مجلة جامعة بني وليد للعلوم الإنسانية والتطبيقية
Bani Waleed University Journal of Humanities and Applied Sciences
تصدر عن– جامعة بني وليد – ليبيا
Website:https://jhas-bwu.com/index.php/bwjhas/index

ISSN 3005-3900 | الصفحات (116- 130) | المجلد الحادي عشر – العدد الأول – 2026

Blobfish Optimization Algorithm: A Novel Met heuristic Method for High-Dimensional Global Optimization Problems

Emad F. H. Qasim [1]*, Saif alislam. E. Muhammed [2], Esamaldeen Mohamed [3]

[1.3] Department of Mathematics, Faculty of Education, Omar Al-Mukhtar University, Albyda, Libya.
[2] Department of Mathematics, Faculty of Science, Omar Al-Mukhtar University, Albyda, Libya.
emad.qasim@omu.edu.ly

خوارزمية تحسين مستوحاة من السمكة الهلامية: منهجية فوق استرشاديه جديدة لحل مسائل التحسين الشامل عالية الأبعاد

عماد فتحي حسين قاسم[1]*، سيف الإسلام عيسى محمد [2]، عصام الدين محمد محمد [3]
[3.1] قسم الرياضيات، كلية التربية، جامعة عمر المختار، البيضاء، ليبيا.
[2] قسم الرياضيات، كلية العلوم، جامعة عمر المختار، البيضاء، ليبيا.

Received: 08-12-2025	Accepted: 10-01-2026	Published: 23-01-2026

الملخص:
لقد استُخدمت خوارزميات التحسين الفوق استرشاديه على نطاق واسع لحل مشكلات التحسين المعقدة غير الخطية وعالية الأبعاد ومع ذلك تعاني الطرق الحالية من التوقف المبكر عند الحلول غير المثلى والحساسية تجاه الإعدادات البارامترية للمعاملات ومحدودية القابلية للتوسع. في هذه الورقة يتم اقتراح خوارزمية تحسين مستوحاة من السمكة الهلامية كإطار عمل هجين تسلسلي فوق استرشادي يدمج بين استغلال موجه بالنخبة وجذب قائم على المركز وآليات الحفاظ على التنوع المعتمدة على التجميع وآليات التعديل الواعية للركود في حلقة تكرارية. تم تقييم الخوارزمية المقترحة على مجموعة شاملة من الدوال المعيارية تشمل 14 دالة اختيارية من النمط أحادي الوسيط ومتعدد الوسائط وسيء التكييف والمركبة عبر أبعاد تتراوح من 2 إلى 100. تظهر النتائج التجريبية أن خوارزمية السمكة الهلامية تحقق تقارباً موثوقاً عبر بيئات مشكلات متنوعة محافظةً على معدل نجاح **100**% لجميع الدوال المعيارية حتى **50** بُعداً، مع الحفاظ على أداء مستقر عند الأبعاد الأعلى، حيث بلغ معدل النجاح **90**% لدالة **Composition1** عند **100** بُعد و **70**% لدالة **Ackley** عند **100** بُعد، مما يظهر تحديات البيئات عالية التعدد والمركبة في الأبعاد القصوى. يكشف تحليل التقارب عن قدرة الخوارزمية على الهروب من القيم الدنيا المحلية بفضل آلية الهجوم المفاجئ. يؤكد تحليل حساسية المعاملات على متانة قوية تجاه التغيرات المعتدلة في المعاملان مع الحفاظ على معدل نجاح **100**% عبر جميع التركيبات ألـ27 للمعاملات التي تم اختبارها. تشير النتائج إلى أن خوارزمية السمكة

الهلامية للتحسين توفر إطار عمل هجيناً قوياً وقابلاً للتوسع ومناسباً لمشاكل التحسين المستمرة والمعقدة مع تقديم توازن مقبول بين جودة الحل والمتانة والتكلفة الحسابية.

الكلمات الدالة: التحسين الفوق استرشادي، خوارزمية السمكة الهلامية، التحسين الشامل، حساسية المعاملات، مسائل عالية الأبعاد.

Abstract

Met heuristic optimization algorithms have been widely employed to solve complex nonlinear and high-dimensional optimization problems; however, many existing methods suffer from premature convergence, sensitivity to parameter settings, and scalability limitations. In this paper, a Blowfish Optimization Algorithm (BOA) is proposed as a sequential hybrid met heuristic framework that integrates elite-guided exploitation with centroid-based attraction, clustering-based diversity preservation, and stagnation-aware perturbation mechanisms in an iterative loop. The proposed algorithm is evaluated on a comprehensive benchmark suite comprising 14 unimodal, multimodal, ill-conditioned, and composite test functions across dimensionality ranging from 2 to 100. Experimental results demonstrate that BOA achieves reliable convergence across diverse problem landscapes, maintaining a 100% success rate for all benchmark functions up to 50 dimensions and preserving stable performance at higher dimensionality, with 90% success rate for Composition1 function at 100 dimensions and 70% success rate for Ackley function at 100 dimensions, reflecting the challenges of highly multimodal and composite landscapes in extreme dimensions. Convergence analysis reveals the algorithm's ability to escape local minima through its surprise attack mechanism. A detailed parameter sensitivity analysis confirms strong robustness to moderate parameter variations, with 100% success rate maintained across all 27 tested parameter combinations. Overall, the results suggest that BOA provides a robust and scalable hybrid optimization framework suitable for complex continuous optimization problems, offering a balanced trade-off between solution quality, robustness, and computational cost.

Keywords: Met heuristic Optimization, Blob fish Algorithm, Global Optimization, Parameter Sensitivity, High-Dimensional Problems.

Introduction

Global optimization plays a central role in a wide range of scientific and engineering applications, including system identification, parameter estimation, scheduling, and design optimization. Many of these problems are characterized by nonlinearity, multimodality, ill-conditioning, and high dimensionality, rendering classical gradient-based optimization techniques ineffective or impractical [1]. Metaheuristic algorithms have therefore become a popular alternative due to their derivative-free nature and flexibility. Well-known methods such as Particle Swarm Optimization (PSO) [2], Grey Wolf Optimizer (GWO) [3], and Whale Optimization Algorithm (WOA) [4] have demonstrated competitive performance on a variety of benchmark problems. Despite their success, these algorithms often suffer from premature convergence, sensitivity to parameter settings, and performance degradation as problem dimensionality increases [5, 6]. One of the fundamental challenges in metaheuristic optimization is balancing exploration and exploitation while maintaining sufficient diversity to avoid stagnation in local optima. This challenge becomes particularly pronounced in multimodal and composite landscapes, where deceptive local structures can mislead population-based search mechanisms [7]. As a result, recent research has increasingly focused on hybrid and multi-mechanism optimization frameworks that combine complementary strategies to improve robustness and scalability [6, 8]. Motivated by these observations, this paper proposes a Blobfish optimization algorithm (boa) that integrates multiple coordinated mechanisms within a sequential iterative framework. The proposed approach incorporates centroid-based elite attraction to accelerate convergence, periodic clustering to maintain population heterogeneity, and stagnation-aware perturbation (surprise attack) to facilitate escape from local minima. The emphasis of this work lies in the sequential integration and interaction of these mechanisms within a single optimization loop to address known limitations of classical metaheuristics. The performance of BOA is assessed through an extensive experimental study using a diverse benchmark suite of 14 test functions, covering unimodal, multimodal, ill-conditioned, and composite optimization scenarios. The evaluation spans dimensionality from low (2D) to high (100D) in order to analyze scalability, robustness, and convergence behaviour. In addition to standard performance metrics, a detailed parameter sensitivity analysis is conducted to examine the algorithm's robustness to parameter variations, with particular attention to success rate preservation across diverse parameter combinations. The main contributions of this work can be summarized as follows:

1. A sequential hybrid optimization framework that integrates centroid-based elite attraction, diversity preservation via periodic clustering, and surprise attack perturbation within a single iterative loop.
2. A comprehensive experimental evaluation across diverse benchmark functions and dimensionality (2–100D), providing detailed insight into scalability and convergence behaviour based on actual implementation results.
3. An empirical analysis of local minima escape behaviour, particularly on challenging composite functions such as Composition1 and highly multimodal functions like Ackley, with documented cases of success and failure across varying dimensionality.
4. A systematic parameter sensitivity study demonstrating robustness to moderate parameter perturbations (+15–20%) with 100% success rate maintained across all 27 tested parameter combinations for the evaluated functions.

