Journal of Transformative Technologies and Sustainable Development (2025) 9:4
https://doi.org/10.1007/s41314-025-00075-4

RESEARCH

Warthog Optimization Algorithm: A New Nature-Derived Solution for Optimization Challenges

Shuvadipta Das[1] · Srijita Baksi[1] · Partha Ghosh[1]

Received: 20 April 2025 / Revised: 17 June 2025 / Accepted: 19 August 2025

Abstract
Data plays an increasingly critical role in today's digital era, driving advancements in Information Technology (IT) and Computer Science (CS). As the volume, complexity, and dimensionality of data have grown manifold, efficient techniques have become the necessity for processing, analyzing, and extracting meaningful data. Challenges such as managing high-dimensional datasets, optimizing distributed resources, and solving complex, nonlinear problems require new adaptable optimization algorithms (OAs) that balance exploration (searching widely for various possible solutions) with exploitation (focusing on and improving the best solutions). Optimization techniques play a critical role across diverse domains including machine learning (ML), engineering design (ED), and image analysis. Traditional algorithms often struggle with the intricacies of modern, complex search spaces, which have led researchers to turn to metaheuristic approaches. Nature-derived optimization techniques, known for their adaptive strategies, have come up as powerful tools in this context. This research introduces the Warthog Optimization Algorithm (WartOA), a novel bio-inspired method that uniquely integrates burrow-based retreat and adaptive foraging strategies for dynamic exploration-exploitation balance. Following evaluation on 23 Benchmark Functions (BFs), multiple ED problems, and feature selection for the NSL-KDD intrusion detection dataset, WartOA demonstrates strong potential and robustness when compared with established optimization techniques.

Keywords Warthog Optimization Algorithm (WartOA) · Bio-inspired optimization algorithm · Metaheuristic algorithms (MAs) · Optimization problems (OPs) · Benchmark functions (BFs) · Exploitation · Exploration

Introduction

Overview

With the exponential growth of technology and data-driven systems in recent decades, the demand for efficient and intelligent optimization techniques has become more crucial than ever. From training machine learning models to making complex engineering designs, modern computational problems often involve navigating vast, high-dimensional, and nonlinear search spaces. Traditional optimization methods such as gradient-driven (GD) [1] and linear programming (LP) work well for structured and convex problems. However, they often struggle when faced with multimodal landscapes, multiple constraints, or the stochastic nature of today's real-world applications.

Aim

This study proposes a new adaptive algorithm inspired by animal behaviour, intended to effectively address numerous complex realistic problems. The aim is to develop a novel optimization approach that achieves a dynamic global–local search balance, enhancing both search capability and solution accuracy. The algorithm is biologically inspired by the behaviour of warthogs and is termed WartOA. It is designed to outperform existing methods by mimicking warthogs' strategic foraging and defence behaviour to efficiently solve real-world optimization problems.

Shuvadipta Das, Srijita Baksi, and Partha Ghosh contributed equally to this work

✉ Partha Ghosh
partha.ghosh@nsec.ac.in

Shuvadipta Das
shuvadiptadas8820@gmail.com

Srijita Baksi
baksisrijita04@gmail.com

[1] Information Technology, NSEC, Technocity, Panchpota, Garia, Kolkata 700152, West Bengal, India

Published online: 30 August 2025

Scope

The scope of this work covers both theoretical and practical aspects of optimization. On the theoretical side, it involves a detailed mathematical formulation of a novel nature-inspired metaheuristic algorithm. Practically, the algorithm is tested against a range of optimization evaluation functions (OEFs) and well-known engineering design (ED) problems, demonstrating its applicability to complex, constrained, and high-dimensional real-world problems. A special focus is given to dimensionality, a critical issue where the search space grows exponentially with the number of variables, making optimization highly challenging. WartOA addresses this by preserving diversity and preventing premature convergence through its biologically motivated strategies.

Motivation

The motivation behind adopting a bio-inspired approach stems from the remarkable problem-solving abilities observed in nature. Biological organisms have evolved over millions of years to survive, adapt, and thrive in highly competitive and unpredictable environments, developing unique survival strategies and responses. Drawing inspiration from these natural processes has led to the development of several metaheuristic algorithms, such as genetic algorithms (GAs) [2] and particle swarm optimization (PSO) [3], which have shown significant performance in avoiding local optima and achieving near-optimal solutions.

These algorithms generally begin with a set of random solutions and evaluate them based on an objective function. The exploration phase involves searching diverse regions of the solution space using stochastic updates, while the exploitation phase focuses on improving the best-found solutions using local refinements or evolutionary operator [4].

This exploration-exploitation trade-off is especially critical in solving complex real-world problems that contain multiple constraints, competing objectives, and uncertain environments [5]. The dimensionality problem leads to vast solution spaces and high computational costs. These challenges inspired the authors to model a novel algorithm after the behaviour of warthogs, animals known for their intelligent foraging, dominance-based interactions, and instinctual retreat to burrows for safety and rest [6].

WartOA incorporates two key behavioural mechanisms: a dominance-based selection strategy that models how warthogs establish hierarchy and compete for resources to drive global search and a burrow retreat strategy that simulates the warthog's instinct to return to safe zones, guiding local exploitation around previously successful solutions. Together, these mechanisms help the algorithm avoid early convergence and maintain variation among the solutions.

Organization

Here, the authors begin by briefly explaining the background of optimization algorithms, warthog behaviour and then introduce the proposed mathematical model and the key elements that define WartOA. This is followed by a detailed explanation of the feeding strategy that governs exploration and the burrow retreat strategy that controls exploitation. The authors then present the complete algorithm and evaluate its performance on a range of OEFs. Subsequently, WartOA is employed to solve a set of well-known ED problems to evaluate its effectiveness in practical scenarios. A comparative analysis with other well-known, nature-derived algorithms is conducted, and the authors conclude by summarizing their findings and outlining future scope.

Related Works

Optimization has long been an important theme of study in computational research. The foundation for many modern optimization techniques comes from classical as well as contemporary reviews that cover the theoretical and practical aspects of optimization. Some of these review papers on OAs by Alridha et al. [7], Abdulrahman et al. [8], etc. surveyed various approaches, including GD strategies and LP methods, that laid the foundation for tackling a wide variety of problems. Milestone contributions in early OAs include Holland's seminal work on GAs [2] and the development of PSO by Kennedy and Eberhart [3]. These classical approaches, while effective in certain contexts, often struggle with the complexities of high-dimensional and multimodal search spaces encountered in modern applications.

To address these issues, MAs had been developed as powerful alternatives. Methods like Ant Colony Optimization(ACO) [9] and Cuckoo Search(CS) [10] had shown significant improvement by incorporating more randomness and grouping behaviour. Their ability to balance exploration and exploitation makes them particularly well-suited for complex, real-world OPs. Building on the success of these methods, a new trend in research started to focus on nature-derived optimization techniques. Nature-derived optimization methods take direct inspiration from biological, physical, or ecological processes. These algorithms imitate natural phenomena such as evolution, swarming, foraging and even chemical reactions, and transforming them into mathematical frameworks for exploration and exploitation, making them highly robust and adaptable. OAs such as the Bat algorithm (BA) [11] and the Whale Optimization Algorithm (WOA) [12] incorporated behavioural patterns observed in these animals to enhance search efficiency and solution quality. These methods demonstrate that taking ideas from natural phenomena can lead to smarter and more adaptive solutions, especially in high-dimensional and dynamically changing environments.

