Journal of Artificial Intelligence and Metaheuristic

♦ ISSN: 2833-5597 Vol: 10, No. 01, 2025 ♦ PP. 37–47

Article DOI: https://doi.org/10.54216/JAIM.100105

FULL LENGTH ARTICLE

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

Ahmed Mohamed Zaki[1,*] **Hala B. Nafea**[1] **Hossam El-Din Moustafa**[1,2]
El-Sayed M. El-Kenawy[3,4,*]

[1] Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[2] Faculty of Artificial Intelligence and Informatics, Horus University, New Damietta 34517, Egypt
[3] Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt
[4] Applied 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

Received: March 09, 2025 Revised: June 08, 2025 Accepted: August 10, 2025 ⋆ Corresponding author

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 the 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 is its adaptive energy-management system, which dynamically regulates exploration–exploitation transitions based on solution-improvement patterns and stagnation detection. Complementary adaptive parameters balance global search and local refinement throughout optimization. The algorithm was evaluated on four engineering design benchmarks: pressure vessel, welded beam, three-bar truss, and cantilever beam. Comparisons with the Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), Harris Hawks Optimization (HHO), and Bat Algorithm (BAT) report that SSO outperforms these methods on the test problems. ANOVA and Wilcoxon signed-rank tests assess the significance of the reported performance differences. The results identify SSO as an effective optimization method, with 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 severe strain on traditional mathematical programming methods. Classical optimization techniques can be ineffective for high-dimensional, multimodal, and discontinuous objective functions. Population-based methods such as adaptive differential evolution and artificial electric-field optimization offer alternatives for global search in these settings [1, 2].

Engineering design optimization seeks feasible solutions that balance competing objectives such as cost, weight, and per-

© 2025 The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

formance. Recent studies address constrained engineering formulations and multi-objective design problems [3, 4].

Metaheuristic optimization algorithms are an important alternative because population-based search can balance global exploration and local search [5, 6, 1]. Inspired by natural phenomena and physical processes, these methods use search rules that can be adapted to different optimization problems [2, 7, 8].

Metaheuristic methods span nature-inspired, physics-based, evolutionary, and swarm-intelligence approaches [5, 1, 2, 7, 8]. Figure 1 summarizes these categories. Recent methods continue to investigate how to balance exploration and exploitation, preserve diversity, and avoid stagnation, with performance depending on the problem and benchmark [6, 1, 3, 4].

The somersaulting spider offers a biological model for addressing these limitations. Its high-energy somersaults enable rapid movement over distance, while rolling allows more precise movement toward a target. These two modes suggest an optimization framework that couples global exploration with local exploitation. The spider's energy-dependent behavioral switching also motivates an adaptive mechanism that responds to solution improvement and stagnation rather than relying only on predefined schedules.

The contributions of this work are as follows:

1. A dual-movement optimization framework inspired by the somersaulting spider, combining somersault-based exploration and rolling-based exploitation.
2. An adaptive energy-control mechanism that guides exploration–exploitation transitions using solution progress and stagnation detection.
3. Complementary adaptive parameters that balance global search diversity and local solution refinement.
4. Experimental evaluation on four engineering design problems against established metaheuristic algorithms.
5. Statistical analyses to evaluate performance differences and robustness across the benchmarks.

The remainder of the paper is organized as follows. Section 2 reviews related optimization algorithms. Section 3 describes SSO, its biological inspiration, mathematical formulation, and algorithmic framework. Section 4 defines the engineering design benchmarks. Section 5 reports comparative and statistical results, and Section 6 concludes and discusses future work.

2. LITERATURE REVIEW

Recent nature-inspired methods model different biological and social processes. Dendritic Growth Optimization uses branching patterns to guide search [7], while the Hippopotamus Optimization Algorithm models three phases of hippopotamus behavior [8]. The Stadium Spectators Optimizer draws on interactions between spectators and players, and the Multi-objective Exponential Distribution Optimizer extends a distribution-based method to multi-objective global and engineering-design problems [9, 4].

The Rat Optimization Algorithm (ROA) models social interactions among rats during prey searching and hunting [5]. An adaptive Sand Cat Swarm Optimization variant has been studied for feature selection and global optimization, using changes to initialization, position updates, and mutation to improve search [6]. Adaptive parameter and dual-mutation strategies have also been introduced in a Differential Evolution framework, together with nonlinear population-size reduction [1].

Physics-motivated methods include the Artificial Electric Field Algorithm (AEFA), based on Coulomb's law and evaluated on global optimization and engineering problems [2]. Hybrid optimization also combines quasi-oppositional learning and chaotic search with the Tunicate Swarm Algorithm (QOCTSA) for global and constrained engineering optimization [3]. The Chaotic Transient Search Optimization algorithm uses chaotic maps and has been evaluated on global optimization, engineering, and feature-selection problems [10].

Metaheuristics have also been applied to engineering applications, including controller tuning and power-quality improvement in hybrid microgrids [11]. Recent work addresses multi-objective global optimization and real-world engineering design, including constrained design cases [3, 4]. Earlier methods include a hybrid Particle Swarm Optimization method, Coronavirus Optimization, a multiparent memetic method, and Pelican Optimization [12, 13, 14, 15].

Other applications include software-cost prediction, construction time–cost planning, photovoltaic parameter estimation and MPPT, microgrid control and scheduling, rock-mass classification, and mechanical engineering design [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Learning-based studies address clinical classification, feature selection, forecasting, neural-network training, student-performance prediction, and sentiment analysis [27, 28, 29, 30, 31, 32, 33]; swarm intelligence has also been used for spectroscopic variable selection [34].

Recent methods include applications and algorithms for optimization across disciplines, the Learning Cooking Algorithm, an enhanced Gorilla Troops Optimizer, and a cooperative heterosis-inspired optimizer [35, 36, 37, 38]. Related work also applies RIME- and coati-based methods to UAV path planning and proposes an enterprise-development-inspired method for structural optimization with frequency constraints [39, 40, 41, 42].

Table 1 compares selected 2024 nature-inspired, physics-based, evolutionary, and hybrid optimization methods, highlighting their target problems and search mechanisms.

