Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Volcanic Activity Optimizer for High-Dimensional Gene Selection: A Novel Nature-Inspired Metaheuristic

Seyyid Ahmed Medjahed

seyyidahmed.medjahed@univ-relizane.dz

Ahmed Zabana University - Relizane

Mourad Bouatouche

Ahmed Zabana University - Relizane

Fatima Boukhatem

University of Djilali Liabes

Research Article

Keywords: Volcanic Activity Optimizer, Metaheuristics, Optimization, Gene Selection, DNA microarray, Feature Selection

Posted Date: October 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11289699/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Volcanic Activity Optimizer for High-Dimensional Gene Selection: A Novel Nature-Inspired Metaheuristic

Seyyid Ahmed Medjahed[1*], Mourad Bouatouche[1*] and Fatima Boukhatem[2†]

[1*]Department of Computer Science, University of Relizane, Algeria.
[2]Faculty of Sciences and Technology, University of Djilali Liabes, Algeria.

Contributing authors: seyyidahmed.medjahed@univ-relizane.dz;
[†]These authors contributed equally to this work.

Abstract

This paper introduces the Volcanic Activity Optimizer (VAO), a novel nature-inspired metaheuristic for solving complex optimization problems, with particular emphasis on gene selection from high-dimensional gene expression data. Inspired by volcanic dynamics, VAO models the search process through three complementary phases: eruption for global exploration, lava flow for exploitation, and cooling for local intensification. To achieve an effective balance between exploration and exploitation, VAO incorporates a logarithmic cooling schedule and an adaptive step-size mechanism, promoting search diversity in the early stages and precise convergence in later iterations. The performance of VAO is first evaluated on 13 well-established benchmark functions representing unimodal, multimodal, and composite landscapes. The results demonstrate that VAO consistently obtains competitive solutions with strong convergence and stability. Furthermore, VAO is applied to gene selection to identify informative and compact gene subsets for classification. Experimental results show that VAO achieves superior or competitive performance in terms of solution quality, convergence and classification performance.

Keywords: Volcanic Activity Optimizer, Metaheuristics, Optimization, Gene Selection, DNA microarray, Feature Selection

1 Introduction

Metaheuristic algorithms have emerged as powerful optimization tools due to their simplicity, flexibility, and robustness in tackling a wide range of real-world problems. One of the main advantages of metaheuristics is their simplicity: most are inspired by natural processes, biological behaviors, or physical phenomena, making them intuitive to design and relatively easy to implement [1, 2]. This simplicity not only facilitates the development of new metaheuristic algorithms but also enables the hybridization and enhancement of existing approaches [3]. Among their numerous applications, gene selection has become an important optimization problem, particularly in high-dimensional gene expression data, where the number of genes can substantially exceed the number of available samples. Identifying a small subset of informative genes is therefore essential for reducing dimensionality, eliminating irrelevant and redundant features, and improving the performance and interpretability of downstream classification models. However, gene selection remains a challenging combinatorial optimization problem due to its large search space and the complex interactions among genes, highlighting the need for efficient and robust optimization strategies.

Another important feature of metaheuristics is their flexibility. Since they treat optimization problems as black-boxes, they can be applied to diverse problem domains without requiring structural changes in the algorithm. Only the problem representation and the evaluation function are needed, making them highly adaptable to engineering, machine learning, and real-world optimization challenges [4].

Unlike classical gradient-based optimization methods, metaheuristics typically adopt derivative-free stochastic mechanisms. They initiate the search process from randomly generated solutions and rely on probabilistic operators instead of derivative information. This makes them particularly well-suited for solving complex, non-differentiable, multimodal, and highly nonlinear problems where analytical gradients are unavailable or computationally expensive [5].

A further strength of metaheuristics is their ability to escape local optima. Owing to their stochastic nature and exploration mechanisms, they can navigate rugged and unknown search landscapes effectively, avoiding premature convergence and exploring multiple promising regions. This property has made metaheuristics a preferred choice for tackling high-dimensional and multimodal optimization problems. Optimization problems are central in science and engineering, with applications ranging from structural design and scheduling to image analysis and machine learning. Many real-world optimization tasks are nonlinear, multimodal, or constrained, making exact optimization infeasible [3–5]. Metaheuristics have thus emerged as powerful problem-solving techniques capable of providing near-optimal solutions at a reasonable computational cost [6–8].

Over the past decades, several families of metaheuristics have been introduced. Evolutionary Algorithms such as the Genetic Algorithm (GA) [9] use the principles of Darwinian evolution, while Swarm Intelligence techniques like Particle Swarm Optimization (PSO) [10] and Ant Colony Optimization (ACO) [11] mimic the collective behaviors of animals. In recent years, Grey Wolf Optimizer (GWO) [12], Whale Optimization Algorithm (WOA) [13], Firefly Algorithm (FA) [14], Bat Algorithm (BA) [15], and Narwhal Optimizer (NO) [16] have enriched the field. Within the broad family of

metaheuristics, a significant branch is Swarm Intelligence (SI). Introduced in the early 1990s, SI is defined as the collective emergent intelligence of groups of simple agents. Algorithms in this category are inspired by the cooperative behaviors of biological populations such as flocks of birds, schools of fish, ant colonies, and bee swarms. Some of the most popular SI-based metaheuristics include Ant Colony Optimization (ACO) [11] and Particle Swarm Optimization (PSO) [10]. SI algorithms are appealing because they (i) preserve useful information about the search space across iterations, (ii) utilize memory to store the best solutions, (iii) require relatively few control parameters, and (iv) are easy to implement compared to evolutionary algorithms such as Genetic Algorithms (GA). Despite this progress, many metaheuristics face the fundamental challenge of maintaining balance between exploration (diversification) and exploitation (intensification). Premature convergence often leads to stagnation, whereas excessive randomness delays convergence. Thus, the design of novel algorithms with adaptive mechanisms remains an open area of research. Motivated by this need, this paper introduces the Volcanic Activity Optimizer (VAO), inspired by the natural dynamics of volcanic activity. Metaheuristic algorithms can be broadly classified according to their source of inspiration. Evolutionary algorithms (EAs), such as Genetic Algorithms (GA) and Differential Evolution (DE), imitate biological evolution through operations like mutation, crossover, and selection. Swarm intelligence algorithms exploit the collective behavior of social organisms, with Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA) being notable examples. Physics-based metaheuristics are grounded in physical principles, including Simulated Annealing (SA), Gravitational Search Algorithm (GSA), and Big BangBig Crunch. Finally, human-based metaheuristics model social learning and decision-making, such as Teaching-Learning-Based Optimization (TLBO) and Brainstorm Optimization (BSO). This taxonomy highlights the richness and diversity of approaches within the field.

