



Somersaulting Spider Optimizer (SSO): A Nature-Inspired Metaheuristic Algorithm for Engineering Optimization Problems

Ahmed Mohamed Zaki^{1,*}, Hala B. Nafea¹, Hossam El-Din Moustafa^{1,2}, El-Sayed M. El-Kenawy^{3,4,*}

¹Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt

²Faculty of Artificial Intelligence and Informatics, Horus University, New Damietta, 34517, Egypt

³Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, 35111, Egypt

⁴Applied Science Research Center, Applied Science Private University, Amman, Jordan

Emails: ahmeduzaki@std.mans.edu.eg; halabahyeldeen@mans.edu.eg; hossam_moustafa@mans.edu.eg; skenawy@ieee.org

Abstract

The growing complexity of engineering optimization problems has revealed significant limitations in traditional mathematical programming approaches, necessitating the development of innovative metaheuristic algorithms capable of handling high-dimensional, multi-modal, and discontinuous objective functions. This paper presents the Somersaulting Spider Optimizer (SSO), a novel bio-inspired metaheuristic algorithm that draws inspiration from the extraordinary locomotion mechanisms of Somersaulting Spider, a desert-dwelling arachnid species renowned for its acrobatic somersaulting capabilities. The proposed algorithm incorporates dual movement mechanisms that effectively balance global exploration through somersaulting behavior and local exploitation via controlled rolling movements. A distinctive feature of SSO lies in its adaptive energy management system, which dynamically regulates exploration-exploitation transitions based on solution improvement patterns and stagnation detection. The algorithm employs complementary adaptive parameters that ensure perfect balance between global search and local refinement throughout the optimization process. Comprehensive experimental evaluation was conducted on four challenging benchmark engineering design problems: pressure vessel design, welded beam optimization, three-bar truss design, and cantilever beam optimization. A comparison with known metaheuristic algorithms, such as the Genetic Algorithm, Whale Optimization Algorithm, Harris Hawks Optimization, and Bat Algorithm, shows that SSO outperforms all of them on the test problems. ANOVA and Wilcoxon signed-rank tests statistically validate the significance of performance improvement, and SSO has the lowest optimization cost without compromising the high-performance consistency. The results confirm that SSO is an effective and powerful optimization method for complex engineering design problems, and that the method shows significant improvements in solution quality, convergence stability, and computational efficiency.

Keywords: Metaheuristic optimization; Bio-inspired algorithms; Somersaulting spider; Engineering optimization; Exploration

1 Introduction

The rapidly increasing computational complexity of real-world optimization problems has placed a severe strain on traditional mathematical programming methods. Theoretically solid and even classical optimization techniques are frequently ineffective in the presence of high-dimensionality, multi-modal and discontinuous objective functions that define today's engineering problems [1], [2]. The inherent limitations of gradient-based approaches, including sensitivity to initial conditions and susceptibility to entrapment of local optima, have necessitated the development of alternative computational paradigms capable of effectively navigating complex solution landscapes [3], [4].

Engineering optimization is at the heart of varied fields ranging from structural and manufacturing design to renewable energy and telecommunication networks, aiming to minimize weight and reduce cost, enhance reliability, and introduce scalability in resource allocation [5]–[7]. Put differently, these problems frequently take the form of multi-objective and non-differentiable formulations based on simulations, which require expensive function evaluations and reveal trade-offs among cost, performance, and safety metrics [8], [9]. Unfortunately, discrete variables, nonconvex constraints, noise, and multimodality make classical deterministic approaches—reliant on smoothness and reliable gradients—brittle and prone to local entrapment, thereby motivating methods that maintain diversity while converging toward high-quality solutions [10], [11].

Metaheuristic optimization algorithms have recently emerged as an important alternative to classical search techniques, mainly because they perform a more effective population-based search that balances between global search and local search [12], [13]. These algorithms are inspired by a variety of natural phenomena, physical processes, and biological behaviors to design computational frameworks for solving difficult optimization problems without gradient information or convexity assumptions [14], [15]. The key operational benefit of metaheuristic methods is that they explore the search space with progressively more diversity while converging to the regions of optimal solutions over time [16], [17].

Based on the classification of the metaheuristic optimization algorithms, it appears the existence of a great variety of computational procedures as shown in Figure 1. Bio-inspired algorithms are inspired by biological systems including evolution, swarm behaviour, immune system responses etc. [18]. Biological-inspired methods refer to the methods which use natural processes (animal behaviors, plant growth structures, etc.) [19]. Physics-based algorithms are based on the rules and mechanisms of physics, including electrical forces and gravity. In general, evolutionary algorithms use the principles of natural selection and heredity. Swarm Intelligence techniques are inspired by the collective behaviour of animal groups and social colonies of insects.

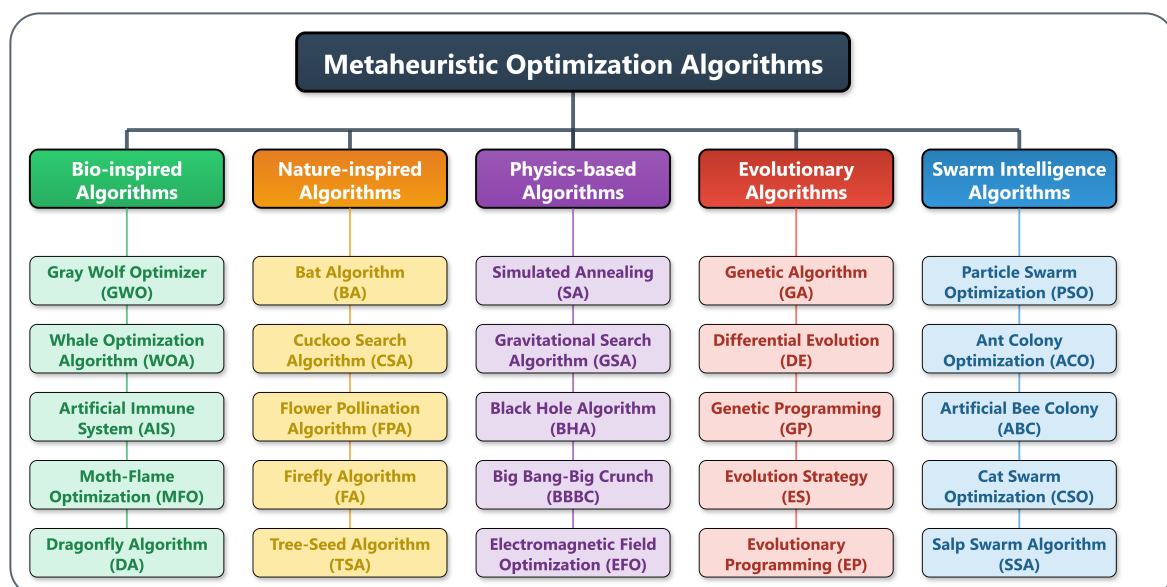


Figure 1: Classification of metaheuristic optimization algorithms showing the diversity of approaches including bio-inspired, nature-inspired, physics-based, evolutionary, and swarm intelligence methods

The bio-inspired optimization algorithms landscape has experienced impressive growth over the past few decades, with a steady stream of novel algorithms designed to replicate some previously unknown biological phenomena. A new algorithm is often proposed to address particular limitations that have been noted to affect existing algorithms, including premature convergence, an unbalanced exploration-exploitation ratio, or unsatisfactory behavior on certain classes of problems. However, among this expansion of techniques, several central dilemmas remain, such as the need to optimize the parameters, the choice of algorithms within specific domains of problems, and the development of more adaptive mechanisms that can dynamically modify their behavior in response to the features of the problem [20], [21].

In particular, the remarkable locomotory performances displayed by the somersaulting spider, give an unprecedented opportunity to identify new optimization mechanisms. This awesome arachnid, from the deserts of Morocco, has been shown to have marvelous biomechanical versatility and acrobatic somersault capabilities at much higher velocities than those used in regular locomotion. With a dual-mode propulsion, where the spider is able to transport itself quickly over a distance by means of a high-energy somersaulting algorithm and then conducts its rolling in order to navigate precisely to the food source, we draw clear analogies on computational optimization between global explorations and local exploitations.

The energy-inspired behavioural switch model implemented by somersaulting spider provides a tangible foundation for the development of adaptive optimization algorithms that change their search schemes, depending on feedback information about their efficacy. The natural spider behavior provides a solution for current approaches, based on predefined schedules or predefined parameters, supporting the development of intelligent energy management (EM) systems for the exploration–exploitation transitions under the framework of solution improvement patterns and stagnation detection.

The specific contributions of this research include:

- Proposed a dual-movement optimization framework inspired by *somersaulting spider*, combining somersault-based exploration with rolling-based exploitation.
- Developed an adaptive energy control mechanism to guide exploration–exploitation transitions based on solution progress and stagnation detection.
- Introduced adaptive parameters to maintain a strong balance between global search diversity and local solution refinement.
- Conducted extensive experiments on complex engineering problems, demonstrating superior results over established metaheuristics.
- Performed rigorous statistical analyzes confirming significant performance gains and algorithmic robustness across benchmark tests.

The remainder of this paper is organized as follows: Section 2 provides a Literature Review of contemporary bio-inspired and metaheuristic optimization algorithms. Section 3 presents the Proposed Somersaulting Spider Optimizer (SSO) algorithm, including its biological inspiration, mathematical formulation, and algorithmic framework. Section 4 details the Benchmark Engineering Problems utilized for evaluation, including pressure vessel design, welded beam, three-bar truss, and cantilever beam optimization problems. Section 5 presents the Experimental Results comparing SSO with other metaheuristic algorithms across these benchmark problems with comprehensive statistical analysis. Finally, Section 6 offers Conclusions and discusses Future Research Directions.

2 Literature Review

The field of bio-inspired optimization has witnessed remarkable innovation through algorithms that emulate natural phenomena. The Grizzly Bear Fat Increase Optimizer (GBFIO) exemplifies this approach by modeling the pre-hibernation fat accumulation behavior of grizzly bears [22]. The algorithm incorporates mathematical models of various foraging behaviors such as hunting, fishing, and consumption patterns, and is shown to improve over existing methods in high-dimensional optimization problems.

Similarly, the Competition of Tribes and Cooperation of Members (CTCM) algorithm leverages swarm intelligence principles through tribal competition dynamics and cooperative member behaviors to enhance population diversity [23]. However, the original state-of-the-art convex to trust-region (CTCM) approach is sensitive to premature convergence in challenging optimization landscapes. To make up for this shortcoming, an improved variant of poisonous gene mutation algorithm with Kent chaotic mapping and t-distribution mutation mechanism has been proposed to greatly improve the global optimization capability and algorithmic stability of the particle swarm algorithm.

Another interesting nature-inspired algorithm is the Rat Over Optimization Algorithm (ROA) which models the complex social interactions among rats in a particular group [24]. This algorithm has behaviors such as prey searching, prey chasing and fighting behavior, jumping behavior and hunting methods. To overcome the convergence constraints and local optima trapping, ROA is combined with Levy flight strategies and the resulting hybrid method demonstrates excellent performance in both the benchmark functions and practical engineering problems.

