

Article

The SIOA Algorithm: A Bio-Inspired Approach for Efficient Optimization

Vasileios Charilogis ¹, Ioannis G. Tsoulos ^{1,*} , Dimitrios Tsalikakis ² and Anna Maria Gianni ¹

¹ Department of Informatics and Telecommunications, University of Ioannina, 47150 Kostaki Artas, Greece; v.charilog@uoi.gr (V.C.); am.gianni@uoi.gr (A.M.G.)

² Department of Engineering Informatics and Telecommunications, University of Western Macedonia, 50100 Kozani, Greece; dtsalikakis@uowm.gr

* Correspondence: itsoulos@uoi.gr

Abstract

The Sporulation-Inspired Optimization Algorithm (SIOA) is an innovative metaheuristic optimization method inspired by the biological mechanisms of microbial sporulation and dispersal. SIOA operates on a dynamic population of solutions (“microorganisms”) and alternates between two main phases: sporulation, where new “spores” are generated through adaptive random perturbations combined with guided search towards the global best, and germination, in which these spores are evaluated and may replace the most similar and less effective individuals in the population. A distinctive feature of SIOA is its fully self-adaptive parameter control, where the dispersal radius and the probabilities of sporulation and germination are dynamically adjusted according to the progress of the search (e.g., convergence trends of the average fitness). The algorithm also integrates a special “zero-reset” mechanism, enhancing its ability to detect global optima located near the origin. SIOA further incorporates a stochastic local search phase to refine solutions and accelerate convergence. Experimental results demonstrate that SIOA achieves high-quality solutions with a reduced number of function evaluations, especially in complex, multimodal, or high-dimensional problems. Overall, SIOA provides a robust and flexible optimization framework, suitable for a wide range of challenging optimization tasks.

Keywords: optimization; SIOA; evolutionary algorithms; global optimization; evolutionary techniques; metaheuristics



Academic Editor: Jun-Juh Yan

Received: 11 August 2025

Revised: 8 September 2025

Accepted: 23 September 2025

Published: 4 October 2025

Citation: Charilogis, V.; Tsoulos, I.G.; Tsalikakis, D.; Gianni, A.M. The SIOA Algorithm: A Bio-Inspired Approach for Efficient Optimization.

AppliedMath **2025**, *5*, 135. <https://doi.org/10.3390/appliedmath5040135>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Global optimization constitutes one of the most important areas of computational science, with applications spanning from engineering and physics to economics and artificial intelligence. Its primary objective is to identify the best possible solution in problems characterized by high complexity, nonlinearity, multiple local minima, and high dimensionality. In contrast to local optimization methods, which often become trapped in suboptimal solutions, global optimization techniques aim to systematically explore the entire search space in pursuit of the true global optimum.

To achieve this objective, numerous algorithms have been proposed, ranging from classical derivative-based approaches to modern metaheuristic techniques inspired by natural and biological processes. Within this context, it is essential to present a formal mathematical formulation of the global optimization problem, which provides the theoretical foundation for the development and evaluation of novel algorithmic approaches.

Mathematical formulation of the global optimization problem:

Let $f : S \rightarrow \mathbb{R}$ be a real-valued function of n variables, where $S \subset \mathbb{R}^n$ is a compact subset. The global optimization problem is defined as the problem of finding

$$x^* = \arg \min_{x \in S} f(x), \quad (1)$$

where the feasible set S is given by the Cartesian product

$$S = \prod_{i=1}^n [a_i, b_i] \subseteq \mathbb{R}^n, \quad (2)$$

with the following conditions:

- $f \in C(S)$, i.e., f is a continuous function on S ($C(S)$ denotes the space of continuous functions on S),
- $[a_i, b_i] \subset \mathbb{R}$ are closed and bounded intervals for $i = 1, \dots, n$,
- S is a compact and convex subset of the Euclidean space $\mathbb{R}^n$,
- $x^* \in S$ is the global minimizer of f over S .

Optimization represents a fundamental discipline in computational mathematics with widespread applications across scientific and industrial domains. Optimization techniques can be broadly categorized into several families. Classical gradient-based methods, such as steepest descent [1,2] and Newton's method [3], remain effective for smooth and convex problems. Stochastic approaches, including Monte Carlo sampling [4] and simulated annealing [5], provide robustness against multimodality.

Among the most influential are the population-based metaheuristics, which have become standard in global optimization. Genetic Algorithms (GA) [6], Differential Evolution (DE) [7–9], and Particle Swarm Optimization (PSO) [10] are widely recognized benchmarks. Over the years, several important variants have been developed to improve performance and adaptability, such as self-adaptive Differential Evolution (SaDE) [11], a self-adaptive DE in which each individual carries and co-evolves its own control parameters F and CR (jDE) [12], and Comprehensive Learning Particle Swarm Optimization (CLPSO) [13]. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14] is also regarded as one of the most powerful evolutionary optimizers.

Derivative-free techniques, such as the Nelder–Mead simplex [15], are often applied when gradient information is unavailable. Finally, modern nature-inspired and hybrid strategies (e.g., ant colony optimization [16], artificial bee colony [17]) represent current research directions that integrate concepts from biology, physics, and social systems.

Within the diverse landscape of metaheuristic optimization algorithms, the Sporulation-Inspired Optimization Algorithm (SIOA) introduces an innovative, biologically motivated approach inspired by microbial sporulation and dispersion. SIOA operates with a dynamic population of solutions, conceptualized as “microorganisms” that undergo processes of sporulation and germination. In this framework, each solution can generate “spores” via adaptive random perturbations, guided by the current best solution, with the intensity of dispersal dynamically regulated through self-adaptive parameters. A key distinguishing feature of SIOA is its two-phase search mechanism, combining the generation of spores (sporulation phase) with a germination process that evaluates and selectively integrates new solutions into the population using a similarity-based (crowding) replacement scheme, where each new spore replaces its most similar population member only if it achieves superior fitness. The algorithm incorporates an additional “zero-reset” mechanism, occasionally forcing solution components to zero, which helps to accelerate convergence towards global optima near the origin. The search is further enhanced by a stochastic, optional local search phase, which promotes the exploitation of promising regions in the solution space. One

of SIOA's main strengths lies in its fully self-adaptive parameter control: not only the dispersal radius, but also the probabilities of sporulation and germination are automatically adjusted according to the algorithm's search progress. This adaptive strategy enables SIOA to balance exploration and exploitation effectively, enhancing its performance in a wide range of complex optimization tasks. Multimodal and high-dimensional optimization problems are particularly challenging because they contain a vast number of local optima and exponentially growing search spaces as dimensionality increases. Such conditions often lead conventional algorithms to premature convergence, trapping the search in suboptimal regions and making it difficult to explore promising areas effectively. To address these difficulties, SIOA integrates diversity-preserving mechanisms such as similarity-based replacement and adaptive parameter control, which together sustain population heterogeneity and guide the search toward unexplored regions. These features equip SIOA with the ability to maintain a balance between exploration and exploitation even in very rugged or large-scale landscapes. Notably, in multimodal and high-dimensional problems, SIOA demonstrates a strong capability to avoid premature convergence and maintain population diversity, contributing to fast and robust convergence. Experimental results confirm that SIOA exhibits high efficiency and stability across a broad suite of benchmark functions, often outperforming established metaheuristics, especially in challenging optimization landscapes. The biological inspiration underpinning SIOA offers natural mechanisms for diversity maintenance and premature convergence avoidance, making it especially suitable for demanding applications where a balance between global exploration and focused exploitation is critical. This work thoroughly examines the theoretical underpinnings of SIOA, including its convergence properties, parameter sensitivity analysis, and practical implementation aspects. Furthermore, the potential for extensions and adaptations of the algorithm is explored, including constrained, multi-objective, and large-scale optimization scenarios. Overall, SIOA emerges as a powerful, modern, and flexible contribution to computational optimization methodology, with significant prospects for both research and real-world applications.

The remainder of this paper is structured as follows. Section 2 introduces the proposed SIOA and its biological motivation. Section 3 describes the experimental setup and presents the benchmark results. Specifically, Section 3.1 reports experiments with traditional methods on classical benchmark problems, while Section 3.2 extends the analysis to advanced methods and real-world applications. The balance between exploration and exploitation is investigated in Section 3.3, followed by a parameter sensitivity study in Section 3.4. Section 3.5 provides an analysis of the computational cost and complexity of the SIOA algorithm. Finally, Section 4 summarizes the main findings and outlines directions for future work.

Overall, this structure ensures a coherent presentation of both the methodological contributions and the experimental validation. The next section introduces the SIOA method in detail, highlighting its biological inspiration and algorithmic design principles.

2. The SIOA Method

The following is the pseudocode of SIOA and the related analysis.

To formalize the sporulation process, each spore s_i is generated from its parent solution x_i according to

$$s_i = x_i + R \cdot \mathcal{U}(-1, 1) \cdot c_1 + \beta \cdot (x_{best} - x_i + \mathcal{U}(-R, R)) \cdot c_2, \quad (3)$$

where R denotes the adaptive dispersal radius, c_1 , c_2 are control coefficients, and $\beta \equiv c_2$ acts as the attraction factor toward the global best solution x_{best} . With probability p_{zero} , selected dimensions of the spore are reset to zero, which enhances the ability of the al-

gorithm to detect global optima near the origin. The generated spore is subsequently bounded to the feasible domain by applying the clamp operator. This explicit formulation improves the mathematical clarity of the sporulation step and highlights the role of adaptive perturbations combined with global guidance.

The SIOA Algorithm 1 begins with an initialization phase, where an initial population of solutions (samples) of size NP is randomly generated within the specified bounds. For each solution, the fitness value is evaluated and stored in the fitness array, while the best solution (x_{best}) and its corresponding fitness (f_{best}) are also tracked. An empty list is initialized to collect spores that will be generated in each iteration.