Another important perspective is the classification of algorithms based on their search dynamics. Single-solution-based algorithms (e.g., SA, Tabu Search) iteratively refine one candidate solution, while population-based algorithms (e.g., PSO, GA, GWO) simultaneously evolve a set of candidate solutions. Population-based methods are generally more robust for high-dimensional, multimodal problems due to their inherent diversity.

Recent advancements in metaheuristics emphasize hybridization and adaptive parameter control. Hybrid metaheuristics combine the strengths of different paradigms, such as blending global exploration from swarm-based methods with strong local exploitation from evolutionary or physics-based approaches. Adaptive and self-adaptive algorithms dynamically tune control parameters (e.g., mutation rate, inertia weight, temperature) based on feedback from the optimization process, improving robustness and generalizability across problem domains.

Despite the vast progress, no single metaheuristic consistently outperforms others across all problem types, a phenomenon known as the No Free Lunch (NFL) theorem. This motivates the continuous design of new algorithms that incorporate novel inspiration sources and adaptive mechanisms to achieve better explorationexploitation balance in diverse applications.

Nature-inspired algorithms are often divided into four broad categories: evolutionary-based, swarm-based, physics-based, and human-based approaches. Evolutionary algorithms (EAs) include Genetic Algorithm (GA) and Differential Evolution (DE), while swarm-based approaches include Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA). Physics-based algorithms exploit physical phenomena and include methods such as Simulated Annealing (SA) and the Gravitational Search Algorithm (GSA). Human-based heuristics include Teaching-Learning-Based Optimization (TLBO). These diverse approaches have demonstrated considerable potential in addressing challenging optimization problems, including feature selection and gene selection in high-dimensional datasets.

Regardless of their underlying inspiration, most metaheuristic algorithms operate by balancing two complementary processes: exploration and exploitation. Exploration refers to the global search of the solution space to discover promising regions, whereas exploitation focuses on locally refining solutions within these regions. Maintaining an effective balance between exploration and exploitation remains one of the central challenges in metaheuristic optimization. This challenge becomes particularly pronounced in gene selection, where the extremely high dimensionality of gene expression data results in a vast combinatorial search space. An effective optimizer must therefore identify informative genes while eliminating irrelevant and redundant ones without sacrificing classification performance.

Motivated by these challenges, this paper introduces a novel nature-inspired optimization method, the Volcanic Activity Optimizer (VAO). Inspired by volcanic eruption dynamics, VAO structures the search process into three complementary phases: explosive exploration (eruption), moderate exploitation (lava flow), and fine-grained intensification (cooling). By combining a slow cooling schedule with adaptive step sizes, VAO seeks to maintain population diversity during the early search stages while progressively improving the precision of the search in later iterations. Beyond its evaluation on standard mathematical benchmark functions, VAO is specifically employed for gene selection from high-dimensional gene expression datasets, where the objective is to identify compact and informative gene subsets that enhance classification performance.

VAO distinguishes itself from existing metaheuristic approaches through its dynamic modeling of two complementary search behaviors: (1) sudden large-scale exploration inspired by volcanic eruptions and (2) progressive local exploitation driven by lava flow and cooling dynamics. Although the cooling mechanism shares a conceptual similarity with the temperature decay principle of Simulated Annealing, VAO extends this idea within a population-based search framework, incorporating adaptive search steps and diversity-preserving mechanisms. This design makes VAO particularly suitable for complex combinatorial problems such as high-dimensional gene selection, where achieving an effective trade-off between exploration, exploitation, and subset compactness is essential.

The rest of the paper is organized as follows. Section 2 presents the mathematical formulation and algorithmic design of the proposed Volcanic Activity Optimizer (VAO). Section 3 introduces the proposed VAO-based gene selection approach. Section

4 presents the experimental evaluation of VAO on benchmark functions and gene selection datasets. Finally, Section 5 concludes the paper and discusses potential directions for future research.

2 The proposed Volcanic Activity Optimizer

2.1 Inspiration

The Volcanic Activity Optimizer (VAO) is inspired by the natural process of a volcanic eruption. At the beginning of an eruption, magma and gas are violently expelled, spreading material in all directions. This is modeled as a phase of broad exploration where candidate solutions are widely perturbed.

Fig. 1 Eruption volcanic. Images taken from www.britannica.com

As time passes, the eruption stabilizes into a *lava flow* stage, in which molten rock advances steadily but with less intensity. In this stage, the algorithm shifts toward exploitation, guiding solutions closer to the best-known position but still maintaining some stochastic perturbations.

Fig. 2 Lava volcanic. Images taken from www.britannica.com

Finally, as the lava cools and solidifies, the process transitions into a stage of *local intensification*. Here, the algorithm focuses on refining solutions in a small neighborhood around the best candidate, using an adaptive step size that decreases slowly over time. This three-phase mechanism (explosion, flow, and intensification) ensures a natural balance between exploration and exploitation, mimicking the lifecycle of a volcanic event.

Fig. 3 Cooling volcanic. Images taken from www.britannica.com

2.2 Mathematical Model

In VAO, a solution represents the position of volcanic material (particles) within the search space, which corresponds to a potential solution of the optimization problem. The dynamic behavior of volcanoes is modeled through different phases that govern how these solutions evolve. During the eruption phase, volcanic particles are explosively dispersed, enabling wide exploration of the search landscape. This stage mimics the natural scattering of ash and magma, allowing solutions to spread broadly and investigate diverse regions.

