e-ISSN : 2600-8793

Available online at
https://jcrinn.com/
https://crinn.conferencehunter.com/

Journal of Computing Research and Innovation 11(2) 2026

Journal of Computing Research and Innovation

Adaptive Elite Cuckoo Search Algorithm with Step-Size Control and Elitism Crossover for Global Optimization

Ong Pauline[1*], Wu Jia Hang[2]

[1]*Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia.*
[2]*Department of Mechanical Engineering, Politeknik Kuching Sarawak, KM 22, Jalan Matang, 93050, Kuching, Sarawak, Malaysia.*

ARTICLE INFO

Article history:
Received 4 March 2026
Revised 15 April 2026
Accepted 21 May 2026
Online first
Published 1 September 2026

Keywords:
Adaptive Step-Size
Crossover
Cuckoo Search Algorithm
Elitism
Metaheuristic Algorithm
Optimization

DOI:
10.24191/jcrinn.v11i2.609

ABSTRACT

Cuckoo Search Algorithm (CSA) has emerged as a powerful bio-inspired metaheuristic, yet its reliance on random-walk search often leads to slow convergence and susceptibility to local optima. This study introduces an Adaptive Elite Cuckoo Search Algorithm (AECSA), designed to enhance both convergence speed and solution precision. The proposed modifications integrate two key mechanisms: (i) an adaptive step-size strategy that dynamically adjusts the Lévy flight scale to balance global exploration and local exploitation, and (ii) an elitist crossover scheme that enables information sharing among the top-performing solutions to accelerate progress toward the global optimum. The AECSA was validated across six classical benchmark functions under extensive simulations. Results demonstrate that AECSA consistently outperforms the standard CSA, achieving up to 85% faster convergence and significantly improved solution accuracy, with statistical validation via a two-tailed t-test. These findings highlight the robustness and efficiency of AECSA, making it a promising approach for tackling complex, high-dimensional optimization problems in engineering and computational intelligence.

1. INTRODUCTION

Optimization underpins a wide spectrum of scientific and engineering applications, including feature extraction, scheduling, manufacturing and path planning (Begum & Valan, 2026; Mechaacha et al., 2026; Wu et al., 2026; Yang et al., 2025). Yet, real-world problems are rarely static; they are often characterized by uncertainty, dynamic constraints, computational expensive and high-dimensional search spaces. Such complexities render many classical optimization techniques inadequate, as their rigid structures prevent effective adaptation to changing problem landscapes (Alibabaei Shahraki, 2025). Moreover, the reliance of gradient-based approaches on derivative information further limits their applicability. In highly nonlinear or ill-posed problems, generating accurate derivatives is either computationally expensive or infeasible, leading to poor convergence and suboptimal solutions (Walton et al., 2011). These limitations highlight the

[1*] Corresponding author. *E-mail address*: ongp@uthm.edu.my
https://doi.org/10.24191/jcrinn.v11i2.609

need for flexible, derivative-free optimization frameworks capable of maintaining performance across diverse problem settings.

Bio-inspired metaheuristic algorithms have emerged as a promising alternative, offering adaptability and robustness by mimicking natural phenomena. Genetic Algorithms (Holland, 1992), grounded in evolutionary theory, Ant Colony Optimization (Dorigo et al., 1991), inspired by pheromone-based foraging, and Carnivorous Plant Algorithm (Ong et al., 2021), motivated by adaptive survival strategies of carnivorous plants in resource-limited environments, exemplify this class of approaches. Their rule-based mechanisms allow efficient exploration and exploitation without explicit gradient information, making them well-suited for complex and dynamic problem spaces.

Within this paradigm, the Cuckoo Search Algorithm (CSA) (Yang & Deb, 2010) has gained attention for its simplicity and effectiveness. Modelled on the brood parasitism behaviour of cuckoos, CSA employs Lévy flight random walks and selection strategies to balance exploration and exploitation. Since its introduction, CSA has been successfully applied across diverse fields, including scheduling (Pavithra & Rekha, 2025), feature extraction (Shi et al., 2025), image segmentation (Subha & Kumaran, 2025) and financial risk prediction (Cai, 2025), to name a few. Despite its successes, CSA remains challenged by slow convergence, weak global exploration, and insufficient exploitation of population knowledge, resulting in premature convergence (Ouyang et al., 2025).

To address these limitations, extensive enhancements to CSA have been proposed. Choudhary and Rajak (Choudhary & Rajak, 2026) introduced an egg-laying radius mechanism to guide solution generation toward an ideal habitat, improving directed exploration. Ouyang et al. (Ouyang et al., 2025) incorporated entropy-based selection, adaptive global search, and hierarchical nest replacement to accelerate convergence and mitigate premature stagnation. Subha and Kumaran (2025) strengthened exploitation by integrating opposition-based learning. Further improvements include chaotic and adaptive control mechanisms, such as integrating logistic maps (Dar et al., 2025), reinforcement learning (Yu et al., 2026), adaptive step-size regulation (Tian et al., 2025), inertia-weight reduction (Safdar et al., 2025), dynamic switching parameters (Mimansha & Kumar, 2025), migration operators (BahraniPour et al., 2025), and adaptive termination strategies (Acharjee & Chaudhuri, 2023). Another line of research emphasizes elite-guided optimization (Tian et al., 2024; Yang et al., 2023), where high-quality solutions are explicitly exploited to accelerate convergence. In addition, population-level structural enhancements, such as diffusion mechanisms (Chen et al., 2023a), Voronoi-based partitioning (Maddaiah & Narayanan, 2023), and dynamic neighbourhood search (Chen et al., 2023b), have been explored to further improve search efficiency and diversity. In parallel, hybrid CSA variants combining complementary metaheuristics, such as Ant Colony Optimization (Yin et al., 2025), Naked Mole Rat Algorithm (Salgotra et al., 2025), Bat Algorithm, Grey Wolf Algorithm (Garg et al., 2025), Simulated Annealing (Saleh et al., 2025), Whale Optimization Algorithm (Mazari et al., 2025) and Big Bang-Big Crunch (Yajid et al., 2025), have been widely investigated to further enhance robustness and solution quality.