Despite their success, many existing algorithms face limitations such as premature convergence, lack of diversity, and poor adaptability to dynamic environments. These challenges highlight the need for novel strategies that can more effectively maintain exploration-exploitation balance and adapt to changing problem landscapes. To address this gap, the development of the Warthog Optimization Algorithm (WartOA) is inspired by the unique ecological behaviours of common warthog ($Phacochoerusafricanus$). Detailed studies on warthog ecology and behaviour such as those by Cumming [13] and Somers et al. [14] have documented their adaptive foraging, social grouping, and defensive strategies, particularly their distinctive use of burrows for shelter. Additional insights provided by Edossa et al. [15] have further clarified how these behaviours contribute to survival in challenging environments. By transforming these natural strategies into a mathematical model, WartOA is designed to balance exploration (through foraging and resting behaviours) and exploitation (via burrow retreat behaviours) in the optimization process. Together, these lines of research, from classical optimization to modern metaheuristics and nature-derived algorithms, highlight how nature-derived ideas can solve modern tasks. WartOA uses the reference of this rich background by incorporating warthog-specific behaviours, offering a new approach that is particularly well-suited for high-dimensional problems in areas such as feature selection, image classification, intrusion detection, and broader applications in machine learning and artificial intelligence.

Inspiration and Warthog Behaviour

Common warthog ($Phacochoerusafricanus$) is member of the Suidae family, widely distributed across the grasslands, savannas, and open woodlands of sub-Saharan Africa. Adult males typically weigh between 60 and 150 kg and females between 50 and 75 kg, with a body length of 0.9 to 1.5 ms and standing at a shoulder height of 63 to 85 cm [16]. Warthogs are easily recognized by their large, flat heads with prominent facial warts (thick growths of skin known as warts hence its name) and curved tusks, which are used for defence and digging.

Warthogs are primarily grazers but exhibit omnivorous tendencies. They forage mainly on rhizomes in dry season and wide variety of grasses in wet season. They also have roots, bulbs, and occasionally have small animals and carrion. Warthogs use their snouts to dig and displace soil in search of food items like bulbs and grubs. Because their mouths are positioned lower on the head, they usually consume the food only after their snout is directly aligned above it [13].

Warthogs exhibit diverse grouping behaviour, forming four main types: lone adult males, bachelor groups, yearling groups, and matriarchal family groups. Bachelor groups consist of two to three adult males that may associate temporarily or stably during non-breeding periods. Yearling groups include young warthogs without adults, while matriarchal groups are led by grown up females with their young ones and piglets, known as sounders, often showing long-term stability [13]. Adult males, in contrast, are often solitary or form temporary bachelor herds, especially during mating seasons [14]. Additionally, group compositions may vary during the mating season, with temporary formations observed as stable groups fragment and re-form. This flexible grouping strategy helps optimize both safety and resource access in dynamic habitats.

Warthogs are also known for their strategic use of burrows, often occupying and modifying abandoned aardvark dens rather than digging their own. These burrows not only offer protection from predators but also sleeping area and thermoregulation at night [15]. Notably, warthogs enter burrows in reverse, allowing them to face outward to monitor threats and escape quickly if needed. This burrow retreat behaviour is a remarkable survival tactic that balances defence and mobility.

Two key behaviours of the warthog, namely dominance-based social structuring and strategic burrow retreat, form the biological inspiration behind the proposed Warthog Optimization Algorithm (WartOA). These behaviours are mathematically modelled within WartOA to establish a dynamic balance between exploration (imitating foraging behaviour) and exploitation (imitating retreat and defence strategies), making it well-suited for navigating complex and high-dimensional optimization landscapes.

Proposed WartOA

WartOA is influenced by the natural behaviours of warthogs, including feeding, resting, and seeking shelter. These behaviours are mathematically modelled to maintain exploration-exploitation equilibrium in the problem space, leading to efficient refinement of solution. Here, the proposed WartOA can be broken down into three main steps: feeding (exploitation), resting (exploration), and seeking shelter (exploration). To initialize each warthog with a random position within the solution domain before commencing their operations, the equation below is employed to cover the $N \times$ Dim solution space. The population size (N) directly affects exploration quality, and larger N enhances global search but increases computation, while smaller N may miss optimal regions.

$$pop_{i,j} = Minimum_j + (Maximum_j - Minimum_j) \times R \quad (1)$$

Here, R is a random value uniformly drawn from the range [0,1], facilitating variation among initial solutions. Each warthog within the group is regarded as a candidate solution. Population matrix *pop* is defined as shown in Eq. 2:

$$pop = \begin{bmatrix} pop_1 \\ \vdots \\ pop_i \\ \vdots \\ pop_N \end{bmatrix}_{N\times Dim} = \begin{bmatrix} pop_{1,1} & \cdots & pop_{1,j} & \cdots & pop_{1,Dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ pop_{i,1} & \cdots & pop_{i,j} & \cdots & pop_{i,Dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ pop_{N,1} & \cdots & pop_{N,j} & \cdots & pop_{N,Dim} \end{bmatrix}_{N\times Dim} \quad (2)$$

Each row in the matrix corresponds to a warthog's position (solution vector) within the solution domain. The OEF is evaluated for each candidate solution to determine its fitness, and the fitness values are stored in a corresponding evaluation vector as in Eq. 3.

$$\mathbf{F} = \begin{bmatrix} F_1 \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(pop_1) \\ \vdots \\ F(pop_N) \end{bmatrix}_{N\times 1} \quad (3)$$

Ages will be assigned to each warthog as well ranging from 1 to 15 according to their average lifespan [13]. There is a possibility of juveniles after the age of 2 years to disperse from the main group on their own. Here, it is signified by the group factor *group* and a random number:

$$\mathrm{AGE}_i = \mathrm{randint}(1, 15) \quad (4)$$

$$\mathbf{group} = \begin{bmatrix} \mathrm{group}_1 \\ \vdots \\ \mathrm{group}_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} g(\mathrm{AGE}_1, r_1) \\ \vdots \\ g(\mathrm{AGE}_N, r_N) \end{bmatrix}_{N\times 1} \quad (5)$$

where

$$g(\mathrm{AGE}_i, r_i) = \begin{cases} \mathrm{rand}(1,2), & \text{if } \mathrm{AGE}_i > 2 \text{ and } r_i < 0.5 \\ \mathrm{rand}(2,3), & \text{otherwise} \end{cases} \quad (6)$$

where rand(a,b) is a random number in the range [a, b], and r_i is a parameter controlling the likelihood of dispersal. As the foraging movements are influenced by the groups, the group factor plays a key role in the algorithm's position-updating equation.

