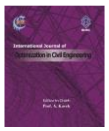
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2025; 15(4):595-620



ENHANCED PRAIRIE DOG METAHEURISTIC OPTIMIZATION ALGORITHM FOR ENGINEERING OPTIMIZATION PROBLEMS

T. PayamiFar, R. Sojoudizadeh[*, †], H. Azizian, and L. Rahimi
Department of Civil Engineering, Mah.C., Islamic Azad University, Mahabad, Iran

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

ABSTRACT

This paper presents an Enhanced Prairie Dog Optimization (IPDO) algorithm for solving complex engineering optimization problems. The proposed improvement integrates Lévy flight dynamics into the original PDO framework to enhance exploration-exploitation balance and accelerate convergence. The performance of IPDO is evaluated against seven established metaheuristics across four challenging civil engineering applications: (1) discrete sizing optimization of a 120-bar truss, (2) structural reliability analysis of a cantilever tube, (3) cost optimization of reinforced concrete beams, and (4) hyperparameter tuning of a Support Vector Machine (SVM) for shear strength prediction of steel fiber-reinforced concrete. Experimental results demonstrate that IPDO consistently achieves superior solution quality, robustness, and convergence speed. Notably, in SVM hyperparameter optimization, IPDO attained the lowest mean squared error (1.4881) with zero variance across runs, outperforming all competitors. The algorithm also proved highly effective in structural design and reliability problems, offering a reliable and efficient tool for real-world engineering optimization.

Keywords: Engineering optimization; Improved Prairie Dog Optimization; Metaheuristic algorithm; Truss structures.

Received: 14 October 2025; Accepted: 3 December 2025

1. INTRODUCTION

Optimization refers to the process of identifying the best possible solution for a given system by evaluating all potential configurations to either maximize or minimize the desired output. As problem complexity has escalated in recent decades, the need for innovative optimization techniques has become increasingly critical [1-3]. Conventional mathematical methods previously employed for such problems were predominantly deterministic but suffered from

[*]Corresponding author: Department of Civil Engineering, Mah.C., Islamic Azad University, Mahabad, Iran
[†]E-mail address: Reza.SojoudiZadeh@iau.ac.ir (R. Sojoudizadeh)

a significant limitation: convergence to local optima. This drawback rendered them inefficient for real-world optimization challenges, thereby fueling a growing interest in stochastic optimization techniques over the last years [4-6].

Many real-world optimization problems—such as those encountered in engineering [7], image processing [8], bioinformatics [9] and hyperparameter tuning in artificial intelligence [10-12]—exhibit high non-linearity and non-convexity. These characteristics arise from their complex constraints and numerous design variables, making them inherently challenging to solve. A major complication is the prevalence of multiple local optima, which often traps traditional methods in suboptimal solutions. Furthermore, achieving a globally optimal solution is never guaranteed [13, 14]. Due to these inherent difficulties, researchers are compelled to explore and develop more robust and efficient procedures to address such complex problems effectively.

To enhance solution quality, researchers have developed novel algorithms and refined existing methods. The metaheuristics community has implemented effective search strategies for global optimization [15-20]. Real-world optimization problems often feature exponentially growing search spaces and highly multimodal landscapes, causing conventional methods to yield suboptimal solutions. This challenge has spurred the development of numerous metaheuristic algorithms over recent decades [21-22], demonstrating robust performance across complex problems.

Recent metaheuristics incorporate diverse nature-inspired search strategies, mimicking swarm intelligence, biological, ethological, and physical principles [17]. Notable examples like Genetic Algorithms [23], Ant Colony Optimization [24], Particle Swarm Optimization (PSO) [25], and Archimedes Optimization Algorithm [26] have gained recognition across scientific disciplines. Their gradient-free mechanism, flexibility, and ability to escape local optima have led to widespread theoretical and practical applications [17]. By eliminating derivative calculations, these methods reduce computational costs while maintaining adaptability for diverse problem domains. Metaheuristic algorithms are broadly classified into two types: single-solution and population-based approaches, with the latter demonstrating superior search space exploration and global optimization capabilities [18]. Population-based algorithms can be further categorized by their inspiration sources:

- Swarm Intelligence (SI): Mimics collective behaviors in nature.
- Evolutionary Algorithms (EA): Based on biological evolution principles.
- Natural Phenomenon (NP): Inspired by physical/chemical laws.

Some algorithms draw from human behavior without fitting strictly into SI or EA categories. SI algorithms specifically model self-organized group behaviors observed in various natural systems, while EAs employ stochastic optimization techniques based on evolutionary mechanisms [11].

Research indicates that despite the high computational time of population-based algorithms, they perform more effectively when dealing with challenging nonlinear and complex engineering problems. Therefore, in recent years, new variants of these algorithms, such as Harris Howk Optimization [17], Gradient-based optimizer [21], Archimedes Optimization Algorithm [26], and etc. have been introduced and successfully applied to solve engineering problems. Examples include the use of modern metaheuristic algorithms for

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

structural design, optimizing parameters of machine learning algorithms, assessing structural reliability, energy optimization, hydraulic structure design, and more. However, according to the No Free Lunch (NFL) theorem [17], no single algorithm is suitable for all problems, as an algorithm may perform well on some problems but not on others. This provides the motivation for proposing new algorithms or improving their performance through hybridization, which obliges researchers to evaluate new algorithms on the specific problem under investigation.

Recently, an algorithm called the Improved Prairie Dogs Optimization (PDO) [28] using Lévy Flight Theory, which is a population-based metaheuristic algorithm, was introduced for solving the optimization problem of steel frames [29]. This algorithm demonstrated significant results in solving the aforementioned problem compared to other metaheuristic algorithms, providing the primary motivation for the present research to evaluate its efficiency in solving complex engineering problems. Accordingly, the aim of the present study is to evaluate the performance of the IPDO algorithm in solving several challenging optimization problems and to compare its performance with other prominent metaheuristic algorithms.

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

2. PRAIRIE DOG ALGORITHM

Prairie dogs (Cynomys genus) are herbivorous, burrowing rodents primarily inhabiting the grasslands of North America [28]. They exhibit distinct physical adaptations, such as powerful forelimbs and long-clawed toes, which have informed the design of the PDO algorithm. These animals construct complex subterranean colonies with specialized chambers, playing a significant ecological role in soil aeration and water infiltration. Their social structure is organized into cohesive family groups called coteries, which maintain stability through specific behavioral interactions. Notably, prairie dogs possess a sophisticated communication system capable of conveying detailed information about predators and other threats through differentiated vocalizations. These unique collective behaviors make them an attractive model for developing swarm-based computational algorithms [29].

2.1. Mathematical Description

This section outlines the core mathematical model and search methodology of the PDO algorithm. As a population-based metaheuristic, PDO initializes its search process by randomly generating the positions of its candidate solutions, referred to as prairie dogs, across the problem's search space. Each prairie dog's location is mathematically represented as a vector within a d-dimensional hyperplane, corresponding to the number of variables in the optimization problem (Equation 1). The initial distribution of these agents across different social groups (colonies) and their individual coordinates is accomplished using a uniform random distribution, ensuring unbiased coverage of the solution space at the start of the optimization process [29].

$$CT = \begin{bmatrix} CT_{1,1} & CT_{1,2} & \cdots & CT_{1,d-1} & \cdots & CT_{1,d} \\ CT_{2,1} & CT_{2,2} & \cdots & CT_{2,d-1} & \cdots & CT_{2,d} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ CT_{i,1} & CT_{i,2} & \cdots & CT_{i,d-1} & \cdots & CT_{i,d} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ CT_{m,1} & CT_{m,2} & \cdots & CT_{m,d-1} & \cdots & CT_{m,d} \end{bmatrix}, \quad \begin{cases} i = 1,2,\dots,m. \\ j = 1,2,\dots,d. \end{cases} \tag{1}$$

Let CTi,j represent the value of the j-th dimension for the i-th coterie in the colony. Accordingly, the location of every member within a coterie can be expressed by the equation below.

$$PD = \begin{bmatrix} PD_{1,1} & PD_{1,2} & \cdots & PD_{1,d-1} & \cdots & PD_{1,d} \\ PD_{2,1} & PD_{2,2} & \cdots & PD_{2,d-1} & \cdots & PD_{2,d} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ PD_{i,1} & PD_{i,2} & \cdots & PD_{i,d-1} & \cdots & PD_{i,d} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ PD_{m,1} & PD_{m,2} & \cdots & PD_{m,d-1} & \cdots & PD_{m,d} \end{bmatrix}, \quad \begin{cases} i = 1,2,\dots,m. \\ j = 1,2,\dots,d. \end{cases} \tag{2}$$

Here, $PD_{i.j}$ denotes the j-th coordinate of the i-th prairie dog within a coterie. Both collective (coterie) and individual positions are initialized uniformly, as described previously. The fitness of each prairie dog's location is evaluated by passing its solution vector into the predefined objective function, and the resulting values are stored as follows [28]:

$$f(PD) = \begin{bmatrix} f(PD_{1,1} & PD_{1,2} & \cdots & PD_{1,d-1} & \cdots & PD_{1,d}) \\ f(PD_{2,1} & PD_{2,2} & \cdots & PD_{2,d-1} & \cdots & PD_{2,d}) \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ f(PD_{i,1} & PD_{i,2} & \cdots & PD_{i,d-1} & \cdots & PD_{i,d}) \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ f(PD_{m,1} & PD_{m,2} & \cdots & PD_{m,d-1} & \cdots & PD_{m,d}) \end{bmatrix}, \quad \begin{cases} i = 1,2,\dots,m. \\ j = 1,2,\dots,d. \end{cases} \tag{3}$$

The fitness value assigned to a prairie dog's location quantifies the quality of the available food sources, the potential for successful burrow construction, and the effectiveness of its anti-predator response. A core collective strategy is the exploration for new food sources, which is modeled as follows:

$$PD_{i+1,j+1} = GBest_{i,j} - eCBest_{i,j} \times \rho - CPD_{i,j} \times Levy(n) \; \forall \; iter < \frac{Max_{iter}}{4} \tag{4}$$

Here, $PD_{i+1.j+1}$ denotes the updated position vector, $GBest_{i.j}$ is the global best solution vector, and $eCBest_{i.j}$ represents the influence of the currently best-known solution. The term $Levy(n)$ corresponds to a step size drawn from the Lévy distribution, while *iter* and *Maxiter* indicate the current and maximum iteration counts, respectively.