Together, these methods illustrate continued efforts to improve convergence reliability, solution diversity, and adaptability in complex optimization tasks.

3. PROPOSED SOMERSAULTING SPIDER OPTIMIZER

The Somersaulting Spider Optimizer (SSO) is a population-based metaheuristic that draws on the locomotion of the desert-dwelling somersaulting spider. Somersaulting supplies high-energy global exploration, while rolling supplies lower-energy local exploitation. An adaptive energy-management system governs transitions between these movements.

3.1 Algorithmic Inspiration

SSO is inspired by the escape mechanism of *Cebrennus rechenbergi*, also known as the somersaulting or flic-flac spider. This arachnid, discovered in the Moroccan desert, per-

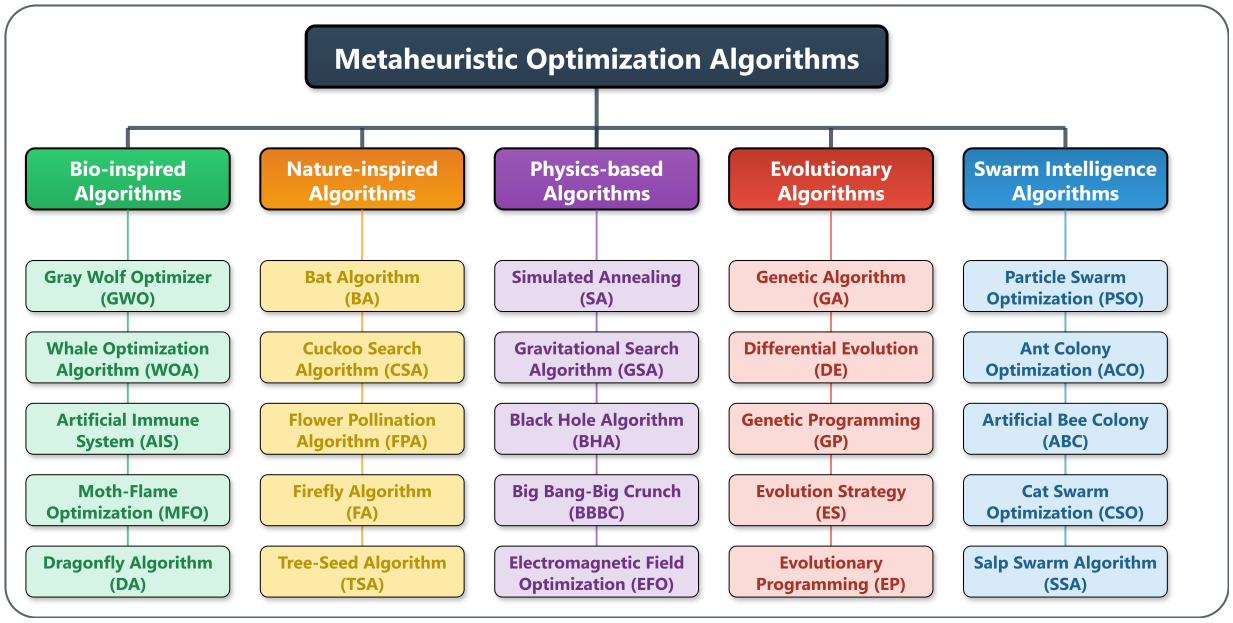


Figure 1. Classification of metaheuristic optimization algorithms, including bio-inspired, nature-inspired, physics-based, evolutionary, and swarm-intelligence methods.



Figure 2. Locomotion behavior of *Cebrennus rechenbergi* spiders demonstrating the somersaulting mechanism that inspired SSO.

forms acrobatic somersaults when threatened and has been reported to move across dunes at speeds of up to 2 m/s. Figure 2 illustrates the locomotion behavior that motivates SSO's adaptive balance between somersaulting and rolling.

The exploration and exploitation factors at iteration t are defined as

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

where $T_{\max}$ is the maximum number of iterations. Thus, $\alpha_{\text{exp}}(t) + \alpha_{\text{expl}}(t) = 1$. When spider i discovers a better solution, its energy is updated as

$$E_i^{t+1} = \min\left(1.0, E_i^t + 0.1\right). \tag{2}$$

Otherwise, its energy decreases according to

$$E_i^{t+1} = 0.95 E_i^t, \tag{3}$$

where E_i^t is the energy level at iteration t.

3.2 Exploration Capability

Somersaulting enables long-distance movements through rotational and translational components. The updated coordinate of spider i in dimension j, directed toward a target $\mathbf{x}_{\text{target}}$, is

$$x_{i,j}^{\text{new}} = x_{i,j}^{\text{current}} + \phi_j (ub_j - lb_j)\,0.1 + \tau_j \left(x_{\text{target},j} - x_{i,j}^{\text{current}}\right), \tag{4}$$

with rotational and translational terms

$$\phi_j = I_s \sin(2\pi r_1), \tag{5}$$

$$\tau_j = 0.5 + 0.5 r_2, \tag{6}$$

where I_s is the somersaulting intensity, $r_1, r_2 \sim U(0,1)$, and ub_j and lb_j are the upper and lower bounds for dimension j. The behavioral threshold is

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

If stagnation lasts more than five iterations, SSO selects a random target and uses $I_s = 0.8\alpha_{\text{exp}}(t)$. Otherwise, it directs the movement toward the global best with $I_s = 0.6\alpha_{\text{exp}}(t)$.

Algorithm 1. Somersaulting movement for exploration

1. Require spider position $\mathbf{x}_i$, energy E_i, stagnation counter s_i, and bounds $[lb, ub]$; return updated position $\mathbf{x}_i^{\text{new}}$.
2. Calculate $\theta_{\text{behav}} = \alpha_{\text{exp}}(t) E_i^t$.
3. Draw random numbers $r_1, r_2 \sim U(0,1)$.
4. If $s_i > 5$, set the somersault intensity to $I_s = 0.8\alpha_{\text{exp}}(t)$.
5. Generate a random target $\mathbf{x}_{\text{target}} \sim U(lb, ub)$.
6. Otherwise, set $I_s = 0.6\alpha_{\text{exp}}(t)$.
7. Set the target to the global best, $\mathbf{x}_{\text{target}} = \mathbf{x}_{\text{best}}$.
8. For each dimension $j = 1, \ldots, D$, calculate $\phi_j = I_s \sin(2\pi r_1)$.
9. Calculate $\tau_j = 0.5 + 0.5 r_2$.
10. Update $x_{i,j}$ using the somersaulting movement equa-

Table 1. Novel bio-inspired and nature-inspired optimization algorithms.