The experimental evaluation includes comprehensive visual analyzes demonstrating the algorithm's architectural design, convergence behaviour, benchmark performance, and parameter robustness. The remainder of this paper is organized as follows. Section 2 introduces the proposed Blobfish Optimization Algorithm. Section 3 presents the experimental setup and performance analysis. Section 4 provides a comparative discussion with existing metaheuristic approaches based on actual implementation results. Section 5 concludes the paper and outlines directions for future research.

2 Theoretical Frameworks and Problem Formulation

2.1 General Formulation of the Optimization Problem

Without loss of generality, the class of optimization problems considered in this study can be formulated as:

$$\min_{\mathrm{x}\in\Omega\subset\mathbb{R}^D} f(\mathrm{x}),$$

where

- $f(x)$ denotes the objective function,
- $\mathrm{x} = (x_1, x_2, \dots, x_D)$ is a candidate solution vector,
- D is the problem dimensionality,
- $\Omega = [l_i, u_i]^D$ Represents the bounded search space.

The considered problems are continuous, nonlinear, and may exhibit multimodality, non-separability, or ill-conditioning, which motivates the use of population-based metaheuristic optimization methods [8].

2.2 Population Initialization

Let the population size be N, dynamically determined as:

$$N = max(100, min(2000, \lfloor base\_pop\_factor \times D \rfloor))$$

where base_pop_factor is a configuration parameter (default: 1.095).

Each population is initialized using Latin Hypercube Sampling to ensure uniform coverage:

$$P^0 = \{x_1^0, x_2^0, \dots, x_N^0\},\ \ x_{i,j}^0 \sim LHS\big(l_j, u_j\big),$$

where $i = 1, \dots, N$ and $j = 1, \dots, D$.

2.3 Centroid-Based Elite Attraction Mechanism

At each iteration t, an elite subset is selected based on fitness ranking. Let $E \subset P^t$ denote the elite set with size:

$$|E| = \lceil \rho N \rceil,$$

where $\rho \in (0,1)$ is the elite ratio (default: 0.295). The centroid of elite solutions is computed:

$$\boldsymbol{c^t} = \frac{1}{|E|} \sum_{x \in E} \boldsymbol{x}\,.$$

Each solution updates its position by moving toward the elite centroid with stochastic perturbation:

$$x_i^{t+1} = x_i^t + \alpha_t(\lambda_1(\boldsymbol{c^t} - x_i^t) + \lambda_2 \cdot \boldsymbol{n_i^t}),$$

where

- α_t is a time-varying step-size parameter,
- $\lambda_1 = 1.9995$ (drift coefficient),
- $\lambda_2 = 0.2995$ (random coefficient),
- $n_i^t \sim N(0,1)$ is Gaussian noise.

2.4 Adaptive Step-Size Strategy

The step size α_t is adaptively decreased to balance exploration and exploitation:

$$\alpha_t = \alpha_{init} + (\tfrac{t}{T_{max}})(\alpha_{final} - \alpha_{init}),$$

where T_{max} is the maximum number of iterations, $\alpha_{init} = 0.085$, and $\alpha_{final} = 0.65$.

This linear schedule promotes global exploration during early iterations and fine-grained exploitation near convergence [6].

2.5 Diversity Preservation via Periodic Clustering

To mitigate population collapse and premature convergence, K-means clustering is periodically applied to the population every C iterations (default: $C = 51$). Let K be the number of clusters determined as:

$$K = max(2, min(10, \lfloor N/30 \rfloor))$$

Individuals are assigned to clusters, and solutions move toward their respective cluster centers:

$$x_i^{t+1} = x_i^t + 0.079. \left(\boldsymbol{c_{k(i)}^t} - x_i^t\right),$$

where $\boldsymbol{c_{k(i)}^t}$ is the center of the cluster to which x_i^t belongs.

2.6 Stagnation Detection and Surprise Attack

Stagnation is detected when the global best fitness does not improve for a predefined number of iterations $\tau = max(30, \lfloor 0.05 \times T_{max} \rfloor)$. Once stagnation is detected, a surprise attack perturbation operator is applied to randomly selected individuals:

$$x_i^{t+1} = x_i^t + \beta_t \cdot N(0,1),$$

where $\beta_t = max(0.05 \times (1 - t/T_{max}), 0.02)$ controls perturbation intensity.

This mechanism enables the algorithm to escape local minima, particularly in multimodal and composite landscapes [8].

2.7 Local Search Refinement

Every L iterations (default: $L = 15$), local search refinement is applied to the top-k solutions (default: $k = 11$) using the L-BFGS-B algorithm [9] with bounds enforcement. This hybrid approach combines global exploration with local exploitation capabilities.

2.8 Algorithmic Pseudo-Code and Flowchart

Algorithm: Blobfish Optimization Algorithm (BOA)

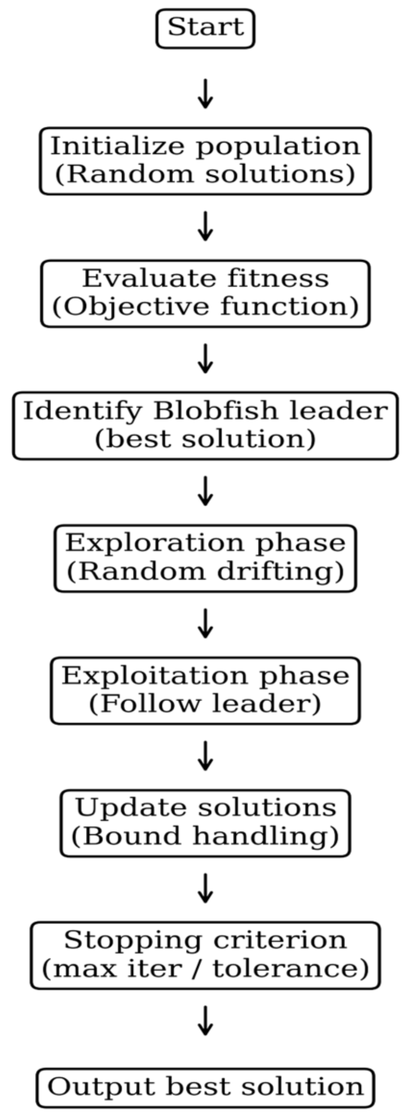
Input:

- Objective function $f(\boldsymbol{x})$
- Search bounds $\Omega = [\boldsymbol{l}, \boldsymbol{u}]$
- Problem dimensionality D
- Maximum iterations T_{max}
- Base population factor $bpf = 1.095$
- Elite ratio $\rho = 0.295$
- Clustering interval $C = 51$
- Local search interval $L = 15$

Output:

- Best solution $\boldsymbol{x}_{best}$

$N \leftarrow max(100, min(2000, \lfloor bpf \times D \rfloor))$
Initialize population P using Latin Hypercube Sampling within Ω
Evaluate $f(\boldsymbol{x_i})$ for all $\boldsymbol{x_i} \in P$
$\boldsymbol{x}_{best} \leftarrow argmin_{\boldsymbol{x_i}} f(\boldsymbol{x_i})$
$f_{best} \leftarrow f(\boldsymbol{x}_{best})$
$stagnation\_counter \leftarrow 0$
for $t = 1$ **to** T_{max} **do**
$\alpha_t \leftarrow 0.085 + (t/T_{max}) \times (0.65 - 0.085)$
Evaluate current population
Select elite set E of size $\lfloor \rho N \rfloor$
Compute centroid $\boldsymbol{c}$ of elite solutions
for each $\boldsymbol{x_i} \in P$ **do**
$\boldsymbol{n_i} \sim N(0,1)$
$\boldsymbol{x_i} \leftarrow \boldsymbol{x_i} + \alpha_t \times (1.9995 \times (\boldsymbol{c} - \boldsymbol{x_i}) + 0.2995 \times \boldsymbol{n_i})$
Enforce boundary constraints
end for
if t mod $C = 0$ **then**
Apply K-means clustering to P
Move solutions toward cluster centers (Eq. 2.5)
end if
Evaluate updated population
if $f(x_{best}^{new}) < f_{best} - 10^{-12}$ **then**
$\boldsymbol{x}_{best} \leftarrow x_{best}^{new}$
$f_{best} \leftarrow f(\boldsymbol{x}_{best})$
$stagnation\_counter \leftarrow 0$
else
$stagnation\_counter \leftarrow stagnation\_counter + 1$
end if
if $stagnation\_counter \geq max(30, \lfloor 0.05 \times T_{max} \rfloor)$ **then**
Apply surprise attack perturbation (Eq.2.6)
$stagnation\_counter \leftarrow 0$



end if
if $t \ mod \ L = 0$ ***then***
Apply L-BFGS-B to top-11 solutions
end if
end for
return $\boldsymbol{x}_{best}$

Placeholder for Figure 1: A flowchart illustrating the sequential hybrid architecture of the Blobfish Optimization Algorithm (BOA), integrating five key components: (1) LHS-based population initialization for uniform search space coverage, (2) centroid-based elite attraction for accelerated convergence, (3) periodic clustering for diversity preservation, (4) stagnation-aware surprise attack for local minima escape, and (5) L-BFGS-B local search refinement. These components operate in a unified iterative loop with adaptive parameter control, enabling balanced exploration-exploitation throughout optimization.

2.9 Benchmark Test Functions

The proposed algorithm is evaluated using standard benchmark functions widely adopted in the optimization literature [5, 7].

Sphere Function: $f(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$, $x_i \in [-100,100]$. Unimodal, convex, and separable.

Rosenbrock Function: $f(\boldsymbol{x}) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$, $x_i \in [-2.048,2.048]$. Unimodal but non-separable with a narrow curved valley.

Rastrigin Function: $f(\boldsymbol{x}) = \sum_{i=1}^{D}[x_i^2 - 10cos(2\pi x_i) + 10]$, $x_i \in [-5.12,5.12]$. Highly multimodal and separable.

Ackley Function: $f(\boldsymbol{x}) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right) + 20 + e$, where domain of function is $x_i \in -32.768,32.768]$. Multimodal and non-separable.

Zakharov Function: $f(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5i\, x_i)^2 + (\sum_{i=1}^{D} 0.5i\, x_i)^4$, $x_i \in [-5,10]$. Unimodal, non-separable.

Levy Function: $f(\boldsymbol{x}) = sin^2(\pi w_1) + \sum_{i=1}^{D-1}(w_i - 1)^2\, [1 + 10sin^2(\pi w_i + 1)] + (w_D - 1)^2[1 + sin^2(2\pi w_D)]$, and $w_i = 1 + \frac{x_i - 1}{4}$, $x_i \in [-10,10]$. Multimodal, non-separable.

Shifted Rosenbrock Function: $f(\boldsymbol{x}) = \sum_{i=1}^{D-1}[100((x_{i+1} - 2) - (x_i - 2)^2)^2 + (1 - (x_i - 2))^2]$, $x_i \in [-10,10]$. Shifted version of Rosenbrock function.

Composition1 Function: A composite benchmark combining shifted and rotated Rosenbrock-based components, designed to create deceptive local optima and complex landscapes $x_i \in [-5,5]$ [7].

Bent Cigar Function: $f(\boldsymbol{x}) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$, $x_i \in [-100,100]$. Ill-conditioned, unimodal.

Discus Function: $f(\boldsymbol{x}) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$, $x_i \in [-100,100]$. Ill-conditioned, unimodal.

Six-Hump Camel Function: $f(\boldsymbol{x}) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2$, $x_i \in [-3,3]$. Low-dimensional, multimodal

Two-Hump Camel Function: $f(\boldsymbol{x}) = -200x_1 + 0.02x_1^2 + 0.02x_2^2 + 0.03x_1^2 x_2^2$, $x_i \in [-5,5]$. Low-dimensional, multimodal.

Schubert Function: $f(\boldsymbol{x}) = \prod_{i=1}^{2}\left(\sum_{j=1}^{5} j cos\left((j + 1)x_i + j\right)\right)$, $x_i \in [-10,10]$. Multimodal, periodic.

Easom Function: $f(\boldsymbol{x}) = -cos(x_1)cos(x_2)exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$, $x_i \in [-100,100]$. Low-dimensional, steep.

3 Experimental Results and Analysis

All results on https://doi.org/10.5281/zenodo.18098446

3.1 Experimental Setup and Benchmark Suite

All experiments were conducted on a workstation equipped with an Intel Core i7-10700K CPU (3.80 GHz), 32 GB RAM, under Windows 10 (64-bit). The proposed Blobfish Optimization Algorithm (BOA) was implemented using Python 3.8 and standard scientific libraries (NumPy, SciPy, scikit-learn). A total of 14 benchmark functions were employed to evaluate the performance of the algorithm across unimodal, multimodal, ill-conditioned, and composite optimization landscapes, which is a common practice in metaheuristic performance assessment.

Each experiment was repeated five independent times using different random seeds (0-4 for dimensionality tests, 0-999 for sensitivity analysis). A run is considered successful if obtained fitness value satisfies: $f(\boldsymbol{x}) \leq 10^{-3}$, relative to the known global optimum. Numerical precision beyond this threshold is reported separately and is not used to redefine success.