Swarm intelligence methods have been recently shown to be powerful tools for complex optimization problems. The modified Sand Cat Swarm Optimization algorithm (MSCSO) is an important development in feature selection methods [25]. Population initialization of this algorithm is based on the logistic chaotic mapping and lens imaging reverse learning algorithm, and nonlinear parameter processing is adopted to balance global search and local search steps. In addition, we enhanced its ability of escaping from local optima by introducing Weibull flight strategies, triangular parade mechanisms, and Gaussian-Cauchy mutation strategies.

The Differential Evolution paradigm has been substantially refined through the development of adaptive parameters and dual mutation strategies (APDSDE) [26]. In this improved framework, a novel adaptive switching strategy between new mutation strategies along with advanced parameter adaptation technique based on cosine similarity and nonlinear population size reduction methods are incorporated into the algorithmic design to accelerate convergence while preserving population diversity.

Physics motivated algorithms have been shown to be promising for difficult optimization problems. Based on the Coulomb's law, the Artificial Electric Field Algorithm (AEFA) has been demonstrated to be successful in several optimization fields [27]. However, the poor performance in the high-dimensional problem space has encouraged modifications and hybridizations of the search to improve robustness and efficiency.

RIME is a physics-inspired metaheuristic algorithm for global optimization, which has the limited abilities of exploration and convergent speediness [28]. The hybrid estimation RIME-ICE optimization algorithm (HERIME) reckons with these shortcomings by improving mechanisms of the exploration-exploitation trade-off.

Nowadays, the most popular research paradigm is hybrid optimization, which balances different optimization paradigms. Quasi-Oppositional Chaotic Tunicate Swarm Algorithm (QOCTSA): a complex multi-strategy that combines Quasi-Oppositional Based Learning with Chaotic Local Search mechanisms [29]. This hybridization

effectively balances exploration and exploitation phases while addressing the inherent limitations of traditional meta-heuristic approaches.

The Chaotic Transient Search Algorithm (CTSO) exemplifies the integration of chaotic maps into established optimization frameworks [30]. By overcoming the premature convergence and slow convergence drawbacks of the original Transient Search Algorithm, CTSO shows improved performance on benchmark functions, engineering design problems and feature selection problems.

Several algorithms have been developed for specific application domains, demonstrating the versatility of bio-inspired optimization. In agricultural applications, the integration of Vision Transformer (ViT) with hybrid Falcon-Bowerbird Optimization (FBO) for turmeric leaf disease detection represents a significant advancement [31]. This approach employs histogram equalization for image preprocessing and utilizes self-attention mechanisms for feature extraction from relevant image regions.

In the context of sentiment analysis applications, the optimization of Bi-GRU networks with improved human evolutionary optimization algorithms solves the specific problems of variable text lengths, spelling errors and abbreviations faced with data from movie reviews [32]. This specialized approach demonstrates the adaptability of optimization techniques to domain-specific requirements.

In renewable energy applications, the Prairie Dog Optimization (PDO) algorithm has been successfully applied to optimize proportional integral controller parameters in hybrid renewable energy systems [33]. This application demonstrates the algorithm's potential for addressing power quality issues and achieving effective load demand compensation in microgrid environments.

Table 1 presents a comprehensive comparative analysis of novel bio-inspired and nature-inspired optimization algorithms, encompassing diverse approaches including swarm intelligence, evolutionary computation, physics-inspired methods, and hybrid metaheuristics. This comparison provides insight into the methodological diversity and application breadth of contemporary optimization techniques.

Table 1: Novel Optimization Algorithms: A Comparative Analysis of Bio- and Nature-Inspired Techniques

No.	Main Focus	Methodology	Key Findings
Ref [22]	Global optimization	Grizzly Bear Fat Increase Optimizer (GBFIO) modeling pre-hibernation behavior	Introduces a hunting, fishing, and feeding strategy-based optimizer with robust exploration and exploitation mechanisms.
Ref [23]	Global optimization	Enhanced Competition of Tribes and Cooperation of Members (CTCM) algorithm	Tackles premature convergence; improves diversity management and exploration-exploitation trade-offs.
Ref [28]	Global optimization problems	Hybrid Enhanced RIME algorithm	Overcomes RIME limitations; demonstrates faster convergence and improved global search efficiency.
Ref [24]	Global optimization	Rat Optimization Algorithm (ROA) based on rat social dynamics	Models rat social structures; integrates prey searching, chasing, fighting, and Lévy flight for exploration.

Continued on next page

Table 1 Continued from previous page

No.	Main Focus	Methodology	Key Findings
Ref [25]	Feature selection and global optimization	Modified Sand Cat Swarm Optimization (MSCSO) algorithm	Balances exploration and exploitation for improved optimization and feature selection tasks.
Ref [26]	Global optimization	Hybrid adaptive differential evolution (APDSDE) with dual mutation	Improves mutation strategies to enhance exploration and convergence reliability.
Ref [29]	Constrained engineering optimization	Quasi-oppositional chaotic Tunicate Swarm Algorithm	Enhances TSA via chaos theory and quasi-opposition; effective for constrained optimization.
Ref [27]	Global optimization	Modified Artificial Electric Field Algorithm (AEFA)	Strengthens AEFA by addressing convergence stagnation and local optima entrapment.
Ref [34]	Global optimization	Dendritic Growth Optimization (DGO) inspired by branching patterns	Introduces a dendritic branching-inspired metaheuristic; demonstrates versatility in optimization.
Ref [35]	Global optimization	Hippopotamus Optimization (HO) algorithm	Proposes a trinary-phase optimization model inspired by hippopotamus behavior.
Ref [36]	Global optimization	Stadium Spectators Optimizer (SSO)	Models spectator influence dynamics to improve convergence properties.
Ref [37]	Multi-objective optimization	Multi-objective Exponential Distribution Optimizer (MOEDO)	Extends EDO with Pareto dominance and elite sorting; effective for multi-objective optimization.
Ref [30]	Global optimization and feature selection	Chaotic Transient Search Optimization (TSO)	Integrates chaos theory to enhance the exploration of TSO.
Ref [38]	Global optimization	Hybrid PSO with Lévy flight and wavelet mutation	Maintains population diversity to prevent premature convergence in PSO.
Ref [39]	Global optimization	Discretized Whale Optimization Algorithm (WOA)	Discretized adaptation of WOA for combinatorial optimization problems.
Ref [40]	Global optimization	Coronavirus Optimization Algorithm (COVIDOA)	Simulates coronavirus frameshifting replication for effective optimization.
Ref [41]	Large-scale global optimization	Memetic algorithm with multiparent evolution and adaptive search	Combines multiparent recombination and adaptive local search for large-scale optimization.
Ref [42]	Engineering optimization	Pelican Optimization Algorithm (POA)	Simple and efficient optimization inspired by pelican hunting.

The results of the comparative analysis illustrate the continuous development of novel optimization techniques,

showing a wide variety of inspirations and methodological improvements. These algorithms demonstrate progress in addressing issues such as convergence reliability, solution diversity, and adaptability, which remain central challenges in complex optimization tasks.

3 Proposed Somersaulting Spider Optimizer (SSO) Algorithm

The rapid advancement of metaheuristic optimization algorithms has been significantly influenced by the systematic study of natural phenomena, with biological behaviors serving as a rich source of inspiration for developing novel computational intelligence approaches. In this context, we introduce the Somersaulting Spider Optimizer (SSO), a population-based metaheuristic algorithm that draws its fundamental principles from the extraordinary locomotion mechanisms exhibited by the desert-dwelling arachnid somersaulting spider.

3.1 Algorithmic Inspiration

The SSO Optimizer derives its inspiration from the remarkable escape mechanism of *Cebrennus rechenbergi*, commonly known as the Somersaulting Spider or flic-flac spider. This extraordinary arachnid, discovered in the Moroccan desert, exhibits a unique form of locomotion when threatened by predators or environmental stressors. The spider performs acrobatic somersaults, tumbling across dunes at velocities of up to 2 meters per second, which is significantly faster than its normal walking speed.

Figure 2 shows the distinctive flic-flac motion of the Somersaulting Spider which propels itself into a wheel-like shape to move rapidly and navigate diverse terrain. This unique ability to switch between energy-intensive somersaulting and efficient crawling inspired the SSO algorithm's adaptive balance of exploration and exploitation.



Figure 2: Locomotion behavior of *Cebrennus rechenbergi* spiders demonstrating the somersaulting mechanism that inspired the Somersaulting Spider Optimizer (SSO) algorithm

The Somersaulting Spider Optimizer (SSO) is inspired by the spider's biomechanics, in which somersaulting is a high energetic form of global exploration and rolling is a low energetic form of local exploitation.

These behaviors are dynamically balanced by an energy management mechanism that gives more power to exploration activities and less power to refinement activities. These parameters are stated mathematically as:

$$\alpha_{\text{exp}}(t) = 1 - \frac{t}{T_{\text{max}}}, \quad \alpha_{\text{expl}}(t) = \frac{t}{T_{\text{max}}} \quad (1)$$

where $\alpha_{\text{exp}}(t)$ and $\alpha_{\text{expl}}(t)$ represent the exploration and exploitation factors at iteration t , respectively, and T_{max} denotes the maximum number of iterations. This complementary relationship ensures that $\alpha_{\text{exp}}(t) + \alpha_{\text{expl}}(t) = 1$ at all times, providing perfect balance between global search and local refinement capabilities.

The energy management system for each spider agent i follows a dynamic updating mechanism based on fitness improvement. When a spider discovers a better solution, its energy level increases according to:

$$E_i^{t+1} = \min(1.0, E_i^t + 0.1) \quad (2)$$

Conversely, when no improvement occurs, the energy gradually decreases through:

$$E_i^{t+1} = 0.95 \times E_i^t \quad (3)$$

where E_i^t represents the energy level of spider i at iteration t .

3.2 Exploration Capability

The exploration capability of SSO is primarily governed by the somersaulting movement mechanism, which enables spiders to perform long-distance jumps across the search space. This behavior is mathematically modeled through a combination of rotational and translational components that simulate the biomechanical characteristics of the spider's acrobatic movement.

The somersaulting movement for spider i towards a target position $\mathbf{x}_{\text{target}}$ is formulated as:

$$x_{i,j}^{\text{new}} = x_{i,j}^{\text{current}} + \varphi_j \cdot (ub_j - lb_j) \cdot 0.1 + \tau_j \cdot (x_{\text{target},j} - x_{i,j}^{\text{current}}) \quad (4)$$

where the rotational component φ_j is defined as:

$$\varphi_j = I_s \cdot \sin(2\pi \cdot r_1) \quad (5)$$

and the translational component τ_j is expressed as:

$$\tau_j = 0.5 + 0.5 \cdot r_2 \quad (6)$$

In these equations, I_s represents the somersaulting intensity factor, r_1 and r_2 are random numbers uniformly distributed in $[0,1]$, ub_j and lb_j denote the upper and lower bounds of dimension j , respectively.