Ref.	Main focus	Methodology	Key findings
[7]	Global optimization	Dendritic Growth Optimization (DGO), inspired by branching patterns	Uses a nature-inspired search strategy and evaluates it on real-world optimization problems.
[8]	Global optimization and engineering design	Hippopotamus Optimization Algorithm (HO)	Models three phases of hippopotamus behavior and evaluates engineering-design problems.
[9]	Global optimization	Stadium Spectators Optimizer	Models spectator influence and evaluates the method on benchmark functions.
[4]	Multi-objective global optimization and engineering design	Multi-objective Exponential Distribution Optimizer (MOEDO)	Uses non-dominated sorting, crowding distance, and information feedback.
[5]	Global optimization	Rat Optimization Algorithm (ROA), based on rat social behavior	Models prey searching and hunting behavior.
[6]	Feature selection and global optimization	Adaptive Sand Cat Swarm Optimization	Combines initialization, position-update, and mutation strategies for global search and feature selection.
[1]	Global optimization	Adaptive Differential Evolution with dual mutation strategies	Uses adaptive parameter switching and nonlinear population-size reduction.
[2]	Global optimization and engineering problems	Artificial Electric Field Algorithm (AEFA)	Applies an electric-field-inspired search to benchmark and engineering problems.
[3]	Global and constrained engineering optimization	Quasi-oppositional Chaotic Tunicate Swarm Algorithm (QOCTSA)	Combines quasi-opposition and chaotic search strategies.

tion.
11. Enforce the boundary constraints on $\mathbf{x}_i^{\text{new}}$.
12. Return $\mathbf{x}_i^{\text{new}}$.

3.3 Exploitation Capability

Rolling performs short-distance local search in promising regions. For dimension j of spider i, the movement is

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

where the rolling radius and random variable are

$$R_j = \alpha_{\text{expl}}(t)(ub_j - lb_j)0.05, \qquad r_3 \sim U(0,1). \tag{9}$$

When energy falls below 0.2, a conservative adjustment moves the spider toward the global best:

$$x_{i,j}^{\text{adjusted}} = 0.9x_{i,j}^{\text{current}} + 0.1x_{\text{best},j}. \tag{10}$$

After this adjustment, the energy is reset to 1.0.

Algorithm 2. Rolling movement for exploitation

1. Require spider position $\mathbf{x}_i$, energy E_i, global best $\mathbf{x}_{\text{best}}$, and bounds $[lb, ub]$; return updated position $\mathbf{x}_i^{\text{new}}$.
2. Draw a random number $r_3 \sim U(0,1)$.
3. If $E_i < 0.2$, process each dimension $j = 1, \ldots, D$.
4. Set $x_{i,j}^{\text{adjusted}} = 0.9x_{i,j} + 0.1x_{\text{best},j}$.
5. End the dimension loop and reset the spider energy to $E_i = 1.0$.
6. Otherwise, for each dimension $j = 1, \ldots, D$, calculate $R_j = \alpha_{\text{expl}}(t)(ub_j - lb_j)0.05$.
7. Apply the rolling update $x_{i,j}^{\text{new}} = x_{i,j} + R_j \cos(2\pi r_3)$.
8. End the dimension loop, enforce boundary constraints, and return $\mathbf{x}_i^{\text{new}}$.

3.4 Complete SSO Algorithm Framework

The complete algorithm coordinates somersaulting and rolling through adaptive energy management, while tracking the global best and stagnation of each spider.

Algorithm 3. Somersaulting Spider Optimizer (SSO)

1. Require population size N, maximum iterations $T_{\max}$, dimension D, and bounds $[lb, ub]$; return $\mathbf{x}_{\text{best}}$ and f_{best}.
2. Initialize population $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ randomly within the bounds.
3. Initialize every spider's energy to 1.0.
4. Initialize all stagnation counters to zero.
5. Evaluate the population and identify the current best position and fitness.
6. For $t = 1, \ldots, T_{\max}$, calculate $\alpha_{\text{exp}}(t) = 1 - t/T_{\max}$.
7. Calculate $\alpha_{\text{expl}}(t) = t/T_{\max}$.
8. For each spider $i = 1, \ldots, N$, calculate $\theta_{\text{behav}} = \alpha_{\text{exp}}(t)E_i^t$.
9. If a random draw is below θ_{behav}, select the exploration movement.
10. Apply somersaulting movement using Algorithm 1.
11. Otherwise, select the exploitation movement.
12. Apply rolling movement using Algorithm 2.
13. End the movement selection.
14. Evaluate the fitness at the updated position.
15. If fitness improves, update the spider's energy.
16. Set $E_i^{t+1} = \min(1.0, E_i^t + 0.1)$.
17. Reset its stagnation counter to zero.
18. Update $\mathbf{x}_{\text{best}}$ and f_{best} if the new solution is globally best.
19. Otherwise, decrease the energy.
20. Set $E_i^{t+1} = 0.95E_i^t$.
21. Increment the spider's stagnation counter.
22. End the fitness-improvement test.
23. End the loop over spiders.
24. End the loop over iterations.
25. Return $\mathbf{x}_{\text{best}}$ and f_{best}.

The adaptive energy mechanism trades off exploration and exploitation. Somersaulting and rolling provide complementary search movements, while energy updates and stagnation counters regulate switching and support convergence.