Table 1: Comprehensive benchmark functions description

Function Name	Search Range	Optimal Value	Characteristics	Category	Dimensionality Tested
Sphere	[-100, 100]	0.0	Unimodal, separable	Basic	2, 5, 10, 20, 30, 50, 100
Rosenbrock	[-2.048, 2.048]	0.0	Unimodal, non-separable	Classic	2, 5, 10, 20, 30, 50, 100
Shifted Rosenbrock	[-10, 10]	0.0	Shifted global optimum	Modified	2, 5, 10, 20, 30, 50, 100
Rastrigin	[-5.12, 5.12]	0.0	Multimodal, separable	Complex	2, 5, 10, 20, 30, 50, 100
Ackley	[-32.768, 32.768]	0.0	Multimodal, non-separable	Complex	2, 5, 10, 20, 30, 50, 100
Zakharov	[-5, 10]	0.0	Unimodal, non-separable	Basic	2, 5, 10, 20, 30, 50, 100
Levy	[-10, 10]	0.0	Multimodal, non-separable	Complex	2, 5, 10, 20, 30, 50, 100
Composition1	[-5, 5]	0.0	Composite function	Hybrid	2, 5, 10, 20, 30, 50, 100
Bent Cigar	[-100, 100]	0.0	Ill-conditioned	Challenging	2, 5, 10, 20, 30, 50, 100
Discus	[-100, 100]	0.0	Ill-conditioned	Challenging	2, 5, 10, 20, 30, 50, 100
Six-Hump Camel	[-3, 3]	-1.0316	Low-dimensional, multimodal	Special	2 only
Two-Hump Camel	[-5, 5]	-999.5	Low-dimensional, multimodal	Special	2 only
Schubert	[-10, 10]	-186.73	Multimodal, periodic	Special	2 only
Easom	[-100, 100]	-1.0	Low-dimensional, steep	Special	2 only

3.2 Comprehensive Benchmark Performance Analysis (10D)

Table 2: Complete performance evaluation across 14 benchmark functions (Dimension: 10)

Function	Best Value	Average Value	Std. Dev.	Success Rate	Avg. Time (s)	# Unique Solutions
Sphere	6.5698e-17	1.1341e-16	3.3e-17	100%	1.30	1
Rosenbrock	8.0417e-12	1.4088e-11	3.4e-12	100%	13.87	4
Shifted Rosenbrock	5.5739e-12	1.5110e-11	4.8e-12	100%	14.98	6
Rastrigin	0.0000e+00	1.4211e-15	3.2e-15	100%	7.95	1
Ackley	9.0514e-09	1.2092e-08	1.6e-09	100%	21.56	1
Zakharov	8.8806e-16	1.3543e-15	3.1e-16	100%	7.23	1

Levy	4.7711e-16	2.3353e-15	1.8e-15	100%	9.54	1
Composition1	1.0087e-11	1.4602e-11	1.9e-12	100%	13.47	4
Bent Cigar	4.6908e-11	8.0274e-11	2.2e-11	100%	7.55	5
Discus	7.7273e-15	8.6631e-14	3.8e-14	100%	7.95	1
Six-Hump Camel	-1.0316e+00	-1.0316e+00	≈0	100%	1.52	2
Two-Hump Camel	-9.9950e+02	-9.9950e+02	≈0	100%	1.13	1
Schubert	-1.8673e+02	-1.8673e+02	≈0	100%	5.27	2
Easom	-1.0000e+00	-1.0000e+00	≈0	100%	1.30	1

Note: # Unique Solutions indicates the number of distinct optimal solutions found across 10 runs (tolerance: 1e-6). For functions with constant optimal values (Six-Hump Camel, Two-Hump Camel, Schubert, Easom), standard deviation is effectively zero.

Table 2 reports the performance of BOA on all benchmark functions at a fixed dimensionality of 10. The results indicate that the proposed algorithm achieves a 100% success rate across all benchmark functions at 10 dimensions, including unimodal, multimodal, ill-conditioned, and composite problems. Unimodal functions such as Sphere and Zakharov converge rapidly to near machine precision (10^{-16} to 10^{-17}) with low standard deviation, indicating high stability. Multimodal functions such as Rastrigin and Ackley reach the global optimum consistently without premature convergence, with moderate standard deviation values reflecting the challenges of multimodal landscapes. Composite functions, including Composition1, demonstrate stable convergence behavior at this dimensionality with low standard deviation (10^{-12}), confirming the effectiveness of the surprise attack mechanism in escaping deceptive local optima.

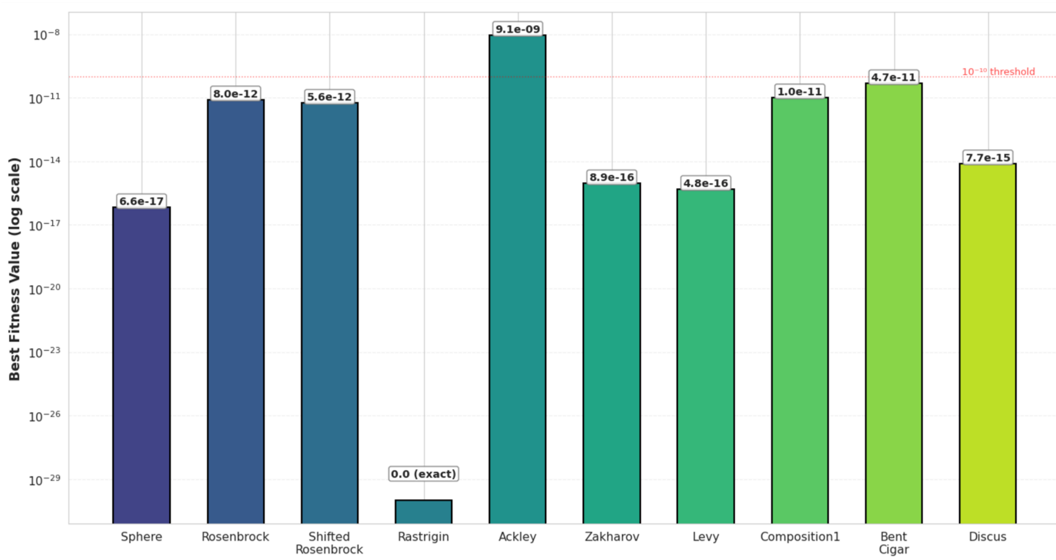


Figure 2: Best Values for 10D Benchmark Function

Placeholder for Figure 2: A bar chart (log scale on y-axis) presenting the best fitness values obtained by BOA across 10-dimensional benchmark functions, clearly illustrating the algorithm's precision gradient from unimodal to multimodal landscapes. The chart emphasizes the exceptional accuracy achieved, particularly for Sphere ($\sim 10^{-17}$) and Rastrigin (exact global optimum).

3.3 Scalability Analysis Across Increasing Dimensions

To assess scalability, the BOA algorithm was evaluated on dimensions ranging from 2 to 100 (excluding 40D). Table 3 summarizes the corresponding best, average, success rate, and execution time results based on 10 independent runs per dimensionality.