Despite extensive modifications, existing CSA variants still face key limitations. Adaptive strategies remain problem-dependent with limited generalizability, hybrid methods increase computational overhead, and elite-guided approaches often induce premature convergence due to excessive selection pressure. Moreover, exploration and exploitation are typically treated separately, leading to weak coordination between global and local search. Consequently, a unified and efficient framework remains lacking. Building on prior advances in adaptive control and elite utilization, this study aims to propose an Adaptive Elite Cuckoo Search Algorithm (AECSA) that integrates adaptive search regulation with elitist exploitation to enhance convergence efficiency. The proposed approach is validated through comprehensive benchmark evaluations. The main contributions of this study are summarized as follows:

- An AESCA is proposed by integrating adaptive Lévy flight step-size control with elitist crossover, enabling coordinated exploration and exploitation while maintaining computational simplicity.

https://doi.org/10.24191/jcrinn.v11i2.609

- Extensive evaluations show that AECSA significantly improves convergence efficiency and solution accuracy compared to the standard CSA and conventional metaheuristic methods.

The remainder of this paper is organized as follows. Section 2 reviews the standard CSA and its limitations. Section 3 presents the proposed AECSA, while Section 4 reports comparative performance results on benchmark functions. Conclusions are drawn in Section 5.

2. CUCKOO SEARCH ALGORITHM

The CSA is inspired by the brood parasitism behaviour of cuckoos. In this strategy, cuckoos lay eggs in host nests by mimicking the appearance of host eggs, thereby increasing the likelihood of survival. Host birds may detect and reject alien eggs or abandon the nest, prompting cuckoos to continuously refine their parasitic strategies. This evolutionary arms race between cuckoos and host birds is abstracted in CSA through selection and replacement mechanisms that favour high-quality solutions while eliminating inferior ones, forming the basis of its search mechanism (Yang & Deb, 2010).

Accordingly, CSA initializes with a randomly generated population of host nests, where each nest represents a candidate solution. New solutions are subsequently generated via Lévy flight and evaluated against randomly selected nests, and those superior solutions are retained and propagated to the next generation through selection-based replacement (Tian et al., 2024).

For simplicity, CSA is formulated under three idealized assumptions (Yu & Luo, 2023), whereby (i) each cuckoo deposits a single egg in a randomly selected host nest representing a candidate solution; (ii) a subset of nests containing high-quality solutions is preserved for subsequent generations; and (iii) the total number of host nests remains fixed while inferior solutions are probabilistically discarded or replaced to model host discovery behaviour.

The first assumption is implemented through stochastic solution generation using Lévy flight, whereby a new candidate solution is produced via a random-walk process defined in Eq.(1) and Eq.(2).

$$x_i(iter+1) = x_i(iter) + \alpha \times Levy(\lambda) \tag{1}$$

$$Levy(\lambda) = \left| \frac{\Gamma(1+\lambda)\times \sin\left(\frac{\pi\lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right)\times \lambda \times 2^{\left(\frac{\lambda-1}{2}\right)}} \right|^{\frac{1}{\lambda}} \tag{2}$$

Here, x_i denotes the solution vector, *iter* represents the current iteration, Γ is the gamma function, and λ is a constant (1< λ ≤3). To ensure convergence, the second assumption enforces elitist selection, allowing only fitter solutions to survive. The third assumption introduces diversification by probabilistically replacing a fraction $p_a \in [1,0]$ of the *n* host nests with newly generated solutions within the prescribed bounds, as defined in Eq.(3):

$$x_i = x_{i,\min} + rand(0,1)(x_{i,\max} - x_{i,\min}), \tag{3}$$

https://doi.org/10.24191/jcrinn.v11i2.609

where $x_{i.\min}$ and $x_{i.\max}$ denote the lower and upper limits of the i-th dimension. The complete CSA procedure is summarized in Algorithm 1 (Yang & Deb, 2010).

Algorithm 1. Cuckoo search algorithm
begin
Generate initial population of n host nest $\mathbf{x}_i$, $i = 1,2,\ldots,n$
for all $\mathbf{x}_i$ **do**
Evaluate the fitness function $F_i = f(\mathbf{x}_i)$
end for
while (*iter<MaxGeneration*) or (*stopping criterion*)
Generate a cuckoo egg $\mathbf{x}_j$ from random host nest by using Lévy flight
Calculate the fitness function $F_j = f(\mathbf{x}_j)$
Get a random nest i among n host nest
if ($F_j > F_i$) **then**
Replace $\mathbf{x}_i$ with $\mathbf{x}_j$
Replace F_i with F_j
end if
Abandon a fraction p_a of the worst nests
Build new nests randomly to replace nests lost
Evaluate the fitness of new nests
end while
end

While the standard CSA provides an effective framework for global optimization, its search behaviour is heavily governed by the stochastic nature of Lévy flight. As indicated in Eq.(1), the step size α critically determines the search scale: large values enhance exploration but may cause inefficient sampling or boundary violations, whereas small values restrict diversity and slow convergence (Chandrasekaran & Simon, 2012). To balance global and local search, α is fixed at 1 in the original CSA formulation (Xin-She & Deb, 2009). However, this static setting lacks adaptability across different search stages. In addition, the absence of structured information exchange among elite solutions limits exploitation efficiency, increasing susceptibility to premature convergence in complex or high-dimensional problems. These limitations motivate the development of an enhanced CSA framework that adaptively regulates search dynamics while explicitly exploiting elite information, leading to the proposed AECSA.