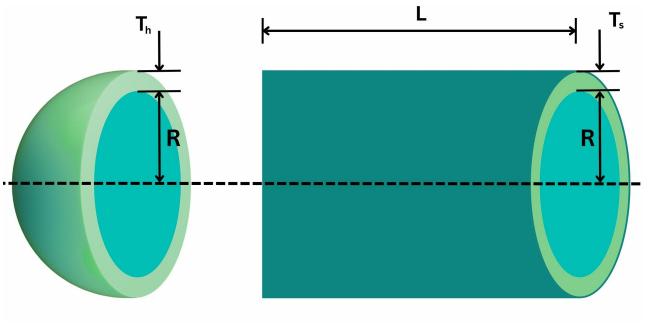


Figure 3. Pressure vessel design.

4. BENCHMARK ENGINEERING DESIGN PROBLEMS

4.1 Pressure Vessel Design Problem

The pressure-vessel problem minimizes material, forming, and welding costs for a cylindrical vessel with hemispherical heads. The decision variables are shell thickness T_s, head thickness T_h, inner radius R, and cylindrical-section length L; the thicknesses are integer multiples of available steel plate thickness.

$$f(T_s,T_h,R,L) = 0.6224T_sRL + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_h^2L. \tag{11}$$

The design constraints are

$$g_1 = -T_s + 0.0193R \le 0, \tag{12}$$
$$g_2 = -T_h + 0.0095R \le 0, \tag{13}$$
$$g_3 = -\pi R^2 L - \frac{4}{3}\pi R^3 + 1{,}296{,}000 \le 0, \tag{14}$$
$$g_4 = L - 240 \le 0, \tag{15}$$

with bounds

$$0 \le T_s, T_h \le 99, \tag{16}$$
$$10 \le R, L \le 200. \tag{17}$$

4.2 Welded Beam Design Problem

The welded-beam problem minimizes fabrication cost subject to shear, bending, buckling, and deflection constraints. Its design variables are weld thickness w, beam length L, beam depth d, and height h.

$$f(w,L,d,h) = 1.10471w^2L + 0.04811dh(14+L). \tag{18}$$

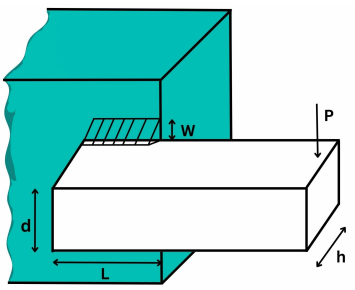


Figure 4. Welded beam design.

The constraints are

$$g_1 = w - h \le 0, \tag{19}$$
$$g_2 = \delta - 0.25 \le 0, \tag{20}$$
$$g_3 = \tau - 13{,}600 \le 0, \tag{21}$$
$$g_4 = \sigma - 30{,}000 \le 0, \tag{22}$$
$$g_5 = 0.125 - w \le 0, \tag{23}$$
$$g_6 = 6000 - P \le 0, \tag{24}$$
$$g_7 = 0.10471w^2 + 0.04811hd(14+L) - 0.5 \le 0, \tag{25}$$

with bounds

$$0.1 \le w, h \le 2.0, \tag{26}$$
$$0.1 \le L, d \le 10.0. \tag{27}$$

The auxiliary variables are

$$\delta = \frac{65856}{30000hD^3}, \qquad \tau = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2},$$
$$\alpha = \frac{6000}{\sqrt{2}wL}, \qquad \beta = \frac{QD}{J},$$
$$Q = 6000\left(14 + \frac{L}{2}\right), \qquad D = \frac{1}{2}\sqrt{L^2 + (w+d)^2},$$
$$J = \sqrt{2}wL\left(\frac{L^2}{6} + \frac{(w+d)^2}{2}\right), \qquad \sigma = \frac{504000}{hd^2},$$
$$P = \frac{0.614232 \times 10^6 dh^3}{6} \times \left(1 - \frac{d\sqrt{30/48}}{28}\right).$$

4.3 Three-Bar Truss Design Problem

The three-bar truss is a classic structural optimization problem that minimizes truss mass subject to stress, displacement, and geometric constraints. Such structures are used in bridges, towers, and roof systems, where optimization can improve stability, reduce material use, and lower construction costs. The decision variables are the edge-rod and central-rod areas, $\mathbf{x} = [x_1, x_2] = [A_1, A_2]$.

$$f(\mathbf{x}) = \left(2\sqrt{2}x_1 + x_2\right)\ell. \tag{28}$$

The constraints are

$$g_1(\mathbf{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0, \tag{29}$$
$$g_2(\mathbf{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0, \tag{30}$$
$$g_3(\mathbf{x}) = \frac{1}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0, \tag{31}$$

with $0 \le x_i \le 99$ for $i = 1,2$, and constants $\ell = 100$, $P = 2$, and $\sigma = 2$.

4.4 Cantilever Beam Design Problem

The cantilever-beam problem minimizes beam weight subject to stress, deflection, and geometric constraints. Cantilever beams occur in architecture, mechanical engineering, and

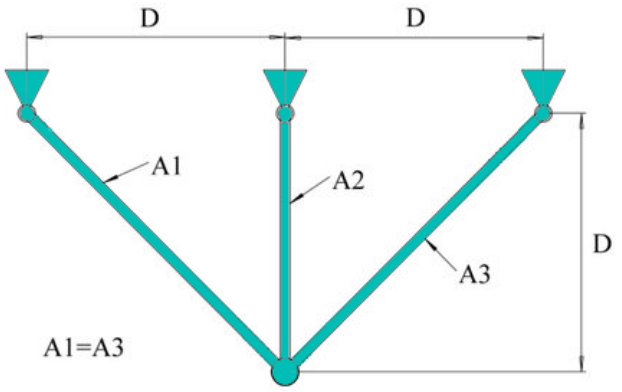


Figure 5. Three-bar truss design.