Feeding Strategy (Exploitation)

In the feeding phase, warthogs search for food in their environment. This behaviour is modelled as exploitation, where warthogs move towards the leader warthog (referred to as the strongest warthog (SW)) in the search domain to find optimal the answer. The intensity of exploitation is influenced by the time of day, with warthogs tending to feed more in the afternoon [13]. The position adjustment formula for the exploitation phase is simulated using the following:

$$P = pop[i] + feeding\_activity \cdot (SW - group_i \cdot pop[i]) \quad (7)$$

where P is the updated location of the warthog, pop[i] signifies the present coordinates, *feeding_activity* is a randomly selected value within the interval [0,1], SW is the best-known solution so far, and group represents the herd size. The *feeding_activity* controls the intensity of movement toward the best solution (SW). Its variation determines how aggressively the solution converges: higher values can accelerate convergence but may risk local optima. The group size modulates social influence; larger groups increase collective behaviour, enhancing exploitation but reducing diversity.

Enhanced feeding activity (late afternoon under normal weather conditions):

$$P = pop[i] + feeding\_activity \cdot a \cdot (SW - group_i \cdot pop[i]) \quad (8)$$

where a is a random activity factor (1 or 2) that enhances activity during the late afternoon. If the time is morning, a check has been added which makes the warthogs revert back to a better grazing area if the latest position is worse than the previous one. This is represented by the following equation:

$$P = \begin{cases} pop[i] & \text{if } f(pop[i]) < f(prev\_locations[i]), \\ prev\_locations[i] & \text{otherwise}, \end{cases} \quad (9)$$

where *pre v_locations* is the previous location of the warthog.

Resting Strategy (Exploration)

After a period of feeding in the morning, at mid-day or if the weather is too hot or rainy, warthogs rest in shades to conserve energy or to get temporary shelter [6]. This resting phase allows warthogs to explore new random positions in the solution domain. The equation (10) below is used to replicate the characteristic foraging strategy of warthogs,

$$P = pop[i] + resting\_activity \cdot (shade - pop[i]) \tag{10}$$

where *shade* is a randomly selected position within the bounds representing a resting area, and *resting_activity* is a random value between 0 and 1. The *resting_activity* introduces randomness and simulates passive search during mid-day or adverse conditions. Its sensitivity lies in how much deviation from current positions occurs: higher values mean greater exploration. The shade position ensures that resting does not stray far from viable zones, maintaining moderate diversity.

Seeking Shelter Strategy (Exploration)

At night, warthogs retreat to the nearest available burrows in the wild [6] (or, in this context, within the search space) to rest and resume their food-searching activity the next day. This is represented by the following equation:

$$P = burrow\_activity \cdot burrows[closest\_burrow\_idx] \tag{11}$$

The distance to the closest burrow is evaluated using the Euclidean-distance formula:

$$d_j = \sqrt{\sum_{k=1}^{n} (pop[i]_k - burrows[j]_k)^2} \tag{12}$$

where *burrow_activity* is a random value between 0 and 1, *burrows[closest_burrow_idx]* is the closest burrow to the warthog, and d_j represents the actual distance between the warthog and the burrow. The *burrow_activity* factor controls movement toward historically successful regions (burrows), supporting convergence. d_j, the distance to the burrow, acts as a stabilizing factor—smaller d_j promotes local refinement. This strategy promotes convergence stability by encouraging warthogs to return to previously successful areas, reducing variance in position updates and maintaining proximity to known good solutions.

For reaching the optimal solution better, *prev_locations* is used to compare the new location in the morning with the location before going to the burrow at night. So before going to the closest burrow, the previous position is saved for future comparisons:

$$prev\_locations[i] = pop[i] \tag{13}$$

Algorithm 1 Warthog Optimization Algorithm(WartOA).

```
Require: Objective function: f(pop), dimension dim, population size
    N, maximum iterations T, lower bound lb, upper bound ub
Ensure: Best-discovered solution and its fitness evaluation
Initialize warthog population pop randomly within [lb, ub] using equation (1)
Calculate fitness F for each warthog position pop
Initialize array of previous locations prev_locations as copy of pop
Assign random ages AGE_i = randint(1, 15) to each warthog
Define time-of-day cycle: morning → mid-day → late-afternoon → night
Set current time index to 0
Generate num_burrows (15-25) random burrow positions within [lb, ub]
for t = 1 to T do
    Find the strongest warthog SW
    if t mod 10 = 0 then
        Regenerate num_burrows (15-25) random burrow positions
    end if
    Determine current time of day based on time index
    Increment time index (cycle back to morning after night)
    Randomly choose weather from {hot, rainy, normal}
    for i = 1 to N do
        Calculate group factor group_i using equations (5) and (6)
        if time is morning or late-afternoon and weather is normal then
            feeding_activity = random(0, 1)
            if time is late-afternoon then
                a = randint(1, 3)                ▷ Enhanced activity factor
                Calculate new position P using equation (8)
            else
                Calculate new position P using equation (7)
                if f(P) > f(prev_locations[i]) and t ≥ 4 then
                    Update position P according to equation (9)
                end if
            end if
        else if time is mid-day or weather is hot or rainy then
            resting_activity = random(0, 1)
            Generate random shade position within bounds
            Calculate new position P using equation (10)
        else if time is night then
            Store current position using equation (13): prev_locations[i] = pop[i]
            burrow_activity = random(0, 1)
            Calculate distances to all burrows using equation (12)
            Find closest burrow to warthog i
            Calculate new position P using equation (11)
        end if
        Clip position P to ensure it's within bounds [lb, ub]
        Calculate fitness F_new = f(P)
        if F_new < F[i] then
            pop[i] = P and F[i] = F_new
        end if
    end for
    Update best solution SW if better solution found
    Shuffle the AGE array
    if t mod 20 = 0 then
        Refresh all ages according to (4)
    end if
end for
return Best solution found and its fitness value
```

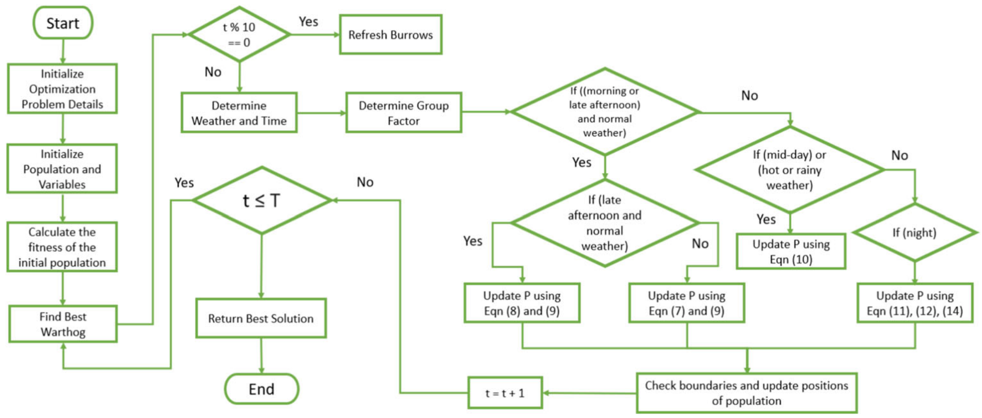

Fig. 1 Flowchart of the Warthog Optimization Algorithm (WartOA)

Results and Discussion

Benchmark Function Evaluation

To assess the performance of the proposed WartOA approach (Fig. 1), a set of 23 well-established test functions were utilized. These functions serve as a standard means for benchmarking optimization techniques. For fairness across comparisons, a uniform setup was used: each algorithm ran with a population size of 30 over 500 iterations. The detailed definitions of the functions are provided in Figs. 2, 3, and 4, while performance comparisons appear in Table 1.