3. ADAPTIVE ELITE CUCKOO SEARCH ALGORITHM

To accelerate convergence while preserving the fundamental characteristics of CSA, the proposed AECSA introduces two key enhancements: (i) an adaptive Lévy flight step-size strategy; and (ii) an elitist crossover mechanism.

3.1 Adaptive search strategy

Appropriate regulation of the Lévy flight step size α is critical to balancing global exploration and local exploitation such that the search strategy is neither too aggressive nor too ineffective. Unlike the original CSA, where α is fixed, AECSA dynamically adjusts α according to the search progress. The adaptive step-size formulation is defined in Eq.(4):

https://doi.org/10.24191/jcrinn.v11i2.609

$$\alpha = \alpha_{\min}\left(1 + \frac{\alpha_0}{\sqrt{G}} \times \tanh\left(\frac{F\left(P_{best}^{t}\right)}{F\left(P_{best}^{0}\right)}\right)\right) \tag{4}$$

where α_{min} denotes the minimum allowable step size, α_0 is the initial step size, G represents the current iteration, $\tanh(\cdot)$ is the hyperbolic tangent function, $F\left(P_{best}^{0}\right)$ is the best fitness in the initial population, and $F\left(P_{best}^{t}\right)$ is the best fitness at iteration G.

At early stages, the population is widely dispersed across the search space; therefore, a relatively large α is employed to promote global exploration. As iterations progress and solutions converge toward promising regions, α is gradually reduced to enable finer local search around the current best solution. This behaviour motivates the inclusion of the scaling parameter $\alpha_0 / \sqrt{G}$ in the adaptive strategy, where a large initial value is assigned to α_0 in accordance with the problem domain to enhance far-field randomization during early iterations. Additionally, the hyperbolic tangent function provides a normalized measure of optimization progress. For a minimization problem, $F\left(P_{best}^{t}\right) \leq F\left(P_{best}^{0}\right)$, and thus the ratio $F\left(P_{best}^{t}\right) / F\left(P_{best}^{0}\right)$ monotonically decreases as the optimization progresses. Consequently, the tanh-based scaling term gradually diminishes, leading to a reduction in step size and a natural transition from global exploration to local exploitation.

From an optimization standpoint, search agents intensify exploitation in regions associated with high-quality solutions, where the probability of generating competitive offspring and locating the global optimum is higher. Conversely, stagnation or degradation in solution quality triggers exploration of unexplored regions to restore diversity (Salinas-Gutiérrez & Muñoz Zavala, 2023). This behaviour is regulated by the adaptive adjustment of the step size α, driven by the fitness improvement rate encoded by the monotonic hyperbolic tangent function in Eq.(4). Higher improvement rates result in smaller step sizes, enabling intensive local search, whereas stagnation increases α to facilitate exploration. Note that since $0 \leq \tanh(\cdot) \leq 1$, to prevent premature stagnation, a lower bound α_{min} is enforced, ensuring nonzero step sizes throughout the search.

3.2 Elitist crossover mechanism

In the standard CSA, search agents operate independently, resulting in limited information exchange among solutions. To address this limitation, AECSA incorporates a genetic algorithm–inspired crossover mechanism that generates new candidate solutions from an elite subset of the population. Specifically, the top 25% of solutions ranked by fitness are designated as elites, from which two individuals are randomly selected to produce offspring using the crossover operation defined in Eq.(5) (Higashi & Iba, 2003):

$$\begin{aligned} x_{i,1}(iter+1) &= rand \times elite_1(iter) + (1-rand) \times elite_2(iter) \\ x_{i,2}(iter+1) &= rand \times elite_2(iter) + (1-rand) \times elite_1(iter) \end{aligned} \tag{5}$$

where rand is a random value from a uniform distribution over the interval [0,1]. This elite-guided recombination enables effective information sharing among high-quality solutions, biasing offspring generation toward promising regions of the search space and thereby accelerating convergence with fewer function evaluations. The 25% elite threshold represents a trade-off between selection pressure and population diversity. A smaller elite set may lead to rapid but potentially premature convergence, whereas

https://doi.org/10.24191/jcrinn.v11i2.609

a larger set may dilute the influence of high-quality solutions. The chosen proportion ensures that crossover is guided by sufficiently fit individuals while preserving diversity for effective exploration.

The procedural steps of the proposed AECSA are summarized in Algorithm 2. To enforce boundary constraints, any newly generated solution that falls outside the feasible domain is discarded, and the corresponding solution position and fitness value are retained.