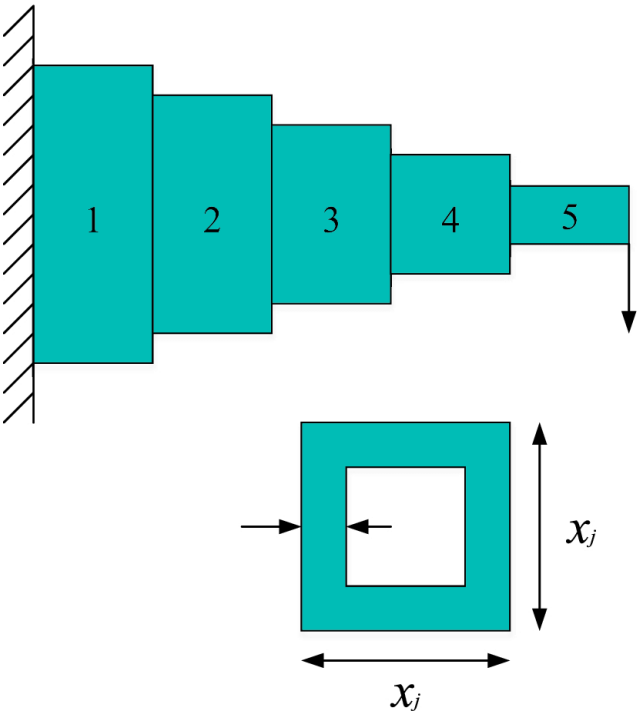


Figure 6. Cantilever beam design, showing the five beam segments and cross-section dimensions.

aerospace applications, including aircraft wings, bridges, and robotic arms. The design uses five segments with rectangular cross-sections and variables $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$.

$$f(\mathbf{x}) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5). \tag{32}$$

The constraint and variable bounds are

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

$$0.01 \leq x_i \leq 100, \qquad i = 1, 2, 3, 4, 5.$$

5. EXPERIMENTAL RESULTS

SSO is compared with the Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), Harris Hawks Optimization (HHO), and Bat Algorithm (BAT) on the four engineering design benchmarks. The evaluation considers solution quality, convergence behavior, execution time, and statistical significance.

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

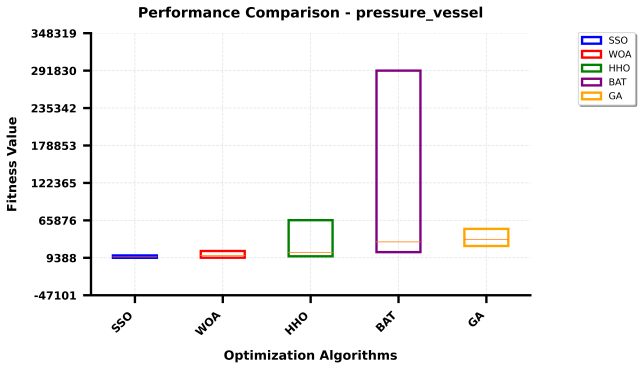


Figure 7. Performance comparison of optimization algorithms for pressure-vessel design using boxplots.

5.1 Benchmark Engineering Design Problem Results

For the pressure-vessel problem, SSO achieves the lowest reported cost (11,032.94), while BAT has the highest (165,486.64). Table 2 gives the design variables and costs. The boxplot in Figure 7 shows SSO with lower fitness values, while BAT has the widest spread and highest median. Figure 8 shows the convergence profiles; the source reports rapid convergence for SSO and a final fitness near -310.

For the welded-beam problem, SSO reports a minimum cost of 2.19, with $w = 0.76$ and $L = 5.96$. GA reports the next-lowest cost (2.65), and WOA the highest (3.64). Table 3 lists the design variables. Figure 9 shows the distributions across runs, and Figure 10 shows the convergence behavior; the source reports an SSO fitness near -2.9.

On the three-bar truss problem, SSO achieves a cost of 263.96 with $A_1 = 0.69$ and $A_2 = 0.74$; GA is second at 265.18, and the differences among methods are relatively small. Table 4 reports the results. Figure 11 shows the SSO distribution with a median near 700, a minimum near 600, and an upper quartile near 800. Figure 12 shows rapid convergence to approximately 0.7 after 10 iterations and 0.5 after 30 iterations, followed by a plateau.

For the cantilever beam, SSO obtains a cost of 1.64 and HHO a cost of 1.68; BAT reports the highest cost (6.76). Table 5 lists the design variables. Figure 13 compares the distributions; the source describes the SSO median fitness as approximately 0.001 with minimal variation. Figure 14 shows near-optimal fitness early in the search and stable subsequent performance.

5.2 Comprehensive Evaluation of Overall Performance

Table 6 summarizes the reported mean costs across the four benchmarks. SSO has the lowest cost in each problem, including 11,032.94 for the pressure vessel. The standard deviations in Table 7 show an SSO standard deviation of 0.054 on the

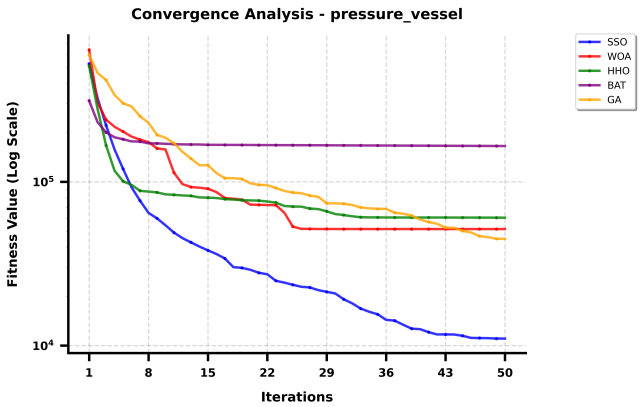


Figure 8. Convergence analysis of optimization algorithms for pressure-vessel design.

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

three-bar truss, indicating stable results there. Table 8 shows WOA has the shortest execution time for the three-bar truss (0.074 s). Figure 15 compares the algorithms; the source identifies GA as fastest overall, followed by WOA and SSO,

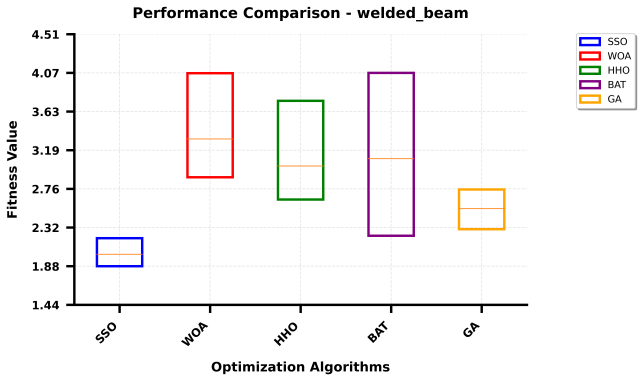


Figure 9. Performance comparison of optimization algorithms for welded-beam design: boxplot analysis.

