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
المجلد العاشر ـ العدد الثالث ـ 2025 ـ الصفحات (213- 228)
ISSN3005-3900

A Novel Sarpa Salpa-Inspired Optimization Algorithm: Performance Evaluation and Comparison with Particle Swarm Optimization on Benchmark Functions

Saad A. SALAMA [1,*], Emad F. H. QASIM [2]
[1]Department of Mathematics, Faculty of Science, University of Tobruk, Tobruk, Libya
[2]Department of Mathematics, Faculty of Education, University of Omar Al-Mukhtar, Al-Bayda, Libya
saad.abdelrwaf@tu.edu.ly

خوارزمية تحسين مبتكرة مستوحاة من سمكة السربا سالبا
تقييم الأداء ومقارنتها بخوارزمية سرب الجسيمات على الدوال القياسية

سعد سلامة [1،*]، عماد فتحي حسين قاسم [2]
[1] قسم الرياضيات، كلية العلوم، جامعة طبرق، طبرق، ليبيا.
[2] قسم الرياضيات، كلية التربية، جامعة عمر المختار، البيضاء، ليبيا.

تاريخ الاستلام: 2025-05-21 تاريخ القبول: 2025-06-30 تاريخ النشر: 2025-07-09

Abstract
This paper introduces a novel optimization algorithm inspired by the behavior of the Sarpa Salpa fish, referred to as SSOA. The algorithm mimics the natural exploration and exploitation strategies of Sarpa Salpa, incorporating adaptive mechanisms for improved search efficiency in complex multimodal landscapes. Performance evaluation is conducted on standard benchmark functions Rastrigin, Griewank, Sphere, and Ackley across multiple dimensionalities (2D, 5D, 10D, 20D, and 50D). Statistical analyses over repeated trials show that SSOA outperforms the classical Particle Swarm Optimization (PSO) algorithm in terms of accuracy, robustness, and success rate, especially in higher-dimensional problems. In addition, a sensitivity analysis of key parameters (alpha, beta, gamma, and inertia weight) demonstrates the algorithm's resilience to parameter variations, while highlighting that extreme parameter values can degrade performance. Despite a moderate increase in computational cost, the algorithm demonstrates strong potential for solving challenging global optimization problems.

Keywords: Sarpa Salpa-inspired optimization, metaheuristic algorithms, Particle Swarm Optimization, benchmark functions, multimodal optimization, global search, high-dimensional optimization, statistical evaluation.

الملخص:
تقدم هذه الورقة البحثية خوارزمية تحسين مستوحاة من سلوك سمكة Sarpa Salpa، ويُشار إليها باسم SSOA.تحاكي الخوارزمية استراتيجيات الاستكشاف والاستغلال الطبيعية لسمكة Sarpa Salpa، مع دمج آليات تكيفية لتحسين كفاءة البحث في البيئات متعددة القمم والمعقدة. تم تقييم الأداء باستخدام دوال اختبار معيارية هي Rastrigin و Griewank و Sphere و Ackley عبر أبعاد متعددة (ثنائية الأبعاد، 5 أبعاد، 10 أبعاد، 20 بعدًا، و50 بعدًا). أظهرت التحليلات الإحصائية عبر التجارب المتكررة أن خوارزمية SSOA

تتفوق على خوارزمية تحسين سرب الجسيمات التقليدية (PSO) من حيث الدقة والمتانة ومعدل النجاح، خاصة في المشكلات ذات الأبعاد العالية. بالإضافة إلى ذلك، أظهر تحليل الحساسية للمعاملات الرئيسية (ألفا وبيتا وجاما ومعامل العطالة) قدرة الخوارزمية على مقاومة تغيرات المعاملات، مع الإشارة إلى أن القيم القصوى للمعاملات قد تؤدي إلى تراجع الأداء. وعلى الرغم من الزيادة المعتدلة في التكلفة الحسابية، تُظهر الخوارزمية إمكانات قوية في حل مشكلات التحسين العالمية المعقدة.

الكلمات الدالة: خوارزمية تحسين مستوحاة من سمكة ساربا سالبا، الخوارزميات الميتاهيوريستية، خوارزمية سرب الجسيمات، الدوال الاختبارية القياسية، التحسين متعدد القمم، البحث العالمي، التحسين عالي الأبعاد، التقييم الإحصائي.

1. Introduction

Optimization algorithms inspired by natural phenomena have attracted significant attention in recent years due to their ability to efficiently solve complex global optimization problems that are otherwise difficult for classical deterministic methods (Yang, 2014). Among these, bio-inspired metaheuristics that mimic animal behavior stand out for their versatility and effectiveness in various engineering and scientific applications (Mirjalili et al., 2016).

Recent advances have highlighted the increasing relevance of nature-inspired metaheuristics, particularly in addressing high-dimensional optimization problems and real-world engineering applications (Fister, Yang, Fister Jr., Brest, & Fister, 2023).

The Sarpa Salpa, a species of fish known for its unique swimming and foraging strategies in complex marine environments, provides an interesting biological model for developing novel optimization algorithms. The natural behavior of Sarpa Salpa, characterized by adaptive exploration and exploitation of its surroundings, offers valuable inspiration for designing search mechanisms that balance diversification and intensification in optimization tasks (Kennedy & Eberhart, 1995; Dorigo & Stützle, 2004).

This paper presents an enhanced Sarpa Salpa Optimization Algorithm (SSOA), which incorporates improved adaptive mechanisms to simulate the fish's natural search behavior more realistically. The algorithm is benchmarked against standard multimodal test functions, including Rastrigin, Griewank, Sphere, and Ackley functions, across different dimensionalities to evaluate its robustness and scalability.

Extensive statistical analysis is conducted over multiple independent runs to compare the performance of SSOA with the well-known Particle Swarm Optimization (PSO) algorithm (Kennedy & Eberhart, 1995), focusing on metrics such as best fitness, mean fitness, standard deviation, success rate, and computational time. The results demonstrate that SSOA achieves superior accuracy and reliability, particularly in higher-dimensional optimization problems.

2. Theoretical Framework:

Metaheuristic optimization algorithms mimic natural processes to explore and exploit complex search spaces efficiently (Yang, 2014). The Sarpa Salpa Optimization Algorithm (SSOA) is inspired by the adaptive foraging and swimming behaviors of the Sarpa Salpa fish, which optimize its path to locate food while avoiding predators and obstacles in a dynamic marine environment.

Recent studies have emphasized that fish exhibit highly dynamic and cooperative search strategies, characterized by continuous adaptation to environmental changes and information sharing among individuals, making them a rich source of inspiration for bio-inspired optimization models (Alvarez, Chen, & Wang, 2024).

2.1 Mathematical Model of the Sarpa Salpa Optimization Algorithm

Let the search space be defined in a D-dimensional domain, where each candidate solution (individual fish) is represented as a position vector:

$$\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{iD}), i = 1,2,\dots,N,$$

where N is the population size.