Algorithm 2. Adaptive elite Cuckoo search algorithm

```
begin
Generate initial population of n host nest x_i, i = 1,2,…,n
Define minimum Lévy flight step size α_min
Define initial Lévy flight step size α_0
for all x_i do
    Evaluate the fitness function F_i = f(x_i)
    Find the minimum fitness value γ among all the host nests
end for
while (iter<MaxGeneration) or (stopping criterion)
for all host nests do
    Sort all host nests by order of fitness
    for all top eggs do
        Current position x_i
        Generate the new cuckoo egg x_j using Eq.(1)
    end for
    for all non-top eggs do
        Current position x_i
        Generate a cuckoo egg x_j from host nest x_i by calculating the step size α for Lévy flight using
        Eq.(4)
    end for
    if (x_j falls outside the bounds)
        Replace x_j with x_i
    end if
    Calculate the fitness function F_j = f(x_j)
    Get a random nest i among n host nest
    if (F_j > F_i) then
        Replace x_i with x_j
        Replace F_i with F_j
    end if
end for
Abandon a fraction p_a of the worst nests
Build new nests randomly to replace nests lost using Eq.(5)
Evaluate the fitness of new nests
end while
end
```

4. RESULTS AND DISCUSSION

In this study, six standard benchmark functions, *i.e.*, Ackley, de Jong, Easom, Griewank, Rastrigin, and Rosenbrock (Ong et al., 2021), are employed to evaluate the effectiveness of the proposed AECSA and to

https://doi.org/10.24191/jcrinn.v11i2.609

compare its performance against the standard CSA. The dimensionality, search domains, and global optima of these benchmark functions are summarized in Table 1.

In practical optimization, the effectiveness of an algorithm is assessed by its ability to achieve meaningful performance gains under reasonable computational cost and acceptable error tolerance. Accordingly, the search performance of CSA and AECSA is evaluated using two criteria: convergence efficiency and solution precision. Precision is quantified by the average Euclidean distance between the best solution and the known global optimum under a fixed number of iterations, while convergence efficiency is measured by the average number of iterations required to satisfy a predefined error tolerance.

For each benchmark function, 30 independent runs are conducted with a population size of 20 host nests. At each iteration, the Euclidean distance between the current best solution and the global optimum (E_f) is recorded, and simulations terminate when the best fitness falls below a minimum error threshold $\varepsilon \leq 10^{-5}$. Mean performance curves are obtained by averaging results of E_f over all runs. Statistical significance of performance differences between CSA and AECSA is assessed using a two-tailed t-test at a 5% significance level.

Table 1: Parameter settings of the benchmark functions

Function	**Equation**	**Dim, *d***	**Search space**	**Global Optimum, $f(x^*)$**	**Optimum Point, x^***
Ackley	$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e$	50	[-32.768, 32.768]	0	(0,0,…,0)
De Jong	$f(\mathbf{x}) = \sum_{i=1}^{d}x_i^2$	50	[-5.12, 5.12]	0	(0,0,…,0)
Easom 2D	$f(\mathbf{x}) = -\cos(x_1)\cos(x_2)\exp\left[-(x_1-\pi)^2-(x_2-\pi)^2\right]$	2	[-100, 100]	-1	(π, π,…, π)
Griewank	$f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{d}x_i^2 - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	100	[-600, 600]	0	(0,0,…,0)
Rastrigin	$f(\mathbf{x}) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]$	10	[-5.12, 5.12]	0	(0,0,…,0)
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{d-1}[(1-x_i)^2 + 100(x_{i+1}-x_i^2)^2]$	10	[-100, 100]	0	(1,1,…,1)

4.1 Performance comparison of AECSA with benchmark algorithms in terms of fixed tolerance error

Table 2 first compares the number of iterations required by AECSA and the baseline CSA to reach the known global optima under a fixed error tolerance ε. In addition, AECSA is further evaluated against Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). The convergence behaviour, expressed as the average distance to the global optimum, is illustrated in Fig. 1 to Fig. 6 for each benchmark function.

https://doi.org/10.24191/jcrinn.v11i2.609

Table 2: Comparison of CSA and AECSA in terms of iterations required to converge to the global optimum

Function	CSA				AECSA				t-Statistic	p-value	Statistically Significant?
	Best	Worst	Mean	SD	Best	Worst	Mean	SD			
Ackley	3799	4856	4249	223	2780	3065	2965	67	30.20	<0.05	Yes
de Jong	1673	1815	1753	42	1104	1317	1222	65	37.58	<0.05	Yes
Easom 2D	311	787	512	96	51	114	74	15	24.69	<0.05	Yes
Griewank	4405	4647	4537	59	3481	3819	3642	88	46.27	<0.05	Yes
Rastrigin	1603	2176	1891	131	1028	1704	1375	156	13.87	<0.05	Yes
Rosenbrock	23218	34916	27635	2450	11328	22701	19687	2247	13.10	<0.05	Yes

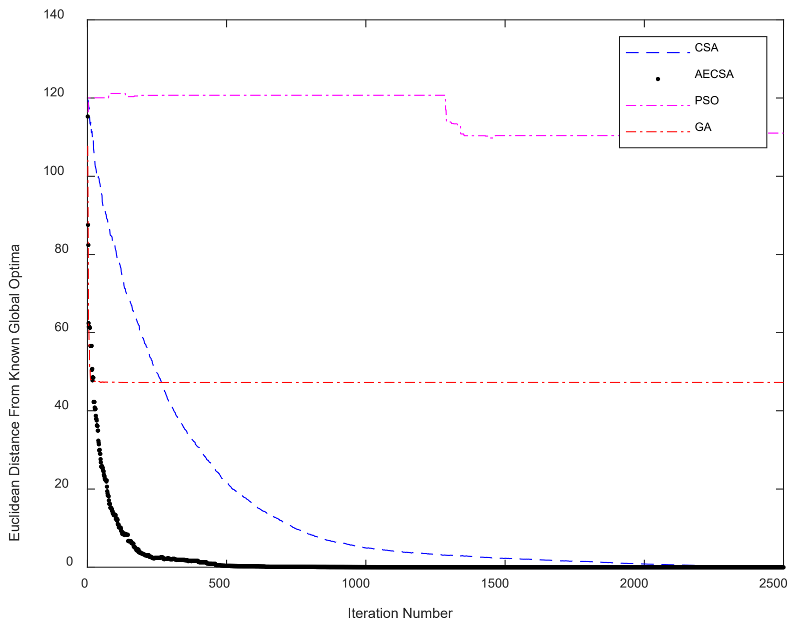

Fig. 1. Convergence comparison of AECSA and benchmark algorithms on the Ackley function