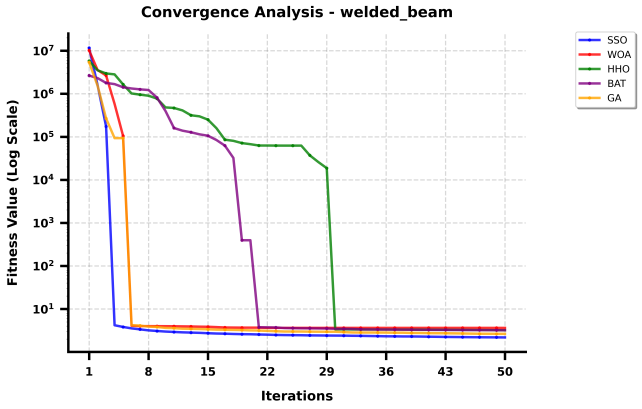


Figure 10. Convergence analysis of optimization algorithms for welded-beam design.

Table 4. Three-bar truss design optimization results.

Algorithm	A_1	A_2	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

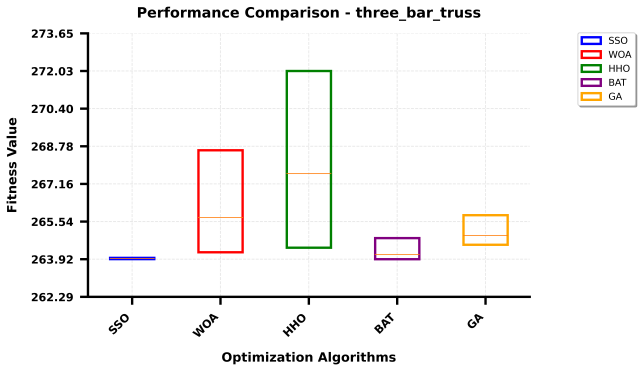


Figure 11. Performance comparison of optimization algorithms on the three-bar truss problem using boxplots.

and describes SSO as balancing solution quality and computational efficiency.

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. 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. 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

The execution-time comparison is shown in Figure 15. The source identifies GA as fastest overall, followed by WOA and SSO, and notes the trade-off between runtime and solution quality.

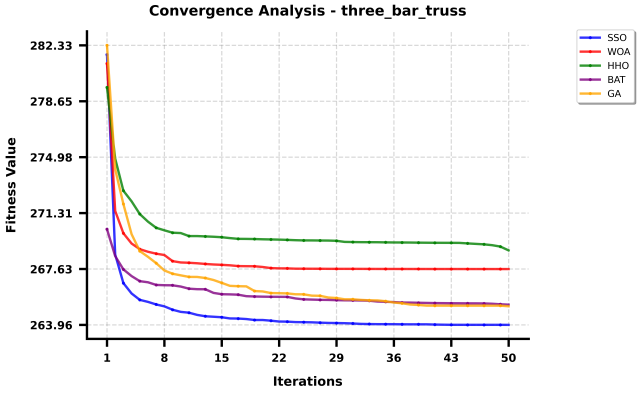


Figure 12. Convergence analysis of the three-bar truss using optimization algorithms.

Table 5. Cantilever beam design optimization results.

Algorithm	x_1	x_2	x_3	x_4	x_5	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

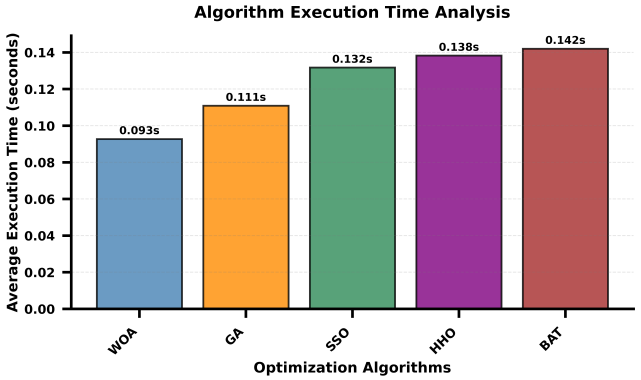


Figure 15. Average execution-time comparison of optimization algorithms.

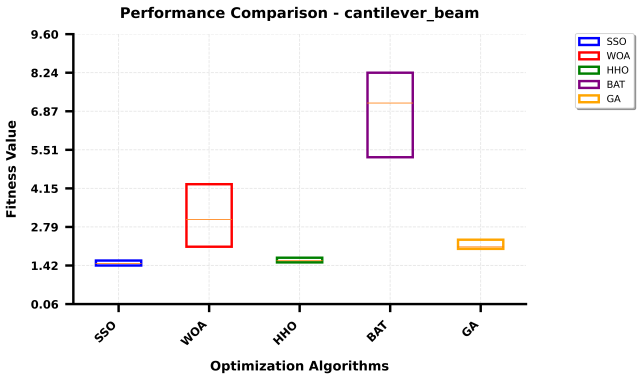


Figure 13. Performance comparison of optimization algorithms for cantilever-beam design.

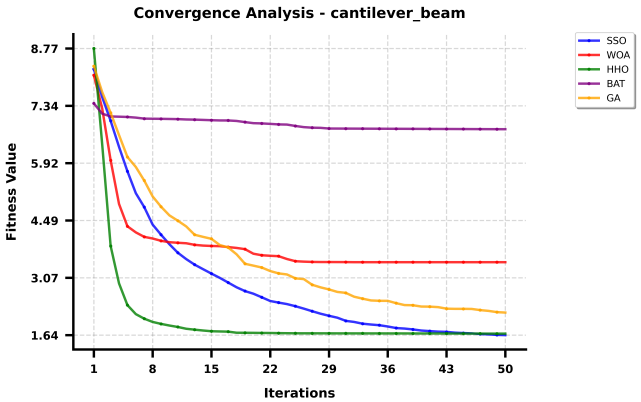


Figure 14. Convergence analysis of optimization algorithms for cantilever-beam design.