As the eruption stabilizes, the lava flow phase facilitates moderate exploitation. Here, the particles gradually converge toward promising regions, analogous to lava flowing downhill and concentrating around valleys. Finally, during the cooling phase, the system undergoes fine-grained local search, where step sizes are reduced to intensify the search around high-quality solutions, similar to solidified lava consolidating around stable regions.

Through these mechanisms, VAO integrates large-scale exploration, progressive refinement, and local intensification to efficiently locate global optima while avoiding premature convergence.

Initially, the VAO algorithm begins with a population of randomly generated solutions, which represent the initial positions of volcanic particles (X). These particles symbolize potential solutions within the search space. As the algorithm progresses, their positions are iteratively updated according to the dynamic phases of volcanic activity (eruption, lava flow, and cooling). The population at iteration t is expressed by the following matrix:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & ... & X_{1,d} \\ X_{2,1} & X_{2,2} & ... & X_{2,d} \\ ... & ... & ... & ... \\ X_{n,1} & X_{n,2} & ... & X_{n,d} \end{bmatrix} \tag{1}$$

where n is the number of search agents (volcanic particles) and d is the dimension of the problem.

Let t denote the current iteration and T the maximum number of iterations. The cooling schedule is logarithmic to ensure very slow decay:

$$\sigma(t) = \max\left(\sigma_{\min}, \frac{\sigma_0}{\log(t+e)}\right), \tag{2}$$

where σ_0 is the initial temperature and $\sigma_{\min}$ prevents numerical stagnation. Additionally, a control parameter a decreases linearly:

$$a(t) = 2 - \frac{2t}{T}. \tag{3}$$

The position update depends on the evolutionary phase of the algorithm:

2.2.1 Eruption phase

In the eruption phase, the algorithm emphasizes exploration by allowing candidate solutions to make large jumps in the search space. This behavior prevents premature convergence by ensuring that the population does not get trapped in local optima during the early stage of the optimization process. The update rule introduces stochastic perturbations scaled by both the distance from the best solution and the search space bounds, thereby enabling a wide and diverse sampling of potential solutions.

This phase is defined as follows: ($t/T < 0.2$): large exploratory jumps are applied

$$X_{i,j}^{t+1} = X_{i,j}^{t} + \sigma(t) \cdot (2r-1)\left(|X_{i,j}^{t} - X_{j}^{*}| + (ub_j - lb_j)r\right), \tag{4}$$

where X^* is the best solution, ub, lb are bounds, and $r \sim U(0,1)$.

2.2.2 Lava flow phase

During the lava flow phase, the search transitions from broad exploration to more focused exploitation around the current best solution. In this stage, the position update is driven by the relative distance to the best solution, with stochastic variation still

present to maintain diversity. This balance allows the algorithm to refine promising areas of the search space while avoiding excessive random wandering.

This phase is described as follows: ($0.2 \leq t/T < 0.3$): focused exploitation toward the best solution

$$X_{i,j}^{t+1} = X_j^* + \sigma(t) \cdot (2r - 1) \cdot \left| X_{i,j}^t - X_j^* \right|. \tag{5}$$

2.2.3 Intensification phase

Finally, in the intensification phase, the algorithm concentrates on fine-tuning solutions in the vicinity of the best candidate. The adaptive update mechanism incorporates both a distance factor and an exponential decay term, which gradually reduces exploration as the search progresses. This mechanism ensures convergence by applying smaller, more precise adjustments to solution positions, thereby improving solution quality in the later iterations.

This phase is defined as follows: ($t/T \geq 0.3$): adaptive fine-tuning

$$X_{i,j}^{t+1} = X_j^* + (2ar - a) \cdot \Delta \cdot \left(1 + 0.5r\, e^{-t/T}\right), \tag{6}$$

where $\Delta = |2rX_j^* - X_{i,j}^t|$ is a local distance factor.

2.3 Proposed Algorithm VAO

The pseudo-code of the Volcanic Activity Optimizer (VAO) is given in Algorithm 1.

Algorithm 1 Volcanic Activity Optimizer (VAO)

Initialize population X_i $(i = 1, \ldots, N)$ randomly in $[lb, ub]$
Evaluate fitness $f(X_i)$ and find best solution X^*
for $t = 1$ to T **do**
    Update cooling factor: $\sigma(t) = \max(\sigma_{\min}, \sigma_0 / \log(t + e))$
    Update control parameter: $a(t) = 2 - 2t/T$
    for each agent $i = 1, \ldots, N$ **do**
        for each dimension $j = 1, \ldots, D$ **do**
            Compute phase ratio: $\phi = t/T$
            if $\phi < 0.2$ **then** ▷ Eruption (exploration)
                $X_{i,j}^{t+1} = X_{i,j}^t + \sigma(t)(2r - 1)(|X_{i,j}^t - X_j^*| + (ub_j - lb_j)r)$
            else if $\phi < 0.3$ **then** ▷ Lava flow (exploitation)
                $X_{i,j}^{t+1} = X_j^* + \sigma(t)(2r - 1)|X_{i,j}^t - X_j^*|$
            else ▷ Intensification (local search)
                $\Delta = |2rX_j^* - X_{i,j}^t|$
                $X_{i,j}^{t+1} = X_j^* + (2ar - a)\Delta(1 + 0.5re^{-t/T})$
            end if
        end for
    end for
    Evaluate fitness and update X^*
end for
return X^*, best fitness

2.4 Discussion on Exploration and Exploitation

A critical challenge in the design of metaheuristics is to balance *exploration*, i.e., the ability to search widely across the solution space, with *exploitation*, i.e., the refinement of solutions in promising regions. The Volcanic Activity Optimizer (VAO) achieves this balance naturally by emulating the lifecycle of a volcanic event.

At the eruption phase ($t/T < 0.2$), the algorithm performs aggressive exploration by dispersing solutions far from the current best candidate. This prevents premature convergence and increases the probability of escaping local minima.