Algorithm 1 Pseudocode of SIOA.

```

Input:
-  $NP$ : Population size
-  $Iter_{max}$ : Maximum iterations
-  $p_{loc}$ : Local search rate
-  $bounds$ : Search space bounds
-  $c1, c2$ : Search coefficients
(Self-adaptive within the loop, initialized with default values:)
-  $R_{min}, R_{max}$ : Min/max dispersal radius
-  $p_{spor}$ : Initial sporulation probability
-  $p_{germ}$ : Initial germination probability
Output:
-  $x_{best}$ : Best solution found
-  $f_{best}$ : Corresponding fitness value
Initialization:
01:  $dim \leftarrow$  Problem dimension
02: Initialize population  $X = x_i | x_i \sim U(bounds), i = 1, \dots, NP$ 
03: Evaluate initial fitness  $F = f_i = f(x_i) | i = 1, \dots, NP$ 
04:  $(x_{best}, f_{best}) \leftarrow \operatorname{argmin}_{(x_i, f_i)} f_i$ 
// Set adaptive parameters:
05:  $R \leftarrow R_{max}$ 
06:  $ps_{ad} \leftarrow p_{spor}$ 
07:  $pg_{ad} \leftarrow p_{germ}$ 
08:  $meanP_{fitness} \leftarrow +\infty$ 
Main Optimization Loop:
09: for  $iter = 1$  to  $Iter_{max}$  do
// Parameter self-adaptation
10:    $t \leftarrow \frac{iter}{Iter_{max}}$ 
11:    $R \leftarrow R_{max} - t \cdot (R_{max} - R_{min})$ 
12:    $meanP_{fitness} \leftarrow \operatorname{mean}(F)$ 
13:    $prog \leftarrow \frac{(best_{prev} - f_{best})}{(|best_{prev}| + \epsilon)}, \epsilon = 1 \times 10^{-10}$ 
14:   if  $prog > 0.001$  then
15:      $ps_{ad} \leftarrow \operatorname{clamp}(ps \cdot 0.98, 0.1, 1.0)$ 
16:      $pg_{ad} \leftarrow \operatorname{clamp}(pg \cdot 1.02, 0.1, 1.0)$ 
17:   else
18:      $ps_{ad} \leftarrow \operatorname{clamp}(ps_{ad} \cdot 1.02, 0.1, 1.0)$ 
19:      $pg_{ad} \leftarrow \operatorname{clamp}(pg_{ad} \cdot 0.98, 0.1, 1.0)$ 
20:   end if
21:    $meanP_{fitness} \leftarrow meanP_{fitness}$ 
// Sporulation phase
22:    $S \leftarrow \emptyset$ 
23:   for each  $x_i$  in  $X$  do
24:     Create vector spore =  $[spore_1, spore_2 \dots spore_{dim}]$ 
25:     for  $d = 1$  to  $dim$  do
26:        $spore_d \leftarrow X_{i,d} + U(-R, R) * c_1 + (x_{best,d} - X_{i,d} + U(-R, R)) * c_2$ 
// Equivalent mathematical form:
//  $s_i \leftarrow x_i + R \cdot U(0, 1) + \beta \cdot (x_{best} - x_i)$  // with  $\beta \equiv c_2$ 
// Special "reset to zero" rule
27:       if  $U(0, 1) < 0.1$  and  $(f_{best} \in (-3, 3))$  then
28:          $spore_d \leftarrow 0$ 
29:       end if
30:        $spore_d \leftarrow \operatorname{clamp}(spore_d, blower_d, bupper_d)$ 
31:     end for
32:      $S \leftarrow S \cup spore$ 
33:   end for
// Germination phase
34:   for each  $spore$  in  $S$  do
35:     if  $U(0, 1) < pg_{ad}$  then
36:        $f_{spore} \leftarrow f(spore)$ 
37:        $idx \leftarrow$  index of sample in  $X$  most similar to  $spore$  (Euclidean distance)
38:       if  $f_{spore} < f_{idx}$  then
39:          $x_{idx} \leftarrow spore$ 
40:          $f_{idx} \leftarrow f_{spore}$ 

```

Algorithm 1 *Cont.*

```

41:         end if
42:         if  $f_{spore} < f_{best}$  then
43:              $x_{best} \leftarrow spore$ 
44:              $f_{best} \leftarrow f_{spore}$ 
45:         end if
46:     end if
47: end for
    // Local search (optional)
48: for each  $x_i$  in  $X$  do
49:     if  $U(0, 1) < p_{loc}$  then
50:          $(x_{ref}, f_{ref}) \leftarrow \text{localSearch}(x_i)$  [18]
51:         if  $f_{ref} < f_i$  then
52:              $x_i \leftarrow x_{ref}$ 
53:              $f_i \leftarrow f_{ref}$ 
54:             if  $f_{ref} < f_{best}$  then
55:                  $x_{best} \leftarrow x_{ref}$ 
56:                  $f_{best} \leftarrow f_{ref}$ 
57:             end if
58:         end if
59:     end if
60: end for
61: if termination criteria met then break:  $\delta_{sim}^{(iter)} = |f_{sim,min}^{(iter)} - f_{sim,min}^{(iter-1)}|$  [19,20] or  $Iter_{max}$  or function evaluations (FEs)
62: end for
63: return  $(x_{best}, f_{best})$ 

```

During the main iteration loop, the algorithm executes three core operations in every cycle:

In the first phase (sporulation), each solution in the population has a probability (p_{spor} , which is self-adaptive) of generating a spore. The new spore is created by applying a combination of adaptive random perturbations and attraction towards the global best solution, with the strength of the perturbation determined by the current value of the adaptive dispersal radius (R). Additionally, with a certain probability, individual dimensions of the spore may be forcibly set to zero, especially when the best fitness value is near zero, enhancing the algorithm's ability to locate optima at or near the origin. All generated spores are ensured to remain within the problem boundaries.

In the second phase (germination), each spore has a probability (p_{germ} , also self-adaptive) to germinate. If so, its fitness is evaluated. The algorithm then uses a crowding (similarity-based) replacement strategy: the spore is compared against the most similar solution in the population (measured by Euclidean distance), and it replaces that solution only if its fitness is superior. If the spore achieves a new best fitness, the x_{best} and f_{best} are updated.

The third phase is optional local search, where each solution in the population has a probability (P_{loc}) of undergoing a specialized local search procedure. If the refined solution is better, it replaces the current one and updates the global best if necessary.

Throughout the process, all critical parameters including dispersal radius and the probabilities of sporulation and germination are dynamically self-adapted based on the search progress, specifically on improvements in the mean fitness of the population. This mechanism ensures that SIOA can automatically balance exploration and exploitation according to the evolving state of the search.

The use of similarity-based (crowding) replacement preserves population diversity and helps prevent premature convergence, while the special zero-reset rule increases the chance of discovering global optima at zero. In practice, the zero-reset mechanism is applied selectively to a fraction of the population when convergence is detected near the coordinate axes or the origin. This targeted reinitialization is particularly effective for benchmark problems with optima located at or close to the origin (e.g., Ackley and Discus), since it enhances the probability of sampling in the true optimum's neighborhood. By introducing controlled diversity only under these conditions, zero-reset improves exploration without disrupting convergence dynamics, thereby increasing accuracy for this important class of optimization problems. The stochastic local search phase further enhances exploitation

capability. Overall, the combination of these mechanisms creates a dynamic, self-adjusting system in which the algorithm continuously tunes its parameters and replacement strategies based on intermediate solution quality, thus maximizing its ability to efficiently explore complex, multimodal, and high-dimensional search spaces.

3. Experimental Setup and Benchmark Results

The experimental framework is structured as follows: First, the benchmark functions used for performance evaluation are introduced, then a thorough examination of the experimental results is provided. A systematic parameter sensitivity analysis is conducted to validate the algorithm’s robustness and optimization capabilities under different conditions. All experimental configurations are specified in Table 1.

Table 1. Parameters and settings.

Parameter	Value	Explanation
NP	100	Population for all methods
p_{spor}	$p_{spor} \in [0, 1]$: adaptive, initial: 0.6	Sporulation probability for SIOA
p_{germ}	$p_{germ} \in [0, 1]$: adaptive, initial: 0.9	Germination probability for SIOA
R_{min}	$R_{min} \in [0, 1]$: adaptive, initial: 0.01	Smaller sporulation radius for SIOA
R_{max}	$R_{max} \in [0, 1]$: adaptive, initial: 0.5	Larger sporulation radius for SIOA
c_1	0.6	Stochastic perturbation
c_2	0.4	Attraction toward the global best
$iter_{max}$	500	Maximum number of iterations for all methods
SR	Similarity of best fitness [19,20] or $iter_{max}$ or FEs	Stopping rule
N_s	12	Similarity $count_{max}$ for stopping rule
P_{loc}	0.005 (0.5%) etc.	Local search rate for all methods (optional)
C_{rate}	double, 0.1 (10%) (classic values)	Crossover for GA
M_{rate}	double, 0.05 (5%) (classic values)	Mutation for GA
cf_1, cf_2	1.193	Cognitive and Social coefficient for PSO
w	0.721	Inertia for PSO
$coef_1, coef_2$	1.494	Cognitive and Social coefficient for CLPSO
w	0.729	Inertia for CLPSO
F	0.8	Initial scaling factor for DE and SaDE
CR	0.9	Initial crossover rate for DE and SaDE
w	$w \in [0.5, 1]$ (random)	Inertia for PSO
NP_C	$Np = 4 + \lfloor 3 \cdot \log(\text{dimension}) \rfloor$	Population for CMA-ES

The computational experiments were conducted using a system equipped with an Debian Linux machine with 128 GB of RAM. The testing framework involved 30 independent runs for each benchmark function, ensuring robust statistical analysis by initializing with fresh random values in every iteration. The experiments utilized a custom-developed tool implemented in ANSI C++ within the OPTIMUS [21] platform, an open-source optimization library available at <https://github.com/itsoulos/GLOBALOPTIMUS> (accessed on 28 July 2025). The algorithm’s parameters, as detailed in Table 1, were carefully selected to balance exploration and exploitation effectively.

3.1. Experiments with Traditional Methods and Classical Benchmark Problems

The evaluation of SIOA was first conducted on established benchmark function sets [22–24], in direct comparison with widely used traditional optimization methods,

in order to assess its computational efficiency, convergence capability, and result stability under standard testing scenarios (Table 2).

Table 2. The benchmark functions used in the conducted experiments.

Name	Formula	Dimension
ACKLEY	$f(x) = -a \exp\left(-b \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + a + \exp(1) \quad a = 20.0$	4
BF1	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) - \frac{4}{10} \cos(4\pi x_2) + \frac{7}{10}$	2
BF2	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) \cos(4\pi x_2) + \frac{3}{10}$	2
BF3	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1 + 4\pi x_2) + \frac{3}{10}$	2
BRANIN	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos(x_1) + 10$ $-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$	2
CAMEL	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad x \in [-5, 5]^2$	2
DIFFERENT POWERS	$f(x) = \sqrt{\sum_{i=1}^n x_i ^{2+4\frac{i-1}{n-1}}}$	10
DIFFPOWER	$f(x) = \sum_{i=1}^n x_i - y_i ^p \quad p = 2, 5, 10$	2, 5, 10
DISCUS	$f(x) = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$	10
EASOM	$f(x) = -\cos(x_1) \cos(x_2) \exp\left((x_2 - \pi)^2 - (x_1 - \pi)^2\right)$	2
ELP	$f(x) = \sum_{i=1}^n (10^6)^{\frac{i-1}{n-1}} x_i^2$	10
EQUAL MAXIMA	$f(x) = \sin^6(5\pi x) \cdot e^{-2\log(2) \cdot \left(\frac{x-0.1}{0.8}\right)^2}$	10
EXP	$f(x) = -\exp(-0.5 \sum_{i=1}^n x_i^2), \quad -1 \leq x_i \leq 1$	10
GKLS [25]	$f(x) = \text{Gkls}(x, n, w) \quad w = 50, 100$	n = 2, 3
GOLDSTAIN	$f(x) = [1 + (x_1 + x_{2+1})^2(19 - 14x_1 + 3x_1^2 14x_2 + 6x_1x_2 + 3x_2^2)]$ $[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2
GRIEWANK ROSENBROCK	$f(x) = \underbrace{\left(\frac{\ x\ ^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1\right)}_{\text{Griewank}} \cdot \underbrace{\left(\frac{1}{10} \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]\right)}_{\text{Rosenbrock}}$	10
GRIEWANK2	$f(x) = 1 + \frac{1}{200} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \frac{\cos(x_i)}{\sqrt{ i }}$	2
GRIEWANK10	$f(x) = 1 + \frac{1}{200} \sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10} \frac{\cos(x_i)}{\sqrt{ i }}$	10
HANSEN	$f(x) = \sum_{i=1}^5 i \cos[(i-1)x_1 + i] \sum_{j=1}^5 j \cos[(j+1)x_2 + j]$	2
HARTMAN3	$f(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	3
HARTAMN6	$f(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	6
POTENTIAL [26]	$V_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6\right]$	9, 15, 30
RARSTIGIN2	$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$	2
ROSENBROCK	$f(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right), \quad -30 \leq x_i \leq 30$	4, 8, 16
ROTATED ROSENBROCK	$f(x) = \sum_{i=1}^{n-1} \left[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2\right], \quad z = Rx$	10
SHEKEL5	$f(x) = -\sum_{i=1}^5 \frac{1}{(x-a_i)(x-a_i)^T + c_i}$	4
SHEKEL7	$f(x) = -\sum_{i=1}^7 \frac{1}{(x-a_i)(x-a_i)^T + c_i}$	4
SHEKEL10	$f(x) = -\sum_{i=1}^{10} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$	4
SINUSOIDAL [27]	$f(x) = -(2.5 \prod_{i=1}^n \sin(x_i - z) + \prod_{i=1}^n \sin(5(x_i - z))),$ $0 \leq x_i \leq \pi$	4, 8, 16