WartOA's effectiveness was contrasted with several established metaheuristic algorithms: the Walrus Optimization Algorithm (WaOA) [17], particle swarm optimization (PSO) [3], gold rush optimizer (GRO) [18], Grey Wolf Optimizer (GWO) [19], and the capuchin search algorithm (CapSA) [20]. Each algorithm was executed 30 times per benchmark to account for randomness, and the outcomes were evaluated using average objective values (avrg) and standard deviations (std_dev).

For measuring fine-tuning ability, the authors focused on uni-modal functions (F1–F7), which are ideal for exploitation analysis due to their single optimum. WartOA consistently yielded superior results in these cases, confirming its strength in local search. The remaining functions (F8–F23) are multi-modal functions, which contain numerous local optima, and were chosen to test exploratory behaviour. WartOA excelled in exploring these complex landscapes, showing adaptability and range. Figures 5 and 6 visualize the topologies of these functions, highlighting the difficulties in locating global optima in each of these functions due to landscape complexity. The results demonstrate WartOA's capacity to balance intensification and diversification effectively, securing high-quality solutions across diverse scenarios. Its performance metrics (avrg, std_dev and bfv) frequently surpassed those of other methods, underlining its competitiveness in solving intricate OPs. For instance, WartOA outperformed PSO on 19 out of 23 functions (approximately 83%), achieving significantly lower average objective values particularly on unimodal benchmarks like F1–F6, demonstrating strong local search capability. Compared to WaOA, WartOA exhibited better consistency (lower standard deviation) on challenging multi-modal functions such as F8 and F20, highlighting its exploration adaptability. However, in certain cases like F7 and F14, WaOA marginally matched or exceeded WartOA in average performance, suggesting room for improvement in specific landscapes. Overall, WartOA achieved the best average result on over 70% of the benchmarks, underscoring its robustness across diverse optimization problems.

Function	Dim	Range	$f_{\min}$
$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n} \lvert x_i\rvert+\prod_{i=1}^{n} \lvert x_i\rvert$	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	30	[-100,100]	0
$F_4(x)=\max_i\left\{\lvert x_i\rvert, 1\leq i\leq n\right\}$	30	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-5,10]	0
$F_6(x)=\sum_{i=1}^{n}\left(\left[x_i+0.5\right]\right)^2$	30	[-100,100]	0
$F_7(x)=\sum_{i=1}^{n} i x_i^4+\text{random}[0,1)$	30	[-1.28,1.28]	0

Fig. 2 Unimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[-500,500]	$-418.9892 \times \text{Dim}$
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_{i+1}}{4},\ u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$F_{13}(x) = 0.1\sin^2(3\pi x_1) + 0.1\sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + 0.1(x_n - 1)^2[1 + \sin^2(2\pi x_n)] + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-50,50]	0

Fig. 3 Multimodal benchmark functions

Evaluation on Dataset

To evaluate the effectiveness of the proposed Warthog Optimization Algorithm (WartOA) in dimensionality reduction, a FS task was performed on the NSL-KDD dataset. The NSL-KDD dataset consists of 41 features extracted from network traffic data, including basic, content-based, and traffic-based attributes, and is labeled as either normal or one of several attack types [21]. It is widely used in intrusion detection system (IDS) research for benchmarking machine learning and feature selection techniques, as it provides a realistic yet manageable environment for evaluating detection accuracy and efficiency across various classification models. The objective was to identify a minimal yet highly informative subset of features trough WartOA that would improve classification accuracy(AC), reduce false positives(FPR), and enhance detection rates(DR). WartOA was applied to generate the optimal feature subset FSs1, consisting of 19 features (Table 2):

This subset was evaluated using four classifiers: neural network (NN), decision tree (DT), random forest (RF), and bagging, under two experimental settings:

1. Train-test split (phase 1)
2. Ten-fold cross-validation (phase 2)

WartOA-based feature selection (FSs1) significantly improved classifier performance in both evaluation phases. In phase 1 as tabulated in Table 3, the decision tree showed the most notable gain, with accuracy rising from 79.35 to 83.12% and FPR dropping from 7.64 to 2.99%, while bagging also improved substantially in accuracy and DR. In phase 2 as tabulated in Table 4, improvements were more modest but stable, with neural network achieving a slight boost in accuracy (99.30 to 99.39%) and FPR decreasing from 0.58 to 0.55%, and bagging further reducing its FPR. Across both phases, WartOA-based feature selection proved to be highly effective, particularly during initial train-test evaluations, where it consistently improved classifier performance. The improvements in detection rate and FPR reduction, especially in decision tree and bagging classifiers, indicate that FSs1 retains critical intrusion-related information while removing noise. The cross-validation results further validate the subset's generalizability, although slight fluctuations suggest classifier-specific sensitivity to feature selection, which could be explored further in future work. Overall, this analysis confirms that WartOA is capable of identifying compact, high-performing feature subsets, making it highly suitable for network intrusion detection tasks in high-dimensional environments.

WartOA Performance on Engineering Design (ED) Problems

In this section, three common ED problems from the structural field are utilized to demonstrate the effectiveness of the proposed WartOA in applied optimization challenges

Function	Dim	Range	$f_{\min}$
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[-5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[-5,5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[0,1]	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