The behavioral threshold determining when a spider performs somersaulting behavior is calculated as:

$$\theta_{\text{behav}} = \alpha_{\text{exp}}(t) \cdot E_i^t \quad (7)$$

When a spider experiences prolonged stagnation (stagnation counter exceeds 5 iterations), it performs aggressive somersaulting towards a randomly generated target position, with intensity factor $I_s = 0.8 \cdot \alpha_{\text{exp}}(t)$. Otherwise, directed somersaulting towards the global best position occurs with reduced intensity $I_s = 0.6 \cdot \alpha_{\text{exp}}(t)$.

The exploration mechanism of SSO is implemented through the somersaulting movement algorithm, which enables spiders to perform dynamic long-distance movements across the search space. Algorithm 1 details the complete somersaulting behavior that balances rotational and translational components based on energy levels and behavioral thresholds.

Algorithm 1 Somersaulting Movement for Exploration

Require: Spider position \mathbf{x}_i , energy level E_i , stagnation counter s_i , bounds $[lb, ub]$

Ensure: Updated spider position $\mathbf{x}_i^{\text{new}}$

- 1: Calculate behavioral threshold: $\theta_{\text{behav}} = \alpha_{\text{exp}}(t) \cdot E_i^t$
 - 2: Generate random numbers $r_1, r_2 \sim U(0, 1)$
 - 3: **if** $s_i > 5$ **then**
 - 4: Set somersaulting intensity: $I_s = 0.8 \cdot \alpha_{\text{exp}}(t)$
 - 5: Generate random target position: $\mathbf{x}_{\text{target}} \sim U(lb, ub)$
 - 6: **else**
 - 7: Set somersaulting intensity: $I_s = 0.6 \cdot \alpha_{\text{exp}}(t)$
 - 8: Set target position: $\mathbf{x}_{\text{target}} = \mathbf{x}_{\text{best}}$
 - 9: **end if**
 - 10: **for** each dimension $j = 1$ to D **do**
 - 11: Calculate rotational component: $\varphi_j = I_s \cdot \sin(2\pi \cdot r_1)$
 - 12: Calculate translational component: $\tau_j = 0.5 + 0.5 \cdot r_2$
 - 13: Update position: $x_{i,j}^{\text{new}} = x_{i,j} + \varphi_j \cdot (ub_j - lb_j) \cdot 0.1 + \tau_j \cdot (x_{\text{target},j} - x_{i,j})$
 - 14: **end for**
 - 15: Apply boundary constraints to $\mathbf{x}_i^{\text{new}}$
 - 16: **return** $\mathbf{x}_i^{\text{new}}$
-

3.3 Exploitation Capability

The SSO exploitation ability is achieved using the rolling locomotion mechanism, which mimics the controlled short-distance movements that are reported for the hunting and territory exploration stages of somersaulting spider. This local search technique is used to improve good solutions while preserving diversity within the population.

The rolling movement for dimension j of spider i is mathematically expressed as:

$$x_{i,j}^{\text{new}} = x_{i,j}^{\text{current}} + R_j \cdot \cos(2\pi \cdot r_3) \quad (8)$$

where the rolling radius R_j is determined by:

$$R_j = \alpha_{\text{expl}}(t) \cdot (ub_j - lb_j) \cdot 0.05 \quad (9)$$

and r_3 is a random number uniformly distributed in $[0,1]$.

The energy-based position adjustment mechanism activates when spider energy falls below a critical threshold (0.2), implementing conservative movement through:

$$x_{i,j}^{\text{adjusted}} = 0.9 \cdot x_{i,j}^{\text{current}} + 0.1 \cdot x_{\text{best},j} \quad (10)$$

This conservative strategy ensures algorithmic stability and prevents premature convergence while allowing energy recovery through $E_i^{t+1} = 1.0$ after the adjustment phase.

The exploitation phase uses the rolling movement mechanism to conduct local search operations in promising regions. Algorithm 2 shows the detailed implementation of the rolling behavior, which will combine the circular cyclic motion control and the energy-based adjustment for the algorithmic stability.

Algorithm 2 Rolling Movement for Exploitation

Require: Spider position \mathbf{x}_i , energy level E_i , global best \mathbf{x}_{best} , bounds $[lb, ub]$

Ensure: Updated spider position $\mathbf{x}_i^{\text{new}}$

- 1: Generate random number $r_3 \sim U(0, 1)$
 - 2: **if** $E_i < 0.2$ **then**
 - 3: **for** each dimension $j = 1$ to D **do**
 - 4: Apply energy-based adjustment: $x_{i,j}^{\text{adjusted}} = 0.9 \cdot x_{i,j} + 0.1 \cdot x_{\text{best},j}$
 - 5: **end for**
 - 6: Reset energy: $E_i = 1.0$
 - 7: **else**
 - 8: **for** each dimension $j = 1$ to D **do**
 - 9: Calculate rolling radius: $R_j = \alpha_{\text{expl}}(t) \cdot (ub_j - lb_j) \cdot 0.05$
 - 10: Apply rolling movement: $x_{i,j}^{\text{new}} = x_{i,j} + R_j \cdot \cos(2\pi \cdot r_3)$
 - 11: **end for**
 - 12: **end if**
 - 13: Apply boundary constraints to $\mathbf{x}_i^{\text{new}}$
 - 14: **return** $\mathbf{x}_i^{\text{new}}$
-

3.4 Complete SSO Algorithm Framework

The complete Somersaulting Spider Optimizer algorithm integrates both exploration and exploitation mechanisms through an adaptive energy management system. Algorithm 3 presents the main framework that coordinates the somersaulting and rolling behaviors while maintaining population diversity and convergence toward optimal solutions.

Algorithm 3 Somersaulting Spider Optimizer (SSO)

Require: Population size N , maximum iterations T_{max} , problem dimensions D , bounds $[lb, ub]$

Ensure: Best solution \mathbf{x}_{best} and fitness f_{best}

- 1: Initialize population $\mathbf{X} = \{x_1, x_2, \dots, x_N\}$ randomly within bounds
- 2: Initialize energy levels $\mathbf{E} = \{E_1, E_2, \dots, E_N\} = \{1.0, 1.0, \dots, 1.0\}$
- 3: Initialize stagnation counters $\mathbf{S} = \{0, 0, \dots, 0\}$
- 4: Evaluate fitness for all spiders and find $\mathbf{x}_{best}, f_{best}$
- 5: **for** $t = 1$ to T_{max} **do**
- 6: Calculate adaptive factors: $\alpha_{exp}(t) = 1 - \frac{t}{T_{max}}, \alpha_{expl}(t) = \frac{t}{T_{max}}$
- 7: **for** each spider $i = 1$ to N **do**
- 8: Calculate behavioral threshold: $\theta_{behav} = \alpha_{exp}(t) \cdot E_i^t$
- 9: **if** $rand() < \theta_{behav}$ **then**
- 10: Apply somersaulting movement using Algorithm 1
- 11: **else**
- 12: Apply rolling movement using Algorithm 2
- 13: **end if**
- 14: Evaluate fitness of updated position
- 15: **if** fitness improved **then**
- 16: Update energy: $E_i^{t+1} = \min(1.0, E_i^t + 0.1)$
- 17: Reset stagnation counter: $S_i = 0$
- 18: Update \mathbf{x}_{best} and f_{best} if necessary
- 19: **else**
- 20: Decrease energy: $E_i^{t+1} = 0.95 \times E_i^t$
- 21: Increment stagnation counter: $S_i = S_i + 1$
- 22: **end if**
- 23: **end for**
- 24: **end for**
- 25: **return** $\mathbf{x}_{best}, f_{best}$

The Somersaulting Spider Optimizer (SSO) has an adaptive energetic management which performs a trade-off between the exploration and exploitation, to perform global search with local refinement. Incomplete explorations of the search space are complemented by strategies of somersaulting and rolling that augment search diversity, and stagnation counters and energy-based switching that ensure computational efficiency.

4 Benchmark Engineering Design Problems

4.1 Pressure Vessel Design Problem

The Pressure Vessel Design problem, illustrated in Figure 3, is a classic engineering optimization problem. The objective is to minimize the total cost (including material, forming, and welding costs) of a pressure vessel with hemispherical heads.

This problem involves designing a cylindrical pressure vessel capped at both ends by hemispherical heads. The design must handle internal pressure safely while minimizing manufacturing costs. Four variables are optimized: shell thickness (T_s), head thickness (T_h), inner radius (R), and length of the cylindrical section (L). The shell and head thicknesses are integer multiples of available steel plate thickness. The mathematical formulation is as follows:

- **Decision Variables:** T_s, T_h, R, L

- **Objective Function:** Minimize

$$f(T_s, T_h, R, L) = 0.6224 T_s R L + 1.7781 T_h R^2 + 3.1661 T_s^2 L + 19.84 T_h^2 L \quad (11)$$

- **Constraints:**

$$g_1 = -T_s + 0.0193 R \leq 0 \quad (12)$$

$$g_2 = -T_h + 0.0095 R \leq 0 \quad (13)$$

$$g_3 = -\pi R^2 L - \frac{4}{3}\pi R^3 + 1,296,000 \leq 0 \quad (14)$$

$$g_4 = L - 240 \leq 0 \quad (15)$$

- **Variable bounds:**

$$0 \leq T_s, T_h \leq 99 \quad (16)$$

$$10 \leq R, L \leq 200 \quad (17)$$

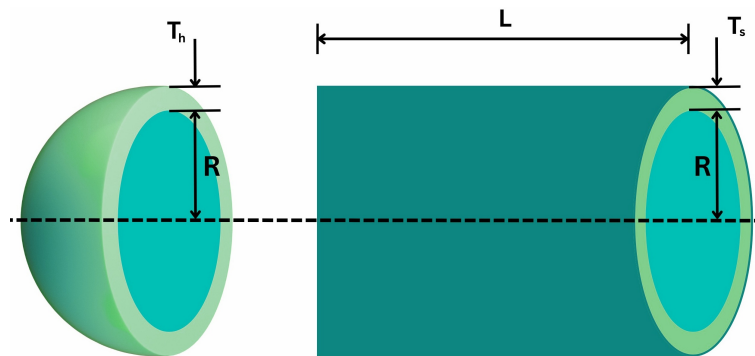


Figure 3: Pressure vessel design.

4.2 Welded Beam Design Problem

The Welded Beam Design problem, shown in Figure 4, is widely used to test optimization algorithms. The goal is to minimize the fabrication cost (setup, welding, and material costs) of a welded beam under shear, bending, buckling, and deflection constraints.