https://doi.org/10.24191/jcrinn.v11i2.609

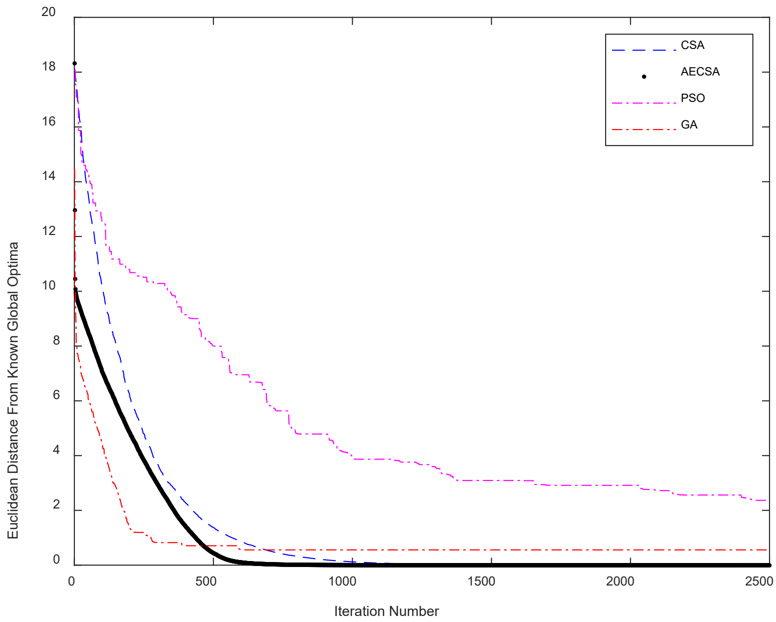

Fig. 2. Convergence comparison of AECSA and benchmark algorithms on the De Jong function

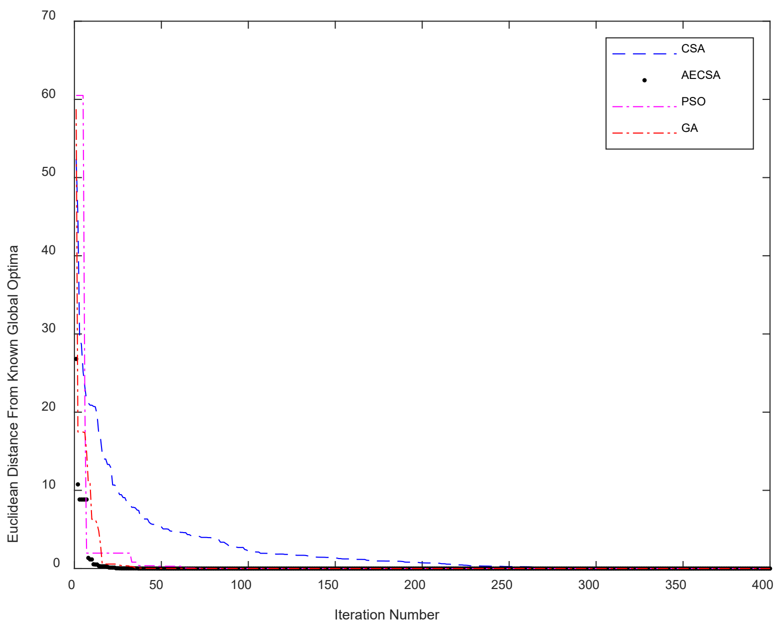

Fig. 3. Convergence comparison of AECSA and benchmark algorithms on the Easom function

https://doi.org/10.24191/jcrinn.v11i2.609

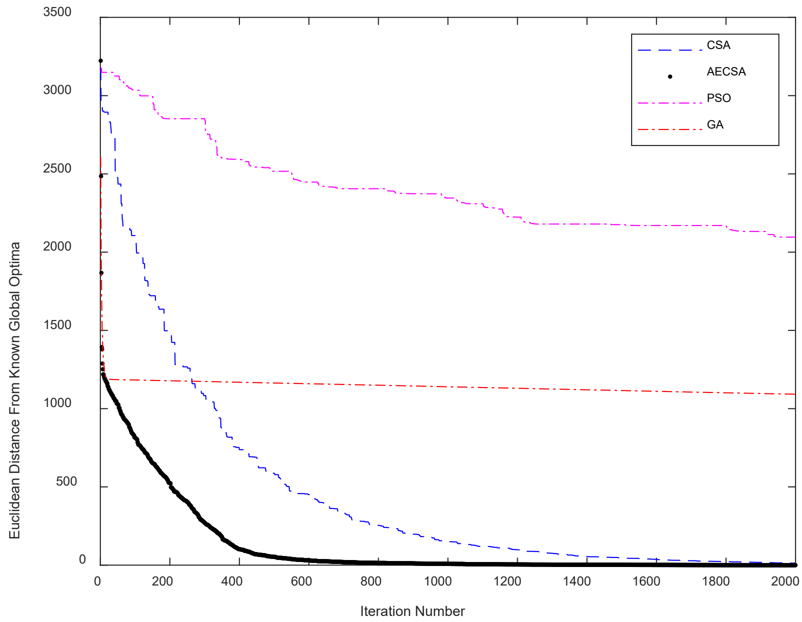

Fig. 4. Convergence comparison of AECSA and benchmark algorithms on the Griewank function