Table 2. *Cont.*

Name	Formula	Dimension
STEP ELLIPSOIDAL	$f(\mathbf{x}) = \sum_{i=1}^n [x_i + 0.5]^2 + \alpha \sum_{i=1}^n \left(10^6 \cdot \frac{i-1}{n-1}\right) x_i^2, \alpha = 1$	4
TEST2N	$f(x) = \frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$	4, 5
TEST30N	$\frac{1}{10} \sin^2(3\pi x_1) \sum_{i=2}^{n-1} \left((x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) \right) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n))$	4, 5

The results presented in Table 3 were obtained using the parameter settings described in Table 1. An important observation is the consistency of the best solution across 12 consecutive runs, which demonstrates a high degree of stability and robustness in the optimization process. This stability was achieved with minimal reliance on local optimization, as the local search procedure was applied in only 0.5% of the cases. Such performance indicates that the algorithm’s global search capabilities are sufficient to consistently identify optimal or near-optimal solutions without heavy dependence on local refinement methods.

Table 3. Comparison of function calls of the SIOA method with others.

Function	SIOA		GA		DE		PSO		ACO	
	Calls	Time	Calls	Time	Calls	Time	Calls	Time	Calls	Time
ACKLEY	3028	0.072	3441	0.089	10694	0.157	5684 (0.86)	0.102	3449	0.103
BF1	1204	0.028	2346	0.056	4963	0.065	2562	0.034	1558 (0.4)	0.033
BF2	1177	0.028	2116	0.051	5139	0.070	2332	0.032	1523 (0.96)	0.040
BF3	1144	0.029	2163	0.053	4730	0.063	2093	0.028	1410	0.044
BRANIN	950	0.027	1668	0.045	2022	0.034	1686	0.025	1054	0.031
CAMEL	1154	0.029	1835	0.048	3161	0.051	2029	0.028	1227	0.029
DIFFERENT POWERS10	2123	0.086	2507	0.100	3897	0.108	2608	0.079	2003	0.088
DIFFPOWER2	1590	0.033	1886	0.053	3239	0.049	2694	0.035	1740	0.036
DIFFPOWER5	3471	0.143	3770	0.134	5620	0.169	4472	0.133	3789	0.140
DIFFPOWER10	4407	0.461	3909	0.352	6546	0.616	5091	0.479	4582	0.464
DISCUS10	931	0.045	1640	0.066	2433	0.049	1658	0.037	1010	0.037
EASOM	776	0.025	1618	0.043	1784	0.032	1576	0.026	977	0.030
ELP10	1126	0.059	1771	0.079	2613	0.070	1867	0.051	1224	0.490
EQUAL MAXIMA10	2649	0.139	2212	0.110	4341	0.174	3401	0.142	2384	0.135
EXP10	1096	0.042	1764	0.064	2625	0.050	1795	0.038	1175	0.033
GKLS250	1202	0.037	1862	0.057	3427	0.065	1996	0.037	1245	0.038
GKLS350	1207	0.043	2038 (0.86)	0.067	3637	0.074	2361	0.047	1550 (0.86)	0.049
GOLDSTEIN	1161	0.030	1925	0.052	2621	0.039	1955	0.026	1249	0.030
GRIEWANK ROSENBROCK10	1684	0.151	2136	0.126	3743	0.212	2437	0.143	1843	0.155
GRIEWANK2	1061	0.028	2956 (0.26)	0.061	4765 (0.46)	0.070	1589 (0.23)	0.052	839	0.066
GRIEWANK10	1899 (0.6)	0.117	2936 (0.2)	0.101	4582 (0.5)	0.124	2209 (0.36)	0.118	2444 (0.33)	0.140

Table 3. Cont.

Function	SIOA		GA		DE		PSO		ACO	
	Calls	Time	Calls	Time	Calls	Time	Calls	Time	Calls	Time
HANSEN	1486	0.041	2143 (0.86)	0.068	3078	0.062	2964	0.061	1424 (0.86)	0.000
HARTMAN3	1067	0.030	1744	0.050	2376	0.040	1760	0.028	1099	0.033
HARTMAN6	1129	0.040	1733 (0.73)	0.055	2558	0.050	1917 (0.7)	0.037	1222 (0.93)	0.037
POTENTIAL3	1156	0.051	1754	0.068	2694	0.064	1875	0.047	1270	0.055
POTENTIAL5	1639	0.108	2106	0.116	3320	0.140	2424	0.114	1749	0.133
POTENTIAL10	3104 (0.6)	0.637	3566 (0.43)	0.558	5583 (0.66)	0.818	4581 (0.5)	0.763	3182 (0.43)	0.801
RASTRIGIN2	933	0.027	2411 (0.93)	0.057	4412	0.060	3017 (0.96)	0.040	1661	0.039
ROSENBROCK4	1422	0.035	1783	0.057	2860	0.046	2069	0.032	1496	0.039
ROSENBROCK8	1558	0.052	2072	0.068	3962	0.076	2501	0.050	1751	0.054
ROSENBROCK16	1833	0.113	2506	0.114	4157	0.145	2781	0.104	2151	0.100
ROTATED ROSENBROCK10	1785	0.081	2237	0.089	3663	0.095	2675	0.074	1918 (0.96)	0.067
SHEKEL5	1220	0.036	1770 (0.66)	0.052	2884	0.050	1990 (0.76)	0.034	1298 (0.76)	0.038
SHEKEL7	1286	0.039	1812 (0.83)	0.054	2890 (0.96)	0.052	2080 (0.83)	0.037	1351 (0.83)	0.040
SHEKEL10	1345 (0.9)	0.043	1867 (0.66)	0.058	3625	0.067	2091 (0.83)	0.041	1335	0.040
SINUSOIDAL4	1358	0.042	1938	0.061	3263	0.064	2213	0.044	1278	0.040
SINUSOIDAL8	1541 (0.96)	0.075	1957	0.080	3241	0.096	2014	0.066	1459	0.062
SINUSOIDAL6	1814 (0.53)	0.239	2319 (0.76)	0.157	4209 (0.7)	0.283	2680	0.209	1979 (0.86)	0.199
STEP ELLIPSOIDAL4	994	0.033	1714 (0.96)	0.052	2102	0.042	1960	0.036	1259	0.038
TEST2N4	1502 (0.73)	0.040	2270 (0.96)	0.070	3619	0.067	2153	0.038	1437 (0.9)	0.042
TEST2N5	1338 (0.5)	0.044	2185 (0.66)	0.072	4556	0.083	2376 (0.86)	0.046	1601 (0.63)	0.042
TEST30N3	1142	0.036	1730	0.055	2381	0.043	1998	0.033	1116	0.029
TEST30N4	1261	0.041	1825	0.062	2408	0.046	2270	0.040	1167	0.059
SUM calls/time	66,953	3.535	93,941	3.880	160,423	4.830	106,484	3.666	72,478	4.198
AVG calls/time	3043.32	3.535	4270.05	3.880	7291.95	4.830	4840.18	3.666	3294.45	4.198
Avg time	0.080		0.088		0.110		0.083		0.095	
AVG Success rate	94.930		90.140		96.000		92.767		92.023	

Figures 1–3 provide a complementary visualization of the numerical data presented in Table 3. These figures highlight both the distribution of function evaluations and the relative execution times across methods, offering a clearer view of convergence dynamics and performance consistency. In this way, the graphical results reinforce the tabular evidence, allowing for a more intuitive comparison of SIOA against GA, DE, PSO, and ACO.

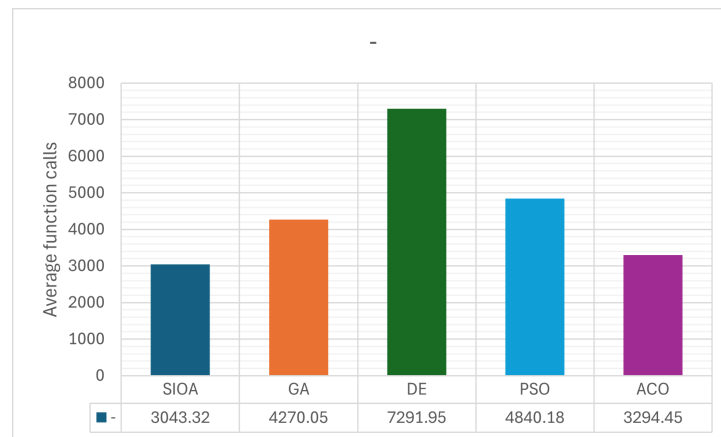


Figure 1. Performance of SIOA and reference methods on benchmark problems: distribution of function evaluations across runs.

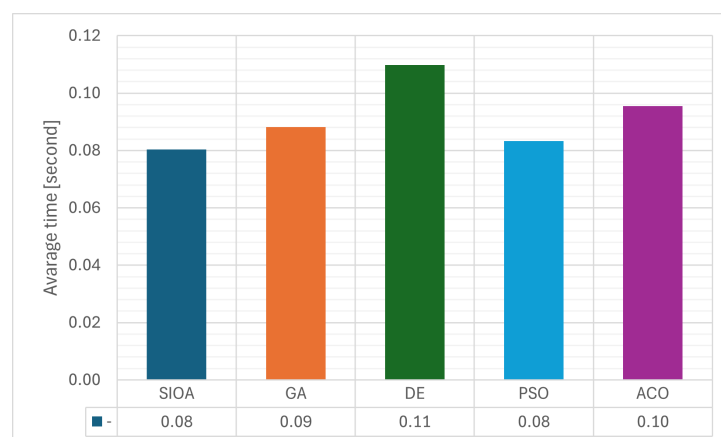


Figure 2. Comparative analysis of SIOA versus GA, DE, PSO, and ACO: average execution times for all test functions.