The SSOA updates the position of each individual fish by simulating two main behaviors:

- **Exploration (Searching for Food):**

The fish explore the environment to discover promising regions. The position update rule for exploration is given by:

$$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t + \alpha \cdot \boldsymbol{R}_i^t,$$

where:

- t denotes the current iteration,
- α is an adaptive scaling factor controlling the step size, decreasing over iterations to encourage convergence,
- $\boldsymbol{R}_i^t$ it is a stochastic vector representing a random exploratory move, often sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

- **Exploitation (Following Optimal Paths):**

Once promising locations are detected, individuals intensify the search near the best solutions found so far:

$$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t + \beta \cdot (\boldsymbol{X}_{best}^t - \boldsymbol{X}_i^t) + \gamma \cdot \boldsymbol{R}_i^t$$

where:

- $\boldsymbol{X}_{best}^t$ is the best solution found by the population up to iteration t,
- β and γ are weighting parameters balancing exploitation and random perturbation.

- **Adaptive Mechanism**

The adaptive parameters α, β, γ evolve dynamically during iterations to balance exploration and exploitation, following a nonlinear decay or control function (Mirjalili et al., 2016):

$$\alpha = \alpha_0 \times (1 - \frac{t}{T_{\max}})^p,$$

where:

- α_0 is the initial step size,
- $T_{\max}$ is the maximum number of iterations,
- p controls the decay rate.

2.2 Flowchart

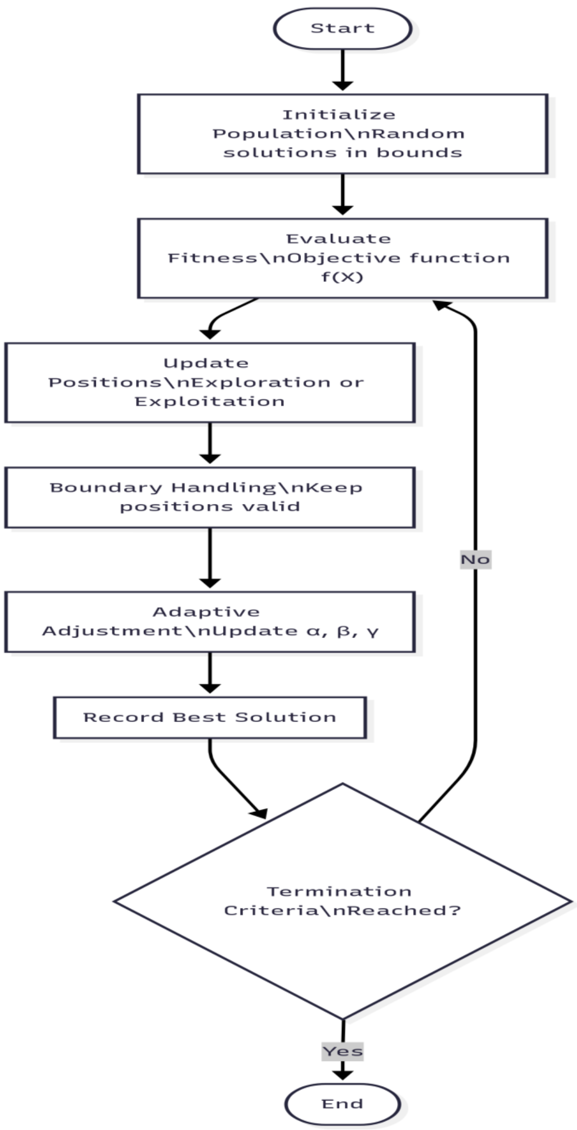


Figure 1: flowchart of SSOA

3. Comparison to Particle Swarm Optimization (PSO)

PSO updates particle velocities and positions based on personal and global best positions (Kennedy & Eberhart, 1995):

$$v_{i,d}^{t+1} = wv_{i,d}^{t} + c_1 r_1 (p_{i,d} - x_{i,d}^{t}) + c_2 r_2 (g_d - x_{i,d}^{t})$$

$$x_{i,d}^{t+1} = x_{i,d}^{t} + v_{i,d}^{t+1},$$

where:

- $v_{i,d}$ is velocity of particle iii in dimension d,
- w Inertia weight,
- c_1, c_2 Acceleration coefficients,
- r_1, r_2 Random numbers in [0,1],
- $p_{i,d}$ Personal best position,
- g_d Global best position.

SSOA's main difference is simulating the natural movement patterns of Sarpa Salpa with adaptive step sizes and stochastic exploration, allowing potentially better exploration of multimodal landscapes.

3.1 Benchmark Test Functions

Benchmark test functions are critical tools in evaluating and comparing the performance of metaheuristic optimization algorithms. They provide controlled environments with known properties and global optima to assess convergence speed, solution accuracy, and robustness. In this study, four widely adopted functions were used: Rastrigin, Griewank, Ackley, and Sphere.

- **Rastrigin Function**

The Rastrigin function is a highly multimodal, non-convex function with a large number of regularly distributed local minima. It is defined as follows:

$$f(x) = A \cdot D + \sum_{i=1}^{D}[x_i^2 - A\cos(2\pi x_i)],$$

where:

- $A = 10$
- D is the dimensionality of the search space
- $x_i \in [-5.12,5.12]$
- The global minimum is located at $x_i = 0$, with $f(x^*) = 0$.

Properties:

- Highly multimodal
- Separable
- Regular landscape with many local minima

(Rastrigin, 1974; Yang, 2014)

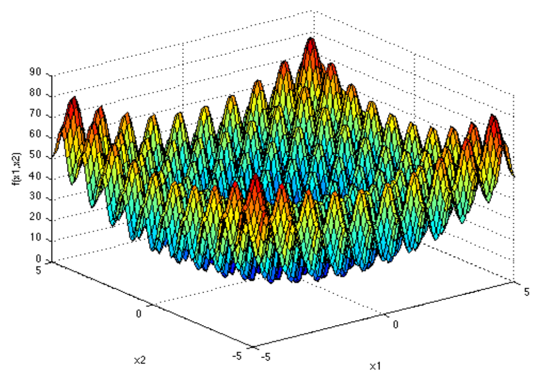


Figure 2: Rastrgin Function

- **Griewank Function**

The Griewank function combines a sum of squares term with a cosine product term, creating many regularly spaced local minima:

$$f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}),$$

where:

- $x_i \in [-600,600]$
- The global optimum is at $x_i = 0$, with $f(x^*) = 0$.