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

$$PD_{i+1,j+1} = GBest_{i,j} \times rPD \times DS \times Levy(n) \; \forall < \frac{Max_{iter}}{4} \leq iter < \frac{Max_{iter}}{2} \quad (5)$$

In this formulation, $PD_{i+1.j+1}$ represents the updated position vector, $GBest_{i.j}$ is the global best solution found so far, and $eCBest_{i.j}$ denotes the influence of the current optimal solution. The step size is determined by $Levy(n)$, a random variable following a Lévy distribution. The iteration counters are given by *iter* (current iteration) and *Maxiter* (maximum iterations). The terms PD and DS are intrinsic parameters of the algorithm [28].

$$eCBest_{i,j} = GBest_{i,j} \times \Delta + \frac{PD_{i,j} \times mean(PD_{n,m})}{GBest_{i,j} \times \left(UB_j - LB_j\right) + \Delta} \quad (6)$$

$$CPD_{i,j} = \frac{GBest_{i,j} - rPD_{i,j}}{GBest_{i,j} + \Delta} \quad (7)$$

$$DS = 1.5 \times r \times (1 - \frac{iter}{Maxiter}) \quad (8)$$

The coefficient r introduces stochasticity to promote exploration, randomly assuming a value of either -1 or 1. Its sign toggles each iteration to maintain equilibrium between exploration and exploitation, determined by the parity (odd/even) of the iteration count. The parameter D represents a small perturbation value accounting for individual variations among prairie dogs, despite the algorithm's baseline assumption of uniformity. The variables *iter* and *Maxiter* indicate the current and the maximum allowed iterations, respectively.

During the exploitation phase, prairie dogs respond differently to distinct auditory signals representing specific environmental cues, such as predator alerts or food source notifications. This differential response mechanism forms the core of the PDO's exploitation strategy. The mathematical formulations governing this phase are presented below:

$$PD_{i+1,j+1} = GBest_{i,j} - eCBest_{i,j} \times \varepsilon - CPD_{i,j} \times rand \; \forall \; \frac{Max_{iter}}{2} \leq iter < 3\frac{Max_{iter}}{4} \quad (9)$$

$$PD_{i+1,j+1} = GBest_{i,j} \times PE \times rand \; \forall 3\frac{Max_{iter}}{4} < iter < Max_{iter} \quad (10)$$

$$PE = 1.5 \times (1 - \frac{iter}{Maxiter})^{\left(2\frac{iter}{Maxiter}\right)} \quad (11)$$

In the equations above, $GBest_{i;j}$ denotes the global optimum solution discovered thus far, while $eCBest_{i;j}$ quantifies the influence of the current best solution. The variable ϵ is a minor scalar representing food source quality, and $CPD_{i;j}$ consolidates the collective influence of all prairie dogs in the coterie. PE models the predator effect, and $rand$ is a uniformly

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

distributed random number in [0,1]. The coefficient $\frac{3}{4}$ was calibrated through sensitivity analysis by the algorithm's principal developer. Furthermore, the exponent in Equation (11) simulates the natural decay in the intensity of predator warnings over time, facilitating a shift towards more concentrated exploitation in later iterations. This decay parameter was also optimized via sensitivity analysis. Figure 1 illustrates the overall flowchart of the PDO algorithm [29].

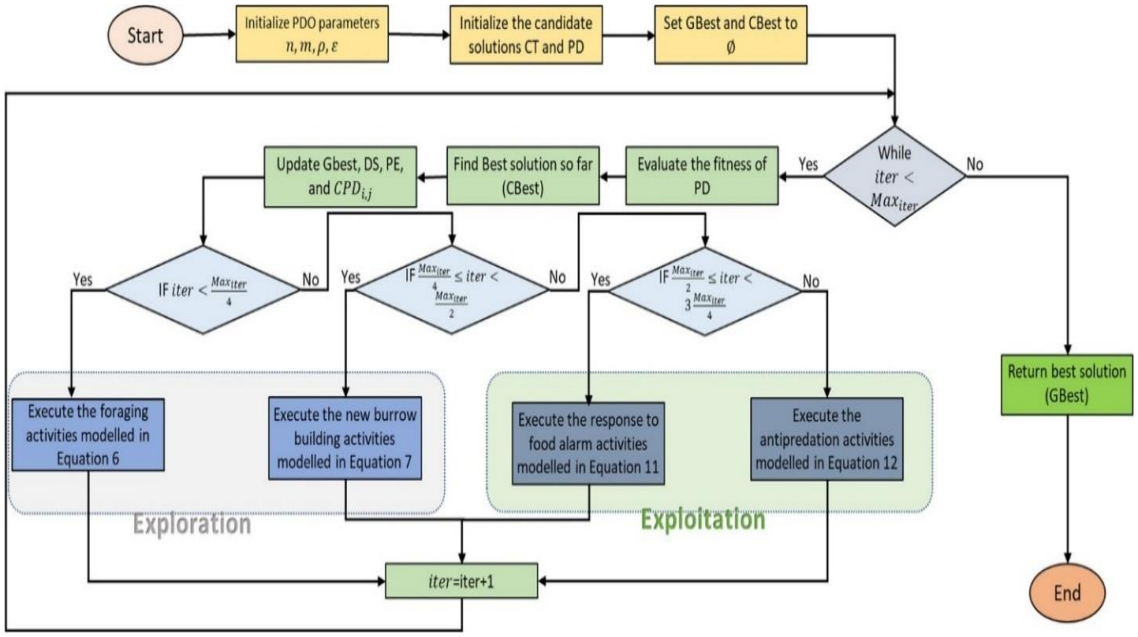


Figure 1:Flowchart of PDO algorithm [28].

2.2. Improved version of PDO

Random movement mechanisms are integral to many modern optimization algorithms, guiding search agents through stochastic position updates along unpredictable trajectories. This principle is inspired by natural stochastic phenomena such as sound diffusion and Brownian motion. Within the standard PDO, random numbers are fundamental to its update rules, perturbing candidate solutions via stochastic increments (see Equations 6–9). PaiamiFar et al. (2024) enhanced the performance of the PDO by replacing its simple uniform random steps with step sizes drawn from the Lévy distribution, thereby improving search efficiency across the solution space [29].

$$Levy(s) = |s|^{-1-\beta} \qquad 0 < \beta \leq 2 \tag{12}$$

$$Levy(s.\gamma.\mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{\frac{3}{2}}} & 0 < \mu < 0 \\ 0 & s \leq 0 \end{cases} \tag{13}$$

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

Here, α, β, σ, and μ represent the scaling parameter, shape factor, distribution factor, and location (position) parameter, respectively. In the Improved Prairie Dog Optimizer (I-PDO), a Lévy flight mechanism replaces the standard random walk within the original PDO's mathematical model.

In the standard PDO algorithm, agents' positions were updated using uniform random numbers, resulting in an approximate time complexity of $O(m \times d \times iter \times C_{\text{rand}})$, where each of the m agents in a d-dimensional space was evaluated per iteration. The improved I-PDO algorithm replaces this simple random walk with Lévy flight, enabling more efficient search trajectories. While the computational cost per Lévy step (C_{levy}) may exceed that of a uniform random step (C_{rand}), the enhanced search strategy significantly reduces the number of effective iterations required to reach the optimum. Consequently, despite the higher cost per step, the faster convergence rate of I-PDO leads to lower overall execution time and improved computational efficiency in complex optimization problems [29].

3. IMPLEMENTATION AND RESULTS

In this section, the results obtained from evaluating the IPDO algorithm and comparing it with other metaheuristic algorithms such as PSO [24], HHO [17], Honey Badger Algorithm (HBA) [30], AOA [26], GWO [15], DE [31], and original PDO for solving four challenging civil engineering problems are presented. Since the tuning parameters of algorithms play a crucial role in reaching the solution, in this research, the values suggested in their original references, as recommended by Arcuri and Fraser (2013) [32], were used. Table 1 shows the tuning parameters of the algorithms.

Table 1:Optimal values for the metaheuristic algorithm's parameters

Algorithm	Parameter	Value
PSO	Inertia weight	Linear reduction from 0.9 to 0.1
	C_1	2
	C_2	2
HHO	E_0	[-1, 1]
HBA	β	6
	C	2
AOA	α	5
	μ	0.05
GWO	Area vector	[0, 2]
	Random vector	[0, 1]
DE	Scaling factor (LB)	0.2
	Scaling factor (UB)	0.8
	PCR	0.8
PDO & IPDO	ρ	0.1
	ε	2.22e-16
	Δ	0.005

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

3.1.120-bar truss optimization

The first problem for evaluating the proposed algorithm is the optimal design of a 120-bar truss […], the objective of which is to find the optimal cross-sectional areas of the truss while considering structural constraints. Given that in reality, design variables are discrete, here the design variables are also considered discrete. Figure 2 shows the schematic of the truss. For the design of the 120-bar truss, the design variables are grouped; therefore, the number of design variables is not equal to the number of truss members, and is taken as 7 here. In other words, the optimization algorithms must select 7 suitable cross-sectional areas for the 120-bar truss from 41 discrete cross-sectional areas $A \in \{0{,}001.0{,}0015.0{,}002. \ldots .0{,}0205.0{,}021\}\ \mathrm{m}^2$. In fact, the decision variables are continuous between 0 and 1, which are converted to discrete values after encoding. The objective function and constraints of the problem are defined as Equation (14):

$$f = M \cdot (1 + C)^{\varepsilon}.C = \sum_{i=1}^{N} \max\left(0.\frac{|\sigma_i|}{\sigma_{allow}} - 1\right) \tag{14}$$

where M is the structural mass, σ_i and σ_{allow} are the maximum stress in member i and the allowable stress, respectively. The material properties and allowable stress are the same for all test problems. The density, modulus of elasticity, and allowable stress are specified as 7850 kg/m^3, 200 GPa, and 400 MPa, respectively. ε is the penalty exponent, a parameter that controls how severely violations are penalized.