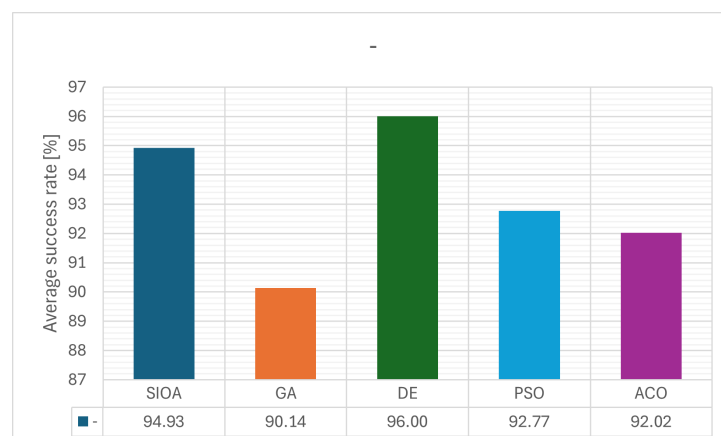


Figure 3. Overall convergence dynamics of SIOA and competing algorithms: stability and robustness across problems.

The comparative analysis of the results of Table 3 shows that the proposed SIOA method outperforms traditional GA, DE, PSO, and ACO methods across a wide range of benchmark functions, both in terms of the number of objective function evaluations and the success rate. In the vast majority of cases, SIOA achieves the minimum or one of the lowest evaluation counts, indicating high computational efficiency and faster convergence. The differences are particularly evident in multidimensional and multimodal problems,

where traditional methods such as GA and DE require significantly more evaluations, often more than double or triple those of SIOA.

The success rate, which is 100% when not shown in parentheses, also presents a positive picture for SIOA. Its overall value reaches 94.9%, surpassing the corresponding rates of GA (90.1%) and PSO (92.8%) and coming very close to the best performances of DE (96%) and ACO (92%), but with considerably lower computational cost. In several challenging cases, such as the GRIEWANK and POTENTIAL functions, SIOA combines low evaluation requirements with competitive or even maximum success rates, demonstrating an ability to maintain a balance between exploration and exploitation.

The overall picture, as reflected in the last row of the table, confirms SIOA's general superiority, as it achieves the lowest total number of evaluations (66,953) compared to other methods, which range from about 72,478 (ACO) to 160,423 (DE). This high efficiency, combined with the stability of the results, suggests that the biologically inspired strategy of sporulation and germination, together with mechanisms for self-adaptation and diversity preservation, offers a clear advantage over classic evolutionary and swarm-based methods across a wide spectrum of optimization problems.

To enhance the numerical accuracy of the final solutions without compromising the validity of cross-method comparisons, we enabled the same lightweight, local search routine for all algorithms reported in Table 3 (SIOA, GA, DE, PSO, ACO). Concretely, at the end of each iteration, each candidate solution had an independent probability $P_{loc} = 0.005$ (0.5%) to invoke the local search procedure (see Algorithm 1, lines 48–60 and parameters summarized in Table 1). This very small activation rate yields occasional refinements near convergence while keeping the computational overhead and any potential bias negligible, importantly, no method received bespoke local search settings or a larger budget. In practice, the routine was triggered only rarely, thus improving accuracy without altering the comparative performance trends observed in Table 3.

The analysis of the results (Friedman test [28]) presented in Figure 4 shows the performance comparison of the proposed SIOA optimization method against other established techniques. The values of the critical parameter p , which indicate the levels of statistical significance, reveal that SIOA demonstrates a very extremely significant superiority over GA, DE, and PSO, with p -values lower than 0.0001. In contrast, the comparison between SIOA and ACO did not show a statistically significant difference, as the p -value is greater than 0.05, indicating that the two methods exhibit a similar level of performance according to this statistical evaluation.

Figure 5 visualizes, on a per-problem basis, the comparative performance of all algorithms as derived from the function-evaluation counts and success rates reported in Table 3. Across most benchmarks, SIOA attains the lowest or among the lowest number of evaluations, with particularly clear margins on multimodal or higher-dimensional families (e.g., GRIEWANK*, POTENTIAL*), while ACO is occasionally comparable, these patterns are consistent with the tabulated trends. The aggregate line of Table 3 is reflected here as well, confirming the overall evaluation load: SIOA (66,953) vs. GA (93,941), PSO (106,484), DE (160,423), and ACO (72,478), underscoring SIOA's efficiency advantage without altering the success-rate profile.

Comparison of Optimization Methods

Friedman Test with Post-hoc Analysis

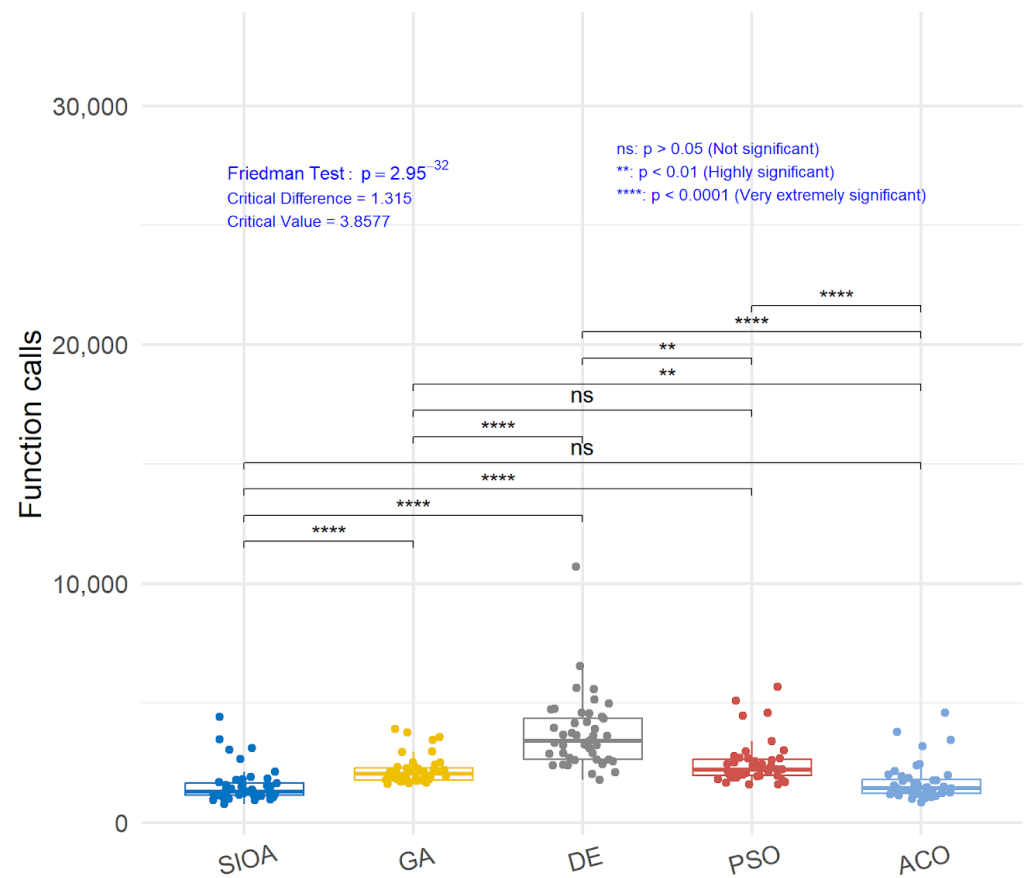


Figure 4. Statistical comparison of SIOA against other methods.



Figure 5. Performance of all methods on each problem.

3.2. Experiments with Advanced Methods and Real-World Problems

Subsequently, SIOA was tested against more sophisticated algorithms on complex, large-scale problems derived from realistic application domains, aiming to evaluate its performance under increased complexity, constraint handling, and uncertainty. These problems are presented in Table 4.

Table 4. Real world problems CEC2011.

Problem	Formula	Dim	Bounds
Parameter Estimation for Frequency-Modulated Sound Waves	$\min_{x \in [-6.4, 6.35]^6} f(x) = \frac{1}{N} \sum_{n=1}^N y(n; x) - y_{\text{target}}(n) ^2$ $y(n; x) = x_0 \sin(x_1 n + x_2 \sin(x_3 n + x_4 \sin(x_5 n)))$	6	$x_i \in [-6.4, 6.35]$
Lennard-Jones Potential	$\min_{x \in \mathbb{R}^{3N-6}} f(x) = 4 \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left[\left(\frac{1}{r_{ij}} \right)^{12} - \left(\frac{1}{r_{ij}} \right)^6 \right]$	30	$x_0 \in (0, 0, 0)$ $x_1, x_2 \in [0, 4]$ $x_3 \in [0, \pi]$ x_{3k-3} x_{3k-2} $x_i \in [-b_k, +b_k]$
Bifunctional Catalyst Blend Optimal Control	$\frac{dx_1}{dt} = -k_1 x_1, \frac{dx_2}{dt} = k_1 x_1 - k_2 x_2 + k_3 x_2 + k_4 x_3,$ $\frac{dx_3}{dt} = k_2 x_2, \frac{dx_4}{dt} = -k_4 x_4 + k_5 x_5,$ $\frac{dx_5}{dt} = -k_3 x_2 + k_6 x_4 - k_5 x_5 + k_7 x_6 + k_8 x_7 + k_9 x_5 + k_{10} x_7$ $\frac{dx_6}{dt} = k_8 x_5 - k_7 x_6, \frac{dx_7}{dt} = k_9 x_5 - k_{10} x_7$ $k_i(u) = c_{i1} + c_{i2}u + c_{i3}u^2 + c_{i4}u^3$	1	$u \in [0.6, 0.9]$
Optimal Control of a Non-Linear Stirred Tank Reactor	$J(u) = \int_0^{0.72} [x_1(t)^2 + x_2(t)^2 + 0.1u^2] dt$ $\frac{dx_1}{dt} = -2x_1 + x_2 + 1.25u + 0.5 \exp\left(\frac{x_1}{x_1+2}\right)$ $\frac{dx_2}{dt} = -x_2 + 0.5 \exp\left(\frac{x_1}{x_1+2}\right)$ $x_1(0) = 0.9, \quad x_2(0) = 0.09, \quad t \in [0, 0.72]$	1	$u \in [0, 5]$
Tersoff Potential for model Si (B)	$\min_{x \in \Omega} f(x) = \sum_{i=1}^N E(x_i)$ $E(x_i) = \frac{1}{2} \sum_{j \neq i} f_c(r_{ij}) [V_R(r_{ij}) - B_{ij} V_A(r_{ij})]$ where $r_{ij} = \ x_i - x_j\$, $V_R(r) = A \exp(-\lambda_1 r)$ $V_A(r) = B \exp(-\lambda_2 r)$ $f_c(r)$: cutoff function with $f_c(r)$: angle parameter	30	$x_1 \in [0, 4]$ $x_2 \in [0, 4]$ $x_3 \in [0, \pi]$ $x_i \in \left[\frac{4(i-3)}{4}, 4\right]$
Tersoff Potential for model Si (C)	$\min_x V(x) = \sum_{i=1}^N \sum_{j>i}^N f_C(r_{ij}) [a_{ij} f_R(r_{ij}) + b_{ij} f_A(r_{ij})]$ $f_C(r) = \begin{cases} 1, & r < R - D \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{\pi(r-R+D)}{2D}\right), & r - R \leq D \\ 0, & r > R + D \end{cases}$ $f_R(r) = A \exp(-\lambda_1 r)$ $f_A(r) = -B \exp(-\lambda_2 r)$ $b_{ij} = \left[1 + (\beta^n) \zeta_{ij}^n\right]^{-1/(2n)}$ $\sum_{k \neq i, j} f_C(r_{ik}) g(\theta_{ijk}) \exp[\lambda_3^3 (r_{ij} - r_{ik})^3]$	30	$x_1 \in [0, 4]$ $x_2 \in [0, 4]$ $x_3 \in [0, \pi]$ $x_i \in \left[\frac{4(i-3)}{4}, 4\right]$
Spread Spectrum Radar Polly phase Code Design	$\min_{x \in X} f(x) = \max\{ \varphi_1(x) , \varphi_2(x) , \dots, \varphi_m(x) \}$ $X = \{x \in \mathbb{R}^n \mid 0 \leq x_j \leq 2\pi, j = 1, \dots, n\} \quad m = 2n - 1$ $\varphi_j(x) = \begin{cases} \sum_{k=1}^{n-j} \cos(x_k - x_{k+j}) & \text{for } j = 1, \dots, n-1 \\ n & \text{for } j = n \\ \varphi_{2n-j}(x) & \text{for } j = n+1, \dots, 2n-1 \end{cases}$ $\varphi_j(x) = \sum_{k=1}^{n-j} \cos(x_k - x_{k+j}), \quad j = 1, \dots, n-1$ $\varphi_n(x) = n, \varphi_{n+\ell}(x) = \varphi_{n-\ell}(x), \quad \ell = 1, \dots, n-1$	20	$x_j \in [0, 2\pi]$
Transmission Network Expansion Planning	$\min \sum_{l \in \Omega} c_l n_l + W_1 \sum_{l \in \Omega_L} f_l - \tilde{f}_l + W_2 \sum_{l \in \Omega} \max(0, n_l - \bar{n}_l)$ $Sf = g - d$ $f_l = \gamma_l n_l \Delta \theta_l, \quad \forall l \in \Omega$ $ f_l \leq \bar{f}_l n_l, \quad \forall l \in \Omega$ $0 \leq n_l \leq \bar{n}_l, \quad n_l \in \mathbb{Z}, \quad \forall l \in \Omega$	7	$0 \leq n_i \leq \bar{n}_i$ $n_i \in \mathbb{Z}$