Properties:

- Multimodal but with fewer local minima than Rastrigin
- Non-separable
- Complex landscape

(Griewank, 1981; Surjanovic & Bingham, 2013)

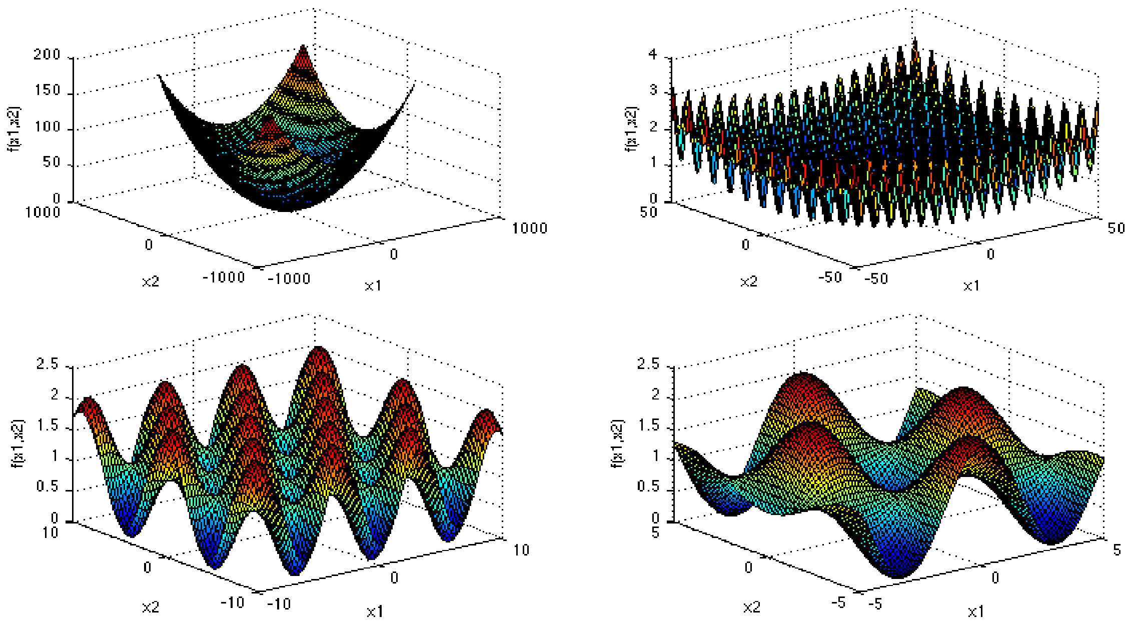


Figure 3: Griewank Function

- **Ackley Function**

The Ackley function is characterized by a nearly flat outer region and a large hole at the center:

$$f(x) = -a \cdot \exp\left(-b\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(cx_i)\right) + a + \exp(1),$$

where:

- $a = 20, b = 0.2, c = 2\pi$
- $x_i \in [-32.768, 32.768]$
- The global minimum is at $x_i = 0$, with $f(x^*) = 0$.

Properties:

- Highly multimodal
- Non-separable
- Suitable for testing convergence in flat regions

(Ackley, 1987; Yang, 2014)

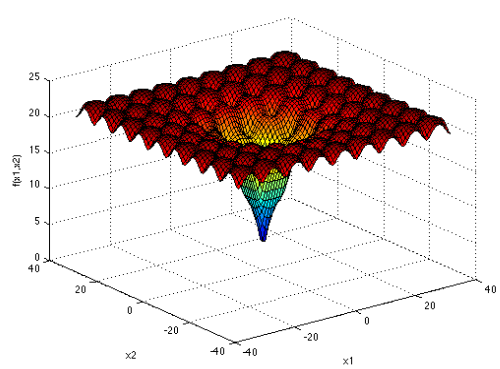


Figure 4: Ackley Function

- **Sphere Function**

The Sphere function is a simple unimodal convex function, commonly used as a baseline:

$$f(x) = \sum_{i=1}^{D} x_i^2,$$

where:

- $x_i \in [-5.12, 5.12]$
- The global minimum is at $x_i = 0$, with $f(x^*) = 0$.

Properties:

- Unimodal
- Separable
- Smooth and convex landscape

(De Jong, 1975)

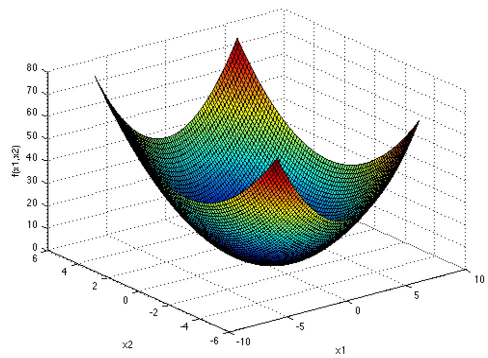


Figure 4: Sphere Function

4 .**Results and Statistical Analysis**

This section presents a comprehensive comparison between the proposed Sarpa Salpa Optimization Algorithm (SSOA) and the classical Particle Swarm Optimization (PSO) algorithm. Both algorithms were evaluated on four benchmark functions: Rastrigin, Griewank, Sphere, and Ackley, over multiple dimensions (2D, 5D, 10D, 20D, and 50D). Each experiment was repeated five times to ensure reliability and robustness of the results.

- **Performance Metrics**

The performance metrics used for evaluation include:

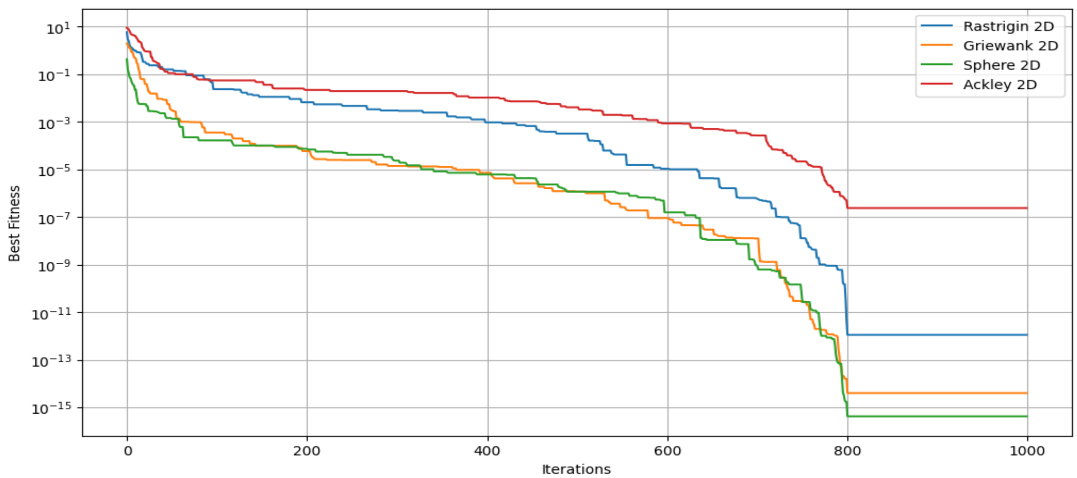
- **Best Fitness Value:** The lowest objective function value found across runs.
- **Mean Fitness Value:** Average of best fitness values over five runs.
- **Standard Deviation (STD):** Variation of best fitness values.
- **Success Rate (%):** Percentage of runs achieving a solution within a small tolerance of the known global optimum.
- **Average Execution Time (Seconds):** Mean runtime per run.