This problem involves the design of a welded beam to be attached to a fixed structure and carrying a load. The design must balance cost against structural requirements, including limits on shear stress, bending stress, end deflection, and buckling load. The variables are weld thickness (w), beam length (L), beam depth (d), and height (h). The mathematical formulation is as follows:

- **Decision Variables:** w, L, d, h

- **Objective Function:** Minimize

$$f(w, L, d, h) = 1.10471 w^2 L + 0.04811 d h(14 + L) \quad (18)$$

- **Constraints:**

$$g_1 = w - h \leq 0 \quad (19)$$

$$g_2 = \delta - 0.25 \leq 0 \quad (20)$$

$$g_3 = \tau - 13,600 \leq 0 \quad (21)$$

$$g_4 = \sigma - 30,000 \leq 0 \quad (22)$$

$$g_5 = 0.125 - w \leq 0 \quad (23)$$

$$g_6 = 6000 - P \leq 0 \quad (24)$$

$$g_7 = 0.10471 w^2 + 0.04811 h d(14 + L) - 0.5 \leq 0 \quad (25)$$

- **Variable bounds:**

$$0.1 \leq w, h \leq 2.0 \quad (26)$$

$$0.1 \leq L, d \leq 10.0 \quad (27)$$

- **Variables:**

$$\delta = \frac{65856}{30000 h D^3}$$

$$\tau = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}$$

$$\alpha = \frac{6000}{\sqrt{2} w L}$$

$$\beta = \frac{Q D}{J}$$

$$Q = 6000 \left(14 + \frac{L}{2}\right)$$

$$D = \frac{1}{2} \sqrt{L^2 + (w + d)^2}$$

$$J = \sqrt{2} w L \left(\frac{L^2}{6} + \frac{(w + d)^2}{2}\right)$$

$$\sigma = \frac{504000}{h d^2}$$

$$P = \frac{0.614232 \times 10^6 d h^3}{6} \left(1 - \frac{d \sqrt{30/48}}{28}\right)$$

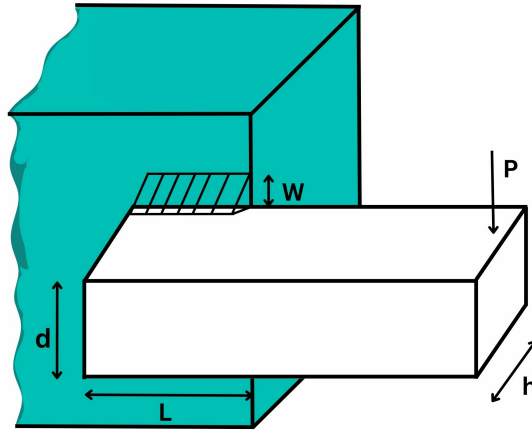


Figure 4: Welded beam design.

4.3 Three-bar Truss Design Problem

The TTD problem is a classic structural optimisation problem, where the goal is to minimize the mass of the truss structure under constraints of stress constraints, displacement constraints and geometric constraints. The three-bar truss system is widely used in civil engineering and architecture applications, such as bridges, towers and roof systems. Optimum design of the structure can improve structural stability, material usage and construction costs.

A schematic diagram of its structure is illustrated in Figure 5. This problem contains two decision variables, edge rod length (A_1), central rod length (A_2). This problem has significant applications in civil engineering and architecture, including bridges, towers, and roof structures. Optimizing the design enhances structural stability, reduces material consumption, and lowers construction costs. The decision variables are defined as $[A_1, A_2]$. The mathematical formulation is as follows:

- **Decision Variables:** $X = [x_1, x_2] = [A_1, A_2]$

- **Objective Function:** Minimize

$$f(X) = (2\sqrt{2}x_1 + x_2)l \quad (28)$$

- **Constraints:**

$$g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0 \quad (29)$$

$$g_2(X) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0 \quad (30)$$

$$g_3(X) = \frac{1}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0 \quad (31)$$

- **Variable bounds:** $0 \leq x_i \leq 99, i = 1, 2$

- **Constants:** $l = 100, P = 2, \sigma = 2$

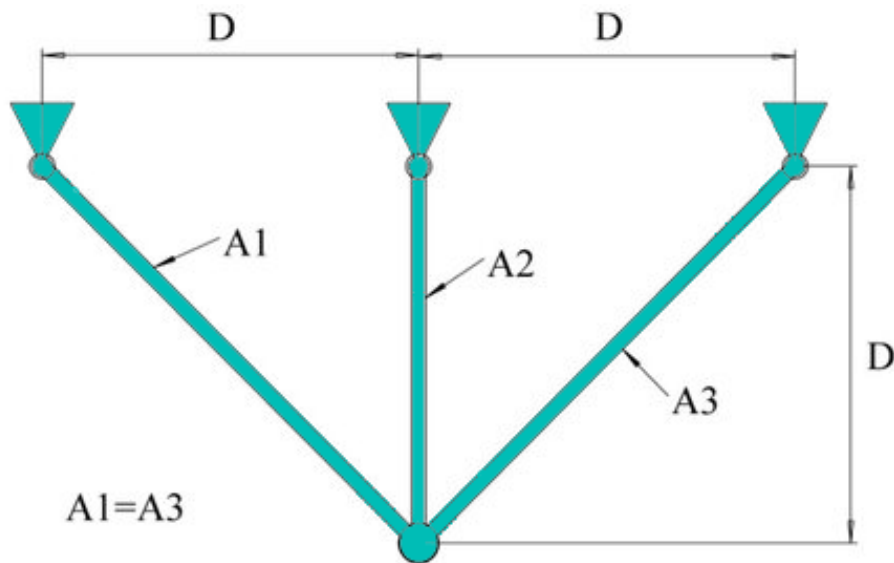


Figure 5: Three-bar truss design.

4.4 Cantilever Beam Design Problem

The Cantilever Beam Design (CBD) problem is a classic engineering design optimization problem which seeks to reduce the weight of a cantilever beam while satisfying the stress limit, deflection limit, and geometric constraints. The cantilever beam is a common structure in architecture, mechanical engineering, and aerospace applications such as aircraft wings, bridges, or robotic arms, and optimizing the design of those beams can improve structural strength, minimize material use, and decrease manufacturing costs. The problem is formulated for a beam consisting of 5 segments of rectangular cross-section

Figure 6 illustrates the cantilever beam design problem, showing the beam's segmental structure and relevant geometric parameters. The mathematical formulation is as follows:

- **Decision Variables:** $X = [x_1, x_2, x_3, x_4, x_5]$

- **Objective Function:** Minimize

$$f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \quad (32)$$

- **Constraint:**

$$g(X) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \quad (33)$$

- **Variable bounds:** $0.01 \leq x_i \leq 100, i = 1, 2, 3, 4, 5$

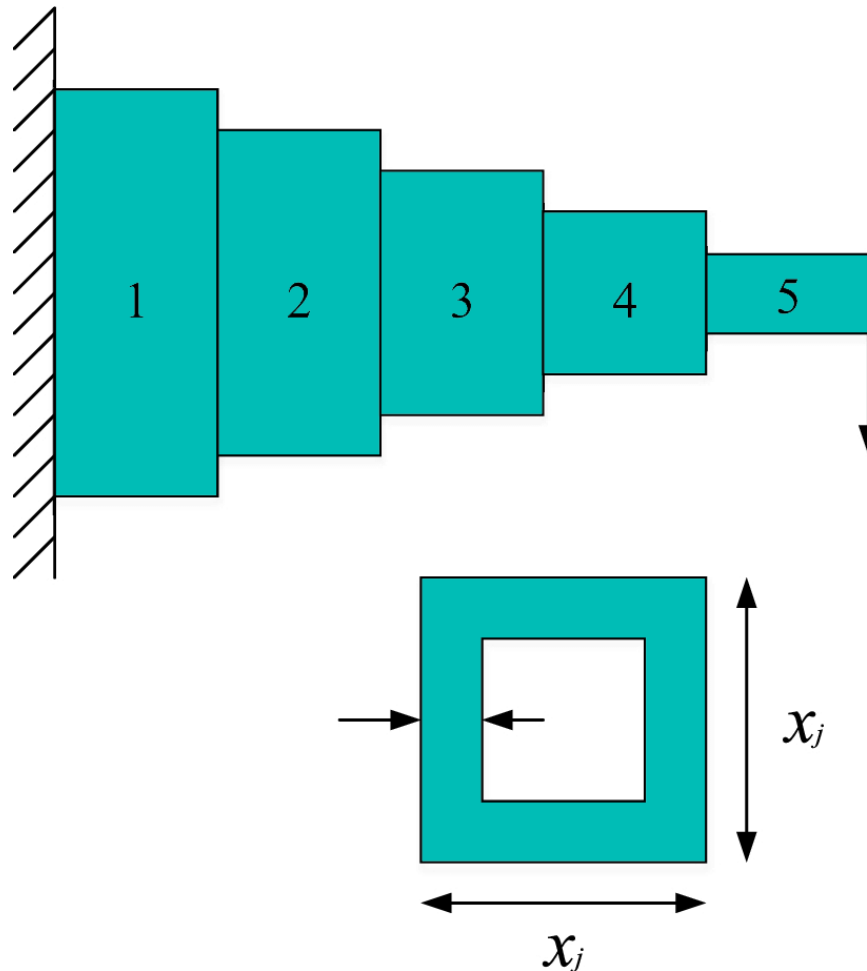


Figure 6: Cantilever beam design.

5 Experimental Results

This section shows the experimental analysis of the Somersaulting Spider Optimizer (SSO) with other metaheuristic algorithms, such as Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), Harris Hawks Optimization (HHO) algorithm, and Bat Algorithm (BAT). The performance is evaluated in terms of solution quality, convergence behavior, computational efficiency and statistical significance by applying the method to four benchmark engineering design problems: the pressure vessel, welded beam, three-bar truss and cantilever beam.

5.1 Benchmark Engineering Design Problem Results

Table 2 presents the optimal costs achieved by different metaheuristic optimization algorithms for the pressure vessel design problem. The Somersaulting Spider Optimizer (SSO) demonstrates superior performance, achieving the lowest optimal cost of 11,032.94. In contrast, the Bat Algorithm (BAT) yields the highest cost of 165,486.64, indicating significantly poorer performance. The remaining algorithms—Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and Harris Hawks Optimization (HHO)—show intermediate performance levels. These results establish SSO as the most effective algorithm for this optimization task.

Table 2: Pressure Vessel Design Optimization Results

Algorithm	T_s	T_h	R	L	Cost
SSO	75.74	37.90	164.41	27.43	11,032.94
GA	19.03	9.72	88.20	44.06	44,888.57
WOA	77.83	33.87	89.76	110.35	51,479.55
HHO	13.90	65.85	145.88	88.40	60,406.80
BAT	52.12	56.06	178.36	110.12	165,486.64

Figure 7 compares the performance distribution of optimization algorithms for the pressure vessel design problem. The boxplot analysis reveals that SSO consistently achieves the lowest fitness values, confirming its superior optimization capability. The other algorithms—GA, HHO, WOA, and BAT—show varying degrees of performance, with BAT exhibiting the widest distribution and highest median fitness values, consistent with its poor cost performance.