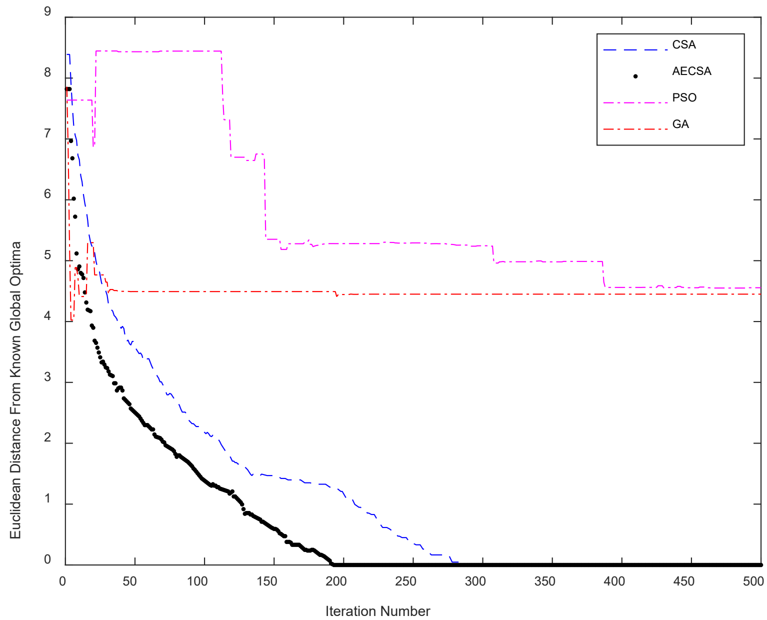

Fig. 5. Convergence comparison of AECSA and benchmark algorithms on the Rastrigin function

https://doi.org/10.24191/jcrinn.v11i2.609

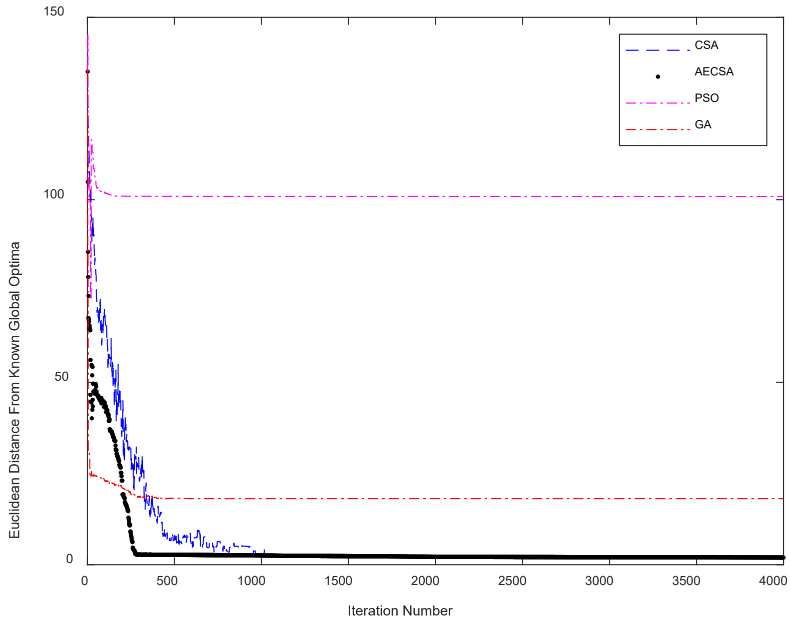

Fig. 6. Convergence performance comparison of AECSA and benchmark algorithms on the Rosenbrock function

A comparison between AECSA and the baseline CSA is first conducted to assess whether the incorporation of adaptive step size and elitism improves convergence performance. As observed from Table 2 and Figs. 1 to 6, both algorithms exhibit an exponential-like reduction in distance error; however, AECSA consistently converges faster than the standard CSA. For the Ackley function, AECSA reaches the global optimum in approximately 3,000 iterations, requiring about 1,300 fewer iterations on average than CSA. For the unimodal de Jong function, both methods converge rapidly due to the simplicity of the test function, yet AECSA still achieves an approximately 30% improvement in convergence speed due to the integration of adaptive step-size control and elitist exploitation.

AECSA significantly outperforms the standard CSA on the two-dimensional Easom function. As reported in Table 2, AECSA converges to the global optimum in an average of 74 iterations, achieving nearly a sevenfold reduction in convergence time compared to CSA. This performance gain is probably attributed to the conservative Lévy flight step size of CSA (where α=1), and the absence of structured information exchange among high-quality solutions, which limit effective exploration and exploitation. The Griewank function in a 100-dimensional space exhibits numerous local minima but a single global optimum at (0,0,…0), with the number of local minima increasing exponentially with dimensionality. Despite this complexity, both CSA and AECSA successfully avoid entrapment in local minima and converge to the global optimum. As shown in Table 2, AECSA achieves convergence in approximately 3,600 iterations, compared to about 4,500 for CSA, reaffirming the superior convergence efficiency of the proposed AECSA.