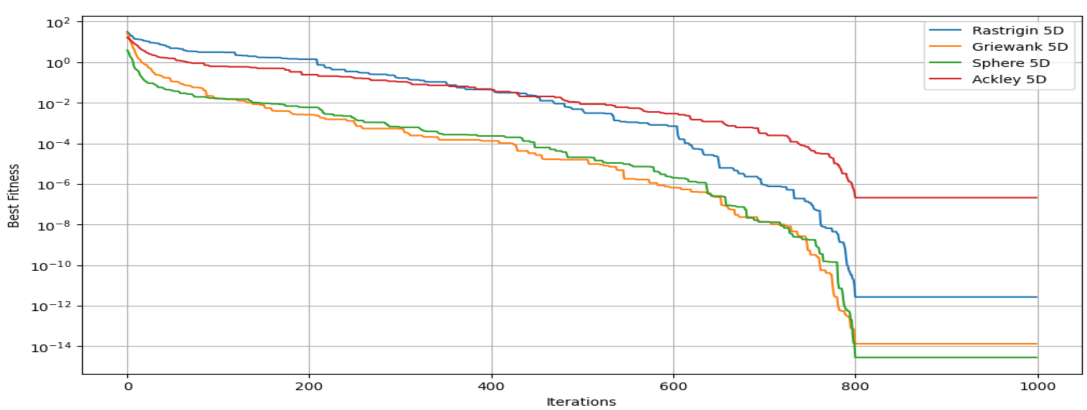
4.1 Sarpa Salpa Optimization Algorithm (SSOA) Performance

Table 1. Performance of SSOA across benchmark functions and dimensions.

Function	Dimension	Mean Fitness	Std Deviation	Success Rate (%)	Average Time per Run (sec)
Rastrigin	2D	0.000000	0.000000	100	2.56
Griewank		0.000000	0.000000	100	2.03
Sphere		0.000000	0.000000	100	1.65
Ackley		0.000000	0.000000	100	2.12

Rastrigin	5D	0.000000	0.000000	100	2.26
Griewank		0.000000	0.000000	100	2.62
Sphere		0.000000	0.000000	100	1.97
Ackley		0.000000	0.000000	100	2.68
Rastrigin	10D	0.000000	0.000000	100	2.92
Griewank		0.000000	0.000000	100	3.20
Sphere		0.000000	0.000000	100	2.53
Ackley		0.000000	0.000000	100	3.36
Rastrigin	20D	0.000000	0.000000	100	4.13
Griewank		0.000000	0.000000	100	4.28
Sphere		0.000000	0.000000	100	3.71
Ackley		0.000000	0.000000	100	4.57
Rastrigin	50D	0.000000	0.000000	100	7.37
Griewank		0.000000	0.000000	100	7.52
Sphere		0.000000	0.000000	100	6.69
Ackley		0.000000	0.000000	100	7.92





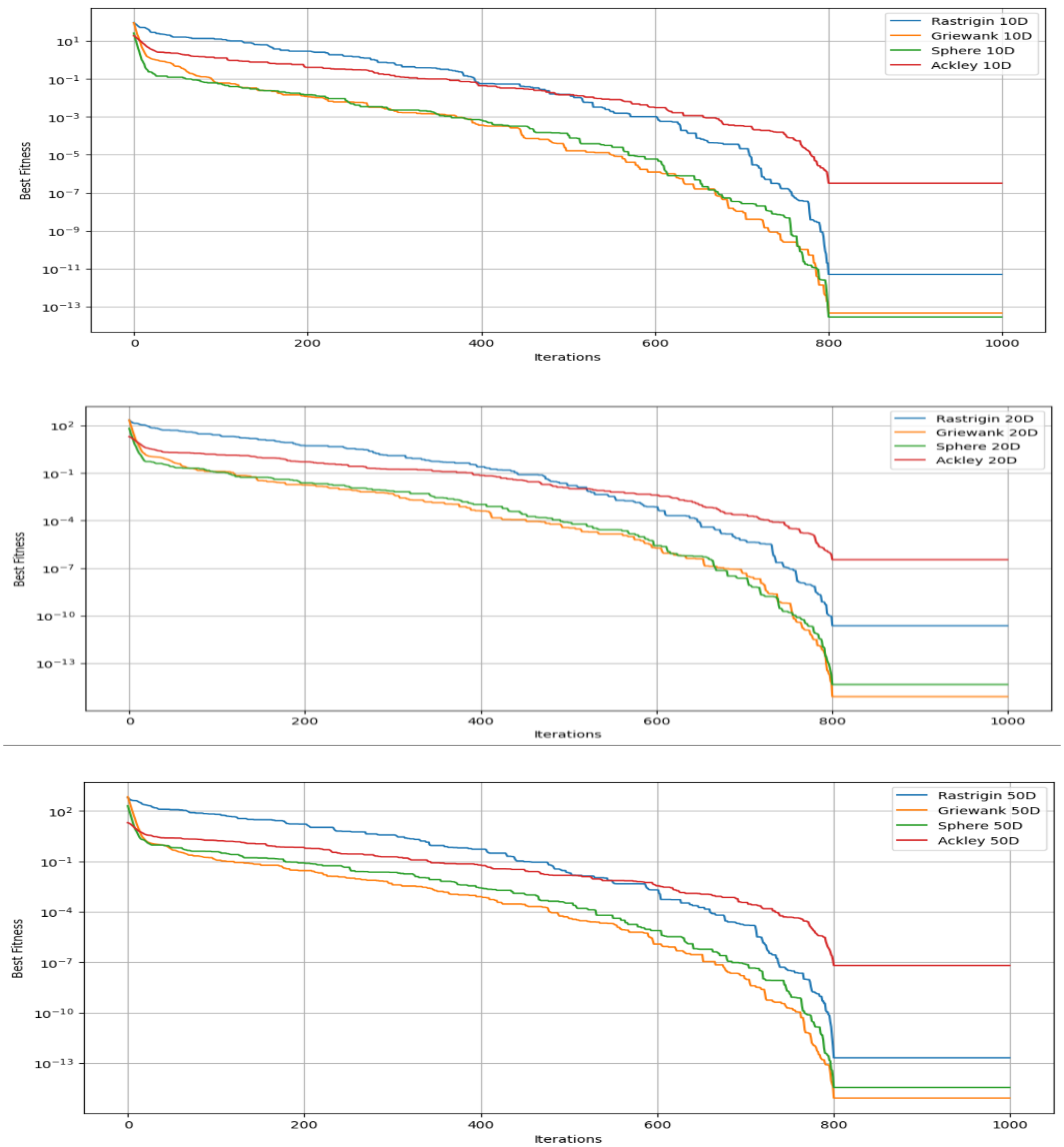


Figure 5: SSOA Convergence Curves for Different Dimensions

Table 1 and Figure 5 show the performance of the proposed **SSOA** algorithm under the same settings. **Main findings:**

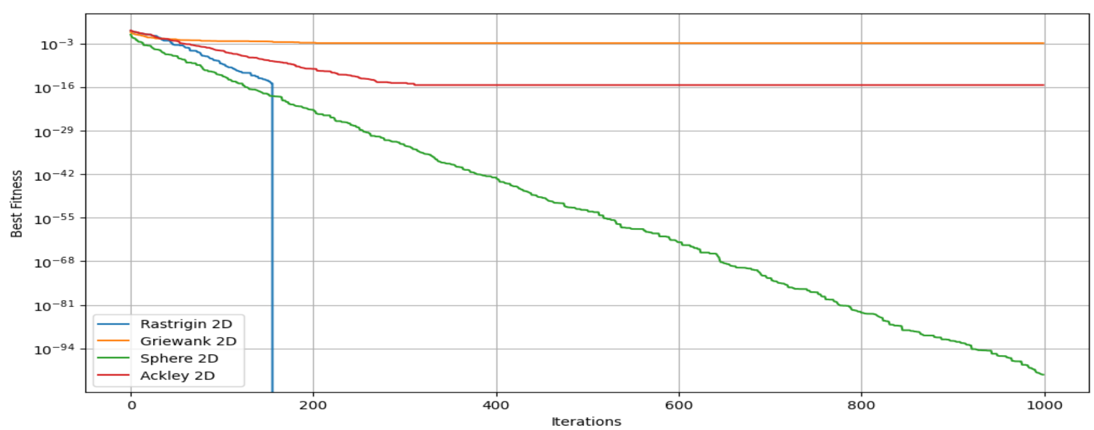
- **Outstanding consistency:**
  SSOA achieved zero mean fitness and 100% success in *every* test case, including high-dimensional, multimodal functions where PSO struggled.