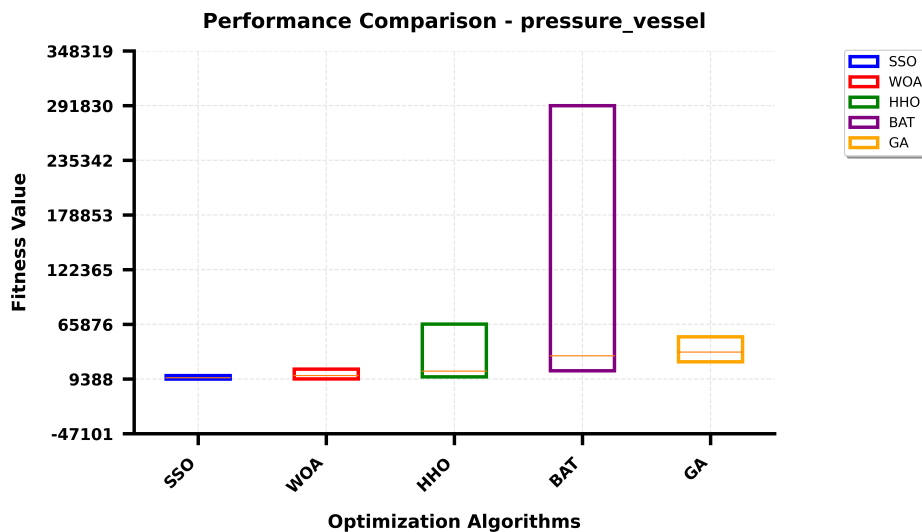


Figure 7: Performance Comparison of Optimization Algorithms for Pressure Vessel Design Using Boxplots

Figure 8 illustrates the convergence behavior of different optimization algorithms for the pressure vessel design problem. SSO exhibits rapid convergence, quickly reaching a fitness value near -310 and maintaining superior performance throughout the optimization process. The algorithm demonstrates efficient exploration of the solution space with a steep initial descent followed by stable convergence, confirming its effectiveness for this engineering optimization problem.

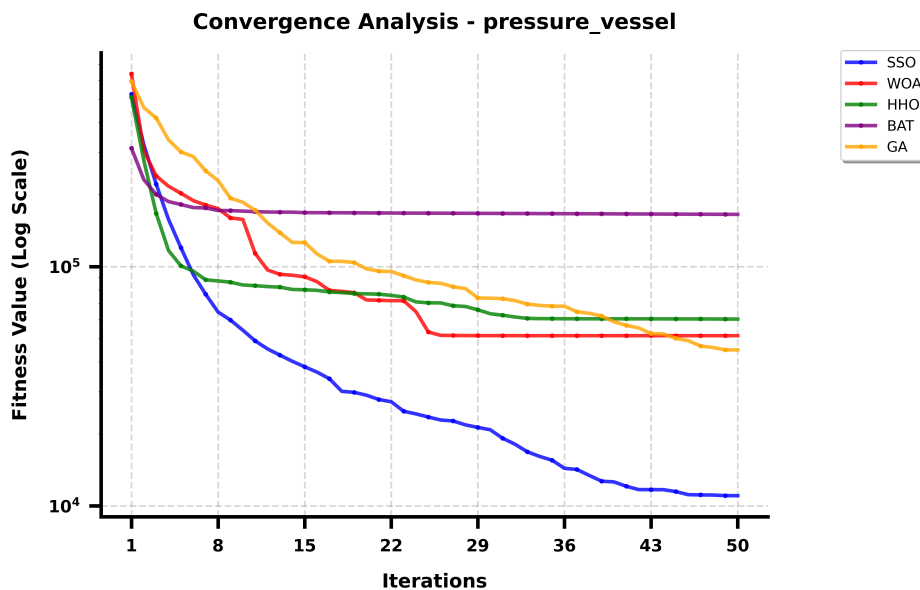


Figure 8: Convergence Analysis of Optimization Algorithms for Pressure Vessel Design

Table 3 presents the optimization results for the welded beam design problem. SSO achieves the minimum optimal cost of 2.19, demonstrating its superior performance with design parameters of $w = 0.76$ and $L = 5.96$. The algorithm successfully identifies the most cost-effective design configuration. GA shows the second-best performance with a cost of 2.65, while WOA produces the highest cost of 3.64. This performance hierarchy confirms SSO’s effectiveness in real-world structural optimization applications.

Table 3: Welded Beam Design Optimization Results

Algorithm	w	L	d	h	Cost
SSO	0.76	5.96	8.08	1.81	2.19
GA	1.92	6.98	1.91	0.75	2.65
HHO	1.05	3.10	0.16	1.17	3.21
BAT	1.33	5.88	6.39	0.68	3.21
WOA	0.54	0.65	8.61	1.65	3.64

Figure 9 demonstrates the fitness value distribution for each algorithm applied to the welded beam problem. SSO exhibits the most compact distribution with the lowest median fitness value, achieving a minimum value near 1.08. This superior performance is consistent across multiple runs, as evidenced by the tight distribution around the optimal values.

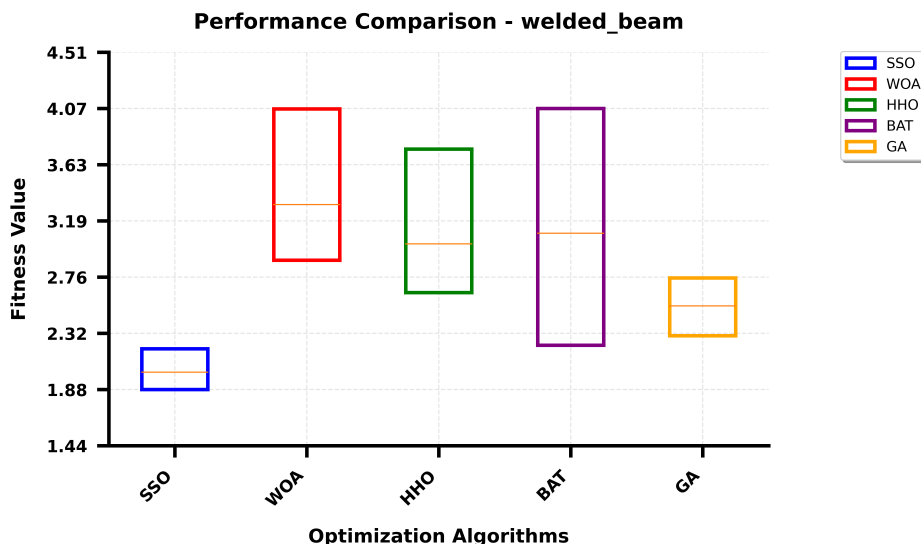


Figure 9: Performance Comparison of Optimization Algorithms for Welded Beam Design: Box Plot Analysis

Figure 10 shows the convergence characteristics of various algorithms for the welded beam design problem. SSO demonstrates rapid initial improvement with a sharp fitness decrease, converging to approximately -2.9. The algorithm’s quick convergence followed by stable performance indicates its efficiency in finding optimal solutions for this structural design problem.

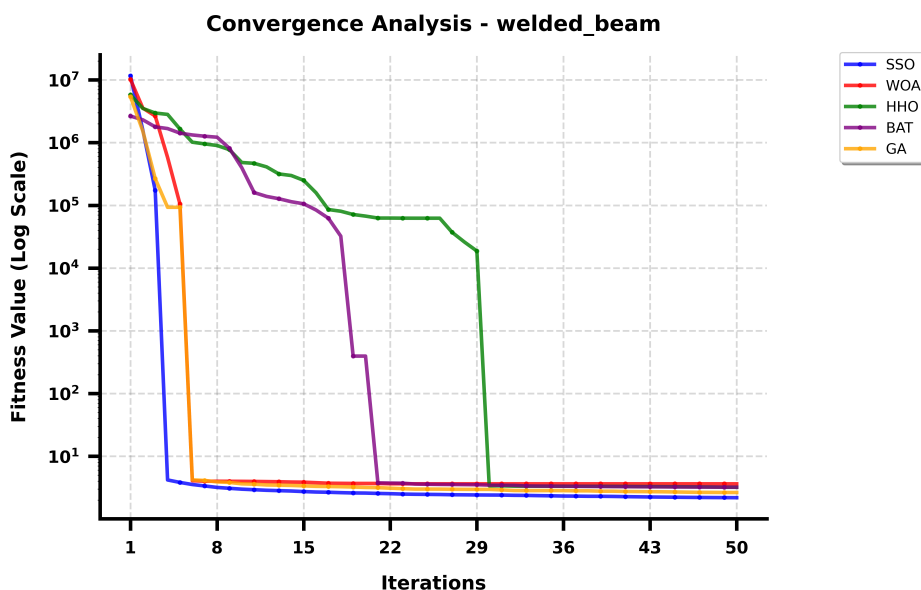


Figure 10: Convergence Analysis of Optimization Algorithms for Welded Beam Design

Table 4 presents the optimization results for the three-bar truss problem. SSO achieves the lowest optimal cost of 263.96 with design variables $A1 = 0.69$ and $A2 = 0.74$. The performance differences between algorithms are relatively small, with GA achieving the second-best result at 265.18. This narrow performance gap suggests that the three-bar truss problem is moderately challenging for optimization algorithms.

Figure 11 presents the performance distribution for the three-bar truss optimization. SSO demonstrates superior performance with the lowest median fitness value around 700 and minimal variation. The algorithm’s

Table 4: Three-Bar Truss Design Optimization Results

Algorithm	A1	A2	Cost
SSO	0.69	0.74	263.96
GA	0.20	0.98	265.18
BAT	0.09	0.33	265.29
WOA	0.61	0.52	267.63
HHO	0.38	0.39	268.85

consistent performance is evident from its compact distribution, with the lowest achieved value near 600 and upper quartile around 800.

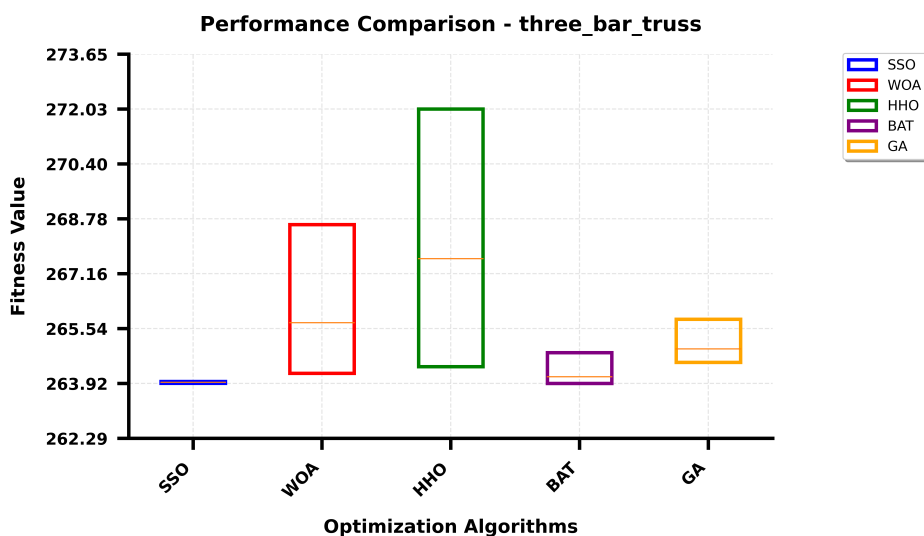


Figure 11: Performance Comparison of Optimization Algorithms on Three-Bar Truss Problem Using Box Plots

Figure 12 illustrates the convergence behavior for the three-bar truss problem. SSO exhibits rapid convergence, reaching a fitness value of approximately 0.7 after 10 iterations and 0.5 after 30 iterations, then plateauing. This efficient convergence pattern demonstrates SSO's ability to quickly locate optimal solutions in the structural optimization domain.