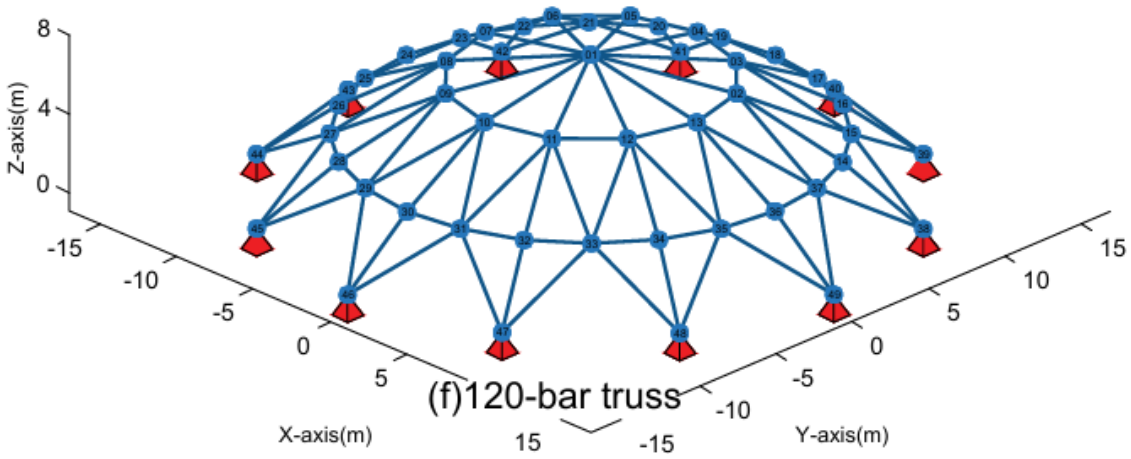


Figure 2:120-bar truss structure.

Considering an initial population size of 50 and the maximum number of iterations for the algorithms, the discrete truss design problem was solved by each algorithm. The obtained results for the mean, standard deviation, minimum, and maximum objective function values over 10 runs are presented in Table 2. As can be seen from the table, all algorithms except the original PDO managed to achieve the best objective value of 4.94E+03 kg. In contrast, PDO obtained a mean objective value of 1.09E+04 kg, indicating its failure to converge within the

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

considered number of iterations. This variation is clearly illustrated in Figure 3, where the other investigated algorithms, along with the enhanced version of this algorithm, demonstrate significantly higher robustness compared to PDO in this problem.

Table 2:Results of algorithms for 120-bar truss structure.

Algorithm	Objective function (kg)			
	Mean	Std	Min	Max
PSO	4.94E+03	0.00E+00	4.94E+03	4.94E+03
HHO	4.94E+03	0.00E+00	4.94E+03	4.94E+03
HBA	4.94E+03	0.00E+00	4.94E+03	4.94E+03
AOA	4.94E+03	0.00E+00	4.94E+03	4.94E+03
GWO	4.94E+03	0.00E+00	4.94E+03	4.94E+03
DE	4.94E+03	0.00E+00	4.94E+03	4.94E+03
PDO	1.09E+04	1.58E+03	8.29E+03	1.31E+04
IPDO	4.94E+03	0.00E+00	4.94E+03	4.94E+03

The convergence curves on the figure 4 demonstrate that most algorithms converge rapidly within the first 25 iterations, with varying levels of final accuracy. Among them, the IPDO algorithm consistently achieved the lowest objective function value and maintained stable performance throughout, indicating superior convergence and robustness. PDO and DE also showed competitive results, while algorithms such as GWO and AOA converged to higher objective values, suggesting less effective optimization in this context. Overall, the IPDO algorithm outperformed others in both convergence speed and final solution quality.

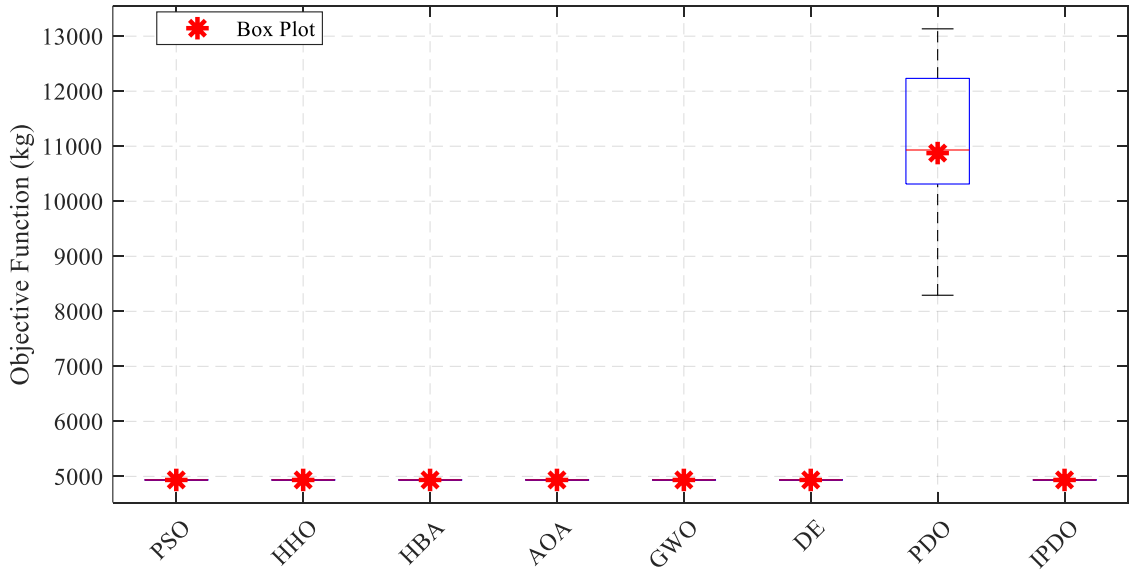


Figure 3:The boxplot of truss structure problem.

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

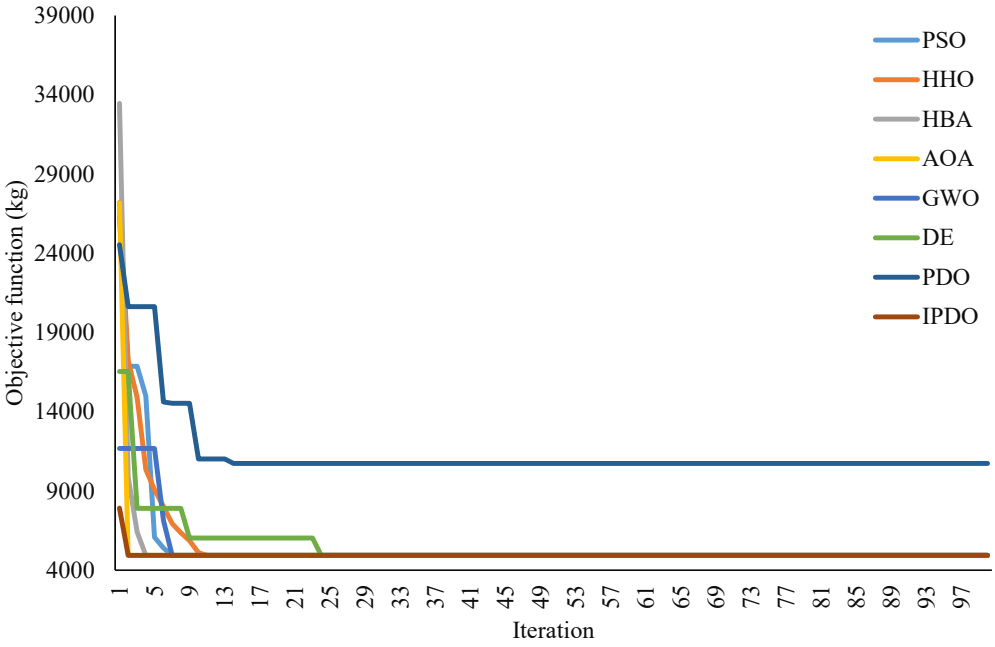


Figure 4:Convergence curves of algorithms for truss structure problem.

3.2. Structural reliability analysis

The second problem for evaluating the proposed optimization algorithm is the identification of design points in structural reliability assessment. In structural reliability analysis, the problem is formulated using a limit state function (LSF), $g(\mathrm{x})$, where x is an n-dimensional vector of random input variables. The condition $g(\mathrm{x}) < 0$ defines the failure region, while $g(\mathrm{x}) > 0$ indicates the safe region. The randomness of x is characterized by its joint probability density function (PDF), $\phi_n(\mathrm{x})$. For simplicity, this study assumes all input variables are independent and follow a standard normal distribution [10].

Under this assumption, the failure domain is:

$$F = \{\mathrm{x}: g(\mathrm{x}) < 0\}, \tag{15}$$

The corresponding indicator function, $I_F(\mathrm{x})$, is defined as:

$$I_F(\mathrm{x}) = \begin{cases} 1 & \text{if } \mathrm{x} \in F. \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

Using this notation, the probability of failure, p_f, is expressed as the following integral over the failure domain:

$$p_f = \int_{\mathbb{R}^n} I_F(\mathbf{x})\ \phi_n(\mathbf{x})\ d\mathbf{x}, \tag{17}$$

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

Estimating the probability of failure in structural reliability analysis is a major challenge, particularly for nonlinear limit state functions and when the failure probability is very small (typically less than 10^{-3}). Over the past fifty years, methods such as First-Order Reliability Method) [34], and Second-Order Reliability Method have been developed to address this problem, primarily based on the concept of the Design Point. The design point is mathematically defined as:

- Statistical Definition: The point within the failure domain F that has the highest probability density function (PDF) value.
- Geometric Definition (for independent standard normal variables): Due to the symmetry of the distribution, this point is equivalent to the point on the failure boundary $g(\mathrm{x}) = 0$ that is closest to the origin in Euclidean distance. This distance is denoted by β, the reliability index.