Table 4. Cont.

Problem	Formula	Dim	Bounds
Electricity Transmission Pricing	$\min_x f(x) = \sum_{i=1}^{N_g} \left(\frac{C_i^{gen}}{P_i^{gen}} - R_i^{gen} \right)^2 + \sum_{j=1}^{N_d} \left(\frac{C_j^{load}}{P_j^{load}} - R_j^{load} \right)^2$ $\sum_j GD_{ij} + \sum_j BT_{ij} = P_i^{gen}, \quad \forall i$ $\sum_i GD_{ij} + \sum_i BT_{ij} = P_j^{load}, \quad \forall j$ $GD_{ij}^{max} = \min(P_i^{gen} - BT_{ij}, P_j^{load} - BT_{ij})$	126	$GD_{ij} \in [0, GD_{ij}^{max}]$
Circular Antenna Array Design	$\min_{r_1, \dots, r_6, \varphi_1, \dots, \varphi_6} f(\mathbf{x}) = \max_{\theta \in \Omega} AF(\mathbf{x}, \theta)$ $AF(\mathbf{x}, \theta) = \left \sum_{k=1}^6 \exp(j[2\pi r_k \cos(\theta - \theta_k) + \varphi_k \frac{\pi}{180}]) \right $	12	$r_k \in [0.2, 1]$ $\varphi_k \in [-180, 180]$
Dynamic Economic Dispatch 1	$\min_{\mathbf{P}} f(\mathbf{P}) = \sum_{i=1}^{24} \sum_{t=1}^5 (a_i P_{i,t}^2 + b_i P_{i,t} + c_i)$ $P_i^{min} \leq P_{i,t} \leq P_i^{max}, \quad \forall i = 1, \dots, 5, t = 1, \dots, 24$ $\sum_{i=1}^5 P_{i,t} = D_t, \quad \forall t = 1, \dots, 24$ $P_{min} = [10, 20, 30, 40, 50]$ $P_{max} = [75, 125, 175, 250, 300]$	120	$P_i^{min} \leq P_{i,t} \leq P_i^{max}$
Dynamic Economic Dispatch 2	$\min_{\mathbf{P}} f(\mathbf{P}) = \sum_{i=1}^{24} \sum_{t=1}^9 (a_i P_{i,t}^2 + b_i P_{i,t} + c_i)$ $P_i^{min} \leq P_{i,t} \leq P_i^{max}, \quad \forall i = 1, \dots, 5, t = 1, \dots, 24$ $\sum_{i=1}^5 P_{i,t} = D_t, \quad \forall t = 1, \dots, 24$ $P_{min} = [150, 135, 73, 60, 73, 57, 20, 47, 20]$ $P_{max} = [470, 460, 340, 300, 243, 160, 130, 120, 80]$	216	$P_i^{min} \leq P_{i,t} \leq P_i^{max}$
Static Economic Load Dispatch (1,2,3,4,5)	$\min_{P_1, \dots, P_{N_G}} F = \sum_{i=1}^{N_G} f_i(P_i)$ $f_i(P_i) = a_i P_i^2 + b_i P_i + c_i, \quad i = 1, 2, \dots, N_G$ $f_i(P_i) = a_i P_i^2 + b_i P_i + c_i + e_i \sin(f_i(P_i^{min} - P_i)) $ $P_i^{min} \leq P_i \leq P_i^{max}, \quad i = 1, 2, \dots, N_G$ $\sum_{i=1}^{N_G} P_i = P_D + P_L$ $P_L = \sum_{i=1}^{N_G} \sum_{j=1}^{N_G} P_i B_{ij} P_j + \sum_{i=1}^{N_G} B_{0i} P_i + B_{00}$ $P_i - P_i^0 \leq UR_i \quad P_i^0 - P_i \leq DR_i$	6 13 15 40 140	See Technical Report of CEC2011

The results shown in Table 5 were obtained using the parameter settings defined in Table 1. Also, a detailed ranking for the algorithms is presented in Table 6. The termination criterion was set to 150,000 function evaluations, ensuring a uniform computational budget across all test cases. No local optimization procedures were applied during the runs, meaning that the reported outcomes reflect solely the global search capabilities of the algorithm without any refinement from local search techniques. This setup allows for an unbiased assessment of the method's performance under purely global exploration conditions.

The comparative analysis of the optimization methods, based on both best and mean performance after 150,000 function evaluations, reveals clear distinctions in their overall effectiveness. CMA-ES achieved the highest ranking, excelling in both peak and consistent performance, followed by EO and CLPSO, which demonstrated strong competitiveness. SIOA ranked closely behind these top methods, showing notable strengths in complex, high-dimensional, and multimodal problems, where its adaptive sporulation and germination mechanisms effectively balanced exploration and exploitation. In certain cases, such as the Tersoff Potential and Static Economic Load Dispatch problems, SIOA's results approached those of CMA-ES, highlighting its capacity to rival advanced evolutionary strategies. However, its slightly higher variance in some problem instances, particularly in less multimodal landscapes, reduced its mean performance score, preventing it from achieving the top overall rank. Despite this, SIOA emerges as a modern and competitive algorithm with strong potential for further improvement, especially through integration with specialized local search schemes aimed at enhancing stability and precision.

Table 5. Algorithm comparison based on best and mean after 1.5×10^5 FEs.

150,000 Fes	CLPSO		SaDE		jDE		CMA-ES		SIOA	
Problem	Best	Mean/st	Best	Mean	Best	Mean	Best	Mean	Best	Mean
Parameter Estimation for Frequency-Modulated Sound Waves	0.1314	0.2124 ± 0.0302	0.1899	0.2025 ± 0.0092	0.1161	0.1460 ± 0.0350	0.1816	0.2568 ± 0.0447	0.2061	0.2599 ± 0.0230
Lennard-Jones Potential	−13.4364	−10.2507 ± 1.0290	−24.8687	−22.6693 ± 1.1272	−29.9812	−27.4925 ± 1.2350	−28.4225	−25.7878 ± 2.2711	−28.5113	−24.1461 ± 2.4893
Bifunctional Catalyst Blend Optimal Control	−0.0002	−0.0002 $\pm 1.1577 \times 10^{-16}$	−0.0002	−0.0002 $\pm 5.5136 \times 10^{-20}$	−0.0002	−0.0002 $\pm 5.5136 \times 10^{-20}$	−0.0002	−0.0002 $\pm 5.5136 \times 10^{-20}$	−0.0002	−0.0002 $\pm 9.1776 \times 10^{-11}$
Optimal Control of a Non- Linear Stirred Tank Reactor	0.3903	0.3903767228 ± 0	0.3903	0.3903 ± 0	0.3903	0.3903 ± 0	0.3903	0.3903 ± 0	0.3903	0.3903 ± 0
Tersoff Potential for model Si (B)	−28.2354	−26.1883 ± 1.0565	−3.1077	25.4711 ± 16.7202	−13.5115	−3.9836 ± 6.6660	−29.2624	−27.5889 ± 1.0406	−28.6359	−27.1151 ± 1.0847
Tersoff Potential for model Si (C)	−30.8520	−28.8734 ± 0.9880	−11.6071	22.0896 ± 18.5809	−18.7621	−8.5060 ± 5.5431	−33.1969	−31.7927 ± 0.8281	−33.5041	−31.0138 ± 1.4206
Spread Spectrum Radar Polly phase Code Design	1.0853	1.34395 ± 0.1487	1.5365	2.1508 ± 0.1986	1.5258	1.8120 ± 0.1712	0.0148	0.1719 ± 0.1378	0.6071	1.0234 ± 0.2286
Transmission Network Expansion Planning	250	250 ± 0	250	250 ± 0	250	250 ± 0	250	250 ± 0	250	250 ± 0
Electricity Transmission Pricing	13,775	13,775,395 ± 222.9723	23,481	30,034,934 $\pm 3,264,767$	13,774,627	14,020,953 $\pm 276,142$	13,775,841	13,787,550 ± 6136	13,774,551	13,775,341 ± 372
Circular Antenna Array Design	0.0069	0.0518 ± 0.0706	0.0214	0.0389 ± 0.0081	0.0068	0.01765 ± 0.0223	0.0072	0.0086 ± 0.0009	0.0074	0.0249 ± 0.0443
Dynamic Economic Dispatch 1	428,607,927	435,250,914 $\pm 2,973,190$	968,042,312	1,034,679,775 $\pm 25,667,290$	968,042,312	1,034,393,036 $\pm 25,445,935$	88,285	102,776 ± 6688	921,434,356	984,699,299 $\pm 23,606,727$
Dynamic Economic Dispatch 2	33,031,590	53,906,147 $\pm 8,492,239$	845,287,898	913,715,793 $\pm 3,0667,287$	340,091,475	397,471,715 $\pm 37,259,947$	502,699	477,720 $\pm 193,951$	768,167,675	768,167,675 $\pm 768,167,675$
StaticEconomic Load Dispatch 1	6554	7668 ± 1245	16,877	101,588 $\pm 81,105$	6163	6778 ± 3004	6657	415,917 $\pm 688,544$	6538	877,097 $\pm 847,631$
Static Economic Load Dispatch 2	19,030	20,699 ± 2922	2,600,565	9,329,466 $\pm 4,019,053$	1,161,578	3,671,587 $\pm 1,542,286$	763,001	1,425,815 $\pm 377,126$	24,026	1,478,534 $\pm 1,063,608$
Static Economic Load Dispatch 3	470,192,288	470,294,703 $\pm 57,822$	478,069,615	541,898,763 $\pm 20,128,777$	471,058,115	471,963,142 $\pm 529,633$	470,023,232	470,023,232 $\pm 1.8487 \times 10^{-07}$	470,825,156	472,256,736 $\pm 608,886$
Static Economic Load Dispatch 4	884,980	1,423,887 $\pm 285,794$	14,170,362	106,749,078 $\pm 73,147,979$	6,482,592	17,527,314 $\pm 53,06,489$	476,053	2,925,852 $\pm 12,68,161$	70,686	580,122 $\pm 340,707$
Static Economic Load Dispatch 5	8,105,947,615	8,110,924,071 $\pm 4,422,895$	1.3127×10^{10}	13,543,754,650 $\pm 213,865,059$	8,453,090	8,459,337,082 $\pm 2874,979$	8,072,077	8,084,017,791 $\pm 4,623,617$	8,002,077,963	8,048,300 $\pm 4,365,204$