Fig. 4 Fixed-dimension multimodal benchmark functions

Table 1 Comparison of optimization results obtained for the benchmark functions

F	WartOA		WaOA		PSO		GRO		CapSA		GWO	
	avrg	std	avrg	std	avrg	std	avrg	std	avrg	std	avrg	std
F1	5.65E−75	1.69E−74	0	0	1.36E−04	2.02E−04	2.02E−61	1.07E−60	1.11E−16	1.34E−14	6.59E−28	6.34E−05
F2	4.17E−39	7.78E−39	1.02E−294	0	0.042	0.045	1.20E−40	3.77E−40	1.06E−09	1.28E−10	7.18E−17	0.029
F3	8.31E−74	2.45E−73	0	0	70.126	22.119	8.18	19.34	1.08E−12	2.97E−11	3.29E−06	79.150
F4	5.45E−39	1.42E−38	2.12E−277	0	1.086	0.317	0.108	0.589	1.91E−08	2.38E−08	5.61E−07	1.315
F5	2.42E−07	3.21E−07	0	0	96.718	60.116	26.67	0.317	1.76E−09	7.45E−11	26.813	69.905
F6	2.55E−07	4.37E−07	0	0	1.02E−04	8.28E−05	0.070	0.051	1.01E−13	7.44E−11	0.817	1.26E−04
F7	0.039	0.030	1.43E−05	1.15E−05	0.123	0.045	0.006	0.006	2.22E−08	1.17E−06	0.002	0.100
F8	−12569	1.12E−06	−8881	153	−4841	1153	−8052	656	1218	762	−6123	−4087
F9	0.0	0.0	0	0	46.704	11.629	0.416	2.28	1.12E−13	1.53E−14	0.311	47.356
F10	4.44E−16	0.0	2.13E−15	1.74E−15	0.276	0.509	4.56E−15	6.49E−16	1.18E−07	1.64E−08	1.06E−13	0.078
F11	0.0	0.0	0	0	0.009	0.008	0	0	8.33E−13	9.67E−14	0.004	0.007
F12	5.14E−10	3.64E−10	1.57E−32	2.81E−48	0.007	0.026	0.004	0.003	4.11E−12	5.42E−11	0.053	0.021
F13	2.49E−09	4.51E−09	1.35E−32	2.81E−48	0.007	0.009	0.153	0.093	1.51E−12	5.47E−10	0.654	0.004
F14	0.998	0.0	0.998	1.02E−16	3.627	2.561	0.998	0	0.998	1.43E−02	4.042	4.253
F15	3.08E−04	4.42E−07	3.08E−04	9.87E−20	5.77E−04	2.22E−04	3.00E−04	3.70E−05	3.06E−04	8.46E−09	3.37E−04	6.25E−04
F16	−1.032	2.22E−16	−1.032	2.28E−16	−1.032	6.25E−16	−1.032	6.52E−16	−1.032	5.57E−12	−1.032	−1.032
F17	0.398	0.0	0.398	0	0.398	0	0.398	0	0.398	1.92E−09	0.398	0.398
F18	3.0	8.19E−16	3.0	5.76E−16	3.0	1.33E−15	3.0	1.06E−15	3.0	3.27E−08	3.000	3.0
F19	−3.863	2.60E−08	−3.863	2.28E−15	−3.863	2.58E−15	−3.863	2.70E−15	−0.768	1.35E−10	−3.863	−3.863
F20	−3.303	1.32E−02	−3.322	4.44E−16	−3.266	0.061	−3.322	6.00E−06	−0.515	5.54E−02	−3.287	−3.251
F21	−10.054	6.30E−09	−10.153	3.21E−15	−6.865	3.020	−10.153	8.10E−07	−3.251	2.58E−01	−10.151	−9.140
F22	−10.064	5.14E−09	−10.403	3.05E−15	−8.457	3.087	−10.403	1.89E−07	−4.208	1.77E−01	−10.402	−8.584
F23	−10.075	9.96E−09	−10.536	1.82E−15	−9.953	1.783	−10.536	4.41E−12	−2.314	4.28E−01	−10.534	−8.559

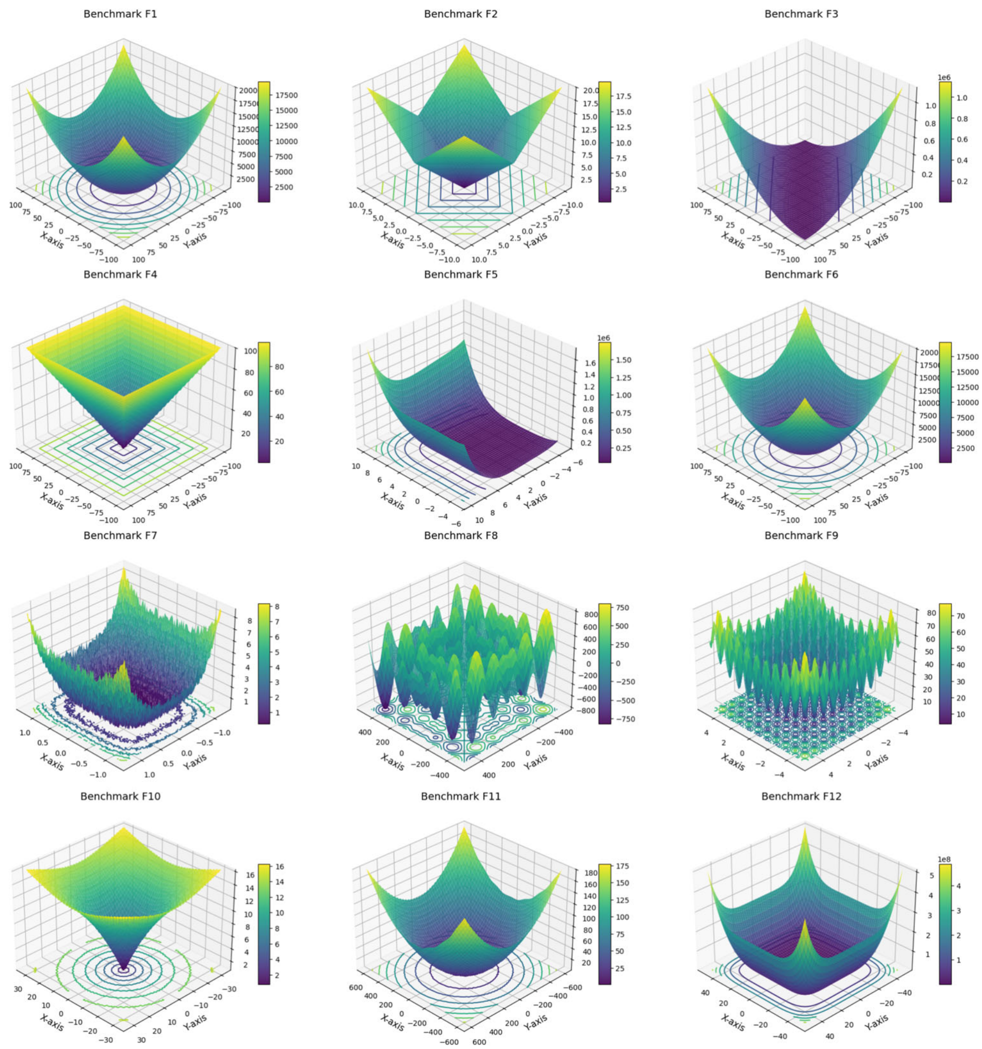

Fig. 5 3D mesh visualization of benchmark functions F1–F12

involving real-world constraints. These include the Cantilever Beam design (CBD), I-beam design (IBD), and Tension/Compression Spring Design (TSD). The experiment is conducted under identical settings with a 30-sized population and 500 iterations per run. WartOA's performance is benchmarked against gradient-based optimizer (GBO) [22], prairie dog optimization (PDO) [23], and Material Generation Algorithm (MGA) [24]. Each problem was executed independently 30 times, followed by an in-depth analysis and interpretation of the results.

Cantilever Beam Design Problem (CBD)

The CBD problem serves as a classic benchmark for evaluating OAs. The beam is divided into five segments, each featuring a hollow rectangular cross-section which can be seen in the Fig. 7. The design variables w_1 to w_5 represent adjustable dimensions (either the width or height) of these segments [25].

Let:

$$\vec{w} = [w_1, w_2, w_3, w_4, w_5] \tag{14}$$

Minimize:

$$p(\vec{w}) = 0.0624(w_1 + w_2 + w_3 + w_4 + w_5) \tag{15}$$

Constraint:

$$q(\vec{x}) = \frac{61}{w_1^3} + \frac{27}{w_2^3} + \frac{19}{w_3^3} + \frac{7}{w_4^3} + \frac{1}{w_5^3} - 1 \leq 0 \tag{16}$$

This inequality limits the downward deflection at the free-end of the cantilever-beam when subjected to a load. It must be ensured that the beam does not deflect more than allowed. The more the value of w_i stiffer is the beam and bends less, but the weight increases too, so there comes a trade-off.