Therefore, finding the design point can be formulated as an optimization problem with the objective of minimizing $\| \mathrm{x} \|$ (or equivalently, maximizing its reciprocal) subject to the failure boundary constraint. While the choice of objective function (based on PDF or distance) does not significantly impact the final algorithm performance, the simpler geometric formulation of minimizing the distance from the origin (β) is adopted in this work for simplicity. It is noted that this formulation is valid as long as the failure probability is not excessively large, e.g., $p_f \leq 0{,}1$.

A cantilever tube structure, as depicted in Figure 5, is analyzed under combined loading conditions including external forces F_1, F_2, P, and a torsional moment T. The limit state function is defined as [35]:

$$g = S_y - \sigma_{\max} \tag{18}$$

where S_y represents the material yield strength and $\sigma_{\max}$ is the maximum von Mises stress on the top surface at the fixed end, expressed as:

$$\sigma_{\max} = \sqrt{\sigma_x^2 + 3\tau_{zx}^2} \tag{19}$$

Here, the normal stress σ_x and shear stress τ_{zx} are given by:

$$\sigma_x = \frac{P + F_1 \sin\,\theta_1 + F_2 \sin\,\theta_2}{A} + \frac{Md}{2I}.\tau_{zx} = \frac{Td}{4I} \tag{20}$$

The bending moment M, cross-sectional area A, and moment of inertia I are:

$$M = F_1 L_1 cos\ \theta_1 + F_2 L_2 cos\ \theta_2 . A = \frac{\pi}{4}[d^2 - (d-2t)^2].I = \frac{\pi}{64}[d^4 - (d-2t)^4] \tag{21}$$

The probabilistic characteristics of the 11 input random variables are provided in Table 3. Based on the complexity of the problem under investigation, the initial population size and maximum number of iterations were set to 50 and 1000, respectively, and each algorithm was

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

independently executed 10 times, consistent with the previous problem setup. The results are presented in Table 4.

Table 3:Statistical properties of the random variables for cantilever beam.

Variable	Distribution	Mean	Standard deviation
$t(mm)$	Normal	5	0.10
$d(m^2)$	Normal	42	0.10
$L_1(mm)$	Normal	120	1.20
$L_2(mm)$	Normal	60	0.60
$F_1(kN)$	Gumbel	3	0.30
$F_2(kN)$	Gumbel	3	0.30
$P(kN)$	Gumbel	12	1.20
$T(N,m)$	Normal	90	9
$S_y(MPa)$	Normal	175	17.50
$\theta_1(deg,)$	Normal	5	0.25
$\theta_2(deg,)$	Normal	10	0.50

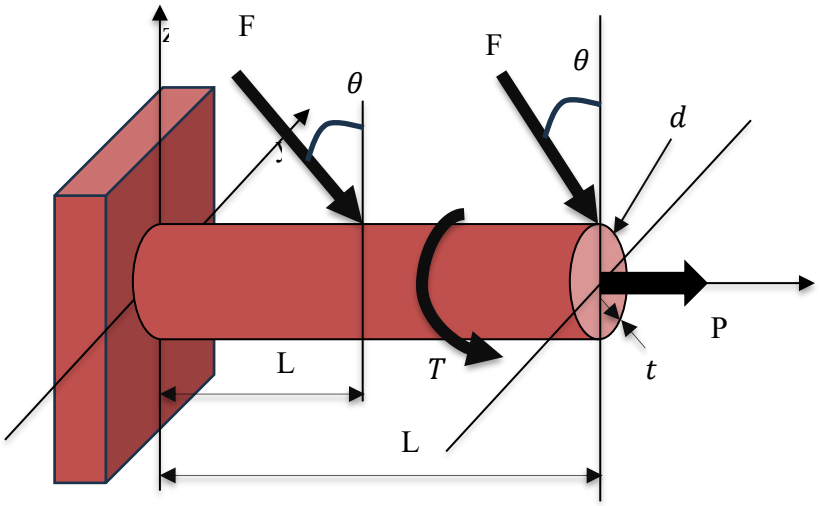


Figure 5:Schematic of cantilever beam.

According to the results of Table 4, which displays the performance of eight different optimization algorithms in solving Problem 3, a comprehensive analysis can be drawn. With a Monte Carlo reference value of 2.3675 established for validation, the proximity of algorithm outputs to this benchmark provides critical insight into their applicability for simulation-based calibration. While the HBA achieved the lowest mean objective value (1.9569) and exhibited the smallest standard deviation (0.41705), indicating superior stability and repeatability, the PDO and HHO produced solutions closest to the Monte Carlo result. Specifically, PDO reached 2.3329 (difference of 0.0346) and HHO reached 2.3226 (difference of 0.0449), representing relative errors of only 1.46% and 1.90%, respectively. PSO also performed reasonably well in approaching the Monte Carlo value with a result of 2.5372 (difference of 0.1697).

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

In contrast, DE and standard PSO showed weaker alignment with the Monte Carlo benchmark, with DE deviating significantly (mean of 44.863). The IPDO ranked second in overall optimization performance with a mean of 2.403, but was less aligned with the Monte Carlo value compared to IPDO and HHO. These findings suggest that IPDO and HHO are the most suitable algorithms for applications requiring consistency with Monte Carlo simulation, despite their slightly higher variance compared to HBA.

Table 4:Results of algorithms for cantilever beam.

	PSO	**HHO**	**HBA**	**AOA**	**GWO**	**DE**	**PDO**	**IPDO**
1	39.358	4.0664	2.0417	5.1719	9.961	56.786	2.8449	1.5687
2	24.859	2.3226	2.3843	8.9804	12.718	33.281	3.5678	1.7802
3	29.199	3.7685	1.695	5.1825	4.6788	24.625	6.2234	2.6485
4	9.8079	4.4999	1.605	9.0853	5.107	47.123	4.2942	2.5448
5	5.8055	6.8006	1.9442	3.9467	10.576	64.43	3.0749	2.9426
6	20.538	3.5105	1.6843	11.983	7.5795	38.707	2.798	2.6018
7	23.694	1.8368	1.5549	7.5897	5.6181	52.555	2.5131	2.82
8	2.5372	2.2251	1.5749	3.9467	8.2182	51.396	2.3329	2.1819
9	15.418	5.3764	2.7827	4.8039	6.7414	40.122	2.5861	2.5828
10	24.889	7.8982	2.302	12.446	3.4986	39.603	4.3655	2.3588
Mean	19.61	4.2305	1.9569	7.3136	7.4697	44.863	3.4601	2.403
SD	11.307	1.9887	0.41705	3.2014	2.9264	11.85	1.2046	0.44121

To further analyze the performance, Figures 6 and 7 respectively present the boxplot diagrams and convergence curves of the algorithms. As clearly observed in the boxplot, the improved prairie dog algorithm exhibits a lower standard deviation compared to its original version. Moreover, according to the convergence plot, the Lévy flight strategy contributes to enhancing the transition from the exploration phase to the exploitation phase in this algorithm, enabling it to reach a suitable solution in fewer iterations.

3.3.Reinforced concrete beam design

The objective of this example is to optimally design continuous reinforced concrete beams by minimizing the total cost function. This cost function, presented in Equation (22), expresses the sum of the costs of concrete and reinforcing steel [36]:

$$\min\ f(x) = V_c C_c + W_s C_s \tag{22}$$

In this equation, V_c is the volume of concrete (m³), C_c is the unit cost of concrete ($/m³), W_s is the total weight of steel (kg), and C_s is the unit cost of steel ($/kg).

To ensure compliance with code requirements and design criteria, a penalty coefficient method was used. The penalty coefficient $p_c = 10^8$ was adopted, and the final penalized objective function is calculated as follows:

$$P(x) = f(x) + p_c \sum \text{violations} \tag{23}$$

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

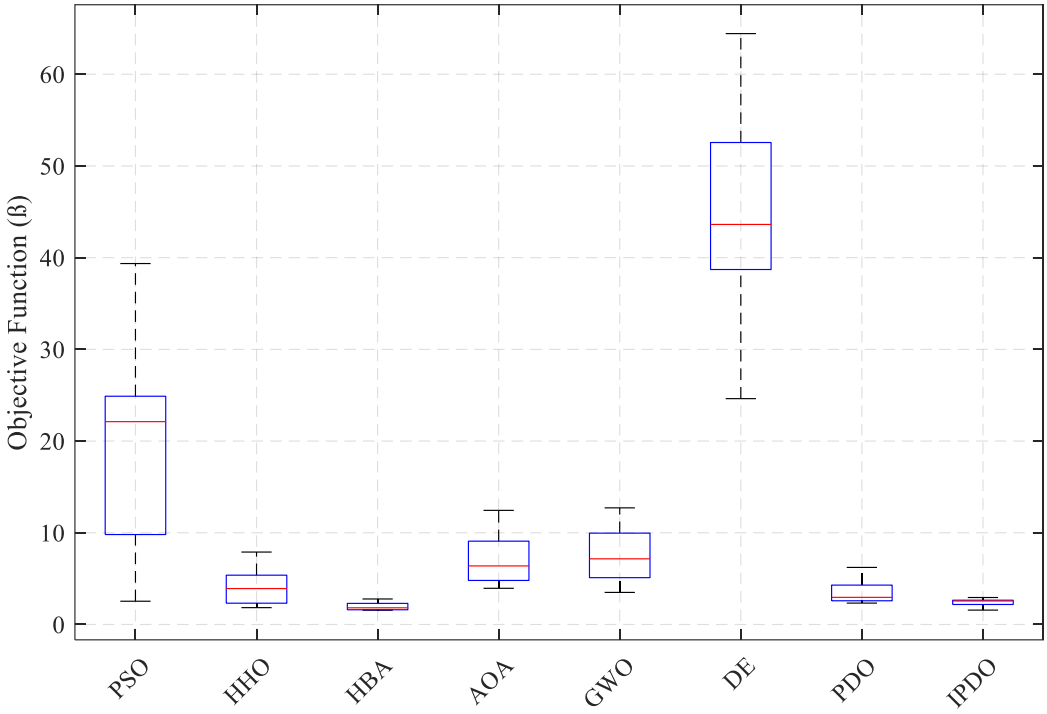


Figure 6: Boxplot of algorithm results for the cantilever beam

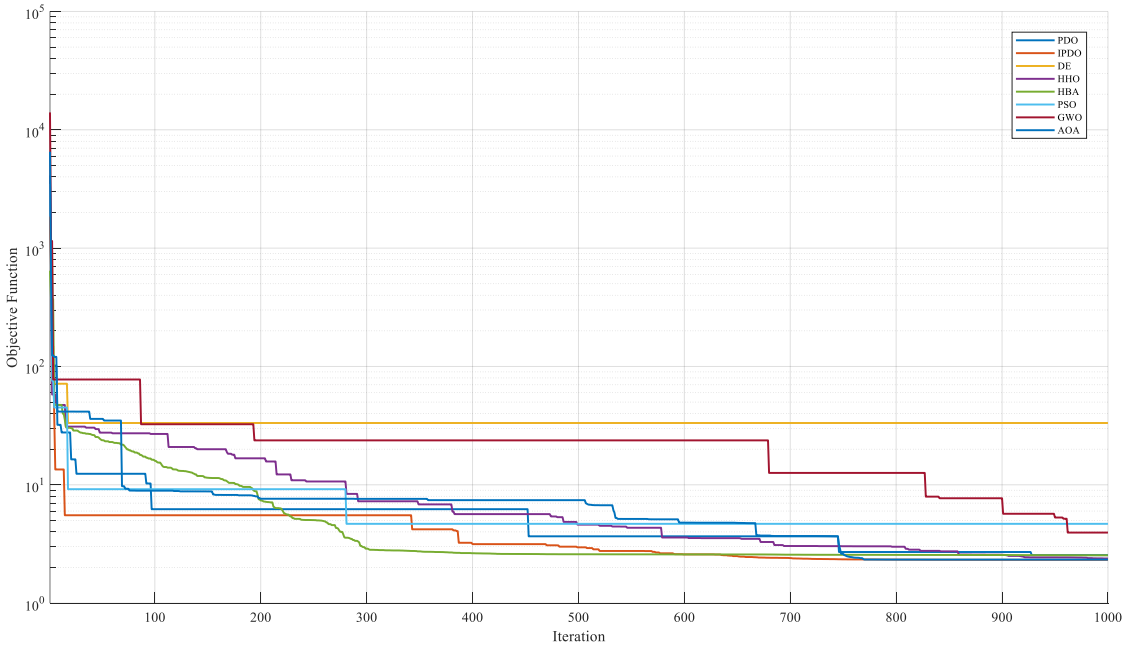


Figure 7: Convergence curves of algorithms for cantilever beam

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

3.3.1.Calculation of Steel Weight

The total weight of steel (W_s) is obtained from the sum of the weights of the main longitudinal reinforcement, stirrups, and web reinforcement according to Equation (24):

$$W_s = W_s^{main} + W_s^{stirrup} + W_s^{web} \tag{24}$$

3.3.1.1. Main Reinforcement

The weight of main reinforcement (W_s^{main}) is Calculated based on the optimal reinforcement layout and development and lap lengths according to the TS500 code.

3.3.1.2. Stirrups

The weight of Stirrups **(**$W_s^{stirrup}$**)** is Determined based on the stirrup spacing in the confinement and middle zones.

3.3.1.3. Web Reinforcement

The weight of Web Reinforcement (W_s^{web}) is Computed when required (based on beam height) in accordance with the TBER and TS500 codes.

3.3.2.Design Constraints

To ensure safety, serviceability, and code compliance, multiple constraints were defined. These constraints are based on the Turkish Reinforced Concrete Standard and earthquake regulations and impose limits on dimensions, reinforcement amounts and ratios, shear and moment capacity checks, and the strong column-weak beam criterion. The general equations for these constraints are as follows:

$$\begin{aligned}
g_1(x) &= \frac{b_w}{h_k + b_x^{column}} - 1 \le 0. \\
g_2(x) &= \frac{h_k}{3{,}5 b_w} - 1 \le 0. \\
g_3(x) &= \frac{L/12 \text{ or } L/15}{h_k} - 1 \le 0. \\
g_4(x) &= \frac{\rho_{min}}{\rho} - 1 \le 0. \\
g_5(x) &= \frac{\rho}{\rho_{max}} - 1 \le 0. \\
g_6(x) &= \frac{(0{,}3 \text{ or } 0{,}5) A_{s.sup}^{top}}{A_{s.sup}^{bot}} - 1 \le 0. \\
g_7(x) &= \frac{M_d}{M_r} - 1 \le 0 \text{(at supports).} \\
g_8(x) &= \frac{M_d}{M_r} - 1 \le 0 \text{(at spans).}
\end{aligned} \tag{25}$$

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

$$g_9(x) = 0{,}3\frac{f_{ctd}}{f_{ywd}} b_w \frac{s}{A_{sw}} - 1 \le 0.$$

$$g_{10}(x) = \frac{s_k}{\min\left(\frac{b_w}{4}.\,8\phi_{\min}.\,150\right)} - 1 \le 0.$$

$$g_{11}(x) = \frac{s_0}{\min\left(\frac{d}{2}.\,300\right)} - 1 \le 0 \text{ (if } V_d \le 3V_c).$$

$$g_{11}(x) = \frac{s_0}{\min\left(\frac{d}{4}.\,200\right)} - 1 \le 0 \text{ (if } V_d > 3V_c).$$

$$g_{12}(x) = \frac{V_d}{V_r} - 1 \le 0.\; g_{13}(x) = \frac{1{,}2(M_{ra} + M_{r\ddot{u}})}{(M_{\ddot{u}a} + M_{a\ddot{u}})} - 1 \le 0$$

In Tables 5 and 6, the detailed design parameters as well as the decision variable ranges are presented. Furthermore, Figure 8 illustrates the schematic configuration of the grillage beam under investigation. As shown, the problem comprises 10 discrete decision variables. After obtaining continuous values from the optimization algorithms, these are converted into integer values through an appropriate coding scheme. Additional information regarding the problem formulation is available in [36].

Table 5: Design Parameters for investigated beam problem

Design Parameter	Value	Unit
Concrete unit weight	25	kN/m³
Floor height	3000	mm
Stirrup diameter	8	mm
Earthquake design class	1a – 3a (only for two-span CBP)	–
Maximum aggregate diameter	16	mm
Beam width	250	mm
Concrete cover	25	mm

Table 6:Lower and upper bounds and increments of design variables

Design Variable	Lower Bound	Upper Bound	Increment
X_1	1	7130	1
X_2	1	7130	1
X_3	250 mm	1050 mm	50 mm
X_4	{C25, C30, C35, C40, C45, C50}	–	–
X_5	{B420C, B500C}	–	–
X_6 to X10	50 mm	350 mm	10 mm

The unit cost of steel, for both grades B420C and B500C, is 0.78 USD per kilogram. For concrete, the costs vary depending on the grade: C25 (25 MPa) costs 47.26 USD per cubic

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

meter, C30 (30 MPa) costs 49.06 USD per cubic meter, C35 (35 MPa) costs 52.35 USD per cubic meter, C40 (40 MPa) costs 55.34 USD per cubic meter, C45 (45 MPa) costs 56.24 USD per cubic meter, and C50 (50 MPa) costs 58.03 USD per cubic meter [36].

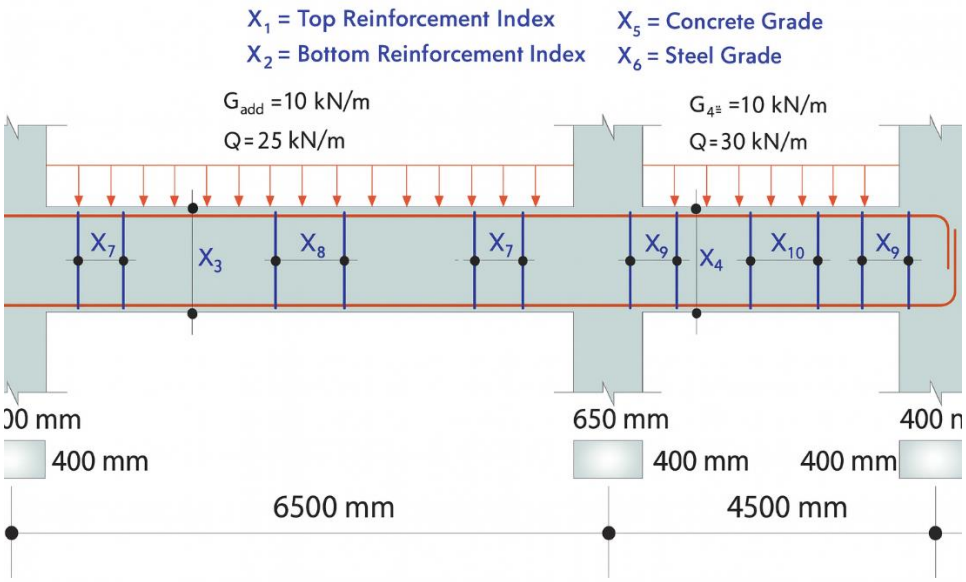


Figure 8:The schematic of investigated R.C beam problem.