Table 6. Detailed Ranking of Algorithms Based on Best and Mean after 1.5×10^5 FEs.

Problem	CLPSO Best	CLPSO Mean	SaDE Best	SaDE Mean	jDE Best	jDE Mean	CMA- ES Best	CMA-ES Mean	SIOA Best	SIOA Mean
Parameter Estimation for Frequency-Modulated Sound Waves	2	3	4	2	1	1	3	4	5	5
Lennard-Jones Potential	5	5	4	4	1	1	3	2	2	3
BifunctionalCatalyst Blend Optimal Control	1	1	1	1	1	1	1	1	1	1
Optimal Control of a Non-Linear Stirred Tank Reactor	1	1	1	1	1	1	1	1	1	1
Tersoff Potential for model Si (B)	3	3	5	5	4	4	1	1	2	2
Tersoff Potential for model Si (C)	3	3	5	5	4	4	2	1	1	2
Spread Spectrum Radar Polly phaseCode Design	3	3	5	5	4	4	1	1	2	2
Transmission Network Expansion Planning	1	1	1	1	1	1	1	1	1	1
Electricity Transmission Pricing	3	1	5	5	2	4	4	3	1	2
Circular Antenna Array Design	2	5	5	4	1	2	3	1	4	3
Dynamic Economic Dispatch 1	2	2	5	5	4	4	1	1	3	3
Dynamic Economic Dispatch 2	2	2	5	5	3	3	1	1	4	4
Static Economic Load Dispatch 1	3	2	5	5	1	1	4	3	2	4
Static Economic Load Dispatch 2	1	1	5	5	4	4	3	2	2	3
Static Economic Load Dispatch 3	2	2	5	5	4	3	1	1	3	4
Static Economic Load Dispatch 4	3	2	5	5	4	4	2	3	1	1
Static Economic Load Dispatch 5	3	3	5	5	4	4	2	2	1	1
TOTAL	40	40	71	68	44	46	34	29	36	42

Also, a comparison of all algorithms and a final ranking is presented in Table 7.

Table 7. Comparison of Algorithms and Final Ranking.

Method	Best	Mean	Overall	Average	Rang
CMA-ES	34	29	63	1.75	1
SIOA	36	42	78	2.16	2
CLPSO	40	40	80	2.22	3
jDE	44	46	90	2.5	4
SaDE	71	68	139	3.86	5

3.3. Exploration and Exploitation

In this study, the trade-off between exploration and exploitation is assessed using a specific set of quantitative indicators: Initial Population Diversity (*IPD*), Final Population Diversity (*FPD*), Average Exploration Ratio (*AER*), Median Exploration Ratio (*MER*), and Average Balance Index (*ABI*). These metrics, although fundamentally grounded in population

diversity measurements, are designed to capture both the temporal evolution of exploration by monitoring diversity changes over the course of the optimization and the degree of exploitation through the level of convergence in the final population. While these indicators provide a structured way to examine algorithmic behavior, further investigation employing more direct analysis tools, such as attraction basin mapping or tracking the clustering of solutions around local or global optima, could yield deeper insights into the search dynamics. Such approaches are considered a promising avenue for extending the current work.

The metrics reported in Table 8 quantify and track the interplay between exploration and exploitation throughout the execution of the SIOA algorithm. Their computation relies on diversity measurements at different stages of the optimization process and on how these values evolve over iterations.

Table 8. Balance between exploration and exploitation of the SIOA method in each benchmark function after 1.5×10^5 FEs.

Problem	Best	Mean	SD	IPD	FDP	AER	MER	ABI
Parameter Estimation for Frequency-Modulated Sound Waves	0.20618586	0.259930863	0.023021357	8.5901	4.16015	4.16015	0	0.49979
Lennard-Jones Potential	−28.51132554	−24.14612379	2.489334694	13.91823	4.466	0.00021	0	0.49967
Bifunctional Catalyst Blend Optimal Control	−0.000286591	−0.000286591	$9.177681044 \times 10^{-11}$	0.0743	0.08161	0.00007	0	0.50005
Optimal Control of a Non—Linear Stirred Tank Reactor	0.390376723	0.390376723	0	49,184,124.11	0.00019	17,134,746.47	0	0.49745
Tersoff Potential for model Si (B)	−28.63594613	−27.11517851	1.084722973	5.52126	1.71041	0.0002	0	0.49968
Tersoff Potential for model Si (C)	−33.50417851	−31.0138182	1.420690601	5.52126	2.74916	0.00017	0	0.49971
Spread Spectrum Radar Polly phase Code Design	0.607180067	1.023498006	0.228610721	8.06994	5.64058	0.00008	0	0.49988
Transmission Network Expansion Planning	250.00	250.00	0	0.96619	0.92498	0.00001	0	0.5
Electricity Transmission Pricing	13,774,551.1	13,775,341.62	372.2433548	6.50993	0.06077	0.00845	0.00078	0.49851
Circular Antenna Array Design	0.007425975	0.024989563	0.044360116	245.62332	26.21386	0.00052	0	0.49938
Dynamic Economic Dispatch 1	921,434,356.7	984,699,299.8	23,606,727.92	530.86265	0.06496	0.54586	0.0025	0.49827
Dynamic Economic Dispatch 2	768,167,675.2	768,167,675.2	768,167,675.2	890.76948	0.08559	0.68792	0.00216	0.49823
Static Economic Load Dispatch 1	6538.455462	877,097.0217	847631.4535	141.01729	149.57653	0.00001	0	0.50002
Static Economic Load Dispatch 2	24,026.88184	1,478,534.024	1,063,608.234	238.24613	207.89492	0.00008	0	0.49982
Static Economic Load Dispatch 3	470,825,156.9	472,256,736.5	608,886.262	218.59546	25.75625	0.00082	0	0.49941
Static Economic Load Dispatch 4	70,686.26733	580,122.834	340,707.3484	410.06721	3.95079	0.01195	0	0.49942
Static Economic Load Dispatch 5	$1.241487588 \times 10^{10}$	$1.284582323 \times 10^{10}$	197,599,745.4	750.05361	0.06971	0.71341	0.00243	0.49825

The *IPD* quantifies the diversity present at the very start of the optimization and is obtained by computing the mean Euclidean distance between all pairs of individuals in the initial population:

$$IPD = \frac{2}{NP(NP-1)} \sum_{i=1}^{NP-1} \sum_{j=i+1}^{NP} d(x_i, x_j) \quad (4)$$

where $d(x_i, x_j)$ is the Euclidean distance between solutions x_i and x_j , and NP denotes the population size.

The *FPD* is computed using the same formulation but applied to the final set of solutions after the algorithm is completed.

The *AER* reflects the average level of exploration across all iterations and is defined as:

$$AER = \frac{1}{G} \sum_{g=1}^{iter_{max}} \frac{IPD_g}{IPD_1} \quad (5)$$

where $iter_{max}$ is the total number of iterations, IPD_g represents the diversity at iteration g , and IPD_1 is the initial diversity value.

The *MER* is the median value of the exploration ratios recorded over all generations:

$$MER = \text{median} \left(\frac{IPD_g}{IPD_1} \right), \quad \text{for } g = 1, \dots, iter_{max} \quad (6)$$

The *ABI* serves as a composite measure of the exploration–exploitation balance. It is typically calculated as a weighted function of *AER* and *FPD* (or other exploitation-related indicators):

$$ABI = \frac{AER}{AER + \epsilon} \cdot \left(1 - \frac{FPD}{IPD} \right) \quad (7)$$

where ϵ is a small constant introduced to avoid division by zero. An *ABI* value close to 0.5 generally indicates a well-balanced interplay between exploration and exploitation.

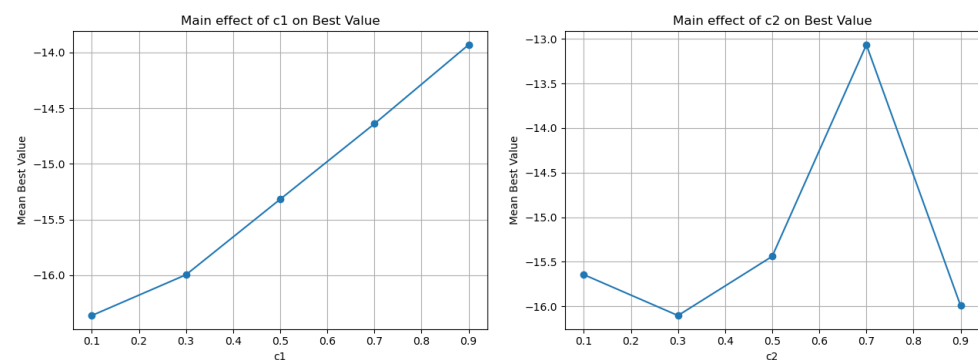
3.4. Parameters Sensitivity

By adopting the parameter sensitivity examination framework proposed by Lee et al. [29], this study provides a solid foundation for understanding how optimization algorithms react to changes in their configuration and sustain their reliability across varying conditions.