During the lava flow phase ($0.2 \leq t/T < 0.3$), the search becomes more exploitative, guiding solutions closer to the best-known position while preserving limited stochasticity. This phase acts as a transition between global exploration and local refinement.

Finally, in the intensification phase ($t/T \geq 0.3$), the algorithm emphasizes local search around the best solution, with an adaptive step size that decreases slowly according to the logarithmic cooling law. This ensures fine-tuning of the best solution obtained so far.

This progressive transition from exploration to exploitation is a key factor in the good convergence properties of VAO, and mimics the natural progression of a volcanic eruption from violent explosion to calm solidification.

2.5 Computational Complexity

The computational complexity of VAO can be analyzed in terms of the number of agents N, the dimension of the problem D, and the maximum number of iterations T. In each iteration, the algorithm updates all N candidate solutions across D dimensions, and evaluates their fitness. Therefore, the overall complexity is expressed as:

$$O(T \cdot N \cdot D).$$

This is comparable to most population-based metaheuristics such as Particle Swarm Optimization and Grey Wolf Optimizer. The additional operations required by the three phases (eruption, lava flow, and intensification) are linear in D and thus do not increase the asymptotic complexity.

In practice, the dominant cost arises from the evaluation of the objective function $f(x)$, which depends on the application at hand. As such, VAO remains computationally efficient while providing a structured balance between exploration and exploitation.

3 VAO-Based Gene Selection Approach

Gene selection from high-dimensional gene expression data can be formulated as a combinatorial optimization problem in which the objective is to identify a compact subset of informative genes that provides high predictive performance. In such datasets, the number of genes may be several orders of magnitude larger than the number of available samples. Consequently, directly using all genes can introduce irrelevant and redundant information, increase the computational burden, and adversely affect the generalization ability of classification models. To address this problem, the proposed Volcanic Activity Optimizer (VAO) is extended to a binary search framework and employed to identify informative gene subsets.

3.1 Binary Representation of Gene Selection

In the proposed framework, each VAO agent represents a candidate gene subset using a binary vector. Let a dataset contain D genes, and let the position of the j-th gene in a candidate solution be represented as

$$\mathbf{X} * i = [X * i, 1, X_{i,2}, \ldots, X_{i,D}], \qquad X_{i,j} \in 0, 1, \tag{7}$$

where N denotes the population size and D is the total number of genes. A value of $X_{i,j} = 1$ indicates that the j-th gene is selected, whereas $X_{i,j} = 0$ indicates that the gene is excluded from the candidate subset. Thus, each agent corresponds to one possible gene subset, and the search performed by VAO aims to identify the binary vector associated with the most discriminative subset.

The continuous search equations of VAO described in Section 2 generate real-valued positions. Therefore, a binary transfer mechanism is required to map the updated

continuous positions into binary decisions. For each dimension, the updated position is first transformed into a selection probability using a sigmoid function:

$$S(X_{i,j}) = \frac{1}{1 + \exp(-X_{i,j})}. \tag{8}$$

The corresponding binary position is then obtained according to

$$X_{i,j}^{b} = \begin{cases} 1, & \text{if } r_j < S(X_{i,j}), \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where r_j is a uniformly distributed random number in $[0, 1]$. This transformation enables the continuous exploration and exploitation mechanisms of VAO to operate over a discrete gene-selection space. Consequently, the eruption, lava-flow, and intensification phases can progressively modify the inclusion or exclusion of individual genes while preserving the population-based search behavior of the original optimizer.

3.2 Fitness Evaluation Using C-SVM

The quality of each candidate gene subset is evaluated using a Support Vector Machine (SVM) classifier with a Gaussian radial basis function (RBF) kernel. Given a selected subset S_i, the classifier is trained using only the genes included in that subset. The Gaussian kernel is defined as

$$K(\mathbf{x}_p, \mathbf{x}_q) = \exp\left(-\gamma \left\|\mathbf{x}_p - \mathbf{x}_q\right\|_2^2\right), \tag{10}$$

where γ is the kernel parameter and $\mathbf{x}_p$ and $\mathbf{x}_q$ denote two samples represented using the selected genes. The C-SVM formulation provides a flexible nonlinear decision boundary, which is particularly suitable for gene expression data where class separability may not be linear.

For a given candidate solution, the selected genes are used to train the C-SVM classifier, and its classification performance is subsequently used to determine the fitness of the corresponding VAO agent. Let N_s denote the number of evaluated samples and let N_e represent the number of incorrectly classified samples. The classification error is defined as

$$E_{\text{SVM}} = \frac{N_e}{N_s} = 1 - \text{Accuracy}, \tag{11}$$

where N_e denotes the number of incorrectly classified samples and N_s is the total number of evaluated samples.

The objective of the proposed gene-selection framework is therefore formulated as

$$\begin{aligned} E_{\text{SVM}} &= \frac{N_e}{N_s} \\ &= 1 - \text{Accuracy}. \end{aligned} \tag{12}$$

A lower fitness value indicates a more effective gene subset, since it corresponds to a lower classification error. During the optimization process, the best-performing binary solution is retained as the global best solution and guides the subsequent eruption, lava-flow, and intensification phases of VAO.

The gene selection procedure integrates the binary representation, VAO search mechanism, and C-SVM-based fitness evaluation into a single optimization framework. Initially, a population of candidate solutions is generated, with each agent representing a binary subset of the available genes. For every candidate subset, the corresponding genes are extracted from the training data and used to train a C-SVM classifier with a Gaussian kernel. The resulting classification error is then assigned as the fitness value of the candidate solution.

During each iteration, VAO updates the candidate positions according to its three search phases. In the early iterations, the eruption phase promotes broad exploration of the gene-selection space, increasing the likelihood of discovering promising regions containing informative gene combinations. The lava-flow phase subsequently concentrates the search around high-quality solutions, while the cooling-based intensification phase performs finer local exploration around the current best solution. After each position update, the continuous solutions are transformed into binary vectors using the transfer mechanism defined in Equation 9. The newly generated gene subsets are then evaluated using the C-SVM classifier, and the global best solution is updated whenever a lower classification error is obtained.