5.3 Statistical Analysis

The ANOVA results in Table 9 give an F-statistic of 5.52 and a p-value of 0.0002, indicating statistically significant differences between algorithms. Table 10 reports Wilcoxon signed-rank summary statistics, with SSO's mean reported as 2,825.1814 and significance marked at $p < 0.001$. Table 11 lists the per-problem ANOVA p-values. Figures 16 and 17 reproduce the source's further statistical comparisons.

Table 9. ANOVA results for GA, HHO, BAT, WOA, and SSO.

Source	Sum of squares	DF	Mean square	F-statistic	p-value	Sig.
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. Wilcoxon signed-rank test results for algorithm performance.

Algorithm	Sample size	Median	Mean	Std. dev.	Sig.	Sig. 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. 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

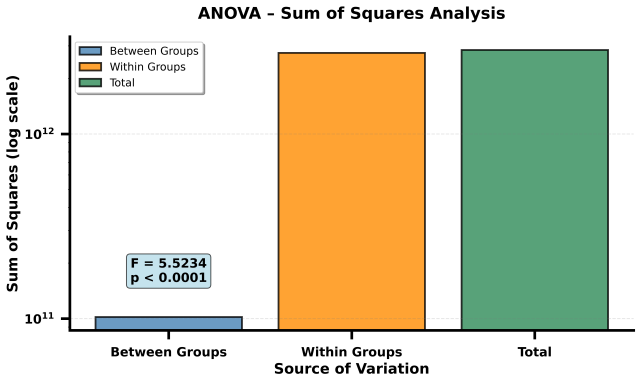


Figure 16. ANOVA sum-of-squares comparison for metaheuristic algorithms.

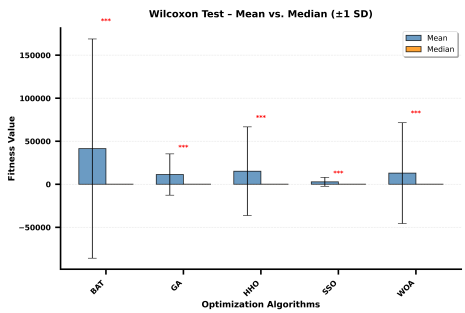


Figure 17. Performance analysis comparing mean, median, and standard deviation.

The results indicate that SSO consistently outperforms the comparison algorithms across the reported engineering benchmarks in solution quality, with statistical tests supporting significant performance differences.

6. CONCLUSION AND FUTURE WORK

This study presents the Somersaulting Spider Optimizer, based on the locomotion of the somersaulting spider. Its dual movements balance global exploration through somersaulting and local refinement through rolling. The adaptive energy-management system controls behavioral transitions according to solution-improvement patterns and stagnation detection. Experiments on pressure-vessel, welded-beam, three-bar truss, and cantilever-beam problems report lower optimization costs for SSO than for GA, WOA, HHO, and BAT. ANOVA and Wilcoxon signed-rank results support the reported performance differences. The complementary adaptive parameters and limited need for hyperparameter tuning are presented as useful properties for engineering applications.

Future work may extend SSO to multi-objective optimization using Pareto dominance and archive management, and to discrete and binary problems such as feature selection, scheduling, and graph optimization. Further theoretical study of convergence, computational complexity, and the no-free-lunch implications could clarify the algorithm's properties and limits. Hybridization with reinforcement learning or machine-learning-based adaptive parameter control may improve performance on selected problem classes. Large-scale optimization may use population decomposition and parallel implementations for high-dimensional problems with populations of thousands. Proposed application areas include renewable-energy optimization, autonomous-vehicle routing, supply-chain management, and neural-architecture search. Sensitivity analysis of energy-management, stagnation, and movement-intensity parameters, together with self-adaptive variants, could guide or automate parameter selection.

REFERENCES

[1] Z. Zhang, J. Zhu, and F. Nie, "A novel hybrid adaptive differential evolution for global optimization," *Scientific Reports*, vol. 14, p. 19697, 2024.

[2] 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*, vol. 9, no. 3, p. 186, 2024.

[3] V. Chandran and P. Mohapatra, "A novel multi-strategy ameliorated quasi-oppositional chaotic tunicate swarm algorithm for global optimization and constrained engineering applications," *Heliyon*, vol. 10, no. 10, p. e30757, 2024.

[4] 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*, vol. 14, p. 1816, 2024.

[5] P. Yan, J. Zhang, and T. Zhang, "Nature-inspired approach: A novel rat optimization algorithm for global optimization," *Biomimetics*, vol. 9, no. 12, p. 732, 2024.

[6] R. Liu, R. Fang, T. Zeng *et al.*, "A novel adaptive sand cat swarm optimization algorithm for feature selection and global optimization," *Biomimetics*, vol. 9, no. 11, p. 701, 2024.

[7] I. Priyadarshini, "Dendritic growth optimization: A novel nature-inspired algorithm for real-world optimization problems," *Biomimetics*, vol. 9, no. 3, p. 130, 2024.

[8] M. H. Amiri, N. Mehrabi Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm," *Scientific Reports*, vol. 14, p. 5032, 2024.

[9] M. Nemati, Y. Zandi, and A. S. Agdas, "Application of a novel metaheuristic algorithm inspired by stadium spectators in global optimization problems," *Scientific Reports*, vol. 14, p. 3078, 2024.

[10] 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*, vol. 9, p. e1526, 2023.

[11] 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*, vol. 23, no. 13, p. 5973, 2023.

[12] 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*, vol. 18, no. 1, p. e0279572, 2023.

[13] A. M. Khalid, K. M. Hosny, and S. Mirjalili, "COVIDOA: A novel evolutionary optimization algorithm based on coronavirus disease replication lifecycle," *Neural Computing and Applications*, vol. 34, no. 24, pp. 22 465–22 492, 2022.

[14] 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*, vol. 2022, p. 3558385, 2022.