In the Potential problem (Table 9 and Figure 6), the mean best value improves as c_1 decreases: the mean best moves from -13.93 ($c_1 = 0.9$) toward -16.36 ($c_1 = 0.1$), with a main effect range of 2.43. This indicates that for this high-dimensional, strongly multimodal potential, excessive stochastic dispersion (high c_1) “blurs” exploitation of promising areas, whereas mild dispersion supports steady improvement. The impact of c_2 is stronger (range 3.04) and non-monotonic: moderate values around 0.3 yield the best mean performance (-16.10), while very low or very high values degrade results. Therefore, in Potential, a clear preference emerges for a “moderate” pull toward the best solution ($c_2 \approx 0.3$) combined with a low stochastic perturbation (small c_1).

Table 9. Sensitivity analysis of the method parameters for the potential problem (Dimension 10).

Potential 10	Value	Mean	Min	Max	Iters	Main Range
c1	0.1	−16.36153	−23.37956	−11.01324	150	2.42989
	0.3	−15.99563	−20.23252	−10.85775	150	
	0.5	−15.31793	−21.19271	−10.81508	150	
	0.7	−14.64094	−20.05168	−10.78772	150	
	0.9	−13.93164	−19.82466	−10.51742	150	
c2	0.1	−15.64469	−18.92493	−12.61004	150	3.03774
	0.3	−16.10363	−23.37956	−10.99769	150	
	0.5	−15.43998	−20.58774	−11.0502	150	
	0.7	−14.64094	−16.71041	−10.51742	150	
	0.9	−13.93164	−20.23252	−11.37413	150	

**Figure 6.** Graphical representation of c_1 and c_2 for the potential problem.

In the Rastrigin problem (Table 10 and Figure 7), the behavior differs: c_1 has a relatively small main effect (0.88), and the best mean value occurs around $c_1 = 0.7$ (Mean Best ≈ 1.25), with similar performance at $c_1 = 0.3$. In contrast, c_2 is more decisive (range 2.44), with the optimal zone around 0.5 (Mean Best ≈ 0.82). The Rastrigin function, with its pronounced symmetric multimodality, benefits from a stronger attraction mechanism toward the best (moderate c_2), which helps “lock in” low-value basins, while a moderate c_1 maintains enough exploration without destabilizing convergence. It is notable that the minima are often 0.00, indicating that all combinations can reach the global minimum, but mean values differentiate reliability and stability.

Table 10. Sensitivity analysis of the method parameters for the Rastrigin (Dimension 4).

Rastrigin 4	Value	Mean	Min	Max	Iters	Main Range
c1	0.1	2.13634	0	11.10018	150	0.88378
	0.3	1.32079	0	8.30785	150	
	0.5	1.52523	0	8.1495	150	
	0.7	1.25256	0	7.10786	150	
	0.9	1.74395	0	6.95643	150	
c2	0.1	3.25941	0	11.10018	150	2.44349
	0.3	1.51291	0	6.55699	150	
	0.5	0.81592	0	5.19549	150	
	0.7	1.34782	0	5.19957	150	
	0.9	1.04281	0	6.60519	150	

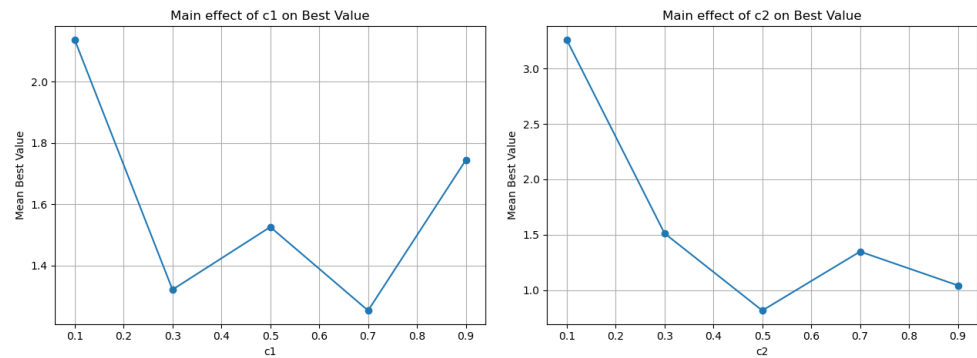


Figure 7. Graphical representation of c_1 and c_2 for the Rastrigin problem.

In the Test2n problem (Table 11 and Figure 8), the picture is even clearer in favor of low c_2 : the main effect of c_2 is very high (8.94), and the best mean performance appears at $c_2 = 0.1$ (Mean Best ≈ -152.41). Increasing c_2 toward 0.7–0.9 significantly worsens mean performance, although the minima remain near -156.664 for all settings. This shows that excessive attraction toward the best induces premature convergence into local basins and increases performance variability. c_1 has a moderate impact (2.82), with a trend suggesting that larger values (e.g., 0.9) may slightly improve mean performance, likely by helping to escape narrow polynomial valleys. Overall, in Test2n4, the guidance is clear: keep c_2 low and allow c_1 to be medium-to-high to maintain consistent solution quality.

Table 11. Sensitivity analysis of the method parameters for the Test2n problem (Dimension 4).

Test2n 4	Value	Mean	Min	Max	Iters	Main Range
c_1	0.1	−146.95432	−156.66451	−128.37343	150	2.81625
	0.3	−146.19977	−156.66454	−128.38355	150	
	0.5	−146.38681	−156.66442	−114.25247	150	
	0.7	−147.60809	−156.6641	−114.25223	150	
	0.9	−149.01602	−156.66437	−114.24072	150	
c_2	0.1	−152.40955	−156.66454	−128.39005	150	8.94298
	0.3	−149.87331	−156.66447	−128.38459	150	
	0.5	−146.48201	−156.66451	−114.25223	150	
	0.7	−143.46657	−156.66437	−128.37376	150	
	0.9	−143.93359	−156.664	−114.24072	150	

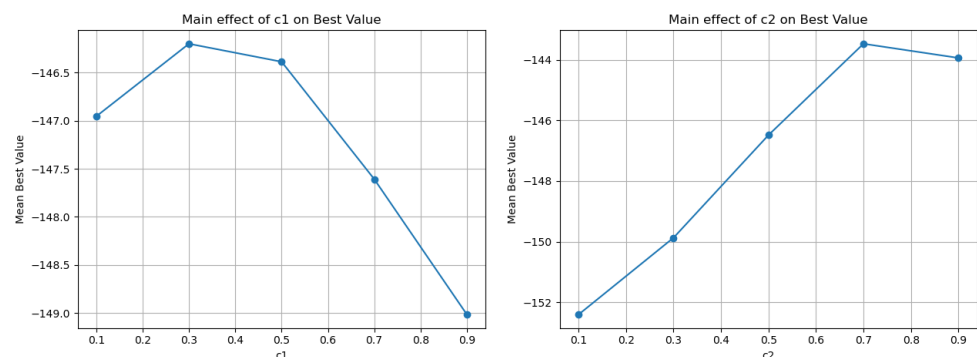


Figure 8. Graphical representation of c_1 and c_2 for the Test2n problem.

In the Rosenbrock4 problem (Table 12 and Figure 9), c_1 has the largest overall effect across all cases (range 30.29), with a dramatic improvement in mean performance as it increases from 0.1 to 0.9 (Mean Best from ~ 35.11 to ~ 4.82). The Rosenbrock function’s

narrow curved valley and anisotropy explain why stronger stochastic perturbation helps maintain mobility along the valley and avoid “dead zones” in step progression. c_2 shows a U-shaped trend: the best mean performance occurs at 0.3 (Mean Best ≈ 4.43), while very low or very high c_2 increases the risk of large outliers, as seen in maximum values that can spike dramatically. Thus, in [rosenbrock4], a high c_1 is recommended to keep search activity within the valley, and a moderate $c_2 \approx 0.3$ helps avoid both over-pulling, which can distort the valley geometry, and overly loose guidance, which delays convergence.

Table 12. Sensitivity analysis of the method parameters for the Rosenbrock problem (Dimension 4).

Rosenbrock 4	Value	Mean	Min	Max	Iters	Main Range
c_1	0.1	35.1061	0	1354.34838	150	30.29039
	0.3	14.66004	0	1038.60285	150	
	0.5	13.3723	0	593.67884	150	
	0.7	9.95311	0	524.37725	150	
	0.9	4.81572	0	314.03933	150	
c_2	0.1	18.00132	0	1354.34838	150	20.91708
	0.3	4.43389	0	235.09807	150	
	0.5	11.78364	0	593.67884	150	
	0.7	18.33745	0	400.1598	150	
	0.9	25.35097	0	1244.51484	150	

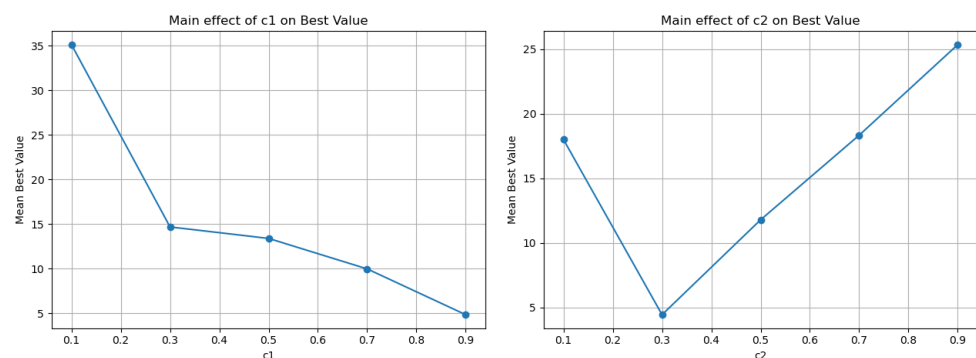


Figure 9. Graphical representation of c_1 and c_2 for the Rosenbrock problem.

Synthesizing these findings, a consistent tuning pattern emerges: in highly multimodal landscapes with many symmetric basins such as Rastrigin, a moderate c_2 around 0.5 and a moderate c_1 around 0.3–0.7 minimize mean values and stabilize convergence. In “parabolic” or polynomial landscapes like Test2n, a low c_2 and medium-to-high c_1 improve stability and mean performance, preventing premature convergence. In narrow-valley problems like Rosenbrock, strong c_1 and moderate $c_2 \approx 0.3$ appear to be the most robust choice. Finally, for dense multimodal potentials like Potential, the optimal zone tends toward low c_1 and moderate $c_2 \approx 0.3$, balancing small, targeted jumps with steady, controlled attraction toward the best.

In practical terms, the ranges that reappear as “safe defaults” are c_2 in the moderate range of 0.3–0.5, and c_1 adapted to landscape morphology: low for Potential-type landscapes, moderate for Rastrigin, high for Rosenbrock, and medium-to-high for polynomial Test2n landscapes. The min/max values per setting highlight the tendency for extreme deviations when c_2 is too high or too low especially in Rosenbrock reinforcing that the “high c_1 —moderate c_2 ” combination is often the most resilient operating point when the goal is high mean performance rather than isolated best cases.