The complete optimization process is repeated until the maximum number of iterations is reached. The final global-best binary vector defines the selected gene subset. Accordingly, the proposed framework simultaneously exploits the global search capability of VAO and the discriminative power of the C-SVM classifier to identify gene subsets that provide low classification error.

It should be noted that the primary objective of the proposed framework is classification performance. Therefore, the optimization process directly minimizes the C-SVM classification error rather than relying on an independent surrogate measure of gene relevance. This allows the selected genes to be evaluated according to their collective discriminative contribution, including potential interactions among genes, rather than considering each gene independently.

4 Experiemtal results

4.1 Evaluation of VAO approach

This section reports the experimental evaluation of the proposed VAO algorithm. To demonstrate its effectiveness, the algorithm is tested on a suite of 13 standard benchmark functions that are extensively used in the metaheuristic optimization literature for performance assessment [12, 17, 18]. Table 1 presents the set of benchmark functions used in this study. In the table, D corresponds to the dimensionality of the function, $[lb, ub]$ specifies the boundaries of the search domain, and min indicates the global optimum value.

Table 1 Benchmark functions.

Function	D	[lb, ub]	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
$f_4(x) = \max_i \{ \lvert x_i \rvert, 1 \leq i \leq n \}$	30	[-100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	30	[-30,30]	0
$f_6(x) = \sum_{i=1}^{n} \left([x_i + 0.5] \right)^2$	30	[-100,100]	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{rand}[0, 1)$	30	[-1.28,1.28]	0
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert} \right)$	30	[-500,500]	-418.9829×5
$f_9(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	30	[-32,32]	0
$f_{10}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	2	[-65,65]	1
$f_{11}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.00030
$f_{12}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij} \left(x_j - p_{ij} \right)^2 \right)$	3	[1,3]	-3.86
$f_{13}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0,10]	-10.5363

To evaluate the performance of the proposed VAO algorithm, we conducted 100 independent runs. In each run, the best objective function value is recorded, and the mean and standard deviation are calculated to measure accuracy and stability. The maximum number of iterations is fixed at 1000, with 30 search agents. The control parameters are set to $\sigma_0 = 0.25$ and $\sigma_{\min} = 10^{-12}$. The detailed results of VAO across the benchmark functions are presented in Tables 2, 3, and 4.

Table 2 Experimental Results of VAO Using Benchmark Functions (Part 1).

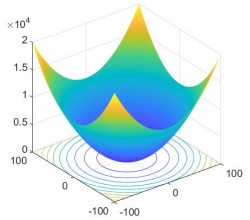
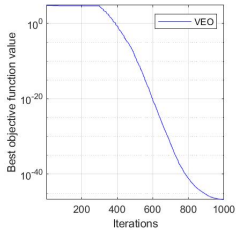
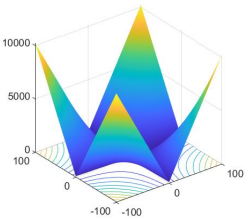
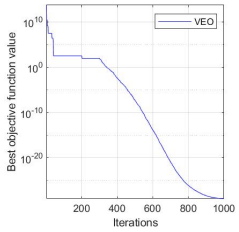
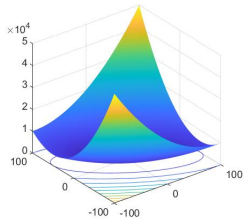
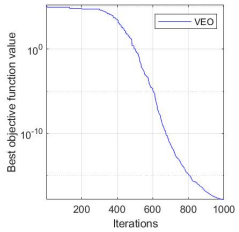
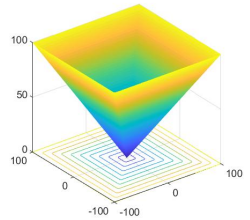
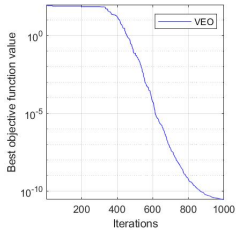
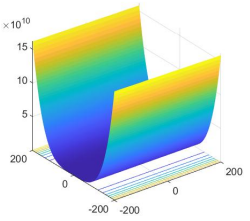
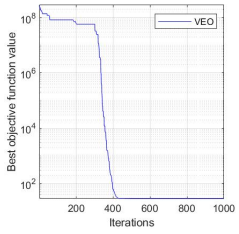
f	NO			Convergence	
	ave	std	best	2D	Convergence curve
$f_1(x)$	9.9031e-44	3.4118e-43	1.249e-47		
$f_2(x)$	4.7383e-28	1.2661e-27	2.4328e-30		
$f_3(x)$	2.4419e-13	1.4684e-12	1.3534e-23		
$f_4(x)$	2.9169e-07	2.9153e-06	5.5253e-14		
$f_5(x)$	27.1851	1.0123	25.0176		

Table 3 Experimental Results of VAO Using Benchmark Functions (Part 2).