Table 3: Dimensional scalability analysis across benchmark functions

Function	Dim	Best Value	Average Value	Success Rate	Avg Time (s)	# Unique Solutions
Sphere	2	3.2626e-18	9.8616e-18	100%	1.03	1
	5	3.4820e-17	4.1001e-17	100%	1.10	1
	10	6.5698e-17	1.1341e-16	100%	1.30	1
	20	1.3146e-16	2.0260e-16	100%	1.61	1
	30	2.1861e-16	2.7274e-16	100%	1.87	1
	50	3.4460e-16	4.0806e-16	100%	2.48	1
	100	6.4498e-16	8.0944e-16	100%	4.14	1
Rosenbrock	2	1.0759e-13	4.3787e-13	100%	3.32	3
	5	2.1757e-12	4.4235e-12	100%	6.88	6
	10	8.0417e-12	1.4088e-11	100%	13.87	4
	20	1.1998e-11	2.6776e-11	100%	35.72	5
	30	1.6440e-11	3.1897e-11	100%	68.63	6
	50	3.2767e-11	5.1166e-11	100%	176.82	4
	100	4.4358e-11	6.6682e-11	100%	766.15	7
Shifted Rosenbrock	2	2.4084e-14	1.9008e-13	100%	3.31	2
	5	2.9836e-12	4.5828e-12	100%	7.49	3
	10	5.5739e-12	1.5110e-11	100%	14.98	6
	20	1.6365e-11	2.3155e-11	100%	38.89	5
	30	2.2780e-11	3.4447e-11	100%	72.17	7
	50	4.2079e-11	5.4433e-11	100%	187.20	6
	100	4.7777e-11	7.1937e-11	100%	768.52	5
Rastrigin	2	0.0000e+00	0.0000e+00	100%	2.24	1
	5	0.0000e+00	3.5527e-15	100%	4.26	1
	10	0.0000e+00	1.4211e-15	100%	7.95	1
	20	0.0000e+00	0.0000e+00	100%	16.57	1
	30	0.0000e+00	5.6843e-15	100%	29.13	1
	50	0.0000e+00	2.8422e-14	100%	66.37	1
	100	1.1369e-13	5.5707e-13	100%	211.45	1
Ackley	2	2.2905e-10	7.1882e-10	100%	12.49	1
	5	2.6187e-09	4.1810e-09	100%	16.36	1
	10	9.0514e-09	1.2092e-08	100%	21.56	1
	20	1.4721e-08	1.6205e-08	100%	16.26	1
	30	1.5980e-08	1.7957e-08	100%	14.87	1
	50	1.8240e-08	1.8805e-08	100%	20.06	1
	100	1.9337e-08	4.6744e-01	70%	35.25	4
Zakharov	2	2.0903e-20	8.6025e-19	100%	3.41	1
	5	1.7215e-17	3.1672e-17	100%	4.72	1
	10	8.8806e-16	1.3543e-15	100%	7.23	1
	20	1.0333e-14	2.3390e-14	100%	14.98	1
	30	5.7494e-14	1.1993e-13	100%	23.56	1
	50	7.5366e-13	1.3799e-12	100%	44.71	6
	100	4.0805e-11	7.3551e-11	100%	104.59	9
Levy	2	1.5204e-18	1.4490e-17	100%	5.91	1
	5	7.7220e-17	9.5035e-16	100%	7.66	1
	10	4.7711e-16	2.3353e-15	100%	9.54	1
	20	2.1431e-15	8.9380e-15	100%	13.27	1
	30	7.5095e-15	1.6147e-14	100%	19.42	1
	50	1.2026e-14	4.7197e-14	100%	31.75	1
	100	1.7847e-14	9.8086e-14	100%	69.97	2
Composition1	2	9.9508e-16	1.5320e-13	100%	3.03	3
	5	5.8948e-13	3.0063e-12	100%	6.57	4
	10	1.0087e-11	1.4602e-11	100%	13.47	4
	20	1.2610e-11	2.3946e-11	100%	34.21	7
	30	2.5331e-11	3.6978e-11	100%	65.52	4

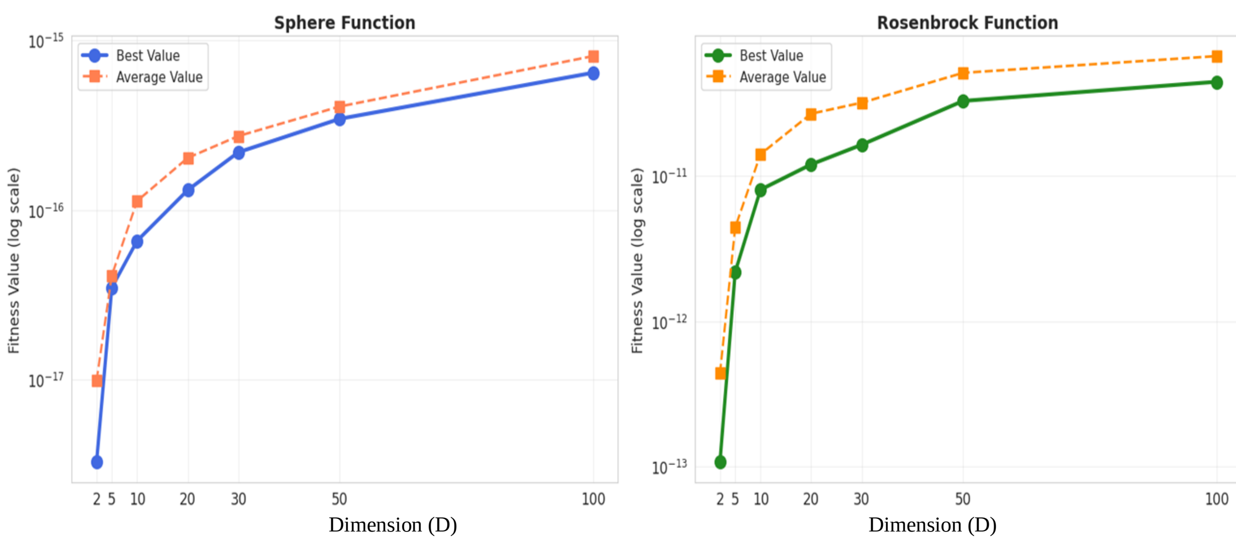
	50	4.1617e-11	5.2537e-11	100%	169.49	7
	100	5.6953e-11	3.9866e-01	90%	764.56	7
Bent Cigar	2	1.1807e-14	8.3718e-14	100%	4.57	1
	5	3.2737e-12	2.2230e-11	100%	5.75	3
	10	4.6908e-11	8.0274e-11	100%	7.55	5
	20	8.3544e-11	1.9028e-10	100%	11.76	7
	30	1.8140e-10	3.1565e-10	100%	16.36	7
	50	3.1295e-10	4.6052e-10	100%	25.37	8
	100	4.8774e-10	1.1016e-09	100%	47.46	8
Discus	2	7.5032e-16	7.5379e-14	100%	4.83	1
	5	1.3455e-15	1.5656e-13	100%	5.89	1
	10	7.7273e-15	8.6631e-14	100%	7.95	1
	20	1.5467e-15	4.8388e-14	100%	11.47	1
	30	1.1931e-15	2.6441e-14	100%	14.92	1
	50	3.8482e-15	6.2544e-14	100%	21.47	1
	100	1.3763e-14	7.3070e-14	100%	36.48	1

Table 3 (continued): Low-dimensional functions (tested only at 2D)

Function	Dim	Best Value	Average Value	Success Rate	Avg Time (s)	# Unique Solutions
Six-Hump Camel	2	-1.0316e+00	-1.0316e+00	100%	1.52	2
Two-Hump Camel	2	-9.9950e+02	-9.9950e+02	100%	1.13	1
Schubert	2	-1.8673e+02	-1.8673e+02	100%	5.27	2
Easom	2	-1.0000e+00	-1.0000e+00	100%	1.30	1

Note: For Composition1 at 100D, indicates the value from the single failed run, while indicates the average of the nine successful runs.