Individual design variables are bounded as follows:

$$0.01 \leq w_1, w_2, w_3, w_4, w_5 \leq 100 \tag{17}$$

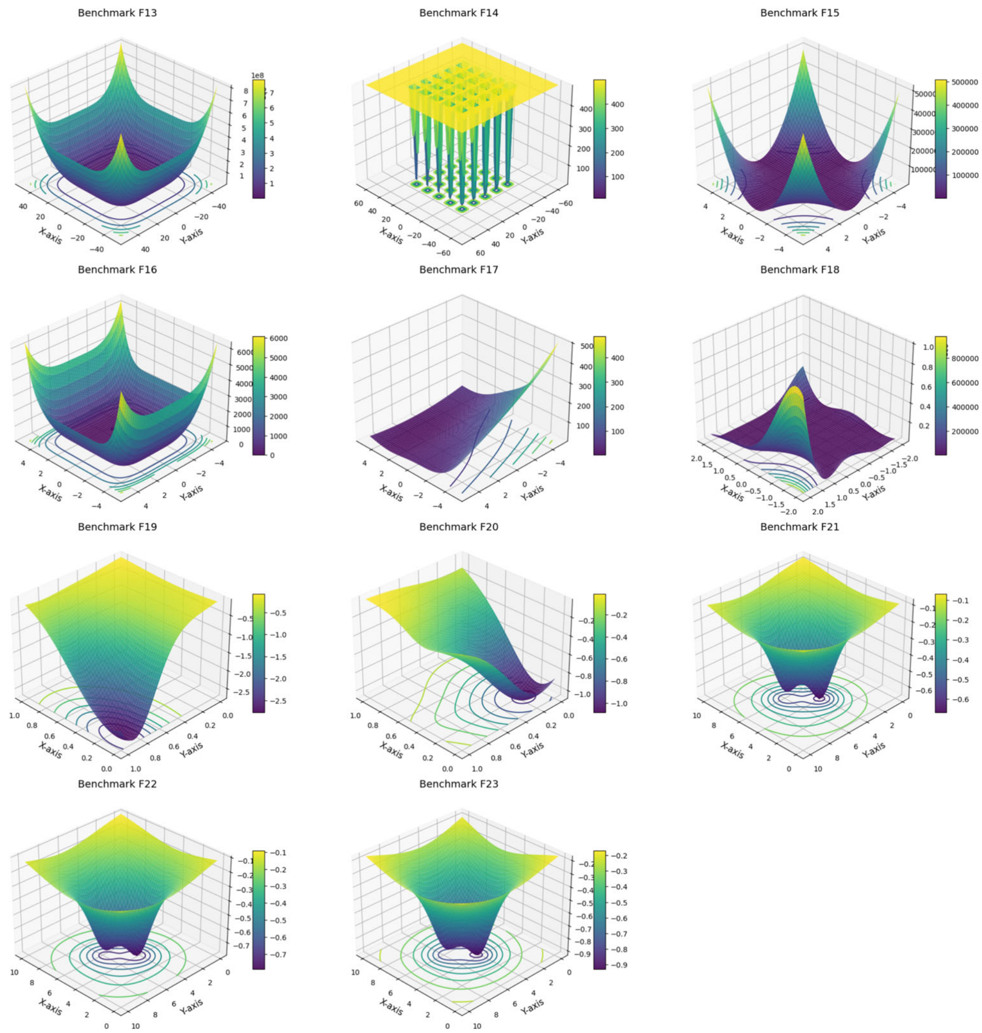

Fig. 6 3D mesh visualization of benchmark functions F13–F23

Table 2 Selected FSs using WartOA on NSL_KDD dataset

Feature subset (FSs)	No. of features	Features
FSs1	19	1, 3, 4, 5, 9, 10, 14, 15, 16, 17, 20, 27, 28, 30, 31, 34, 38, 39, 40

Table 3 Performance comparison in evaluation phase 1 using the NSL-KDD dataset

Classifier	Without FS			With FS			
	AC	DR	FPR	Feature subset	AC	DR	FPR
Neural network	77.182	62.417	3.306	FSs1	80.371	67.809	3.027
Decision tree	79.352	69.508	7.641	FSs1	83.117	72.601	2.986
Random forest	77.803	63.243	2.955	FSs1	80.402	67.583	2.656
Bagging	80.820	69.914	4.768	FSs1	83.632	73.552	3.048

Table 4 Performance comparison in evaluation phase 2 using NSL-KDD dataset

Classifier	Without FS			With FS			
	AC	DR	FPR	Feature subset	AC	DR	FPR
Neural network	99.303	99.207	0.584	FSs1	99.395	99.352	0.554
Decision tree	99.808	99.816	0.200	FSs1	99.727	99.743	0.289
Random forest	99.841	99.86	0.183	FSs1	99.445	99.407	0.511
Bagging	99.833	99.855	0.192	FSs1	99.784	99.720	0.158

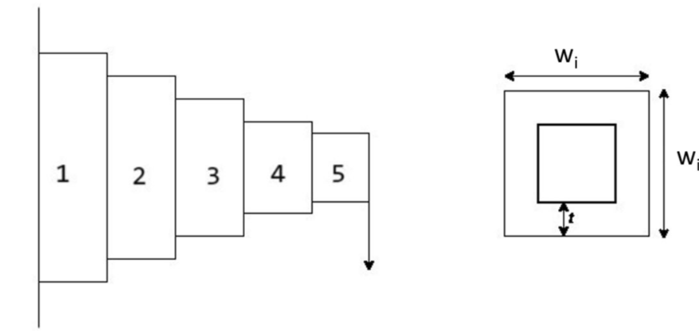

Fig. 7 Schematic view of CBD problem

Table 5 Performance comparison of CBD problem across various algorithms

Algorithm	Optimal solutions for variables					Minimized weight
	w_1	w_2	w_3	w_4	w_5	
GBO [22]	6.0124	5.3129	4.4941	3.5036	2.1506	1.3400
PDO [23]	5.6896	5.0208	4.2617	3.3130	2.0409	1.3004
MGA [24]	6.0117	5.3157	4.5107	3.4857	2.1503	1.3400
WartOA	5.9831	4.9153	4.4922	3.5101	2.3103	1.3202

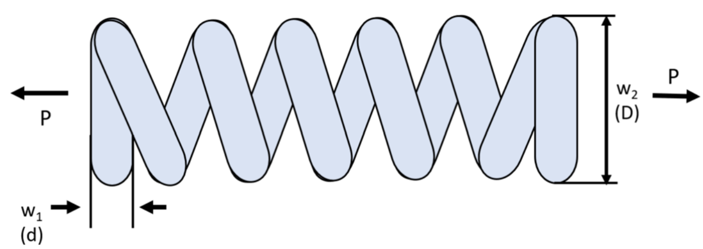

Fig. 8 Schematic view of TSD problem

Table 6 Algorithm-wise performance analysis of TSD problem

Algorithm	Optimal solutions for variables			Minimized weight
	w_1 (d)	w_2 (D)	w_3 (N)	
GBO [22]	0.0520	0.3651	10.8146	0.0127
PDO [23]	0.0517	0.3582	11.2038	0.0127
MGA [24]	0.0500	0.3104	15.0000	0.0126
WartOA	0.0530	0.3889	9.8445	0.0127

Table 5 summarizes the numerical performance of WartOA.

WartOA successfully identified the optimal configuration for the beam, yielding a minimum objective function value of 1.3201786190742018. The optimal design variables obtained by WartOA are as follows: $(w_1, w_2, w_3, w_4, w_5) = (5.98307794, 4.9153079, 4.49224618, 3.5101183, 2.31034478)$.