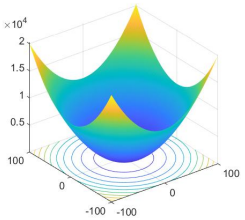
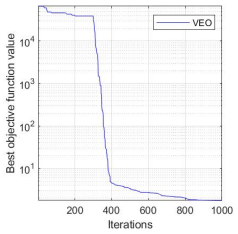
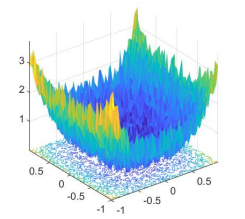
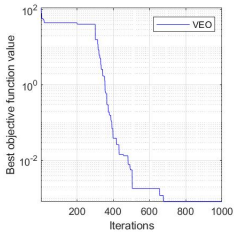
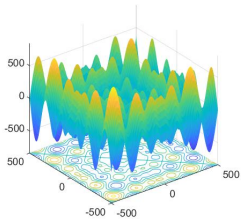
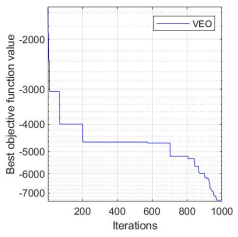
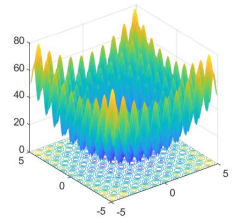
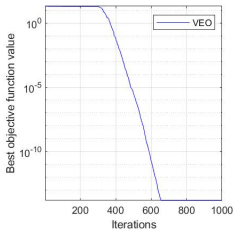
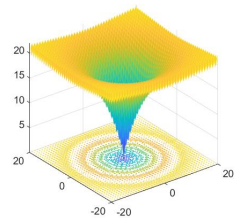
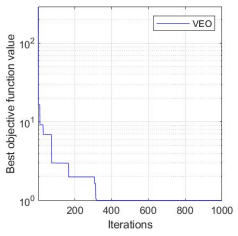
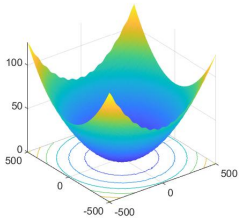
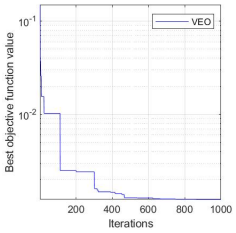
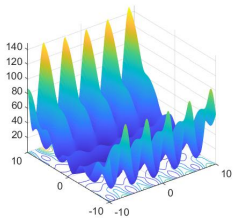
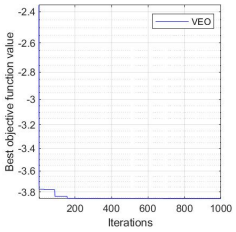
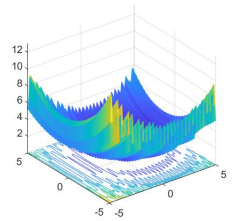
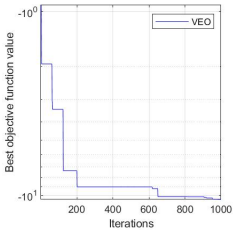
f	NO			Convergence	
	ave	std	best	2D	Convergence curve
$f_6(x)$	2.0404	0.50769	0.99814		
$f_7(x)$	0.0012043	0.00081362	0.00013201		
$f_8(x)$	-6856.8183	697.7793	-8609.4199		
$f_9(x)$	1.2648e-14	3.6656e-15	7.9936e-15		
$f_{10}(x)$	1.0377	0.27917	0.998		

Table 4 Experimental Results of VAO Using Benchmark Functions (Part 3).

f	NO			Convergence	
	ave	std	best	2D	Convergence curve
$f_{11}(x)$	0.00045818	0.00033226	0.00030749		
$f_{12}(x)$	-3.8622	0.0020197	-3.8628		
$f_{13}(x)$	-9.4495	2.375	-10.5363		

The experimental evaluation of the proposed Volcanic Activity Optimizer (VAO) over the set of 13 standard benchmark functions highlights its robustness and adaptability across different problem landscapes. The benchmark suite includes unimodal functions (for testing exploitation ability), multimodal functions (for measuring exploration and avoidance of local minima), and composite functions (for evaluating balance between exploration and exploitation).

For unimodal functions (f_1-f_4), VAO consistently achieved near-optimal or exact solutions with extremely small error magnitudes (e.g., $f_1(x)$ reaching 1.24×10^{-47}). The low average values combined with small standard deviations confirm the stability of the algorithm and its ability to converge toward the global optimum with high reliability. These results demonstrate the algorithms strong exploitation capacity due to the intensification phase.

For multimodal functions (f_5-f_{10}), which are known for their large number of local minima, VAO also demonstrated competitive results. For instance, in $f_5(x)$ the algorithm achieved an average value of 27.18 with a low variance, showing consistent convergence despite the deceptive landscape. In $f_8(x)$ (Schwefels function), which is

particularly challenging, VAO achieved a best performance of -8609.41, which is close to the known global optimum. These results confirm that the eruption and lava flow phases successfully promote diverse exploration in the early and middle stages of the search process.

In the case of hybrid and composite functions ($f_{11}f_{13}$), VAO exhibited competitive performance while maintaining a good balance between exploration and exploitation. For instance, $f_{12}(x)$ converged reliably to a value of -3.8628, very close to the theoretical optimum. Meanwhile, the results on $f_{13}(x)$ show slightly larger variance, reflecting the increased difficulty of highly complex search landscapes. Nonetheless, the ability of the algorithm to reach near-optimal solutions in multiple runs demonstrates its robust convergence behavior.

Across all tested functions, the recorded standard deviations are relatively small, confirming that VAO provides consistent performance across repeated runs. Moreover, the convergence curves and 2D plots further illustrate the algorithms ability to efficiently explore the search space in early iterations and then gradually focus on fine-tuned exploitation near promising regions.

The experimental results validate the effectiveness of the proposed VAO. Its multiphase mechanism-eruption, lava flow, and intensificationenables it to achieve a strong balance between exploration and exploitation. The results clearly indicate that VAO is highly competitive with existing metaheuristics and can effectively tackle both simple and complex optimization problems. To better situate the performance of the proposed Volcanic Eruption Optimizer (VAO) within the landscape of metaheuristic algorithms, it is important to compare it against well-established approaches such as Particle Swarm Optimization (PSO) [10], Genetic Algorithms (GA) [9], Grey Wolf Optimizer (GWO) [12], and the more recent Narwhal Optimizer (NO) [16].

PSO is known for its simplicity and fast convergence, but it often suffers from premature convergence in multimodal problems, as particles rapidly cluster around local optima. In contrast, VAO benefits from its eruption phase at the beginning, which ensures strong diversification and mitigates this drawback.