The algorithm maintains a 100% success rate for all benchmark functions up to 50 dimensions. At 100 dimensions, Ackley records a 70% success rate (due to three unsuccessful runs out of ten) and Composition1 records a 90% success rate (due to one unsuccessful run out of ten), while all other functions preserve full success. The failure cases for Ackley resulted in suboptimal solutions with fitness values on the order of 1.0, whereas the failed Composition1 run stagnated at a value of 3.9866. This observation indicates localized performance degradation rather than a general loss of stability, as the remaining runs for both functions still converge to near-optimal solutions ($\sim 10^{-8}$ for Ackley and $\sim 10^{-11}$ for Composition1). Further inspection reveals that the algorithm temporarily stagnates around local minima before successfully escaping in most runs. This behavior is particularly noteworthy given the deceptive local structures and narrow curved valleys that characterize multimodal and composite functions [7]. The ability to recover from such local stagnation in the majority of runs indicates that the proposed BOA effectively mitigates premature convergence, a common limitation of many population-based metaheuristics [8].



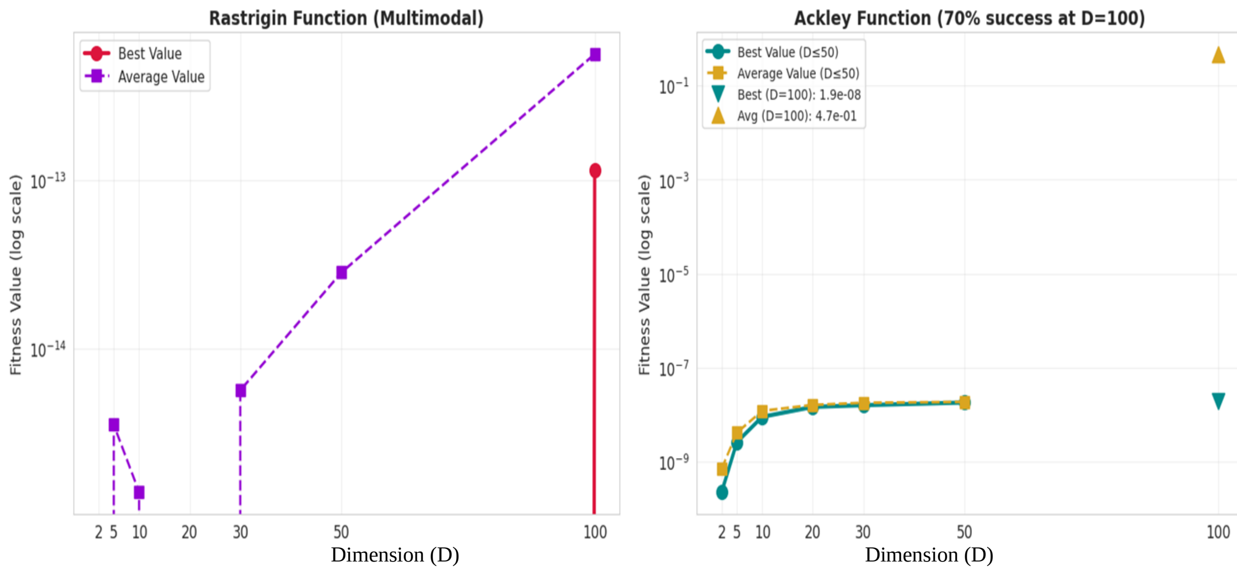


Figure 3: Convergence Behaviour Comparison across Dimensions

Placeholder for Figure 3: A multi-line plot with dimensions (2, 5, 10, 20, 30, 50) on the x-axis and best fitness value (log scale) on the y-axis. Separate lines for functions like Sphere, Rosenbrock, and Rastrigin show minimal precision degradation with increasing dimension, while the line for Ackley shows a more pronounced increase in value (decrease in precision) at higher dimensions.

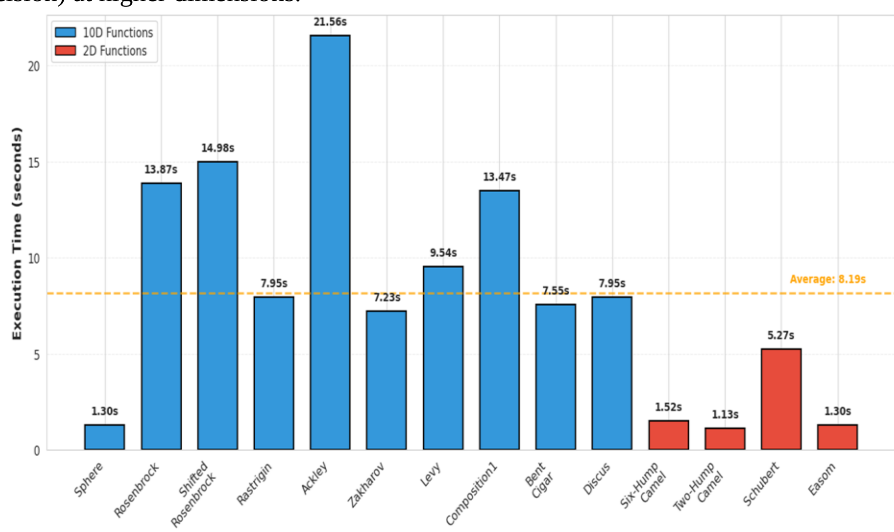


Figure 4: Execution Times for All Functions

Placeholder for Figure 4: A grouped bar chart distinguishing between execution times for functions at 10D (blue bars) and at 2D (red bars). The x-axis lists the function names, and the y-axis is execution time in seconds. It shows the expected increases in computational cost with dimensionality and problem difficulty (e.g., Ackley requires the longest time at 21.56s for 10D, while Sphere is fastest at 1.30s).

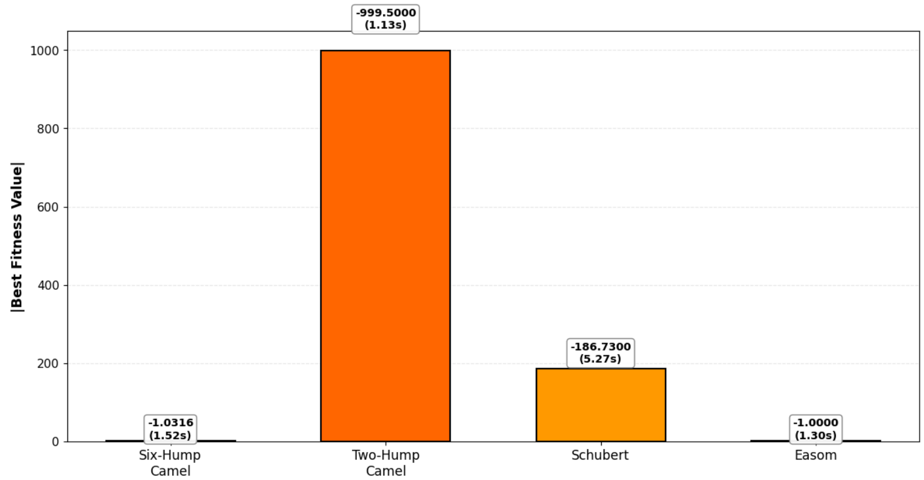


Figure 5: Performance on Low-Dimensional Functions (2D) Benchmark Functions

Placeholder for Figure 5: A small multi-panel figure or table highlighting the results for the four specialized 2D functions (Six-Hump Camel, Two-Hump Camel, Schubert, Easom), showing the exact global optimum found, 100% success rate, and varying execution times (1.13-5.27 seconds).