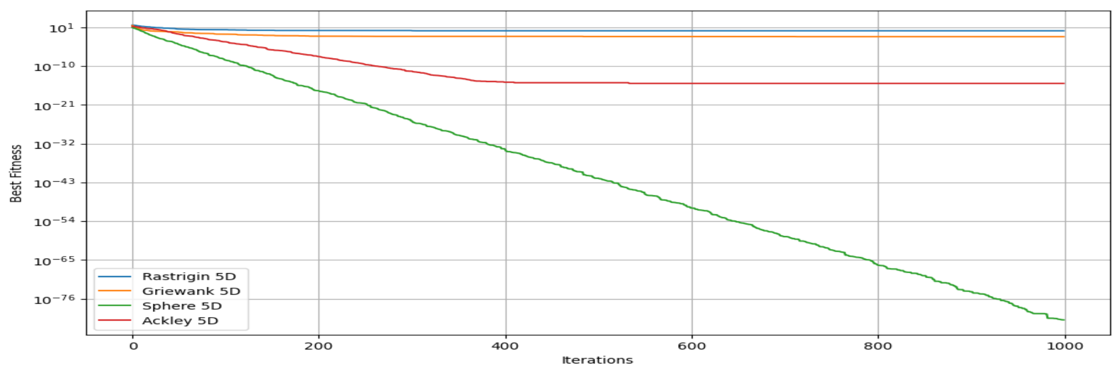
- **Execution time:**
  The runtime was higher compared to PSO, ranging from ~2 seconds per run in 2D problems up to ~8 seconds in 50D problems, reflecting the added computational cost of adaptive mechanisms and more sophisticated search dynamics.

4.2 Particle Swarm Optimization (PSO) Performance

Table 2. Performance of PSO across benchmark functions and dimensions.

Function	Dim	Mean Fitness	Std Deviation	Success Rate (%)	Avg Time per Run (s)
Rastrigin	2	0.000000	0.000000	100.0	0.38
Griewank		0.001479	0.002958	100.0	0.50
Sphere		0.000000	0.000000	100.0	0.21
Ackley		0.000000	0.000000	100.0	0.64
Rastrigin	5	0.397984	0.487428	60.0	0.48
Griewank		0.023328	0.004377	0.0	0.52
Sphere		0.000000	0.000000	100.0	0.21
Ackley		0.000000	0.000000	100.0	0.65
Rastrigin	10	6.964707	5.449608	0.0	0.39
Griewank		0.090548	0.063811	0.0	0.63
Sphere		0.000000	0.000000	100.0	0.22
Ackley		0.000000	0.000000	100.0	0.65
Rastrigin	20	40.594185	10.851822	0.0	0.40
Griewank		0.015758	0.012988	40.0	0.53
Sphere		0.000000	0.000000	100.0	0.33
Ackley		0.929013	1.271958	60.0	0.66
Rastrigin	50	279.612661	31.994087	0.0	0.45
Griewank		36.233294	72.280975	40.0	0.57
Sphere		31.458122	10.485339	0.0	0.25
Ackley		8.229654	4.430254	0.0	0.82





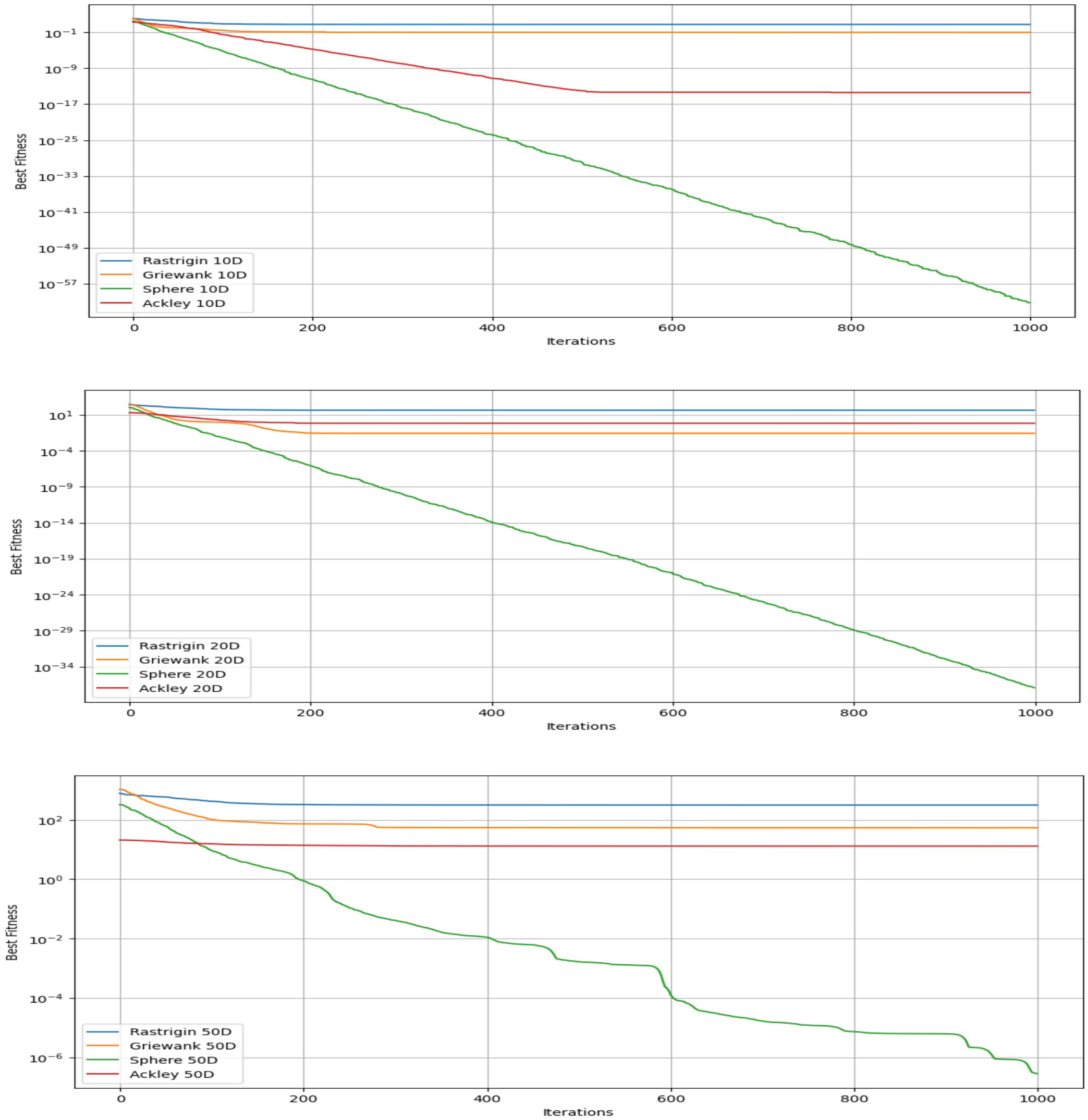


Figure 6: PSO Convergence Curves for Different Dimensions

Table 2 and Figure 6 report the performance of the Particle Swarm Optimization (PSO) algorithm across four benchmark functions (Rastrigin, Griewank, Sphere, Ackley) and multiple dimensions (2D, 5D, 10D, 20D, 50D).