3.5. Analysis of Computational Cost and Complexity of the SIOA Algorithm

Figure 10 illustrates the complexity of the proposed method, showing the number of objective function calls and the execution time (in seconds) for problem dimensions ranging from 20 to 260. The experimental settings follow the parameter values specified in Table 1, with the termination criterion based on the homogeneity of the best value. In addition, a limited local optimization procedure is applied at a rate of only 0.5%, enhancing the exploitation of promising regions in the search space without significantly affecting the overall global exploration strategy.

More specifically, in the ELLIPSOIDAL problem, the execution time increases gradually from 0.111 s at dimension 20 to 41.714 s at dimension 260, while the corresponding objective function calls range from 1398 to 6457. Similarly, for the ROSENBROCK problem, the execution time rises from 0.144 s at dimension 20 to 40.873 seconds at dimension 260, with the number of calls increasing from 2830 to 9200. The results indicate that both execution time and the number of calls grow as the problem dimensionality increases, with ROSENBROCK generally requiring greater computational effort in higher dimensions compared to ELLIPSOIDAL. This observation highlights the sensitivity of the method's complexity to the nature of the problem, while also confirming its ability to scale efficiently across a wide range of search space sizes.

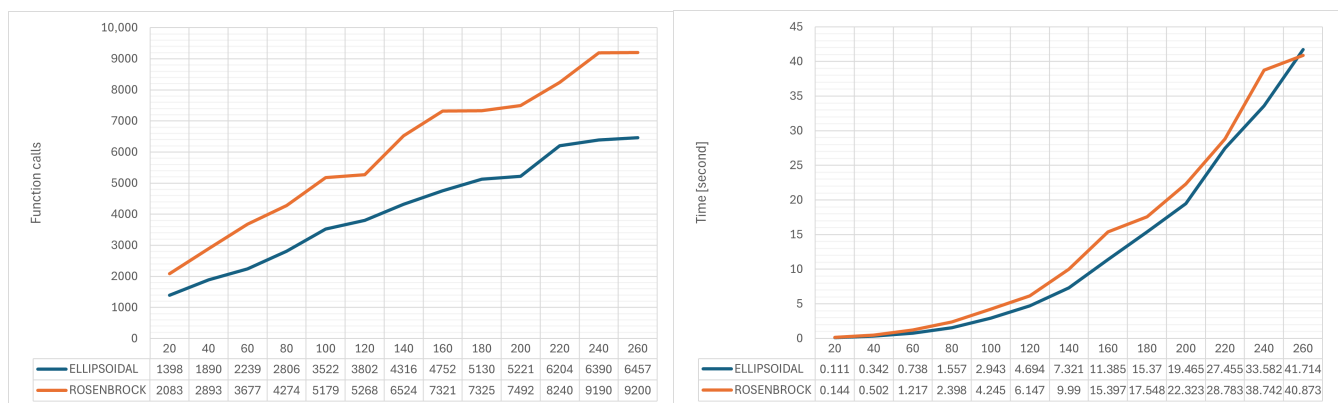


Figure 10. Computational performance (Calls and Time) of the proposed method on ELLIPSOIDAL and ROSENBROCK across dimensions 20–260.

4. Conclusions

Based on the experiments conducted, SIOA proves to be a mature, competitive, and efficient metaheuristic. In classical benchmark problems, it consistently outperforms GA, DE, PSO, and ACO in terms of required objective function calls while maintaining a high success rate; the overall evaluation footprint is significantly lower than that of traditional methods, translating into faster convergence for a given computational budget. This performance profile supports the view that the biologically inspired “sporulation–germination” mechanism, combined with self-adaptive parameter control and similarity-based replacement, provides a tangible advantage across a wide range of problem types.

The method also demonstrates notable stability: with the parameter settings of Table 1, the best result was reproduced uniformly in 12 consecutive runs, while local optimization was used minimally (only 0.5%), indicating that SIOA’s global search is sufficient to locate optimal or near-optimal solutions without relying heavily on exploitation. The algorithm’s core components—stochastic perturbation around an adaptive radius, attraction toward the global best, the “zero-reset” rule when the optimum lies near the origin, and replacement through crowding—collectively explain both the maintenance of diversity and the ability to avoid premature convergence.

In more demanding, realistic scenarios with a uniform budget of 150,000 function evaluations and no local optimization, SIOA remains highly competitive against advanced techniques. Although CMA-ES achieved the top overall rank, SIOA came very close, with results in certain cases (e.g., Tersoff Potential and Static Economic Load Dispatch) approaching the best of the leading competitors. A slightly higher variance in some less multimodal landscapes limited the mean performance, highlighting a margin for improvement in stability without undermining the overall strength of the method.

The scalability analysis shows that both runtime and function evaluations increase with problem dimension and landscape ruggedness, with problems such as Rosenbrock generally requiring more computational effort than smoother ellipsoidal forms—an observation consistent with the expected behavior of metaheuristics in difficult, poorly scaled valleys. In all cases, SIOA maintains an economical evaluation profile compared to competing approaches, a feature of direct value in costly simulations.

Overall, the method is realistically ready for application: fast in terms of evaluations, stable without relying on intensive local search, and sufficiently flexible to dynamically adapt critical parameters as the search progresses. At the same time, clear opportunities for further improvement remain. Realistic next steps include integrating more specialized, problem-sensitive local optimizers to reduce variance and improve final accuracy, as well as extending SIOA to constrained, multi-objective, and large-scale problems, where the combination of self-adaptation, crowding, and “zero-reset” may yield even greater benefits. Equally promising are explorations of hybrid versions augmented with surrogate modeling for expensive problems, further parallelization and GPU/multi-threaded implementations, the use of restart strategies and dynamic similarity thresholds, and the development of fully parameter-free versions with stronger theoretical convergence guarantees. The indicated extensions to constrained, multi-objective, and large-scale applications, along with reinforcement via dedicated local search schemes, underscore SIOA’s realistic potential as a modern foundation for further research and practical deployment.

Author Contributions: Conceptualization, I.G.T. and V.C.; methodology, V.C.; software, V.C.; validation, I.G.T., D.T. and A.M.G.; formal analysis, D.T.; investigation, I.G.T.; resources, D.T.; data curation, A.M.G.; writing—original draft preparation, V.C.; writing—review and editing, I.G.T.; visualization, V.C.; supervision, I.G.T.; project administration, I.G.T.; funding acquisition, I.G.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been financed by the European Union: Next Generation EU through the Program Greece 2.0 National Recovery and Resilience Plan, under the call RESEARCH—CREATE—INNOVATE, project name “iCREW: Intelligent small craft simulator for advanced crew training using Virtual Reality techniques” (project code: TAEDK-06195).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cauchy, A.-L. Méthode générale pour la résolution des systèmes d’équations simultanées. *Compte Rendus Hebd. Séances L’Académie Sci.* **1847**, *25*, 536–538.
2. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2006.
3. Newton, I. *Method of Fluxions*; Colson, J., Translator; Henry Woodfall: London, UK, 1736; Original work written in 1671.
4. Metropolis, N.; Ulam, S. The Monte Carlo method. *J. Am. Stat. Assoc.* **1949**, *44*, 335–341. [[CrossRef](#)] [[PubMed](#)]
5. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)] [[PubMed](#)]

6. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
7. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
8. Charillogis, V.; Tsoulos, I.G.; Tzallas, A.; Karvounis, E. Modifications for the Differential Evolution Algorithm. *Symmetry* **2022**, *14*, 447. [[CrossRef](#)]
9. Charillogis, V.; Tsoulos, I.G. A Parallel Implementation of the Differential Evolution Method. *Analytics* **2023**, *2*, 17–30. [[CrossRef](#)]
10. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks 1995, Perth, Australia, 27 November–1 December 1995; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
11. Qin, A.K.; Huang, V.L.; Suganthan, P.N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol.* **2009**, *13*, 398–417. [[CrossRef](#)]
12. Brest, J.; Greiner, S.; Boskovic, B.; Mernik, M.; Zumer, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **2006**, *10*, 646–657. [[CrossRef](#)]
13. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [[CrossRef](#)]
14. Hansen, N.; Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **2001**, *9*, 159–195. [[CrossRef](#)] [[PubMed](#)]
15. Nelder, J.A.; Mead, R. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308–313. [[CrossRef](#)]
16. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milano, Italy, 1992.
17. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization (Technical Report TR06)*; Erciyes University, Engineering Faculty, Computer Engineering Department: Kayseri, Turkey, 2005.
18. Lam, A. *BFGS in a Nutshell: An Introduction to Quasi-Newton Methods Demystifying the Inner Workings of BFGS Optimization*; Towards Data Science: San Francisco, CA, USA, 2020.
19. Charillogis, V.; Tsoulos, I.G. Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions. *Information* **2022**, *13*, 217. [[CrossRef](#)]
20. Gianni, A.M.; Tsoulos, I.G.; Charillogis, V.; Kyrou, G. Enhancing Differential Evolution: A Dual Mutation Strategy with Majority Dimension Voting and New Stopping Criteria. *Symmetry* **2025**, *17*, 844. [[CrossRef](#)]
21. Tsoulos, I.G.; Charillogis, V.; Kyrou, G.; Stavrou, V.N.; Tzallas, A. OPTIMUS: A Multidimensional Global Optimization Package. *J. Open Source Softw.* **2025**, *10*, 7584. [[CrossRef](#)]
22. Siarry, P.; Berthiau, G.; Durdin, F.; Haussy, J. Enhanced simulated annealing for globally minimizing functions of many-continuous variables. *ACM Trans. Math. Softw. (TOMS)* **1997**, *23*, 209–228 [[CrossRef](#)]
23. Koyuncu, H.; Ceylan, R. A PSO based approach: Scout particle swarm algorithm for continuous global optimization problems. *J. Comput. Des. Eng.* **2019**, *6*, 129–142. [[CrossRef](#)]
24. LaTorre, A.; Molina, D.; Osaba, E.; Poyatos, J.; Del Ser, J.; Herrera, F. A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. *Swarm And Evol. Comput.* **2021**, *67*, 100973. [[CrossRef](#)]
25. Gaviano, M.; Kvasov, D.E.; Lera, D.; Sergeyev, Y.D. Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Trans. Math. Softw. (TOMS)* **2003**, *29*, 469–480. [[CrossRef](#)]
26. Jones, J.E. On the determination of molecular fields.—II. From the equation of state of a gas. *Proc. R. Soc. London. Ser. Contain. Pap. Math. Phys. Character* **1924**, *106*, 463–477.
27. Zabinsky, Z.B.; Graesser, D.L.; Tuttle, M.E.; Kim, G.I. Global optimization of composite laminates using improving hit and run. In *Recent Advances in Global Optimization*; Princeton University Press: Princeton, NJ, USA, 1992; pp. 343–368.
28. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]
29. Lee, Y.; Filliben, J.; Micheals, R.; Phillips, J. *Sensitivity Analysis for Biometric Systems: A Methodology Based on Orthogonal Experiment Designs*; National Institute of Standards and Technology Gaithersburg (NISTIR): Gaithersburg, MD, USA, 2012; p. 20899. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.