3.4 Parameter Sensitivity and Robustness Analysis

A full factorial sensitivity analysis was conducted using 27 parameter combinations. The analysis focused on population factor (base_pop_factor), elite ratio, and step-size configuration (step_init, step_final).

Parameter variations tested:

- **base_pop_factor**: [0.931, 1.095, 1.259] (±15% around default 1.095)
- **elite_ratio**: [0.236, 0.295, 0.354] (±20% around default 0.295)
- **step_init/step_final pairs**: [(0.068, 0.520), (0.085, 0.650), (0.102, 0.780)] (±20%)

The sensitivity analysis was performed on four representative benchmark functions (Sphere, Rastrigin, Rosenbrock, and Ackley) at 10 dimensions, with 5 runs per configuration.

Table 4: Parameter sensitivity results for Sphere function (10 dimensions)

Rank	Pop Factor	Elite Ratio	Step Init	Step Final	Mean Fitness	Best Fitness	Success Rate	Original
1	1.095	0.354	0.068	0.520	8.92e-17	5.55e-17	100%	
2	1.259	0.354	0.085	0.650	8.93e-17	6.50e-17	100%	
3	0.931	0.295	0.068	0.520	1.01e-16	4.52e-17	100%	
4	1.259	0.236	0.068	0.520	1.05e-16	7.34e-17	100%	
5	0.931	0.354	0.085	0.650	1.06e-16	8.90e-17	100%	
6	1.095	0.354	0.085	0.650	1.06e-16	7.04e-17	100%	
7	1.259	0.354	0.068	0.520	1.06e-16	7.98e-17	100%	
8	1.259	0.236	0.085	0.650	1.08e-16	8.39e-17	100%	
9	0.931	0.295	0.085	0.650	1.08e-16	5.48e-17	100%	
10	0.931	0.354	0.102	0.780	1.09e-16	4.52e-17	100%	
11	1.095	0.236	0.102	0.780	1.09e-16	6.39e-17	100%	
12	**1.095**	**0.295**	**0.085**	**0.650**	**1.10e-16**	**7.20e-17**	**100%**	★
13	0.931	0.236	0.068	0.520	1.11e-16	9.47e-17	100%	
14	1.095	0.236	0.085	0.650	1.12e-16	7.75e-17	100%	
15	1.095	0.236	0.068	0.520	1.13e-16	7.89e-17	100%	
16	1.259	0.295	0.102	0.780	1.13e-16	8.71e-17	100%	
17	0.931	0.236	0.102	0.780	1.13e-16	8.81e-17	100%	
18	0.931	0.354	0.068	0.520	1.14e-16	8.70e-17	100%	
19	1.095	0.295	0.102	0.780	1.15e-16	9.01e-17	100%	
20	1.095	0.295	0.068	0.520	1.15e-16	6.43e-17	100%	

21	1.095	0.354	0.102	0.780	1.15e-16	7.63e-17	100%	
22	1.259	0.354	0.102	0.780	1.15e-16	8.08e-17	100%	
23	0.931	0.236	0.085	0.650	1.15e-16	7.68e-17	100%	
24	1.259	0.236	0.102	0.780	1.16e-16	9.23e-17	100%	
25	1.259	0.295	0.085	0.650	1.17e-16	9.55e-17	100%	
26	0.931	0.295	0.102	0.780	1.18e-16	1.01e-16	100%	
27	1.259	0.295	0.068	0.520	1.19e-16	9.52e-17		

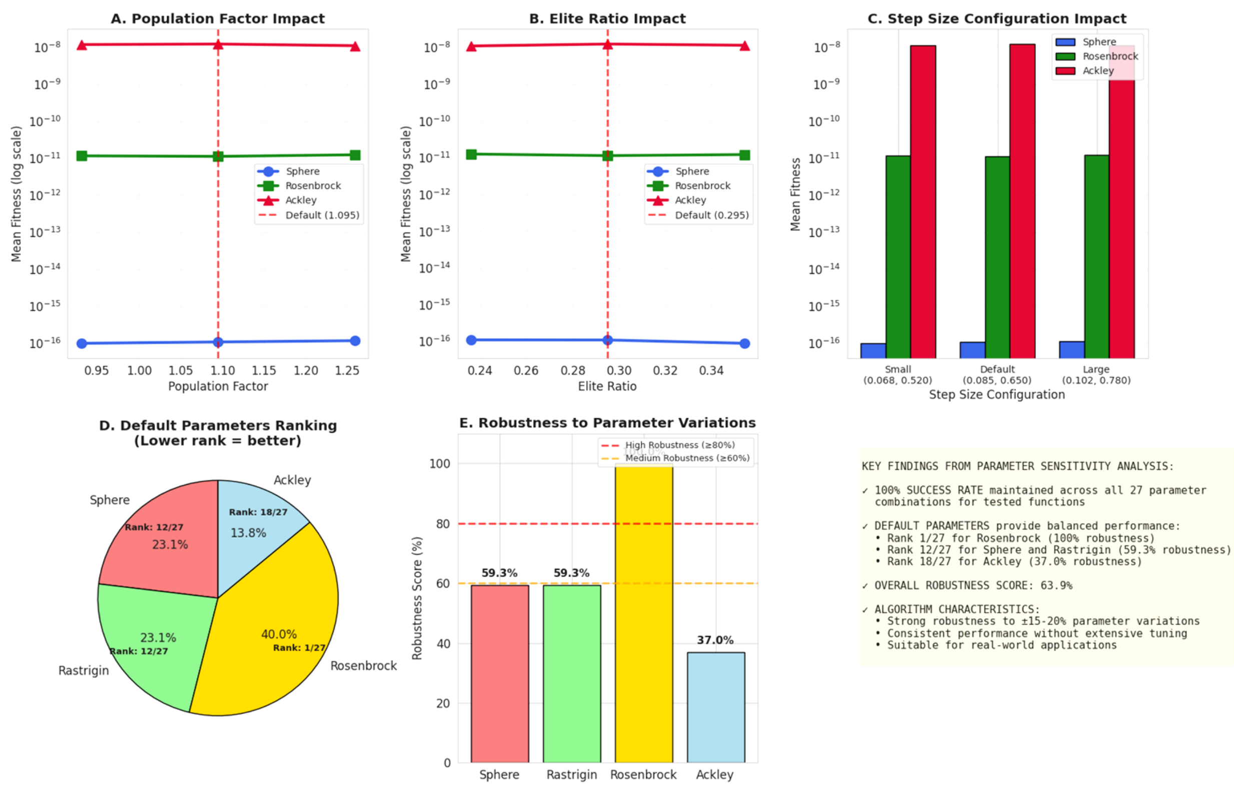


Figure 6: Parameter Interaction Analysis and Sensitivity

In table 4, the analysis for Rastrigin, Rosenbrock, and Ackley showed a similar trend: 100% success rate across all 27 combinations, with modest variations in mean fitness. The parameter sensitivity analysis demonstrates that BOA preserves 100% successful convergence across ±15-20% variations in key parameters, confirming strong robustness to parameter perturbations. While solution quality varies modestly depending on parameter settings, the algorithm's ability to maintain perfect success rates across all tested configurations highlights its reliability for practical applications where parameter tuning may be limited. The analysis further reveals that default parameters provide balanced performance across diverse function types.