The results of evaluating different optimization algorithms based on the objective function of problem number 3 are presented in Table 7, which shows their distinct performance. The IPDO emerged as the most effective method, achieving the lowest mean objective function value (196.77$) and the lowest standard deviation (6.21), which highlights its high stability and accuracy in reaching optimal solutions. Following IPDO, the HBA (204.02$) and HHO (213.99$) algorithms secured the second and third ranks, respectively, demonstrating their competitive performance in solving the problem. In contrast, the PSO algorithm showed the weakest performance, with a mean of 302.51$ and a standard deviation of 24.867, reflecting its greater sensitivity to initial conditions and relative instability during the optimization process. Algorithms such as AOA, GWO, DE, and PDO exhibited moderate performance, all proving less efficient compared to IPDO. Overall, the results underscore the significant superiority of the improved PDO algorithm over the other examined methods in terms of both accuracy and reliability.

Based on the obtained results, the IPDO achieved not only the best mean objective function value (196.77$) but also the lowest minimum value (187.85$) among all runs, demonstrating its superior capability in locating the deepest and most desirable solutions in the search space. In contrast, PSO recorded the highest maximum value (345.29$) in its worst run, indicating the largest deviation from the optimal solution and revealing its high sensitivity to initial conditions and relative instability during the optimization process.

A direct comparison between the standard PDO (mean: 235.15$) and its enhanced version IPDO (mean: 196.77) shows a significant and meaningful improvement in performance. This notable enhancement; both in the mean (a reduction of approximately 16%)

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

and in the minimum value obtained; confirms the effectiveness of the proposed improvement approach for the PDO algorithm. The concurrent reduction in the standard deviation from 4.3163 to 6.2109 (although slightly higher numerically, it remains acceptable given the substantial improvement in mean and minimum values, indicating relative stability in the improved version) and the narrowing of the value range (from 229.29–240.98 to 187.85–205.83) further validate that the implemented modifications; likely including refined search mechanisms, a better balance between exploration and exploitation, or the addition of more efficient operators; have successfully guided the algorithm toward higher-quality regions while mitigating the risk of entrapment in local optima.

In summary, the results clearly indicate that IPDO outperforms all other compared algorithms, including its original version, in terms of solution accuracy, search depth, and reliability. The proposed enhancement strategy has proven to be a highly effective approach for increasing the efficiency and robustness of metaheuristic-based optimization algorithms.

Table 7:The results of algorithms on investigated R.C beam problem.

Algorithm	Objective function ($)			
	Mean	Std	Min	Max
PSO	302.51	24.868	275.5	345.29
HHO	213.99	7.7618	201.93	229.66
HBA	204.02	8.282	192.88	217.44
AOA	210.96	9.0305	200.9	226.17
GWO	229.67	6.057	221.47	239.01
DE	225.59	13.852	206.25	250.8
PDO	235.15	4.3163	229.29	240.98
IPDO	196.77	6.2109	187.85	205.83

To illustrate the variations in objective function values and the robustness of the algorithms in solving this complex engineering problem, the box plot of the algorithms over 10 independent runs is presented in Figure 9. As discussed earlier, the HHO, PDO, and its improved version demonstrated greater robustness compared to the other algorithms, as evidenced by the fact that the variations in the objective function are confined to a narrow range. Furthermore, since all algorithms converged in the initial iterations, their convergence curves based on the average responses are shown in Figure 10 for fewer than 30 iterations. It can be inferred that the proposed improvement to the desert dog algorithm did not affect its convergence speed; rather, it only contributed to achieving a more optimal solution. Overall, the findings of this study indicate that the problem under investigation is a complex engineering challenge, and its solution using metaheuristic algorithms requires the development of effective approaches to enhance their performance.

3.4.Hyperparameter optimization of Support Vector Machine

In the final example, this study focuses on optimizing the parameters of a Support Vector Machine (SVM) model for predicting the shear strength of steel fiber–reinforced concrete (SFRC). A database of 241 records was compiled through a comprehensive review of

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

experimental studies, ensuring the accuracy and reliability of the collected information [37]. The dataset encompasses key material, geometric, and structural parameters essential for model training and validation, including steel fiber ratio (F%), concrete compressive strength (fc), effective beam depth (d), reinforcement ratio (ρ), and span-to-depth ratio (a/d). These variables were selected due to their critical impact on the shear behavior of SFRC beams, while the target output is the experimentally measured shear strength reported in the literature. Table 8 shows a summary of the statistical distribution of the data.

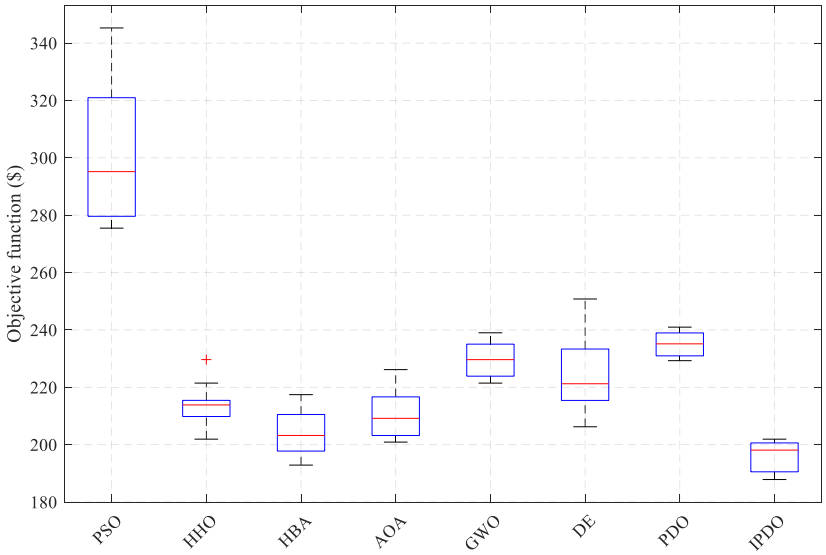


Figure 9: The boxplot of algorithms for the R.C. beam optimization problem

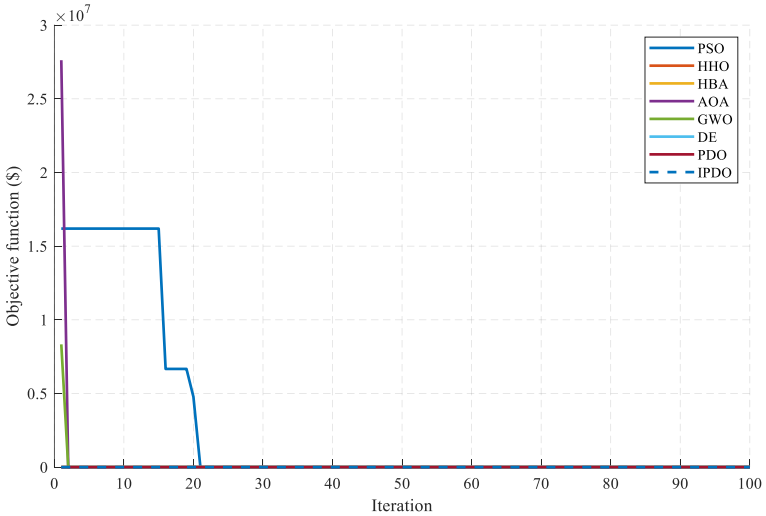


Figure 10:Convergence curves of algorithms for the R.C. beam optimization problem

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

Table 8:Statistical properties of the data sets

	$V_f(\%)$	$F = V_f \frac{l}{d}$	$\rho(\%)$	$d(mm)$	$f_c(Mpa)$	a/d	$v_{exp}(Mpa)$
Average	0.79	0.59	2.71	243.8	47.85	3.08	3.96
Standard deviation	0.38	0.33	1.15	145.9	22.22	1.12	2.59
Minimum	0.22	0.10	0.02	102.00	19.60	0.46	0.56
Maximum	2.00	2.00	5.72	1118.00	111.50	6.00	15.05

In SVM-based regression, the goal is to model the functional relationship between a dependent variable y and independent variables x. As with other regression techniques, the underlying assumption is that y = f(x) + noise. The core objective is to derive a function f capable of accurately predicting new, unseen patterns beyond the training set. This function is learned by training the SVM on available data and iteratively minimizing a designated loss function. The present research employs ε-SVM for its recognized efficacy in regression applications. The model utilizes an ε-insensitive loss function, formulated as:

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{26}$$

Minimization of this loss is subject to the following constraints:

$$\begin{aligned} \mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i &\leq \epsilon + \xi_i^* \\ y_i - \mathbf{w}^T\phi(\mathbf{x}_i) - b &\leq \epsilon + \xi_i \\ \xi_i.\xi_i^* &\geq 0.\, i = 1.\ldots.N \end{aligned} \tag{27}$$

In these expressions, $\mathbf{w}$ and $\mathbf{w}^T$ represent the coefficient vector and its transpose, respectively; $C > 0$ is the regularization (capacity) parameter; ξ_i and ξ_i^* are slack variables; b is the bias term; N denotes the number of training samples; and $\phi(\cdot)$ signifies the feature mapping induced by the chosen kernel function $k(\cdot.\cdot)$. Given the complexity and time-consuming nature of selecting an appropriate kernel type, regularization parameter C, and kernel scale (e.g., γ for an RBF kernel), this study employs optimization algorithms to identify the optimal combination of these hyperparameters. To evaluate the SVM-IPDO model developed, three performance metrics were employed: root mean square error (RMSE), mean absolute error (MAE), and the coefficient of determination (R^2), as defined in Equations (28)–(30). In addition, the dataset was partitioned into 70% for training, 10% for validation, and 20% for testing the developed models [38].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{mea}^i - y_{pre}^i\right)^2} \tag{28}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_{mea}^i - y_{pre}^i}{y_{mea}^i}\right) \tag{29}$$

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_{mea}^{i} - y_{pre}^{i}\right)^2}{\sum_{i=1}^{n}\left(y_{mea}^{avg} - y_{pre}^{i}\right)^2} \tag{30}$$

It should be noted that the objective function was defined as MSE according to Equation (31).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_{mea}^{i} - y_{pre}^{i}\right)^2 \tag{31}$$

where, y_{mea}^{i} and y_{pre}^{i} denote the measured and the predicted values, respectively, and n is the number of measurements. y_{mea}^{avg} and y_{pre}^{avg} are the corresponding average values of the measured and predicted parameters, respectively.