Fig. 7 illustrates the three-dimensional surface of the Rastrigin function, which is characterized by a large number of local extrema, making it a challenging benchmark for optimization algorithms, particularly gradient-based methods. To further assess convergence capability, a more challenging 10-dimensional Rastrigin function is examined. As indicated in Fig. 5, AECSA exhibits a faster convergence rate than CSA.

https://doi.org/10.24191/jcrinn.v11i2.609

On average, AECSA reaches the global optimum at (0,0,…,0) within approximately 1,300 iterations, whereas CSA requires about 1,600 iterations to achieve convergence.

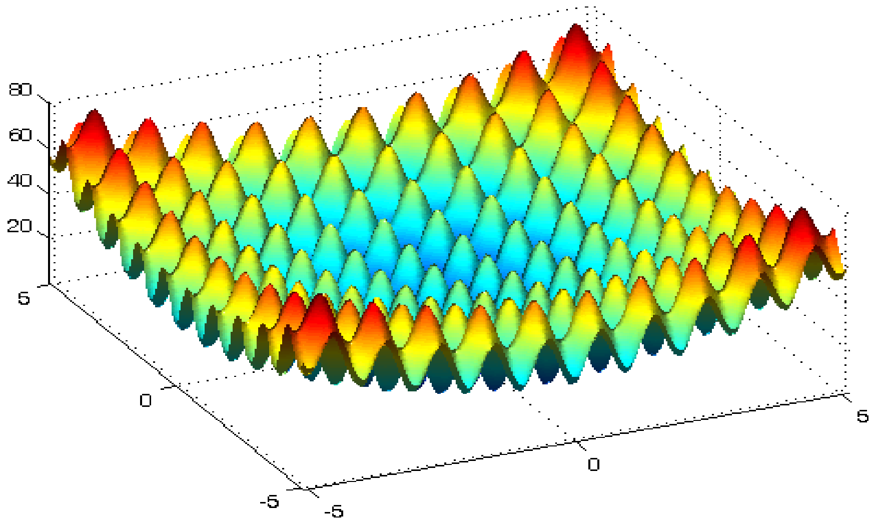

Fig. 7. The 3-dimensional surface plot of the Rastrigin's function

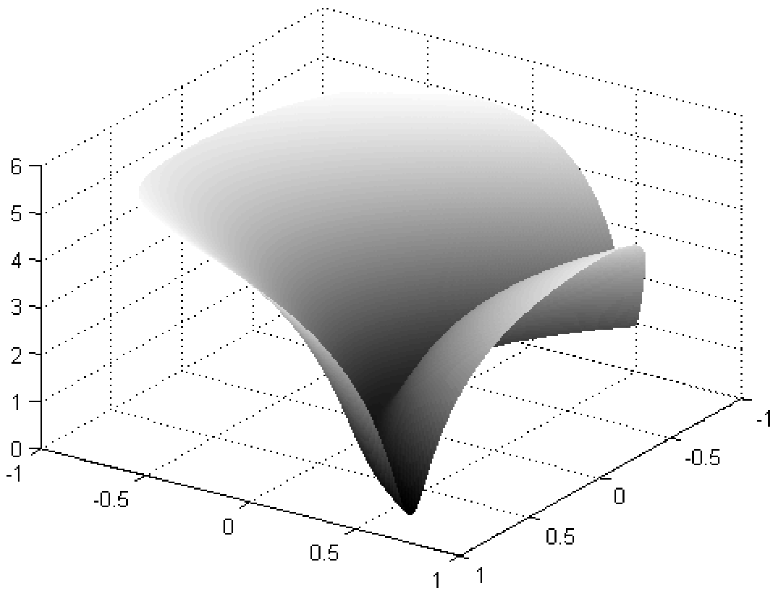

Fig. 8. The 3-dimensional surface plot of the Rosenbrock's function

https://doi.org/10.24191/jcrinn.v11i2.609

Fig. 8 depicts the three-dimensional surface of the Rosenbrock function, characterized by a narrow, curved valley containing the global minimum. Although the valley is relatively easy to locate, convergence to the global optimum within this region is challenging, typically resulting in a large number of iterations (Walton et al., 2011). This behaviour is reflected in Table 2, where the standard CSA requires approximately 28,000 fitness evaluations on average to converge. By contrast, the incorporation of adaptive step-size control and elite information exchange in AECSA reduces the required evaluations to about 20,000, corresponding to an improvement of approximately 30%.

Next, AECSA is compared with two widely used classical bio-inspired metaheuristic algorithms, namely PSO and GA. As illustrated in Figs. 1–6, both PSO and GA exhibit significantly slower convergence and inferior performance. Except for the Easom function, neither PSO nor GA attains the predefined tolerance error, even when AECSA has already converged much earlier. This indicates that AECSA possesses superior search efficiency and robustness, effectively balancing exploration and exploitation, and is more capable of escaping local optima compared to conventional approaches.

4.2 Performance comparison with baseline CSA in terms of fixed iterations

From a precision perspective, performance is assessed based on solution quality achieved within a fixed number of iterations. For each benchmark function, the average distance between the best solution and the known global optimum obtained by CSA and AECSA at 1, 500, 1,000, 1,500, and 2,000 iterations is reported in Table 3. Lower distance values indicate higher solution accuracy.