These values give a lightweight, adaptable, and optimized beam with no constraint violations, indicating WartOA's capability to effectively navigate the solution domain and reach more accurate and stable outcomes.

Tension/Compression Spring Design Problem (TSD)

The TSD problem (as per Fig. 8) is a well-known mechanical optimization task aimed at reducing the weight of a helical-spring while ensuring it meets key performance constraints. These constraints include allowable shear stress, natural frequency, and required deviation limits. The spring design involves three primary design variables: wire thickness (d), the average coil diameter (D), and the number of functioning coils (N) [5]. The objective of TSD is reducing the total weight (or volume) of the spring, which is directly proportional to its geometry.

Let:

$$\vec{w} = [w_1, w_2, w_3] = [d, D, N] \tag{18}$$

where w_1 is d, w_2 is D, and w_3 is N. The function to minimize is as follows:

$$p(\vec{w}) = (w_3 + 2)w_2 w_1^2 \tag{19}$$

Constraints:

$$q_1(\vec{w}) = 1 - \frac{w_2^3 w_3}{71785 w_1^4} \leq 0 \tag{20}$$

$$q_2(\vec{w}) = \frac{4w_2^2 - w_1 w_2}{12566(w_2 w_1^3 - w_1^4)} + \frac{1}{5108 w_1^2} \leq 0 \tag{21}$$

$$q_3(\vec{w}) = 1 - \frac{140.45 w_1}{w_2^2 w_3} \leq 0 \tag{22}$$

$$q_4(\vec{w}) = \frac{w_1 + w_2}{1.5} - 1 \leq 0 \tag{23}$$

Individual design variables are bounded as follows:

$$0.05 \leq w_1 \leq 2.0, \quad 0.25 \leq w_2 \leq 1.30, \quad 2.00 \leq w_3 \leq 15.00 \tag{24}$$

Table 6 summarizes the numerical performance of WartOA.

The Warthog Optimization Algorithm (WartOA) was applied to this problem and successfully found an optimal design configuration that meets all constraints while achieving a minimized weight objective. The optimal solution parameters obtained are as follows: $(w_1, w_2, w_3) = (0.052961, 0.38885515, 9.84452432)$ with a corresponding objective function value of 0.0127 and zero constraint violations.

These findings reflect WartOA's ability to reliably produce lightweight and feasible spring designs while maintaining strong convergence behaviour, competing against existing algorithms.

I-Beam Design (IBD) Problem

The structural optimization of an I-beam represents a well-known engineering problem, where the primary objective is to reduce the beam's overall mass without compromising its ability to withstand bending stresses. The I-beam geometry comprises three main components: the upper-flange, the central-web, and the lower-flange, as illustrated in Fig. 9. By

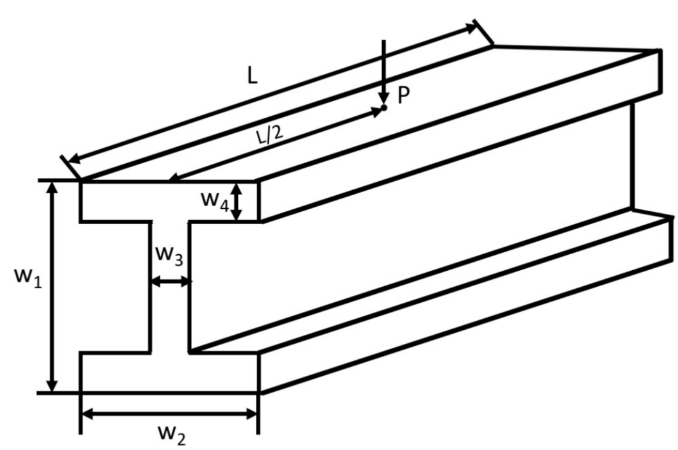

Fig. 9 Schematic view of IBD problem

Table 7 Evaluation of algorithmic performance for IBD problem

Algorithm	Optimal solutions for variables				Minimized weight
	w_1 (v)	w_2 (u)	w_3 (W_t)	w_4 (F_t)	
GBO [22]	50.0000	80.0000	0.9000	2.3217	0.0131
PDO [23]	80.0000	50.0000	0.9000	2.3218	0.0131
MGA [24]	49.9999	79.9999	0.9000	2.3217	0.0130
WartOA	80.0000	28.0000	5.0000	5.0000	0.0093

adjusting the dimensions of these components, one can balance between minimizing weight and satisfying structural performance criteria. I-beam's weight is dependent on its cross-sectional area. The optimization parameters include the flange width (v), the overall height of the beam section (u), the web thickness (W_t), and the flange thickness (F_t) [26].

Let:

$$\vec{w} = [w_1, w_2, w_3, w_4] = [v, u, W_t, F_t] \tag{25}$$

where w_1 is v, w_2 is u, w_3 is W_t, and w_4 is F_t. Assuming constant material density and beam length, the design problem can be formulated by minimizing the cross-sectional area:

$$p(\vec{w}) = 5000\left(\frac{1}{12w_3(w_1-2w_4)^3} + \frac{1}{6}w_2w_4^3 + 2w_2w_4\left(\frac{w_1-w_4}{2}\right)^2\right) \tag{26}$$

Constraints:

$$q_1(\vec{w}) = 2w_2w_4 + w_3(w_1-2w_4)^3 \le 300 \tag{27}$$

This constraint restricts the vertical tip deflection under load, ensuring that the beam does not bend excessively.

$$q_2(\vec{w}) = \frac{180,000w_1}{w_3(w_1-2w_4)^3} + \frac{2w_2w_4[4w_4^2+3w_1(w_1-w_4)]}{15,000w_2} + \frac{1}{(w_1-2w_4)w_3^3+2w_4w_2^3} \le 6 \tag{28}$$

This inequality ensures that the combined stress effects (which include contributions from bending and axial loads) remain within acceptable limits.

Individual design variables are bounded as follows:

$$10 \le w_1 \le 80, \quad 10 \le w_2 \le 50, \quad 0.9 \le w_3 \le 5, \quad 0.9 \le w_4 \le 5 \tag{29}$$

Table 7 summarizes the numerical performance of WartOA.

WartOA successfully identified the optimal configuration for the I-beam, yielding a minimum objective function value of 0.0093. The optimal design variables determined by WartOA are: $(w_1, w_2, w_3, w_4) = (80.0000, 27.99994, 5.0000, 5.0000)$.

These dimensions give a lightweight, robust I-beam with zero bending stress violations, showing WartOA's competitiveness with other existing algorithms in solving this problem.

Conclusion and Potential Advancements in Future

The authors here proposed a new nature-derived MA called the Warthog Optimization Algorithm (WartOA), designed by modelling the intelligent behavioural patterns of warthogs in the wild particularly their strategies for feeding, resting, and retreating into burrows for defence. These behaviours were mathematically formulated to guide the process of optimization through exploration-exploitation stages, enabling the algorithm to navigate complex problem domains with improved efficiency. To validate its effectiveness, WartOA was rigorously tested on 23 diverse OEFs, including unimodal and multi-modal, and was further evaluated by optimizing several real-world ED problems, such as CBD, TSD, and IBD. The experimental findings highlighted that WartOA consistently delivered strong performance across various problem scenarios. In many cases, it even outperformed several widely recognized optimization methods. This is largely due to its adaptive mechanism that intelligently shifts between scanning the wider solution space for new possibilities and concentrating on fine-tuning the best candidates already discovered. This approach not only improves the reliability of the solutions but also ensures they are reached in fewer steps, making the algorithm both time-efficient and robust against diverse problem landscapes.