Table 9 presents the statistical outcomes of ten independent runs for each metaheuristic algorithm under identical population size and maximum iteration settings, thereby enabling a fair comparison of their performance in solving the engineering optimization problem. The PSO and DE algorithms yielded relatively high mean objective values (2.1081 and 2.0282, respectively) accompanied by large standard deviations, which reflect their unstable convergence behavior and sensitivity to initial conditions. In contrast, the HHO and GWO demonstrated remarkable robustness, both achieving a mean of 1.4891 with a very small standard deviation (0.0015), indicating consistent convergence across all runs. The HBA and AOA produced moderate mean values of approximately 1.607, but their higher variability (standard deviation ≈0.315) suggests less stability compared to HHO and GWO. The PDO algorithm showed a mean of 2.0438 with moderate variability, while its improved version (IPDO) achieved the best performance, consistently converging to the global minimum of 1.4881 in all runs, as evidenced by its negligible standard deviation (≈0). Taken together, these results highlight that HHO, GWO, and particularly IPDO are the most reliable and robust algorithms for this problem, whereas PSO and DE exhibit weaker stability and higher dispersion, and HBA and AOA occupy an intermediate position in terms of accuracy and consistency.

Table 9:The results of algorithms for the SVM problem

	PSO	**HHO**	**HBA**	**AOA**	**GWO**	**DE**	**PDO**	**IPDO**
1	2.869393	1.488148	1.488148	1.488148	1.491353	2.292825	1.488148	1.488148
2	1.491353	1.488148	1.488148	1.488148	1.488148	2.292825	2.333763	1.488148
3	1.488148	1.488148	1.488148	1.488148	1.488148	1.488148	2.869393	1.488148
4	2.869393	1.488148	1.488148	1.488148	1.488148	1.488148	2.292825	1.488148
5	2.869393	1.488148	1.488148	1.678652	1.488148	1.488148	2.333763	1.488148
6	1.488148	1.491353	1.678652	2.489168	1.491353	2.333763	1.488148	1.488148
7	1.488148	1.488148	2.489168	1.488148	1.488148	3.43887	1.914714	1.488148
8	2.695093	1.491353	1.488148	1.488148	1.491353	2.292825	2.740542	1.488148
9	1.488148	1.491353	1.488148	1.488148	1.488148	1.678652	1.488148	1.488148
10	2.333763	1.488148	1.488148	1.491353	1.488148	1.488148	1.488148	1.488148
Mean	2.1081	1.4891	1.6073	1.6076	1.4891	2.0282	2.0438	1.4881
SD	0.67098	0.001548	0.31559	0.31545	0.001548	0.63089	0.54251	2.34E-16

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

The box plot presented in Figure 11 further illustrates the distribution and variability of the objective function values (MSE) obtained from the ten independent runs of each algorithm. It provides a visual comparison of robustness and consistency across methods. Notably, the IPDO algorithm exhibits perfect stability, with all values converging to the same minimum, resulting in a flat box with no visible spread or outliers. Similarly, HHO and GWO demonstrate high robustness, as indicated by their narrow interquartile ranges and tightly clustered values around the median. In contrast, PSO, DE, and PDO show wider boxes and longer whiskers, reflecting greater variability and less reliable convergence behavior. The presence of outliers in these algorithms suggests sensitivity to initial conditions and potential instability in performance. HBA and AOA occupy an intermediate position, with moderate spread and consistent medians, indicating acceptable but less robust behavior. Overall, the box plot confirms that IPDO, HHO, and GWO are the most stable and reliable algorithms for this problem, while PSO and DE exhibit significant dispersion and weaker convergence characteristics.

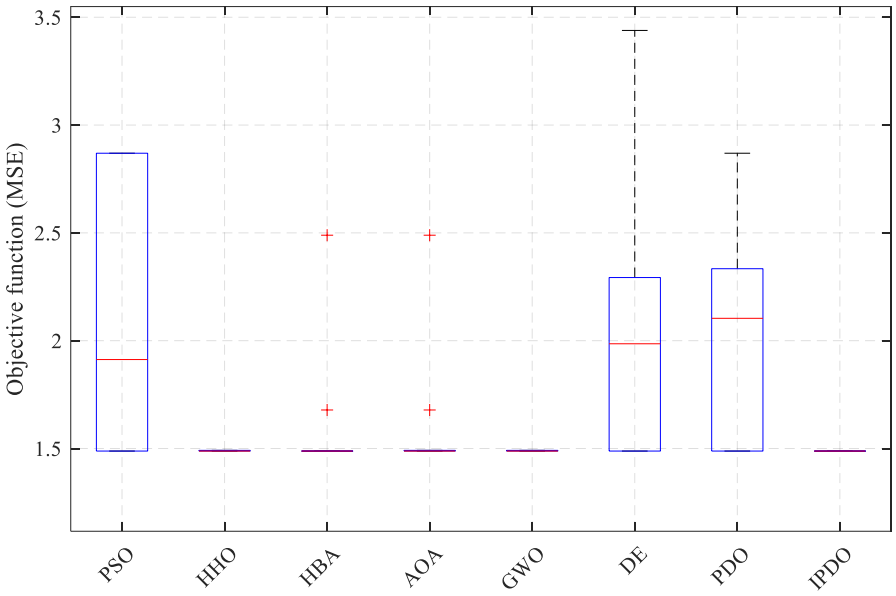


Figure 11:Boxplot of algorithm results for the SVM problem

Based on the above results, the IPDO algorithm demonstrated the best performance for SVM optimization, both in terms of accuracy and robustness (with the best average over 10 runs). Notably, the optimal objective function value (training MSE = 1.4884) was the same for all algorithms. Consequently, the optimal decision variables were identical across all algorithms in their best runs, specifically: Kernel type = Polynomial, Kernel scale = 2, and Regularization parameter = 2. Therefore, only the results of the SVM-IPDO model are presented for estimating the shear strength of concrete beams.

Figures 12 and 13 illustrate the error distribution and the correlation between predicted and experimental values, respectively. Furthermore, according to Table 10, the comparison of the proposed model with previously established empirical methods demonstrates that SVM-IPDO, with lower RMSE, MAE, MAPE, and higher R^2, provides a robust and reliable

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

approach for predicting the shear strength of beams.

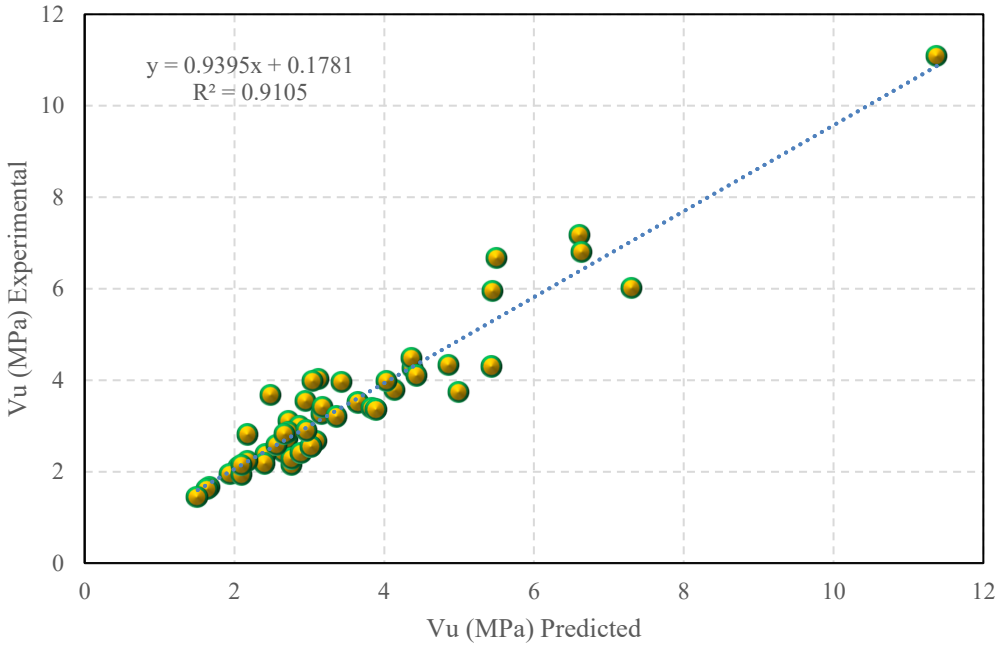


Figure 12:Scatter plot of SVM-IPD

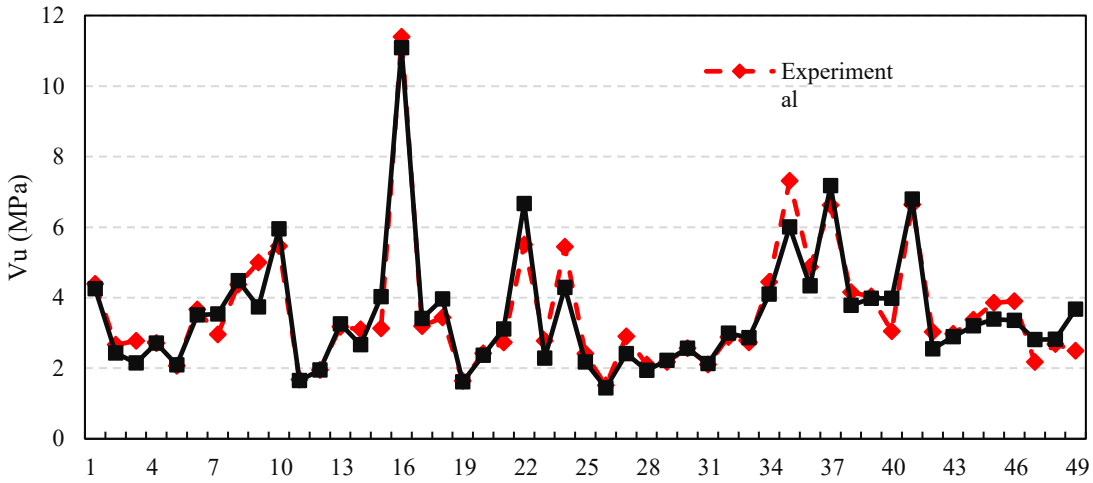


Figure 13:Time series plot for SVM-IPDO performance.