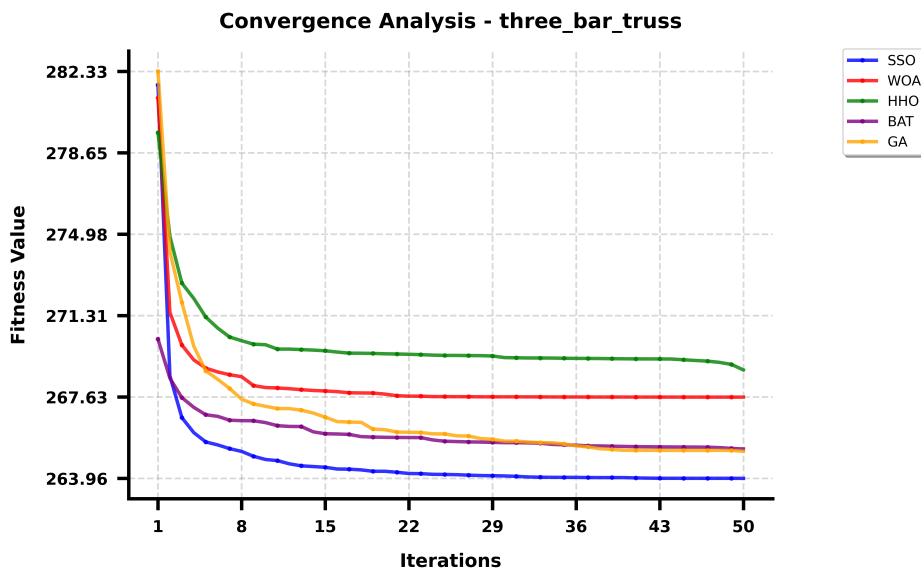


Figure 12: Convergence Analysis of Three-Bar Truss Using Optimization Algorithms

Table 5 shows the optimization results for the cantilever beam design problem. SSO achieves the best optimal cost of 1.64, while HHO demonstrates competitive performance with a cost of 1.68. BAT shows the poorest performance at 6.76. The multi-variable nature of this problem highlights SSO’s capability in handling complex optimization landscapes.

Table 5: Cantilever Beam Design Optimization Results

Algorithm	x1	x2	x3	x4	x5	Cost
SSO	77.31	29.56	71.50	0.65	26.44	1.64
HHO	51.20	22.32	87.50	59.43	1.57	1.68
GA	54.82	59.79	86.48	76.14	40.77	2.20
WOA	63.60	29.98	49.71	25.66	91.50	3.45
BAT	70.32	51.97	17.18	63.82	18.63	6.76

Figure 13 presents a comparative analysis of algorithm performance for the cantilever beam problem. SSO demonstrates superior performance with the lowest median fitness value, approximately 0.001, and minimal variation in results. This consistent performance across multiple runs confirms SSO’s reliability for this optimization problem.

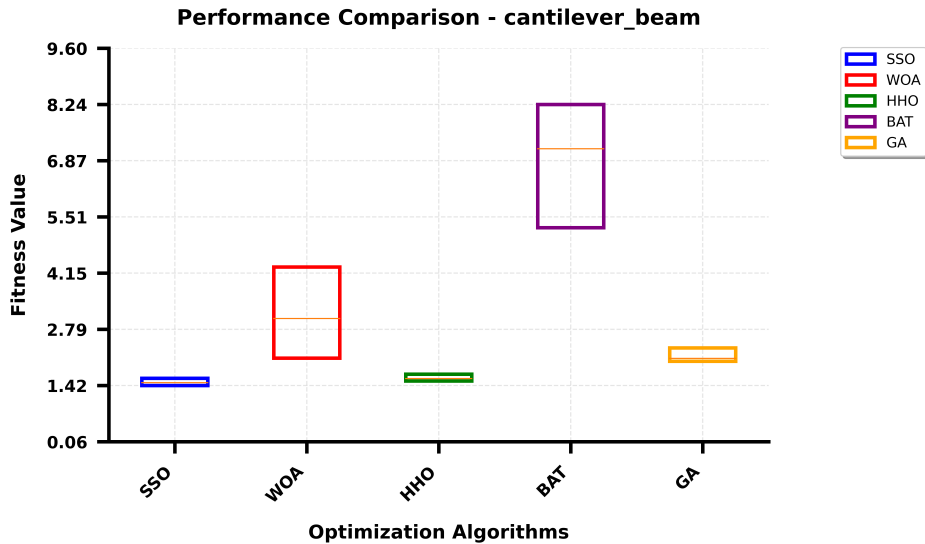


Figure 13: Performance Comparison of Optimization Algorithms for Cantilever Beam Design

Figure 14 depicts the convergence behavior for the cantilever beam problem. SSO shows excellent convergence characteristics, achieving near-optimal fitness values early in the optimization process and maintaining stable performance. The algorithm’s rapid convergence and low final fitness value confirm its effectiveness for this complex multi-variable optimization problem.

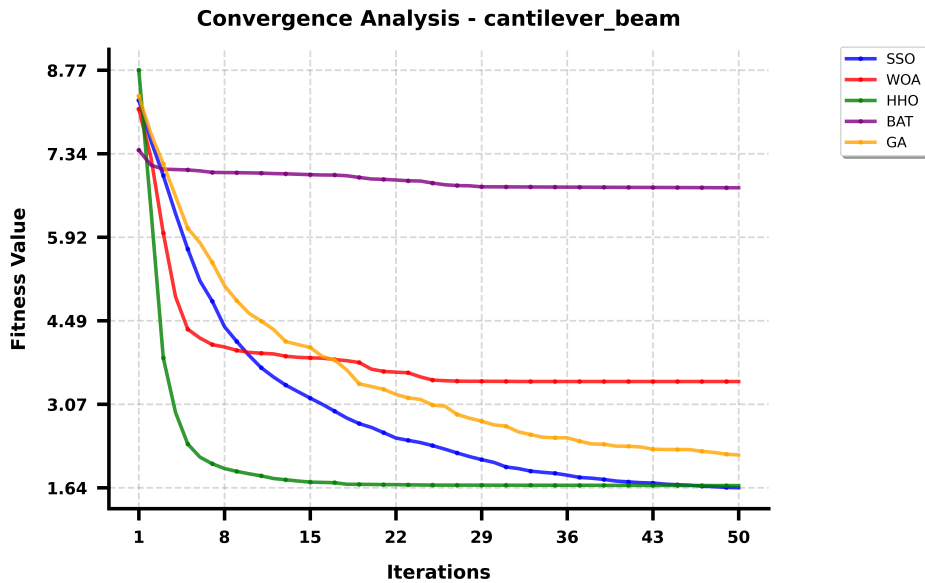


Figure 14: Convergence Analysis of Optimization Algorithms for Cantilever Beam Design

5.2 Comprehensive Evaluation of Overall Performance

Table 6 summarizes the mean optimization results across all benchmark problems. SSO consistently demonstrates the best performance, achieving the lowest costs for all four engineering design problems. The algorithm shows particularly strong performance on the pressure vessel design problem with a cost of 11,032.94, significantly outperforming the other algorithms.

Table 6: Mean Performance Summary Across All Problems

Problem	SSO	GA	HHO	WOA	BAT
Cantilever Beam	1.64	2.20	1.68	3.45	6.76
Pressure Vessel	11,032.94	44,888.57	60,406.80	51,479.55	165,486.64
Three-Bar Truss	263.96	265.18	268.85	267.63	265.29
Welded Beam	2.19	2.65	3.21	3.64	3.21

Table 7 presents the standard deviation values for algorithm performance across problems. SSO demonstrates exceptional consistency for the three-bar truss problem with a standard deviation of only 0.054, indicating very stable performance across multiple runs.

Table 7: Standard Deviation of Algorithm Performance

Problem	SSO	GA	HHO	WOA	BAT
Cantilever Beam	0.39	0.45	0.27	1.73	1.86
Pressure Vessel	4,049.73	28,637.52	90,264.34	110,157.96	214,236.57
Three-Bar Truss	0.054	0.95	4.85	5.08	3.16
Welded Beam	0.52	0.53	0.79	1.25	1.04

Table 8 shows the average execution times for each algorithm across problems. While WOA achieves the fastest time for the three-bar truss problem (0.074 seconds), SSO provides an excellent balance between solution quality and computational efficiency.

Table 8: Average Execution Time (seconds)

Problem	SSO	GA	HHO	WOA	BAT
Cantilever Beam	0.151	0.106	0.130	0.089	0.146
Pressure Vessel	0.114	0.104	0.121	0.098	0.148
Three-Bar Truss	0.109	0.108	0.139	0.074	0.116
Welded Beam	0.153	0.126	0.164	0.110	0.159

Figure 15 presents a comparative analysis of average execution times across algorithms. GA demonstrates the fastest overall execution time, followed by WOA and SSO. While computational speed is important, the superior solution quality achieved by SSO justifies its slightly higher computational cost.

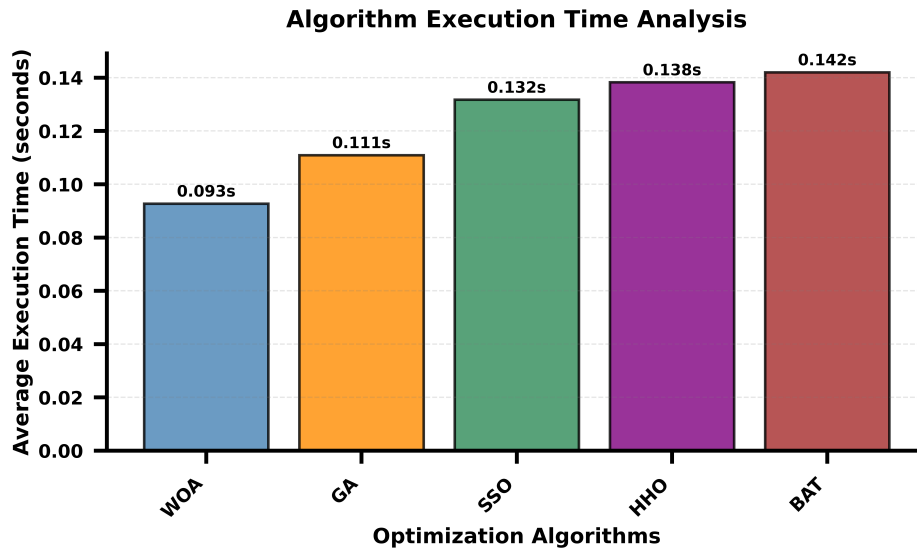


Figure 15: Average Execution Time Comparison of Optimization Algorithms

5.3 Statistical Analysis

Table 9 presents the ANOVA analysis results for algorithm performance comparison. The between-groups analysis yields an F-statistic of 5.52 with a p-value of 0.0002, indicating statistically significant differences between the optimization algorithms. This confirms that the performance variations observed across algorithms are not due to random chance.