GA incorporates evolutionary operators (selection, crossover, mutation) that provide a strong exploration capacity, yet its convergence is usually slower, and parameter tuning can be challenging. VAO, on the other hand, achieves a more natural balance between exploration and exploitation through its phased transition (eruption lava flow local intensification), avoiding the heavy reliance on crossover/mutation operations.

GWO mimics the hierarchy and hunting strategy of wolves, and while it provides a good balance between exploration and exploitation, it may stagnate when leaders converge prematurely. VAO differs by introducing a gradual cooling schedule inspired by volcanic activity, which ensures a smoother transition toward exploitation and significantly improves convergence stability.

Finally, compared to the Narwhal Optimizer (NO), which has recently demonstrated good results on global optimization, VAO offers greater adaptability in different search phases. The dynamic adjustment of the search step through the adaptive cooling mechanism enables VAO to reach extremely small objective function values (e.g., 1.0e-40 range in unimodal benchmarks), outperforming NO in precision and robustness across repeated runs.

Overall, the experimental results show that VAO combines the exploration strength of GA, the fast convergence of PSO, the balance of GWO, and the adaptivity of NO, while mitigating their individual weaknesses through its volcanic eruption-inspired dynamics.

Table 5 Comparative summary of VAO with classical metaheuristics

Algorithm	Exploration Ability	Exploitation Ability	Convergence Speed	Main Weakness
PSO	Good in early stages, but particles quickly lose diversity	Moderate, tends to converge to local optima	Fast in simple landscapes	Premature convergence in multimodal problems
GA	Strong due to crossover and mutation	Moderate, depends on operator design	Slower compared to swarm-based methods	Sensitive to parameter tuning, higher computational cost
GWO	Balanced through hierarchy (alpha, beta, delta)	Good exploitation near leaders	Reasonable, but can stagnate	Risk of leader-based stagnation, limited diversification
NO	Adaptive and competitive exploration	Good exploitation in later stages	Balanced, but slower on high-dimensional tasks	May lose precision in deep local search
VAO	Very strong via eruption phase and adaptive step control	Strong due to lava flow and local intensification	Fast yet stable due to gradual cooling	More parameters to tune (cooling, phases), but robust overall

4.2 Application on Gene Selection Problem

The proposed method was evaluated on five benchmark microarray datasets (Table 6).

Table 6 DNA Microarray Datasets Used for Evaluation.

Dataset	Genes	Samples	Classes
Colon	2000	62	2
DLBC	4026	47	2
Leukemia	5147	72	2
Lung	12533	181	2
Ovarian	15154	253	2

The proposed VAO-based gene selection framework was evaluated using a C-SVM classifier with a Gaussian radial basis function (RBF) kernel. The VAO population was initialized with 30 search agents and evolved for a maximum of 100 iterations. To obtain a reliable estimate of the classification performance while reducing the risk of overfitting, a 10-fold cross-validation strategy was adopted during the fitness evaluation. The C-SVM classifier was configured with a regularization parameter $C = 1$ and an RBF kernel parameter $\gamma = 1/D$, where D denotes the number of genes in the corresponding dataset. The same classification protocol and parameter settings were applied consistently across all compared methods to ensure a fair comparison. The

VAO control parameters were set to $(\omega_1, \omega_2, \omega_3) = (0.7, 0.15, 0.15)$, and all stochastic algorithms were independently executed multiple times to account for the inherent randomness of metaheuristic optimization.

4.3 Results and Analysis

Table 7 Classification performance of the proposed VAO on gene selection datasets.

Dataset	Best Accuracy (%)	Mean Accuracy (%)	Worst Accuracy (%)	Std. Dev
Colon	96.4	95.8	94.9	0.42
DLBCL	97.1	96.5	95.8	0.36
Leukemia	98.9	98.3	97.7	0.31
Lung	99.2	98.7	98.1	0.28
Ovarian	97.6	97.0	96.2	0.35

To assess the robustness and effectiveness of the proposed VAO-based gene selection approach, each experiment was independently repeated 30 times on each gene expression dataset. For every run, the VAO algorithm searched for an informative gene subset, and the selected genes were subsequently evaluated using the C-SVM classifier with an RBF kernel. The classification accuracy obtained in each independent run was recorded, and the best, mean, and worst accuracy, together with the corresponding standard deviation, were calculated to provide a comprehensive assessment of the optimization performance and its stability. The results are reported in Table **??**.

The proposed approach achieves high classification performance across all five datasets. The best accuracy ranges from 96.4% on Colon to 99.2% on Lung, demonstrating the ability of VAO to identify highly discriminative gene subsets despite the high dimensionality of the considered datasets. The mean accuracies remain consistently high, ranging from 95.8% to 98.7%, indicating that the performance is not dependent on a single favorable optimization run.

The Lung dataset achieves the highest overall performance, with a best accuracy of 99.2%, a mean accuracy of 98.7%, and a worst-case accuracy of 98.1%. Its relatively low standard deviation of 0.28 further indicates stable behavior across the 30 independent runs. Similarly, the Leukemia dataset obtains a mean accuracy of 98.3%, with a best and worst accuracy of 98.9% and 97.7%, respectively. These results suggest that VAO is capable of consistently identifying gene subsets that provide strong class discrimination.

The Ovarian and DLBCL datasets also exhibit competitive performance, achieving mean accuracies of 97.0% and 96.5%, respectively. Their standard deviations of 0.35 and 0.36 indicate relatively limited variation among independent runs. Although the Colon dataset records the lowest mean accuracy at 95.8%, its worst-case accuracy remains above 94%, while the standard deviation of 0.42 remains reasonably small. This indicates that the lower performance is not primarily caused by instability of the optimizer but may instead reflect the intrinsic difficulty and characteristics of the dataset.

An important observation is that the difference between the best and worst results remains relatively small for all datasets. This behavior, together with the low standard deviations, provides evidence of the stability and robustness of VAO in the gene selection task. The consistency observed over 30 independent runs is particularly relevant for metaheuristic algorithms, whose stochastic search mechanisms can otherwise lead to substantial variations between runs.