Table 10:Comparison between the different existing formulas and SVM-IPDO.

Model	MAE	RMSE	R2
Sharma et al. (1986)	1.45	2.50	0.40
Ashour et al. (1992)	1.94	2.75	0.65
Kuntia et al. (1999)	1.69	2.47	0.52
Momani et al. (2021)	1.00	1.50	0.70
SVM-IPDO	0.82	1.59	0.91

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

4. CONCLUSIONS

This study investigated the efficiency of an Improved Prairie Dog Optimization (IPDO) algorithm through four distinct and challenging civil engineering optimization problems: discrete truss design, structural reliability assessment, cost minimization of reinforced concrete beams, and SVM hyperparameter tuning for shear strength prediction. Across all benchmarks, IPDO consistently demonstrated superior performance in terms of solution accuracy, convergence speed, and robustness compared to seven well-established metaheuristic algorithms (PSO, HHO, HBA, AOA, GWO, DE, and standard PDO). Key findings include:

- In the 120-bar truss problem, IPDO achieved the optimal design weight (4.94E+03 kg) with perfect consistency (zero standard deviation), unlike the standard PDO which showed significant variability.
- For the structural reliability problem, IPDO provided stable and accurate approximations of the design point, with competitive alignment to Monte Carlo simulation results.
- In the reinforced concrete beam cost optimization, IPDO yielded the lowest mean cost (196.77 $) and exhibited strong robustness, highlighting its practical applicability in structural design.
- Most notably, in the SVM hyperparameter tuning task for shear strength prediction, IPDO consistently reached the global optimum (MSE = 1.4881) across all independent runs, demonstrating unparalleled reliability and precision. The resulting SVM-IPDO model outperformed existing empirical formulas, achieving lower MAE (0.82), RMSE (1.59), and higher R^2 (0.91).

The results affirm that the Lévy flight integration successfully enhances PDO's search dynamics, enabling more efficient navigation of complex, multimodal landscapes. The proposed IPDO algorithm proves to be a robust, accurate, and computationally effective metaheuristic, suitable for a wide range of engineering optimization problems. Future work may explore adaptive parameter control, hybridizations with local search methods, and applications to other interdisciplinary optimization challenges.

REFERENCES

1. Siadatpour S, Aghamolaei Z, Jafari-Asl J, Moghadam AB. Single-objective optimization of water quality sensors in water distribution networks using advanced metaheuristic techniques. *Water*. 2025;**17**(8):1221.
2. Jafari-Asl J, Dong Y, Li Y. Advanced computational strategy for damage identification of offshore jacket platforms. *Civil Engineering Design*. 2025;**7**(1):36–48.
3. Jafariasl J, Spyridis P, Ploennigs J. Automated screening for cracks in concrete structures using an optimized convolutional neural network. *Ce/Papers*. 2025;**8**(3–4):368–73.
4. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. *Acta Mechanica*. 2010;**213**(3–4):267–89.
5. Kaveh A, Hamedani KB, Kamalinejad M, Joudaki A. Quantum-based jellyfish search

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

optimizer for structural optimization. *IJOCE*. 2021;**11**(2):329–56.
6. Kaveh A, Kamalinejad M, Arzani H. Quantum evolutionary algorithm hybridized with enhanced colliding bodies for optimization. *Structures*. 2020;**28**:1479–501.
7. Gholizadeh S, Sojoudizadeh R. Modified sine-cosine algorithm for sizing optimization of truss structures with discrete design variables. *IJOCE*. 2019;**9**(2):195–212.
8. Aslam Y, Santhi N. A comprehensive survey on optimization techniques in image processing. *Materials Today Proceedings*. 2020;**24**:1758–65.
9. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*. 2015;**10**(3):e0118432.
10. Jafari-Asl J, Dong Y, Guo H. Enhancing structural reliability through AI-driven control variates and subset simulation. *ASCE-ASME J Risk Uncertain Eng Syst Part A Civ Eng*. 2025;**11**(4).
11. Rahmanshahi M, Jafari-Asl J, Fathi-Moghadam M, Ohadi S, Mirjalili S. Metaheuristic learning algorithms for accurate prediction of hydraulic performance of porous embankment weirs. *Appl Soft Comput*. 2023;**151**:111150.
12. Khastar S, Bashirizadeh F, Jafari-Asl J, Hamzehkolaei NS. Predicting the cooling and heating loads of energy efficient buildings: a hybrid machine learning approach. *Cluster Comput*. 2025;**28**(5).
13. Baradaran M, Madhkhan M. Determination of optimal configuration for mega bracing systems in steel frames using genetic algorithm. *KSCE J Civ Eng*. 2019;**23**(8):3616–27.
14. Zheng Y, Da D, Li H, Xiao M, Gao L. Robust topology optimization for multi-material structures under interval uncertainty. *Appl Math Model*. 2019;**78**:627–47.
15. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw*. 2014;**69**:46–61.
16. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw*. 2016;**95**:51–67.
17. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst*. 2019;**97**:849–72.
18. Mafarja MM, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*. 2017;**260**:302–12.
19. Kaveh A, Yousefpoor H. Chaos-based swarm intelligence algorithms for optimal design of truss structures. *Period Polytech Civ Eng*. 2025;**69**(3):884–903.
20. Kaveh A, Hamedani KB. Success-history based adaptive differential evolution algorithm for discrete structural optimization. *Iran J Sci Technol Trans Civ Eng*. 2024;**49**(1):409–31.
21. Ahmadianfar I, Bozorg-Haddad O, Chu X. Gradient-based optimizer: a new metaheuristic optimization algorithm. *Inf Sci*. 2020;**540**:131–59.
22. Maleki A, Roayaei M, Mirjalili S. Enhancing leadership-based metaheuristics using reinforcement learning: a case study in grey wolf optimizer. *Knowl Based Syst*. 2025;**330**:114471.
23. Genetic algorithms in search, optimization, and machine learning. *Choice Rev Online*. 1989;**27**(02):27–0936.
24. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Comput Intell Mag*. 2006;**1**(4):28–39.
25. Poli R, Kennedy J, Blackwell T. Particle swarm optimization. *Swarm Intell*. 2007;**1**(1):33–57.

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]

26. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl Intell*. 2020;**51**(3):1531–51.
27. Wolpert D, Macready W. No free lunch theorems for optimization. *IEEE Trans Evol Comput*. 1997;**1**(1):67–82.
28. Ezugwu AE, Agushaka JO, Abualigah L, Mirjalili S, Gandomi AH. Prairie dog optimization algorithm. *Neural Comput Appl*. 2022;**34**(22):20017–65.
29. PayamiFar T, Sojoudizadeh R, Azizian H, Rahimi L. Seismic optimization of steel mega-braced frame with improved prairie dog metaheuristic optimization algorithm. *Struct Des Tall Spec Build*. 2025;**34**(3):e2207.
30. Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W. Honey badger algorithm: new metaheuristic algorithm for solving optimization problems. *Math Comput Simul*. 2021;**192**:84–110.
31. Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim*. 1997;**11**(4):341–59.
32. Arcuri A, Fraser G. Parameter tuning or default values? An empirical investigation in search-based software engineering. *Empir Softw Eng*. 2013;**18**(3):594–623.
33. Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables. *J Constr Steel Res*. 2009;**65**(8–9):1558–68.
34. Zhong C, Wang M, Dang C, Ke W, Guo S. First-order reliability method based on Harris hawks optimization for high-dimensional reliability analysis. *Struct Multidiscip Optim*. 2020;**62**(4):1951–68.
35. Hong L, Li H, Peng K. A combined radial basis function and adaptive sequential sampling method for structural reliability analysis. *Appl Math Model*. 2020;**90**:375–93.
36. Öztürk HT. Research on optimal solutions and algorithm stability analyses in RC continuous beam problems. *Structures*. 2024;**62**:106239.
37. Momani Y, Tarawneh A, Alawadi R, Momani Z. Shear strength prediction of steel fiber-reinforced concrete beams without stirrups. *Innov Infrastruct Solut*. 2022;**7**:107.
38. Vatani A, Jafari-Asl J, Ohadi S, Hamzehkolaei NS, Ahmadabadi SA, Correia JFO. An efficient surrogate model for reliability analysis of the marine structure piles. *Proc Inst Civ Eng Marit Eng*. 2023;**176**(4):176–92.
39. Ashour SA, Hasanain GS, Wafa FF. Shear behavior of high-strength fiber reinforced concrete beams. *ACI Struct J*. 1992;**89**(2).
40. Khuntia M, Stojadinovic B, Goel SC. Shear strength of normal and high-strength fiber reinforced concrete beams without stirrups. *ACI Struct J*. 1999;**96**.

[Downloaded from ijoce.iust.ac.ir on 2026-09-08]

[DOI: 10.22068/ijoce.2025.15.4.657]