Given the notable effectiveness of WartOA, there is a wide scope for its future development and application. One of the most significant challenges in modern computing lies in the management and processing of high-dimensional data. Dimensionality reduction through effective feature selection has become essential for achieving more accurate models, efficient storage utilization, and reduced computational cost. WartOA demonstrates strong potential to tackle high-dimensional problems and enhancing model performance in big data scenarios. WartOA also has strong potential in image classification, network intrusion detection, bioinformatics, and sensor selection where intelligent reduction and optimization can significantly impact accuracy and efficiency. Investigating hybrid forms of WartOA, particularly by blending it with other optimization techniques such as GA or PSO, may lead to faster convergence and improved solution quality.

Furthermore, WartOA could be extended into multi-objective, binary, and discrete versions, making it suitable for a broader class of real-world problems. In machine learning, WartOA could contribute to hyperparameter tuning, model structure optimization, and automated feature engineering. It can also be applied to deep learning and AI tasks like neural network architecture searching. Its lightweight yet powerful design also makes it a suitable candidate for autonomous systems, IoT device optimization, and real-time adaptive control systems. Given its adaptive capabilities, WartOA may also be explored for optimizing parameters in post-quantum cryptographic(PQC) algorithms, especially those involving complex structures like lattice problems and error-correcting codes.

As data continues to grow exponentially in modern IT and CS domains, and as optimization remains at the core of automation, AI, and decision-making processes, the significance of such adaptive, intelligent algorithms becomes even more critical. The biologically inspired WartOA, with its demonstrated robustness and versatility, stands as a valuable addition to the growth of optimization methods, offering real-world impact and a strong foundation for future research.

Author Contributions All the authors contributed equally to this work.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Rodríguez N, Gupta A, Zabala PL, Cabrera-Guerrero G (2018) Optimization algorithms combining (meta)heuristics and mathematical programming and its application in engineering. Math Probl Eng 2018:3967457. https://doi.org/10.1155/2018/3967457. https://arxiv.org/abs/onlinelibrary.wiley.com/doi/pdf/10.1155/2018/3967457
2. Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. The MIT Press, Cambridge, MA, USA. https://doi.org/10.7551/mitpress/1090.001.0001
3. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - international conference on neural networks, vol 4, pp 1942–19484. https://doi.org/10.1109/ICNN.1995.488968
4. Sadeeq HT, Abdulazeez AM (2022) Giant Trevally Optimizer (GTO): a novel metaheuristic algorithm for global optimization and challenging engineering problems. IEEE Access 10:121615–121640. https://doi.org/10.1109/ACCESS.2022.3223388
5. Xiao Y, Guo Y, Cui H, Wang Y, Li J, Zhang Y (2022) Ihaoavoa: an improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems. Math Biosci Eng 19:10963–11017. https://doi.org/10.3934/mbe.2022512
6. Furstenburg D (2009) Focus on the warthog (Phacochoerus africanus) 12035:38–41
7. Alridha A, Alsharify F, Al-Khafaji Z (2024) A review of optimization techniques: applications and comparative analysis. Iraqi J Comput Sci Mathem 5:122–134. https://doi.org/10.52866/ijcsm.2024.05.02.011
8. Abdulrahman A, Ali ZM, Zaki AM, H.Rizk FH, Eid MM, EL-Kenawy E-SME (2024) Exploring optimization algorithms: a review of methods and applications. J Artif Intell Metaheuristics 7(2):08–17. https://doi.org/10.54216/JAIM.070201
9. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern, Part B (Cybernetics) 26(1):29–41. https://doi.org/10.1109/3477.484436
10. Yang X-S, Deb S (2009) Cuckoo search via lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC), pp 210–214. https://doi.org/10.1109/NABIC.2009.5393690
11. Yang X-S (2010) In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) A new metaheuristic bat-inspired algorithm, pp 65–74. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12538-6_6
12. Mirjalili S, Lewis A (2016) The Whale Optimization Algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
13. Cumming D (1975) A field study of the ecology and behaviour of warthog. Museum Memoir No 7, Natl Mus Monuments Rhod 7:1–179
14. Somers MJ, Rasa OAE, Penzhorn BL (1995) Group structure and social behaviour of warthogs Phacochoerus aethiopicus 40(3):257–281. Data are presented on aspects of the group structure and social behaviour of warthogs Phacochoerus aethiopicus (Pallas, 1767) in the Andries Vosloo Kudu Reserve (AVKR). Eastern Cape Province, South Africa
15. Edossa A, Bekele A, Debella HJ (2020) Social organization and activity patterns of common warthog (Phacochoerus africanus Gmelin, 1788) in dabena valley forest, western ethiopia. Ecol Evol Biol 5(4):173–181. https://doi.org/10.11648/j.eeb.20200504.18
16. Mason DR (1982) Studies on the Biology and Ecology of the Warthog'Phacochoerus Aethiopicus Sundevalli'Lonnberg, 1908 in Zululand. https://repository.up.ac.za/handle/2263/25861

17. Trojovský P, Dehghani M (2022) Walrus Optimization Algorithm: a new bio-inspired metaheuristic algorithm. https://doi.org/10.21203/rs.3.rs-2174098/v1
18. Zolfi K (2023) Gold Rush Optimizer. A new population-based metaheuristic algorithm. Oper Res Decisions 33(1):113–150. https://doi.org/10.37190/ord230108
19. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
20. Braik M, Sheta A, Al-Hiary H (2021) A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. Neural Comput Appl 33(7):2515–2547. https://doi.org/10.1007/s00521-020-05145-6
21. Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD cup 99 data set. In: 2009 IEEE Symposium on computational intelligence for security and defense applications, pp 1–6. https://doi.org/10.1109/CISDA.2009.5356528
22. Ahmadianfar I, Bozorg-Haddad O, Chu X (2020) Gradient-based optimizer: a new metaheuristic optimization algorithm. Inf Sci 540:131–159. https://doi.org/10.1016/j.ins.2020.06.037
23. Ezugwu AE, Agushaka JO, Abualigah L, Mirjalili S, Gandomi AH (2022) Prairie dog optimization algorithm. Neural Comput Appl 34(22):20017–20065. https://doi.org/10.1007/s00521-022-07530-9
24. Talatahari S, Azizi M, Gandomi AH (2021) Material generation algorithm: a novel metaheuristic algorithm for optimization of engineering problems. Processes 9(5). https://doi.org/10.3390/pr9050859
25. Gezici H, Livatyalı H (2022) Chaotic Harris hawks optimization algorithm. J Comput Des Eng 9(1):216–245. https://doi.org/10.1093/jcde/qwab082
26. Rani R, Jain S, Garg H (2024) A review of nature-inspired algorithms on single-objective optimization problems from 2019 to 2023. Artif Intell Rev 57(5):126. https://doi.org/10.1007/s10462-024-10747-w