. Valian, E., Valian, E.: A cuckoo search algorithm by Lévy flights for solving reliability redundancy allocation problems. Eng. Optim. **45**(11), 1273–1286 (2013)
61. Hsieh, Y.C., You, P.S.: An effective immune based two-phase approach for the optimal reliability-redundancy allocation problem. Appl. Math. Comput. **218**(4), 1297–1307 (2011)
62. Garg, H.: An efficient biogeography based optimization algorithm for solving reliability optimization problems. Swarm Evol. Comput. **24**, 1–10 (2015)
63. Arora, S., Singh, S.: Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput. **23**, 715–734 (2019)
64. Kaur, S., Awasthi, L.K., Sangal, A.L., Dhiman, G.: Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng. Appl. Artif. Intell. **90**, 103541 (2020)
65. Dhiman, G., Garg, M., Nagar, A., Kumar, V., Dehghani, M.: A novel algorithm for global optimization: rat swarm optimizer. J. Ambient Intell. Hum. Comput. **12**, 8457–8482 (2021)
66. Trojovský, P., Dehghani, M.: Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications. Sensors **22**(3), 855 (2022)
67. Mohammed, H., Rashid, T.: FOX: A FOX-inspired optimization algorithm. Appl. Intell. **53**(1), 1030–1050 (2023)
68. Mirjalili, S.: The ant lion optimizer. Adv. Eng. Software **83**, 80–98 (2015)
69. Dhiman, G., Kumar, V.: Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. Knowl.-Based Syst. **165**, 169–196 (2019)
70. Dhiman, G., Kaur, A.: STOA: a bio-inspired based optimization algorithm for industrial engineering problems. Eng. Appl. Artif. Intell. **82**, 148–174 (2019)
71. Huang, C.L.: A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems. Reliab. Eng. Syst. Saf. **142**, 221–230 (2015)
72. He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng. Appl. Artif. Intell. **20**(1), 89–99 (2007)
73. Liu, Y., Qin, G.: A modified particle swarm optimization algorithm for reliability redundancy optimization problem. J. Comput. **9**(9), 2124–2131 (2014)
74. Kim, H.G., Bae, C.O., Park, D.J.: Reliability-redundancy optimization using simulated annealing algorithms. J. Qual. Maint. Eng. **12**(4), 354–363 (2006)
75. Yeh, W.C., Hsieh, T.J.: Solving reliability redundancy allocation problems using an artificial bee colony algorithm. Comput. Oper. Res. **38**(11), 1465–1473 (2011)
76. Garg, H., Rani, M., Sharma, S.P.: An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique. Comput. Oper. Res. **40**(12), 2961–2969 (2013)
77. Garg, H.: An approach for solving constrained reliability-redundancy allocation problems using cuckoo search algorithm. Beni-Suef Univ. J. Basic Appl. Sci. **4**(1), 14–25 (2015)
78. Kanagaraj, G., Ponnambalam, S.G., Jawahar, N.: A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems. Comput. Ind. Eng. **66**(4), 1115–1124 (2013)
79. He, Q., Hu, X., Ren, H., Zhang, H.: A novel artificial fish swarm algorithm for solving large-scale reliability-redundancy application problem. ISA Trans. **59**, 105–113 (2015)
80. Liu, Y., Qin, G.: A DE algorithm combined with levy flight for reliability redundancy allocation problems. Int. J. Hybrid Inf. Technol. **8**(5), 113–118 (2015)
81. Zou, D., Liu, H., Gao, L., Li, S.: A novel modified differential evolution algorithm for constrained optimization problems. Comput. Math. Appl. **61**(6), 1608–1623 (2011)
82. Liu, Y., Qin, G.: A hybrid TS-DE algorithm for reliability redundancy optimization problem. J. Comput. **9**(9), 2050–2057 (2014)
83. Kundu, T., Garg, H.: INNA: an improved neural network algorithm for solving reliability optimization problems. Neural Comput. Appl. **34**(23), 20865–20898 (2022)

Parul Punia is currently a PhD. scholar in the Department of Mathematics at the Central University of Haryana, Mahendergarh, India. She received her bachelor's degree in science and master's degree in mathematics from MCM DAV College for Women, Panjab University, Chandigarh. Her research interests include computational mathematics, mathematical modeling, nature-inspired optimization.

Amit Raj is a research scholar in the Department of Mathematics at the Central University of Haryana, Mahendergarh, India. He did his graduation and post-graduation from Kirrori Mal College, University of Delhi. His area of interest is optimization.

Pawan Kumar is currently an assistant professor, at the Department of Mathematics, Central University of Haryana, Jant-Pali, Mahendergarh, Haryana, India-123031. He did his Master's degree in mathematics from Kurukshetra University, Kurukshetra. He did his Ph.D. degree from the Department of Mathematics, National Institute of Technology, Kurukshetra, Haryana. He is currently working in the field of nature-inspired optimization.