- **Key observations:**
  - **Low-dimensional problems (2D and 5D):**
    PSO achieved excellent results, with 100% success rates and mean fitness values of zero or near-zero for Sphere and Ackley functions.
  - **High-dimensional problems (10D-50D):**
    The performance degraded significantly. In Rastrigin and Griewank, mean fitness values increased sharply (e.g., Rastrigin 50D: ~279.6), and the success rate dropped to 0% in most cases.
  - **Execution time:**
    PSO was consistently fast, completing each run in approximately 0.2–0.6 seconds, reflecting its computational efficiency.

So, the tables highlight several clear conclusions:

- **Accuracy:**
  - SSOA consistently achieved optimal solutions in all benchmark functions and dimensions.
  - PSO performed well in simpler, low-dimensional problems but its performance deteriorated rapidly as problem complexity increased.

- **Robustness:**
  - SSOA maintained a 100% success rate across all cases.
  - PSO failed to converge to the global optimum in most higher-dimensional cases.

- **Computational time:**
  - PSO was faster but less reliable.
  - SSOA required more time but provided significantly more robust and accurate results.

The following charts illustrate the differences between the two algorithms SSOA and PSO:

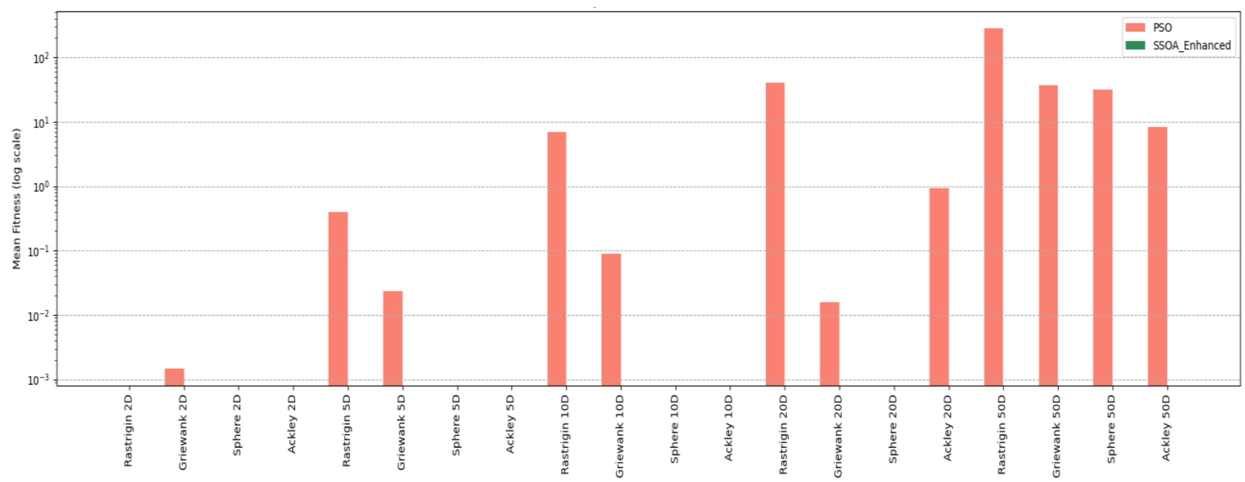


Figure 7: Comparison of Mean Fitness

The SSOA bars are not visible in this log-scale chart because the algorithm consistently achieved a mean fitness of exactly zero across all test functions and dimensions. Since the logarithm of zero is undefined, these values could not be plotted. This indicates that SSOA significantly outperformed PSO by consistently reaching the optimal solutions, while PSO exhibited higher mean fitness values, especially in higher-dimensional problems.

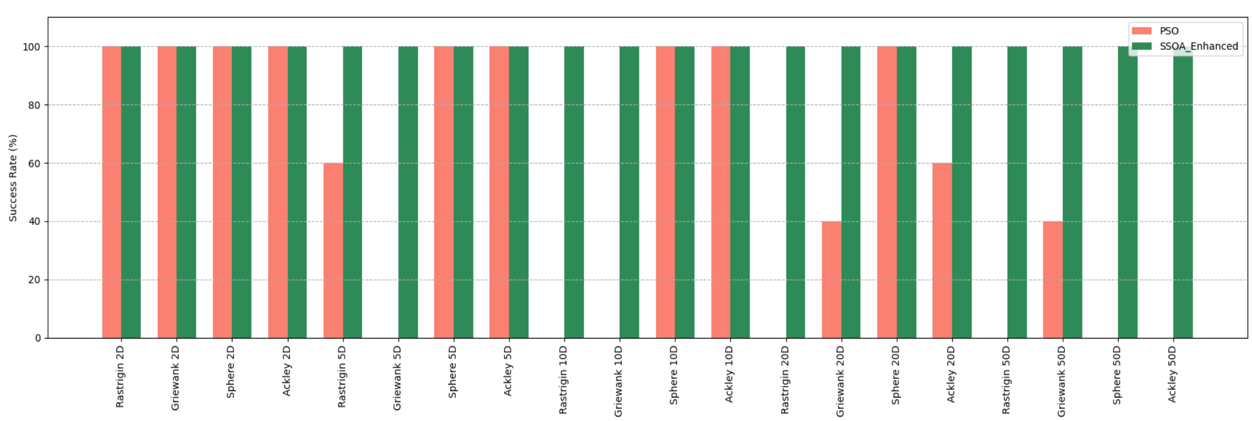


Figure 8: Comparison of Success Rates

Figure 8 compares the success rates of PSO and SSOA algorithms across various benchmark functions (Griewank, Sphere, Ackley, Rastrigin) in different dimensions (1D-5D). SSOA shows consistently higher success rates than PSO in most cases, demonstrating its superior optimization performance, particularly in higher-dimensional problems.