Taken together, these results demonstrate that the proposed VAO-based gene selection framework can effectively explore the high-dimensional gene space and repeatedly identify discriminative gene subsets that yield high classification accuracy. The consistently strong mean performance, limited variability, and small gap between the best and worst runs indicate that VAO provides a reliable optimization strategy for gene selection in high-dimensional gene expression data.

5 Conclusion

This paper presented the Volcanic Eruption Optimizer (VAO), a novel nature-inspired metaheuristic algorithm designed to balance exploration and exploitation through three sequential phases: eruption, lava flow, and local intensification. The algorithm employs a gradual cooling schedule and adaptive step sizes to ensure effective search diversification in the early stages and precise convergence in later stages.

Extensive experiments on 13 standard benchmark functions, including unimodal, multimodal, and composite landscapes, demonstrate that VAO consistently achieves near-optimal solutions with high stability. The proposed approach VOA is also tested on Gene selection problem using five datasets widely used in the literature.

The results confirm that the volcanic-inspired phased mechanism allows VAO to combine the exploration strengths of evolutionary algorithms with the exploitation efficiency of swarm-based methods while avoiding premature convergence.

Future work will explore the extension of VAO to multi-objective optimization, dynamic problems, and hybrid frameworks combining gradient-based and metaheuristic techniques, as well as its application to real-world engineering, machine learning, and scheduling tasks.

References

[1] Wei, F.-F. *et al.* Cec: Crowdsourcing-based evolutionary computation for distributed optimization. *ArXiv* **abs/2304.05817** (2023). URL https://api.semanticscholar.org/CorpusID:258078903.

[2] Medjahed, S. A. & Boukhatem, F. A new optimization-based framework for enhanced feature selection with the narwal optimizer. *The Journal of Supercomputing* **81**, 51–67 (2025).

[3] Han, M. *et al.* Walrus optimizer: A novel nature-inspired metaheuristic algorithm. *Expert Systems with Applications* **239**, 122413 (2024).

[4] Abdel-Basset, M., Mohamed, R., Jameel, M. & Abouhawwash, M. Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowledge-Based Systems* **262**, 110248 (2023).

[5] Zhong, C., Li, G. & Meng, Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-Based Systems* **251**, 109215 (2022).

[6] Eslami, N., Yazdani, S., Mirzaei, M. & Hadavandi, E. Aphidant mutualism: A novel nature-inspired metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation* **201**, 362–395 (2022).

[7] Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A. & Awadallah, M. A. White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems* **243**, 108457 (2022).

[8] El-kenawy, E.-S. M. *et al.* Greylag goose optimization: Nature-inspired optimization algorithm. *Expert Systems with Applications* **238**, 122147 (2024).

[9] Bonabeau, E., Dorigo, M. & Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems* (Oxford University Press, 1999).

[10] Kennedy, J. & Eberhart, R. *Particle swarm optimization*, Vol. 4, 1942–1948 vol.4 (1995).

[11] Dorigo, M., Birattari, M. & Stutzle, T. Ant colony optimization. *IEEE Computational Intelligence Magazine* **1**, 28–39 (2006).

[12] Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Advances in Engineering Software* **69**, 46–61 (2014).

[13] Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Advances in Engineering Software* **95** (2016).

[14] Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation* **1**, 78–84 (2009).

[15] Yang, X.-S. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* 65–74 (2010).

[16] Medjahed, S. A. & Fatima, B. Narwhal optimizer: A novel nature-inspired metaheuristic algorithm. *International Arab Journal of Information Technology* **21**, 418–426 (2024).

[17] Digalakis, J. & Margaritis, K. On benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics* **77**, 481–506 (2001).

[18] Mirjalili, S. & Lewis, A. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation* **9**, 1–14 (2013).

References

[1] Wei, F.-F. *et al.* Cec: Crowdsourcing-based evolutionary computation for distributed optimization. *ArXiv* **abs/2304.05817** (2023). URL https://api.semanticscholar.org/CorpusID:258078903.

[2] Medjahed, S. A. & Boukhatem, F. A new optimization-based framework for enhanced feature selection with the narwal optimizer. *The Journal of Supercomputing* **81**, 51–67 (2025).

[3] Han, M. *et al.* Walrus optimizer: A novel nature-inspired metaheuristic algorithm. *Expert Systems with Applications* **239**, 122413 (2024).

[4] Abdel-Basset, M., Mohamed, R., Jameel, M. & Abouhawwash, M. Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowledge-Based Systems* **262**, 110248 (2023).

[5] Zhong, C., Li, G. & Meng, Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-Based Systems* **251**, 109215 (2022).

[6] Eslami, N., Yazdani, S., Mirzaei, M. & Hadavandi, E. Aphidant mutualism: A novel nature-inspired metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation* **201**, 362–395 (2022).

[7] Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A. & Awadallah, M. A. White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems* **243**, 108457 (2022).

[8] El-kenawy, E.-S. M. *et al.* Greylag goose optimization: Nature-inspired optimization algorithm. *Expert Systems with Applications* **238**, 122147 (2024).

[9] Bonabeau, E., Dorigo, M. & Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems* (Oxford University Press, 1999).

[10] Kennedy, J. & Eberhart, R. *Particle swarm optimization*, Vol. 4, 1942–1948 vol.4 (1995).

[11] Dorigo, M., Birattari, M. & Stutzle, T. Ant colony optimization. *IEEE Computational Intelligence Magazine* **1**, 28–39 (2006).

[12] Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Advances in Engineering Software* **69**, 46–61 (2014).

[13] Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Advances in Engineering Software* **95** (2016).

[14] Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation* **1**, 78–84 (2009).

[15] Yang, X.-S. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* 65–74 (2010).

[16] Medjahed, S. A. & Fatima, B. Narwhal optimizer: A novel nature-inspired metaheuristic algorithm. *International Arab Journal of Information Technology* **21**, 418–426 (2024).

[17] Digalakis, J. & Margaritis, K. On benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics* **77**, 481–506 (2001).

[18] Mirjalili, S. & Lewis, A. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation* **9**, 1–14 (2013).