[15] P. Trojovsky and M. Dehghani, "Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications," *Sensors*, vol. 22, no. 3, p. 855, 2022.

[16] M. M. Draz, O. Emam, and S. M. Azzam, "Software cost estimation prediction using a convolutional neural network and particle swarm optimization algorithm," *Scientific Reports*, vol. 14, p. 13129, 2024.

[17] V. H. S. Pham, N. T. Nguyen Dang, and N. V. Nam, "Optimizing time and cost in construction projects with a hybridized multi-verse optimizer and opposition-based learning," *Engineering, Construction and Architectural Management*, vol. 32, no. 7, pp. 4852–4886, 2024.

[18] R. Mohamed, M. Abdel-Basset, K. M. Sallam, I. M. Hezam, A. M. Alshamrani, and I. A. Hameed, "Novel hybrid kepler optimization algorithm for parameter estimation of photovoltaic modules," *Scientific Reports*, vol. 14, p. 3453, 2024.

[19] H. Zhang, X. Wang *et al.*, "Mppt control of photovoltaic array based on improved marine predator algorithm under complex solar irradiance conditions," *Scientific Reports*, vol. 14, p. 19745, 2024.

[20] H. Rezk, A. Ghani Olabi, T. Wilberforce, and E. Taha Sayed, "Metaheuristic optimization algorithms for real-world electrical and civil engineering application: A review," *Results in Engineering*, vol. 23, p. 102437, 2024.

[21] A. Mathur, R. Kumari, V. P. Meena, V. P. Singh, A. T. Azar, and I. A. Hameed, "Data-driven optimization for microgrid control under distributed energy resource variability," *Scientific Reports*, vol. 14, p. 10806, 2024.

[22] N. Karthik, A. Rajagopalan, M. Bajaj *et al.*, "Chaotic self-adaptive sine cosine multi-objective optimization algorithm to solve microgrid optimal energy scheduling problems," *Scientific Reports*, vol. 14, p. 18997, 2024.

[23] L. Li, "Lightgbm integration with modified data balancing and whale optimization algorithm for rock mass classification," *Scientific Reports*, vol. 14, p. 23028, 2024.

[24] N. Aditya and S. S. Mahapatra, "An adaptive gravitational search algorithm for optimizing mechanical engineering design and machining problems," *Engineering Applications of Artificial Intelligence*, vol. 138, no. Part A, p. 109298, 2024.

[25] Y. Chun, X. Hua, C. Qi, and Y. X. Yao, "Improved marine predators algorithm for engineering design optimization problems," *Scientific Reports*, vol. 14, p. 13000, 2024.

[26] E. H. Houssein, M. K. Saeed, M. M. Al-Sayed, and G. Hu, "Metaheuristics for solving global and engineering optimization problems: Review, applications, open issues and challenges," *Archives of Computational Methods in Engineering*, vol. 31, pp. 4485–4519, 2024.

[27] A. Bilal, A. Alzahrani, H. Long *et al.*, "Advanced ckd detection through optimized metaheuristic modeling in healthcare informatics," *Scientific Reports*, vol. 14, p. 12601, 2024.

[28] J. L. Potharlanka *et al.*, "Feature importance feedback with deep q process in ensemble-based metaheuristic feature selection algorithms," *Scientific Reports*, vol. 14, p. 2923, 2024.

[29] B. Gulsun and M. R. Aydin, "Optimizing a machine learning algorithm by a novel metaheuristic approach: A case study in forecasting," *Mathematics*, vol. 12, no. 24, p. 3921, 2024.

[30] S. M. Azzam, O. E. Emam, and A. S. Abolaban, "An improved differential evolution with sailfish optimizer for handling feature selection problem," *Scientific Reports*, vol. 14, p. 13517, 2024.

[31] F. Berchez-Moreno *et al.*, "A memetic dynamic coral reef optimisation algorithm for simultaneous training, design, and optimisation of artificial neural networks," *Scientific Reports*, vol. 14, p. 6961, 2024.

[32] 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," *Proceedings of the 2024 International Telecommunications Conference (ITC-Egypt)*, pp. 32–37, 2024.

[33] M. S. Rani and S. Subramanian, "Meta-heuristic approaches for sentiment analysis," *Expert Systems*, vol. 41, no. 8, p. e13592, 2024.

[34] 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*, vol. 284, p. 121788, 2023.

[35] E. H. Cui, Z. Zhang, C. J. Chen *et al.*, "Applications of nature-inspired metaheuristic algorithms for tackling optimization problems across disciplines," *Scientific Reports*, vol. 14, p. 9403, 2024.

[36] S. Gopi and P. Mohapatra, "Learning cooking algorithm for solving global optimization problems," *Scientific Reports*, vol. 14, p. 13359, 2024.

[37] M. H. Hassan, S. Kamel, and A. W. Mohamed, "Enhanced gorilla troops optimizer powered by marine predator algorithm for global optimization and engineering design," *Scientific Reports*, vol. 14, p. 7650, 2024.

[38] T. Cai, S. Zhang, Z. Ye, W. Zhou, M. Wang, Q. He, Z. Chen, and W. Bai, "Cooperative metaheuristic algorithm for global optimization and engineering problems inspired by heterosis theory," *Scientific Reports*, vol. 14, p. 28876, 2024.

[39] X. Wang, Y. Feng, J. Tang, Z. Dai *et al.*, "A uav path planning method based on the framework of multi-objective jellyfish search algorithm," *Scientific Reports*, vol. 14, p. 28058, 2024.

[40] D.-N. Truong and J.-S. Chou, "Metaheuristic algorithm inspired by enterprise development for global optimization and structural engineering problems with frequency constraints," *Engineering Structures*, vol. 318, p. 118679, 2024.

[41] C. Xie, S. Li, X. Qin, S. Fu *et al.*, "Multiple elite strategy enhanced rime algorithm for 3d uav path planning," *Scientific Reports*, vol. 14, p. 21734, 2024.

[42] C. Jia, L. He, D. Liu, and S. Fu, "Path planning and engineering problems of 3d uav based on adaptive coati optimization algorithm," *Scientific Reports*, vol. 14, p. 30717, 2024.