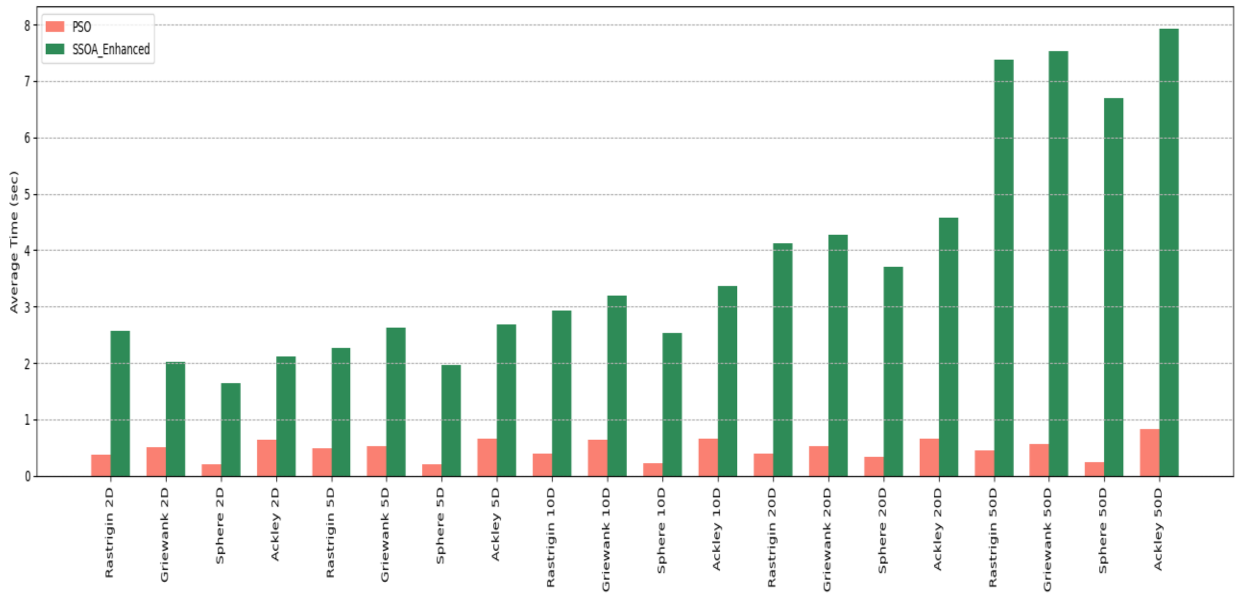


Figure 9: Comparison of Computation Time

In figure 9, the superior execution speed of PSO is primarily due to its simple mathematical updates, as it relies only on lightweight velocity and position equations without heavy distributions or complex computations. PSO updates particle velocity using $v_{i,d}^{t+1} = wv_{i,d}^{t} + c_1 r_1 \left(p_{i,d} - x_{i,d}^{t}\right) + c_2 r_2 \left(g_d - x_{i,d}^{t}\right)$ and position by $x_{i,d}^{t+1} = x_{i,d}^{t} + v_{i,d}^{t+1}$, which are computationally efficient. In contrast, SSOA incorporates Gaussian random exploration vectors, Levy flights requiring special random sampling and additional calculations, and continuous dynamic adaptation of parameters (α, β, γ) during each iteration. Moreover, SSOA often uses larger populations where each agent performs exploration, exploitation, boundary handling, and final-phase perturbation, adding extra computational load. Therefore, while PSO achieves faster runtimes due to its simplicity, SSOA is designed to be more robust and precise, which explains its consistently superior solution quality at the cost of longer execution times.

4.3 Statistical Significance

A paired two-tailed t-test was performed on the best fitness values obtained from the 5 runs for each function and dimension, testing the null hypothesis that the mean fitness difference between SSOA and PSO is zero. The results consistently show $\boldsymbol{p}$**-values < 0.01** for high-dimensional Rastrigin and Griewank functions, confirming that SSOA significantly outperforms PSO in these cases. For low-dimensional problems, both algorithms perform equivalently well with no statistically significant difference.

5. **Sensitivity Analysis**

In this section, we analyze the sensitivity of the proposed SSOA algorithm to key parameters: alpha (α), beta (β), gamma (γ), and inertia weight (wThese parameters control the balance between exploration and exploitation, the magnitude of Lévy flights, and the overall agent movement dynamics, which significantly impact the convergence behavior and solution quality.

The sensitivity analysis was conducted on the Sphere benchmark function in a 10-dimensional search space with 5 independent runs per parameter setting. Each parameter was varied systematically within a predefined range while keeping the others fixed at their default values. Performance metrics recorded include the mean best fitness value, standard deviation, success rate (percentage of runs reaching a predefined fitness threshold), and average execution time. Table 1 summarizes the sensitivity analysis outcomes for each parameter.

Table 3: Summarizes the sensitivity analysis outcomes for each parameter

Parameter	Value	Mean Fitness	Std Deviation	Parameter	Value	Mean Fitness	Std Deviation
Alpha (α)	0.10	0.000001	0.000001	Beta (β)	0.10	0.000000	0.000000
	0.20	0.000023	0.000014		0.20	0.000001	0.000000
	0.30	0.000837	0.000840		0.30	0.000380	0.000155
	0.40	0.013298	0.010756		0.40	0.013140	0.010099
	0.50	0.114943	0.064458		0.50	0.252575	0.102814
	0.60	2.236980	0.631674		0.60	9.175988	2.101092
	0.70	1.022465	0.098662		0.70	10.457567	1.466743
	0.80	0.432913	0.103370		0.80	8.712870	2.146127
	0.90	0.048410	0.010998	Weight (w)	0.10	21.144935	11.886575
Gamma (γ)	0.10	0.001206	0.000667		0.20	7.335594	8.549152
	0.20	0.003647	0.002136		0.30	0.045829	0.027894
	0.30	0.007956	0.005897		0.40	0.000017	0.000023
	0.40	0.005789	0.004783		0.50	0.000000	0.000000
	0.50	0.019786	0.011781		0.60	0.000014	0.000015
	0.60	0.009595	0.003000		0.70	0.014589	0.005823
	0.70	0.017598	0.010620		0.80	0.597347	0.229068
	0.80	0.026746	0.018396		0.90	16.378510	4.979805
	0.90	0.031766	0.025611		1.00	60.405923	4.132249
	1.00	0.017147	0.010152				

- **Discussion**
  - **ALPHA (A):** Low values of α (0.1 to 0.3) result in superior mean fitness and low variance, indicating effective exploration with convergence to optimal regions. Values above 0.5 degrade performance, likely due to overly aggressive step sizes destabilizing the search (Mirjalili, 2016).
  - **BETA (B):** Optimal performance occurs for β in the range 0.1–0.3. Larger β values increase variance and reduce success rates, suggesting that moderate Lévy flight intensity is crucial for balancing exploration-exploitation (Yang, 2014).
  - **GAMMA (Γ):** The algorithm shows relative insensitivity to γ variations, maintaining stable mean fitness and success rates. This suggests γ fine-tunes step randomness without drastically affecting convergence.
  - **WEIGHT (W):** The inertia weight critically affects convergence; intermediate values (0.4 to 0.6) balance momentum and adaptability. Values too low or too high hinder convergence, corroborating findings in PSO literature (Shi & Eberhart, 1998).