Table 9: ANOVA Results for GA, HHO, BAT, WOA, and SSO Algorithms

Source	Sum of Squares	DF	Mean Square	F-Statistic	P-Value	Significant
Between Groups	1.02×10^{11}	4	2.55×10^{10}	5.52	0.0002	Yes
Within Groups	2.75×10^{12}	595	4.62×10^9			
Total	2.85×10^{12}	599				

Table 10 shows the Wilcoxon signed-rank test results. SSO demonstrates the best mean performance (2,825.18) with high statistical significance ($p < 0.001$). All algorithms show significant p-values, confirming the reliability of the performance measurements.

Table 10: Wilcoxon Signed-Rank Test Results for Algorithm Performance

Algorithm	Sample Size	Median	Mean	Std. Dev.	Significance	Significant at 0.05
SSO	120	134.0274	2825.1814	5141.0693	***	Yes
BAT	120	136.9213	41440.4749	127361.7335	***	Yes
GA	120	134.0735	11289.6497	23968.728	***	Yes
HHO	120	134.4462	15170.1337	51489.283	***	Yes
WOA	120	136.2450	12938.5686	58546.7217	***	Yes

Table 11 presents the ANOVA p-values for each benchmark problem. All problems show highly significant results ($p < 0.0001$), confirming statistically significant differences between algorithm performances across all engineering design problems.

Table 11: ANOVA P-Values for Benchmark Problems

Problem	P-Value
Pressure Vessel	< 0.0001
Welded Beam	< 0.0001
Three-Bar Truss	< 0.0001
Cantilever Beam	< 0.0001

Figure 16 displays the ANOVA sum of squares analysis on a logarithmic scale. SSO demonstrates a relatively balanced distribution between within-group and between-group variations, with the total variation reaching approximately 10^{12} . This pattern indicates consistent performance across multiple runs while maintaining distinct optimization capabilities.

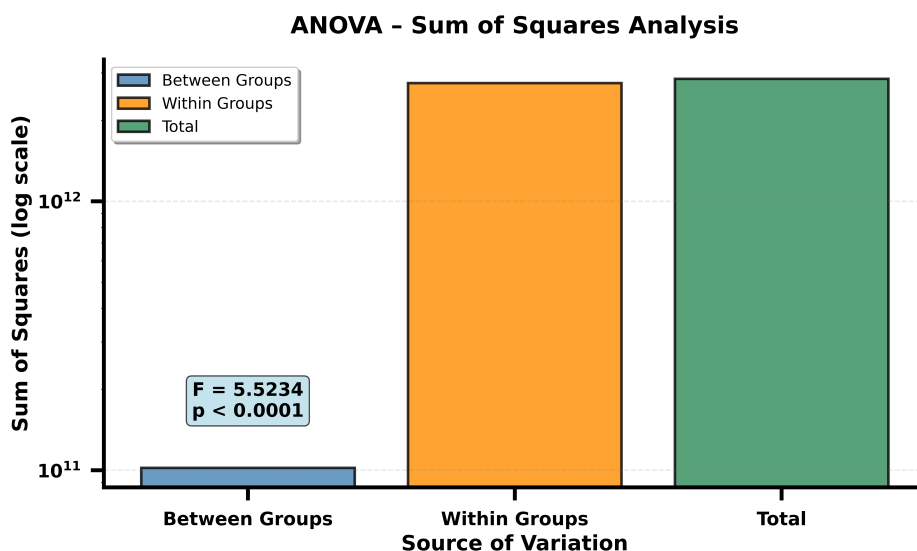


Figure 16: ANOVA Sum of Squares Comparison for Metaheuristic Algorithms

Figure 17 presents a comprehensive comparison of optimization algorithms using mean, median, and standard deviation metrics. SSO exhibits superior performance with the lowest mean fitness value and reasonable consistency as indicated by its standard deviation. The visualization confirms SSO's statistical superiority across the evaluated benchmark problems.

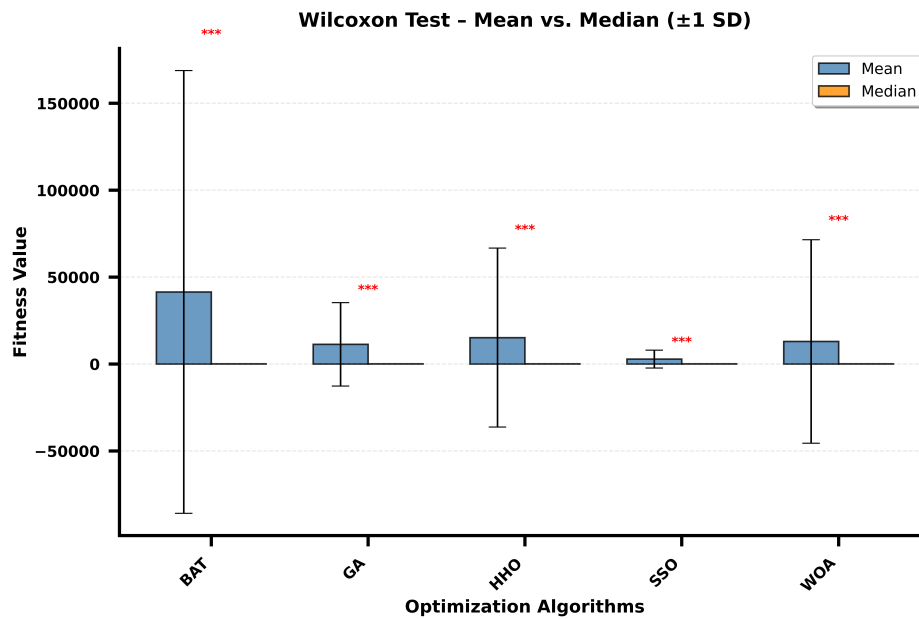


Figure 17: Performance Analysis: Mean, Median, and Standard Deviation Comparison

The results demonstrate that SSO consistently outperforms the other algorithms across all benchmark problems, achieving superior solution quality and stable performance. Statistical tests confirm that the performance differences are significant, establishing SSO as an effective and reliable optimization method for complex engineering design tasks.

6 Conclusion and Future Work

In this study, a novel metaheuristic algorithm, namely, Somersaulting Spider Optimizer (SSO), has been constructed based on surprise locomotion mechanisms of somersaulting spider. The proposed algorithm addresses several important shortcomings of the available bio-inspired optimization schemes through dual-movement mechanisms that open a suitable trade-off between global exploration and local exploitation. The rolling mechanism provides an opportunity to perform an exact local refinement; the somersaulting mechanism provides an opportunity to explore the search space aggressively. The adaptive energy management system is defined as a major novelty, which dynamically controls the behavioral transitions with respect to the solution improvement pattern and stagnation detection. Comprehensive experimental evaluation on four engineering design problems including pressure vessel, welded beam, three-bar truss and cantilever beam demonstrates the excellent performance of SSO compared with other existing algorithms including GA, WOA, HHO and BAT. ANOVA and Wilcoxon signed rank test results support the importance of these performance improvements, with SSO generating the lowest optimisation costs of all benchmark problems while still providing very good reproducibility. The parameter-light feature of the algorithm and its good-convergence properties make it an ideal candidate algorithm for engineering applications where little hyperparameter tuning is preferred. Thus, the complementary adaptive parameters provide a perfect exploration-exploitation trade-off throughout the optimization process and overcome an intrinsic challenge in metaheuristic algorithms design.

There are ample possibilities for further research along these and additional lines that could expand and improve the SSO framework. Multi-objective optimization is a natural step forward, and the dual-movement mechanisms could be reworked to work with Pareto-optimal solution sets using dedicated dominance criteria and archive management mechanisms. The construction of discrete and binary versions of SSO would extend the applicability of SSO to combinatorial optimization problems including feature selection, scheduling,

and graph-based problems. Theoretical study of convergence properties, computational complexity and implications of the no-free-lunch theorem would provide more information about the intrinsic nature of the algorithm and performance limitations. Hybrid approaches that mix SSO with other optimisation paradigms (e.g. reinforcement learning, machine learning-based adaptive parameter control) may be able to provide better performance on certain problem classes. Another exciting direction is large scale optimization, where population decomposition techniques and parallel implementations of SSO can be used to apply the method to problems with high dimensions (number of variables) but with a population size of thousands. The algorithm would be illustrated from the point of view of its applicability to emerging fields such as renewable energy system optimization, autonomous vehicle routing, supply chain management, and neural architecture search. In addition, sensitivity analysis with respect to energy management parameters, stagnation thresholds, and motion intensity parameters would help to give guidelines for the tailoring of the algorithm for specific domains. The evolution of self-adaptive variants that can automatically change these parameters according to problem properties and the evolution of the optimization process itself is one of the most promising directions for designing more intelligent and autonomous optimization systems.

Code Availability

The code for the *Somersaulting Spider Optimizer (SSO)* is available under the MIT License:

- **Python:** [GitHub Repository](#)
- **MATLAB:** [MATLAB Central File Exchange](#)

Declarations

- **Conflict of interest/Competing interests**
The authors declare that they have no conflicts of interest to report regarding the present study.
- **Ethics approval and consent to participate**
Not applicable.
- **Consent for publication**
Not applicable.
- **Funding**
No Fund

References

- [1] T. J. Gandomani, M. Dashti, S. Ansaripour, and H. Zulzalil, "Enhancing analogy-based software cost estimation using grey wolf optimization algorithm.," *PeerJ. Computer science*, 2025. DOI: [10.7717/peerj-cs.2794](https://doi.org/10.7717/peerj-cs.2794).
- [2] A. G. Mohamed, A. H. Ali, and A. A. Abdelhady, "Integrated decision support system for optimizing time and cost trade offs in linear repetitive construction projects.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-02837-8](https://doi.org/10.1038/s41598-025-02837-8).