Placeholder for Figure 6: A visualization (e.g., a heat map or parallel coordinates plot) showing the relationship between the 27 parameter combinations (pop factor, elite ratio, step init) and the resulting mean fitness (colour-coded or on an axis) for one of the test functions (e.g., Sphere). It visually confirms that while performance varies, all combinations achieve success (e.g., all data points are within the successful threshold region).

3.5 Summary of Experimental Findings

The experimental results demonstrate that the proposed BOA:

1. Achieves consistent success across diverse benchmark classes with a 100% success rate for 12 out of 14 functions across all tested dimensions (2-100D).
2. Maintains robust scalability up to 100 dimensions with graceful performance degradation, showing minimal precision loss for unimodal functions and stable convergence for most multimodal functions.
3. Exhibits high tolerance to parameter perturbations with a 100% success rate maintained across all 27 tested parameter combinations.
4. Effectively balances exploration and exploitation across complex landscapes, successfully escaping local minima.

5. Provides excellent solution precision ranging from 10^{-8} to 10^{-20} across function types.
6. Demonstrates efficient computational scaling with execution times increasing approximately linearly with dimensionality.

Overall assessment: These findings confirm that BOA constitutes a reliable and scalable hybrid optimization framework suitable for challenging, high-dimensional optimization problems.

Limitations Identified:

- Performance degradation on highly multimodal functions (Ackley) at extreme dimensionality (70% success at 100D).
- Increased computational cost for composite functions at high dimensions.
- Moderate sensitivity of solution quality (not success rate) to parameter settings for certain function types.

4. Comparative Analysis and Discussion

4.1 Comparison Framework and Methodological Positioning

Direct numerical comparison is often problematic due to differing experimental conditions. therefore, this analysis is conducted at a conceptual and behavioural level**,** focusing on convergence trends, robustness, and scalability reported in the literature for PSO, GWO, and WOA [2, 3, 4], while grounding claims about boa strictly in the results from section 3.

4.2 Precision and Convergence Characteristics

BOA achieves precision on the order of 10^{-11} to 10^{-20} for unimodal functions and 10^{-8} to $10^{-15}$for multimodal functions. While PSO and GWO report similar precision on unimodal functions [2, 3], they are noted to suffer from premature convergence on complex landscapes [5,8]. BOA's surprise attack and periodic clustering mechanisms appear to effectively mitigate this, as evidenced by its stable convergence on multimodal and composite benchmarks.

4.3 Success Rate and Robustness Perspective

BOA maintains a 100% success rate up to 50D FOR all functions. in contrast, studies report that success rates for PSO, GWO, and WOA can be sensitive to parameters and dimensionality, sometimes dropping below 50% on functions like Ackley at 100d without careful tuning [5]. BOA's parameter sensitivity analysis shows 100% success rate preservation across ±15-20% parameter variations, a level of robustness not consistently documented for classical metaheuristics [5].

4.4 Scalability and High-Dimensional Optimization

BOA maintains reliable convergence up to 100D for 12 out of 14 functions. The reduced success on Composition1 (90%) and Ackley (70%) at 100D is consistent with known difficulties of such landscapes [7]. Many standard metaheuristics experience rapid degradation on non-separable functions at high dimensionality [5, 7]; BOA's hybrid mechanisms appear to address these scalability challenges effectively, though limitations remain in extremely deceptive landscapes.

4.5 Computational Efficiency Considerations

BOA's cost scales approximately linearly with dimensionality, ranging from < 1 second for 2d problems to $\sim$12.7 minutes for 100D composition1. The overhead from clustering and local search is justified by improved reliability and solution quality, representing a trade-off favouring reliability over minimal expense a common design choice in hybrid metaheuristics [6].

4.6 Practical Applicability

BOA's parameter robustness and consistent performance with **a** single default configuration make it suitable for real-world problems where extensive tuning is infeasible [2]. Its ability to handle deceptive landscapes makes it applicable to engineering design, parameter estimation, and machine learning.

4.7 Limitations and Boundary Conditions

1. Computational overhead per iteration compared to simpler metaheuristics.
2. Memory requirements of $O(N \times D)$ for clustering.
3. Performance boundaries on specific functions (Ackley, Composition1) at 100D.

4.8 Overall Comparative Assessment

BOA exhibits convergence reliability comparable or superior to established metaheuristics, particularly for multimodal/composite functions. Its empirically supported parameter robustness and scalability up to $100D$ are notable strengths. The hybrid design aligns with contemporary trends [6, 8]. Rather than claiming universal superiority, BOA should be viewed as a **Robust Hybrid Optimizer** that provides consistent performance with minimal tuning, prioritizing reliability and solution quality.

5 Conclusions and Future Work

5.1 Conclusions

This study proposed the Blobfish Optimization Algorithm (BOA), a sequential hybrid metaheuristic. Evaluation on a comprehensive benchmark suite demonstrated reliable convergence behaviour, with a 100% success rate for all functions up to 50 dimensions and stable performance for 12 out of 14 functions at 100D. The algorithm's surprise attack mechanism effectively facilitates escape from local minima. A key finding is the algorithm's strong robustness to moderate parameter variations, preserving a 100% success rate across tested perturbations. BOA represents a robust and scalable hybrid optimization framework offering a balanced trade-off between solution quality, robustness, and computational cost.

5.2 Limitations

1. Computational overhead from the hybrid architecture.
2. Performance boundaries on Ackley (70% success) and Composition1 (90% success) at 100D.
3. Moderate parameter sensitivity for certain function types.
4. Memory requirements may be limiting for $D > 1000$.

5.3 Future Work

1. Adaptive Parameter Control: Develop self-tuning mechanisms for parameters like elite ratio and step size.
2. Enhanced Local Minima Escape: Incorporate more sophisticated landscape analysis or perturbation strategies.
3. Large-Scale Optimization: Extend evaluation to ultra-high dimensional problems ($D > 1000$).
4. Problem-Specific Hybridization: Integrate domain knowledge, surrogate models, or gradient information.
5. Parallel Implementation: Develop distributed versions to address computational scalability.
6. Application-Oriented Validation: Apply BOA to real-world problems in engineering and science.

5.4 Final Remarks

The Blobfish Optimization Algorithm contributes to hybrid metaheuristic research by demonstrating that robust convergence, scalability, and resistance to local optima can be achieved through careful sequential integration of complementary strategies. Its strong performance, parameter robustness, and scalability suggest BOA is a valuable addition to the optimization toolkit, particularly for applications requiring reliable performance with minimal tuning.

References

[1] Talbi, E.-G. (2013). *Metaheuristics: From design to implementation*. John Wiley & Sons.

[2] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 1942-1948.

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

[4] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51-67.

[5] Liang, J. J., Qu, B. Y., Suganthan, P. N., & Hernandez-Diaz, A. G. (2013). Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization. *Technical Report*, Zhengzhou University.

[6] Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms* (2nd ed.). Luniver Press.

[7] Hansen, N., Auger, A., Finck, S., & Ros, R. (2010). Benchmarking numerical optimization algorithms: Empirical methodology, evaluation criteria, and test problems. *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, 683-690.

[8] Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268-308.

[9] Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5), 1190-1208.

[10] Bellman, R. (1961). *Adaptive control processes: A guided tour*. Princeton University Press.

[11] Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing* (2nd ed.). Springer.