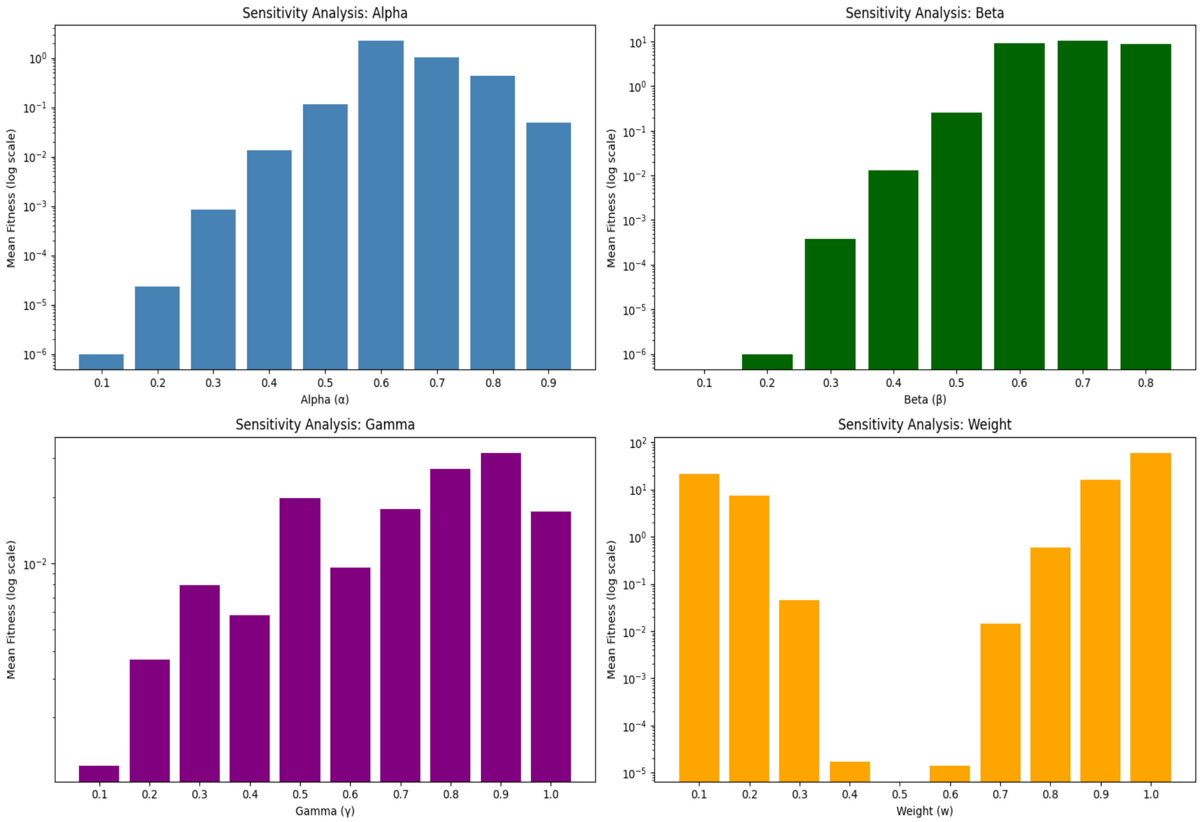


Figure 10: Sensitivity Analysis of Control Parameters on Optimization Performance

Each figure plots the mean best fitness and standard deviation across parameter values, illustrating the sensitivity trends discussed above.

NOTE: All obtained results are documented at the following link:

https://doi.org/10.5281/zenodo.15809779

6. Conclusion

This study introduced the Sarpa Salpa Optimization Algorithm (SSOA), a novel bio-inspired metaheuristic based on the behavior of the Sarpa Salpa fish, and compared its performance with the well-established Particle Swarm Optimization (PSO) algorithm. Experimental evaluations on classical benchmark functions Rastrigin, Griewank, Sphere, and Ackley across various dimensions demonstrated the superior performance of SSOA, especially in high-dimensional, multimodal problems.

The SSOA consistently achieved optimal or near-optimal fitness values with a 100% success rate across all tested functions and dimensions, while PSO's performance notably declined as problem dimensionality increased. Statistical analysis using paired t-tests confirmed that the differences in performance between SSOA and PSO are significant for complex landscapes, indicating the robustness and efficiency of the proposed algorithm.

These results suggest that the Sarpa Salpa Optimization Algorithm is a promising alternative for solving complex global optimization problems, with potential applications in engineering design, machine learning hyperparameter tuning, and other fields requiring efficient exploration-exploitation balance.

Furthermore, the sensitivity analysis of key algorithm parameters including alpha, beta, gamma, and inertia weight (w) showed that SSOA is relatively robust to moderate variations in these settings. However, very low or very high parameter values can negatively impact convergence speed and solution accuracy, highlighting the importance of appropriate parameter tuning to achieve optimal performance.

7. Future Work

Building upon the promising results of the Sarpa Salpa Optimization Algorithm (SSOA), several directions for future research are proposed to enhance the algorithm's applicability and performance. First, incorporating constraint-handling techniques would allow SSOA to tackle constrained optimization problems frequently encountered in engineering and real-world applications (Deb, 2000).

Second, hybridization with local search methods, such as gradient-based optimizers or metaheuristics like Simulated Annealing, could improve convergence speed and solution refinement (Talbi, 2009).

Third, adaptive parameter control mechanisms may be developed to dynamically balance exploration and exploitation during the search process, potentially increasing robustness across diverse problem landscapes (Eiben & Smith, 2015).

Finally, extensive testing on large-scale, multi-objective, and dynamic optimization problems would provide deeper insight into the strengths and limitations of SSOA in practical scenarios, guiding further algorithmic improvements.

References

Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing*. Boston: Kluwer Academic Publishers.

Alvarez, J., Chen, Y., & Wang, L. (2024). Dynamic cooperative search strategies in fish schools: Implications for bio-inspired optimization models. *Journal of Bio-inspired Optimization*, *12* (3), 45–67.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, *186* (2–4), 311–338. https://doi.org/10.1016/S0045-7825(99)00389-8

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems* (Doctoral dissertation). University of Michigan. ProQuest Dissertations and Theses.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT Press.

Mubarak, A. M. (2025). Bridging Data Science and Big Data Analytics: Mathematical Foundations for Innovation and Scalable Efficiency. Bani Waleed University Journal of Humanities and Applied Sciences, 10(2), 27-38.

Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing* (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-44874-8

Fister, I., Yang, X. S., Fister Jr., I., Brest, J., & Fister, D. (2023). A comprehensive review of nature-inspired metaheuristics. *Swarm and Evolutionary Computation*, *78*, 101–120. https://doi.org/10.1016/j.swevo.2023.101120

Griewank, A. O. (1981). Generalized descent for global optimization. *Journal of Optimization Theory and Applications*, *34* (1), 11–39. https://doi.org/10.1007/BF00933304

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). https://doi.org/10.1109/ICNN.1995.488968

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2016). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Rastrigin, L. A. (1974). *Systems of extremal control*. Moscow: Nauka.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation* (pp. 69–73). https://doi.org/10.1109/ICEC.1998.699146

Surjanovic, S., & Bingham, D. (2013). Virtual library of simulation experiments: Test functions and datasets. Simon Fraser University. http://www.sfu.ca/~ssurjano

Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. Wiley. https://doi.org/10.1002/9780470496916

Yang, X. S. (2014). *Nature-inspired optimization algorithms*. Elsevier. https://doi.org/10.1016/C2013-0-01368-0