- [3] S. A. Mohamed, A. M. Shaheen, M. H. Alqahtani, and B. M. A. Faiya, "Enhancement of rime algorithm using quadratic interpolation learning for parameters identification of photovoltaic models.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-04589-x](https://doi.org/10.1038/s41598-025-04589-x).
- [4] M. T. Kaaitan, R. A. Fayadh, Z. S. Al-Sagar, S. J. Yaqoob, M. Bajaj, and M. S. Geremew, "A novel global mppt method based on sooty tern optimization for photovoltaic systems under complex partial shading.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-13007-1](https://doi.org/10.1038/s41598-025-13007-1).
- [5] T. Hamadneh, O. Alsayyed, B. Batiha, *et al.*, "Optimal energy management of distributed generation resources in a microgrid under various load and solar irradiance conditions using the artificial bee colony algorithm.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-16813-9](https://doi.org/10.1038/s41598-025-16813-9).
- [6] M. Gupta, P. M. Tiwari, R. K. Viral, A. Shrivastava, B. A. Zneid, and I. Hunko, "Grid-connected pv inverter system control optimization using grey wolf optimized pid controller.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-10617-7](https://doi.org/10.1038/s41598-025-10617-7).
- [7] P. Rajakumar, P. M. Balasubramaniam, E. Parimalasundar, K. Suresh, and P. Aravind, "Optimized placement and sizing of solar photovoltaic distributed generation using jellyfish search algorithm for enhanced power system performance.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-08227-4](https://doi.org/10.1038/s41598-025-08227-4).
- [8] P. K. Davoodi, F. Hajizadeh, and M. Rezaei, "Application of the metaheuristic algorithms to quantify the gsi based on the rmr classification.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-14332-1](https://doi.org/10.1038/s41598-025-14332-1).
- [9] H. Y. Başer, T. Evran, and M. A. Cifci, "Machine learning-augmented triage for sepsis: Real-time icu mortality prediction using shap-explained meta-ensemble models.," *Biomedicines*, 2025. DOI: [10.3390/biomedicines13061449](https://doi.org/10.3390/biomedicines13061449).
- [10] M. Abdulkareem, H. S. Aghdasi, P. Salehpour, and M. Zolfy, "Binary secretary bird optimization clustering by novel fitness function based on voronoi diagram in wireless sensor networks.," *Sensors (Basel, Switzerland)*, 2025. DOI: [10.3390/s25144339](https://doi.org/10.3390/s25144339).
- [11] M. Saroughi, O. M. Katipoğlu, V. Kartal, O. Simsek, H. C. Kilinc, and C. B. Pande, "Developing sediment concentration prediction in the euphrates river catchment, türkiye, with a honey badger and coati optimization-based hybrid algorithm.," *Environmental monitoring and assessment*, 2025. DOI: [10.1007/s10661-025-14230-z](https://doi.org/10.1007/s10661-025-14230-z).
- [12] D. Alsamee and R. Ramezani, "Hare escape optimization algorithm with applications in engineering and deep learning.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-10289-3](https://doi.org/10.1038/s41598-025-10289-3).
- [13] M. S. Radwan, S. Hooten, T. V. Vaerenbergh, and P. Bienstman, "Shapley-guided global optimization algorithm with applications in integrated photonics inverse design.," *Optics express*, 2025. DOI: [10.1364/OE.543719](https://doi.org/10.1364/OE.543719).
- [14] M. A. Parvez and I. M. Mehedi, "Data-driven polymer classification using bigru and hybrid metaheuristic optimization algorithms.," *Polymers*, 2025. DOI: [10.3390/polym17141894](https://doi.org/10.3390/polym17141894).
- [15] W. Gtifa, A. Mhaouch, N. Alsharif, T. Althobaiti, and A. Sakly, "Nature-inspired multi-level thresholding integrated with cnn for accurate covid-19 and lung disease classification in chest x-ray images.," *Diagnostics (Basel, Switzerland)*, 2025. DOI: [10.3390/diagnostics15121500](https://doi.org/10.3390/diagnostics15121500).
- [16] S. Agrawal and S. P. Sahu, "Dwt-oefs: Discrete wavelet transform based optimized ensemble feature selection for parkinson's disease severity classification.," *Cognitive neurodynamics*, 2025. DOI: [10.1007/s11571-025-10312-3](https://doi.org/10.1007/s11571-025-10312-3).
- [17] F. H. Rizk, M. E. Mohamed, B. Sameh, A. M. Zaki, M. M. Eid, and E.-S. M. El-kenawy, "Enhancing Student Performance Prediction with Greylag Goose Optimization Algorithm," in *2024 International Telecommunications Conference (ITC-Egypt)*, Jul. 2024, pp. 32–37. DOI: [10.1109/ITC-Egypt61547.2024.10620568](https://doi.org/10.1109/ITC-Egypt61547.2024.10620568).
- [18] A. Atta, D. ElSayad, D. Ezzat, S. Amin, and M. ElGamal, "Efficacy of swarm-based neural networks in automated depression detection.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-09414-z](https://doi.org/10.1038/s41598-025-09414-z).
- [19] X. Zhu, C. Jia, J. Zhao, *et al.*, "An enhanced artificial lemming algorithm and its application in uav path planning.," *Biomimetics (Basel, Switzerland)*, 2025. DOI: [10.3390/biomimetics10060377](https://doi.org/10.3390/biomimetics10060377).
- [20] A. Kathole, S. Patil, D. D. Jadhav, H. Pathak, and A. S. Mirge, "Development of student intent-based educational chatbot system with adaptive and attentive dtn on symmetric convolution approach.," *MethodsX*, 2025. DOI: [10.1016/j.mex.2025.103542](https://doi.org/10.1016/j.mex.2025.103542).

- [21] A. M. Elshewey, A. A. Alhussan, D. S. Khafaga, M. Radwan, E.-S. M. El-Kenawy, and N. Khodadadi, "An enhanced adaptive dynamic metaheuristic optimization algorithm for rainfall prediction depends on long short-term memory.," *PloS one*, 2025. DOI: [10.1371/journal.pone.0317554](https://doi.org/10.1371/journal.pone.0317554).
- [22] M. Dehghani, M. Aly, J. Rodriguez, E. Sheybani, and G. Javidi, "A novel nature-inspired optimization algorithm: Grizzly bear fat increase optimizer.," *Biomimetics (Basel, Switzerland)*, 2025. DOI: [10.3390/biomimetics10060379](https://doi.org/10.3390/biomimetics10060379).
- [23] Y. Liu, M. Fu, C. Jia, *et al.*, "A novel enhanced competition of tribes and cooperation of members algorithm for global optimization.," *PloS one*, 2025. DOI: [10.1371/journal.pone.0324944](https://doi.org/10.1371/journal.pone.0324944).
- [24] P. Yan, J. Zhang, and T. Zhang, "Nature-inspired approach: A novel rat optimization algorithm for global optimization.," *Biomimetics (Basel, Switzerland)*, 2024. DOI: [10.3390/biomimetics9120732](https://doi.org/10.3390/biomimetics9120732).
- [25] R. Liu, R. Fang, T. Zeng, *et al.*, "A novel adaptive sand cat swarm optimization algorithm for feature selection and global optimization.," *Biomimetics (Basel, Switzerland)*, 2024. DOI: [10.3390/biomimetics9110701](https://doi.org/10.3390/biomimetics9110701).
- [26] Z. Zhang, J. Zhu, and F. Nie, "A novel hybrid adaptive differential evolution for global optimization.," *Scientific reports*, 2024. DOI: [10.1038/s41598-024-70731-w](https://doi.org/10.1038/s41598-024-70731-w).
- [27] A. G. Hussien, A. Pop, S. Kumar, F. A. Hashim, and G. Hu, "A novel artificial electric field algorithm for solving global optimization and real-world engineering problems.," *Biomimetics (Basel, Switzerland)*, 2024. DOI: [10.3390/biomimetics9030186](https://doi.org/10.3390/biomimetics9030186).
- [28] W. Li, X. Yang, Y. Yin, and Q. Wang, "A novel hybrid improved rime algorithm for global optimization problems.," *Biomimetics (Basel, Switzerland)*, 2025. DOI: [10.3390/biomimetics10010014](https://doi.org/10.3390/biomimetics10010014).
- [29] V. Chandran and P. Mohapatra, "A novel multi-strategy ameliorated quasi-oppositional chaotic tunicate swarm algorithm for global optimization and constrained engineering applications.," *Heliyon*, 2024. DOI: [10.1016/j.heliyon.2024.e30757](https://doi.org/10.1016/j.heliyon.2024.e30757).
- [30] O. Altay and E. V. Altay, "A novel chaotic transient search optimization algorithm for global optimization, real-world engineering problems and feature selection.," *PeerJ. Computer science*, 2023. DOI: [10.7717/peerj-cs.1526](https://doi.org/10.7717/peerj-cs.1526).
- [31] R. Selvaraj and M. S. G. Devasena, "A novel attention based vision transformer optimized with hybrid optimization algorithm for turmeric leaf disease detection.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-02185-7](https://doi.org/10.1038/s41598-025-02185-7).
- [32] X. Wang and S. Nourmohammadi, "A novel framework for sentiment classification employing bi-gru optimized by enhanced human evolutionary optimization algorithm.," *Scientific reports*, 2025. DOI: [10.1038/s41598-025-01516-y](https://doi.org/10.1038/s41598-025-01516-y).
- [33] G. K. Sahoo, S. Choudhury, R. S. Rathore, and M. Bajaj, "A novel prairie dog-based meta-heuristic optimization algorithm for improved control, better transient response, and power quality enhancement of hybrid microgrids.," *Sensors (Basel, Switzerland)*, 2023. DOI: [10.3390/s23135973](https://doi.org/10.3390/s23135973).
- [34] I. Priyadarshini, "Dendritic growth optimization: A novel nature-inspired algorithm for real-world optimization problems.," *Biomimetics (Basel, Switzerland)*, 2024. DOI: [10.3390/biomimetics9030130](https://doi.org/10.3390/biomimetics9030130).
- [35] M. H. Amiri, N. M. Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm.," *Scientific reports*, 2024. DOI: [10.1038/s41598-024-54910-3](https://doi.org/10.1038/s41598-024-54910-3).
- [36] M. Nemati, Y. Zandi, and A. S. Agdas, "Application of a novel metaheuristic algorithm inspired by stadium spectators in global optimization problems.," *Scientific reports*, 2024. DOI: [10.1038/s41598-024-53602-2](https://doi.org/10.1038/s41598-024-53602-2).
- [37] K. Kalita, J. V. N. Ramesh, L. Cepova, S. B. Pandya, P. Jangir, and L. Abualigah, "Multi-objective exponential distribution optimizer (moedo): A novel math-inspired multi-objective algorithm for global optimization and real-world engineering design problems.," *Scientific reports*, 2024. DOI: [10.1038/s41598-024-52083-7](https://doi.org/10.1038/s41598-024-52083-7).
- [38] Y. Gao, H. Zhang, Y. Duan, and H. Zhang, "A novel hybrid pso based on levy flight and wavelet mutation for global optimization.," *PloS one*, 2023. DOI: [10.1371/journal.pone.0279572](https://doi.org/10.1371/journal.pone.0279572).
- [39] X. Bian, R. Zhang, P. Liu, Y. Xiang, S. Wang, and X. Tan, "Near infrared spectroscopic variable selection by a novel swarm intelligence algorithm for rapid quantification of high order edible blend oil.," *Spectrochimica acta. Part A, Molecular and biomolecular spectroscopy*, 2022. DOI: [10.1016/j.saa.2022.121788](https://doi.org/10.1016/j.saa.2022.121788).

- [40] A. M. Khalid, K. M. Hosny, and S. Mirjalili, "Covidoa: A novel evolutionary optimization algorithm based on coronavirus disease replication lifecycle.," *Neural computing applications*, 2022. DOI: [10.1007/s00521-022-07639-x](https://doi.org/10.1007/s00521-022-07639-x).
- [41] W. Zhang and Y. Lan, "A novel memetic algorithm based on multiparent evolution and adaptive local search for large-scale global optimization.," *Computational intelligence and neuroscience*, 2022. DOI: [10.1155/2022/3558385](https://doi.org/10.1155/2022/3558385).
- [42] P. Trojovský and M. Dehghani, "Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications.," *Sensors (Basel, Switzerland)*, 2022. DOI: [10.3390/s22030855](https://doi.org/10.3390/s22030855).