Table 3: Comparison of CSA and AECSA based on solution accuracy at fixed iteration counts

Function	Generation	Average E_f over 30 Independent Runs	
		CSA	**AECSA**
Ackley	1	20.9015	20.9126
	500	11.1213	0.5015
	1000	3.9515	0.0361
	1500	1.9738	0.0044
	2000	0.6089	0.0005
de Jong	1	336.3923	337.0073
	500	1.8928	0.0050
	1000	0.0156	5.8419e-05
	1500	0.0001	9.9746e-07
	2000	1.0394e-06	1.7962e-08
Easom 2D	1	52.4996	49.2165
	500	0.0118	1.2549e-08
	1000	3.1400e-06	3.8041e-09
	1500	6.2143e-09	3.8041e-09
	2000	4.8688e-09	3.8041e-09
Griewank	1	2460.9166	2483.6485
	500	90.2198	10.3679
	1000	7.4227	0.7493
	1500	1.4941	0.1170
	2000	1.0232	0.0144
Rastrigin	1	788.3416	788.4332
	500	281.3610	79.7475
	1000	223.2405	68.7607
	1500	193.6881	60.0045
	2000	174.4563	52.1707
Rosenbrock	1	134.1209	138.6492
	500	11.4478	6.4302
	1000	6.6084	2.5646
	1500	4.8169	2.0547
	2000	3.7297	1.8498

https://doi.org/10.24191/jcrinn.v11i2.609

Consider all benchmark functions, AECSA consistently yields solutions closer to the global optimum than the standard CSA. The improvement in solution accuracy is particularly pronounced for the Ackley, de Jong, and two-dimensional Easom functions, where AECSA attains near-optimal solutions within 500 iterations, while CSA remains far from convergence. For the Griewank and Rastrigin functions, AECSA exhibits a more rapid reduction in distance error after 500 evaluations, indicating superior exploitation capability. Notably, even after 2,000 iterations on the Griewank function, CSA lags behind AECSA in solution quality. In contrast, performance differences on the Rosenbrock function are marginal, likely due to its narrow, curved valley structure and increased optimization difficulty.

4.3 Discussion

Overall, AECSA consistently demonstrates faster convergence and higher solution accuracy than the baseline CSA. This improvement can be attributed to the synergistic integration of adaptive step-size control and elitist information exchange. Unlike the standard CSA, where a fixed step size limits adaptability, the proposed adaptive mechanism enables dynamic adjustment of the search scale, allowing efficient exploration in early iterations and refined exploitation in later stages. Similar observations have been reported in adaptive metaheuristics, where dynamic parameter control improves convergence behaviour across complex landscapes (Abu-Ein et al., 2026; Gaborit et al., 2026; Zhou et al., 2026). The incorporation of elitist crossover further contributes to the improved performance by enabling structured information sharing among high-quality solutions. In contrast to the independent search behaviour in standard CSA, the recombination of elite individuals biases the search toward promising regions while preserving diversity. Similar elite-guided strategies have been shown to improve solution in evolutionary and population-based algorithms (Ekinci et al., 2025; Torres-Cardenas et al., 2026; Wang & Xu, 2026). When compared with PSO and GA, AECSA demonstrates superior convergence behaviour and robustness across most benchmark functions. While PSO and GA are effective general-purpose optimizers, their performance is often sensitive to parameter settings and may suffer from premature convergence or slow exploitation (Chauhan et al., 2025; Katoch et al., 2021). In contrast, AECSA achieves a more balanced search process, resulting in faster convergence and improved accuracy.

Despite these advantages, it is important to note that the performance of AECSA may still depend on the choice of parameters such as the initial step size and elite proportion. Although these parameters are less sensitive than in many hybrid approaches, further investigation into parameter self-adaptation could enhance generalizability across different problem domains.

5. CONCLUSIONS

This study proposed an AECSA that enhances the standard CSA by jointly integrating adaptive Lévy flight step-size control and elite-guided information exchange. The adaptive mechanism enables a dynamic balance between global exploration and local exploitation, while the elitist crossover strategy promotes effective knowledge sharing among high-quality solutions without increasing algorithmic complexity. Evaluations on six benchmark optimization functions demonstrate that AECSA consistently outperforms the standard CSA in terms of both convergence efficiency and solution precision. Specifically, AECSA accelerates convergence by approximately 30% for the Ackley, de Jong, Rastrigin, and Rosenbrock functions, and by about 85% and 20% for the Easom and Griewank functions, respectively. In terms of solution precision, AECSA consistently achieves higher accuracy than CSA under fixed iteration budgets. Overall, the results confirm that synergistically combining adaptive search control with elite exploitation provides an effective and computationally efficient enhancement to CSA. Future work will focus on extending AECSA to constrained, multi-objective, and real-world engineering optimization problems.

https://doi.org/10.24191/jcrinn.v11i2.609

6. ACKNOWLEDGEMENTS/FUNDING

The author(s) have no acknowledgments to declare and received no funding from any agency in the public, commercial or not-for-profit sectors for this study.

7. CONFLICT OF INTEREST STATEMENT

The author(s) declare that they have no conflicts of interest relevant to this study.

8. AUTHORS' CONTRIBUTIONS

Pauline Ong: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Jia Hang Wu:** Data curation.

9. REFERENCES

Abu-Ein, A. A., Al-Hazaimeh, O. M., Tawfik, M., & Fathi, I. S. (2026). Dynamic fog node placement optimization using adaptive dynamic pufferfish optimization for real-time IoT networks. *Scientific Reports*, *16*(1), 11624. https://doi.org/10.1038/s41598-026-41740-8.

Acharjee, S., & Chaudhuri, S. S. (2023). Modified Cuckoo Search algorithm for motion vector estimation. *International Arab Journal of Information Technology*, *20*(3), 340-348. https://doi.org/10.34028/iajit/20/3/6.

Alibabaei Shahraki, M. (2025). Cloud drift optimization algorithm as a nature-inspired metaheuristic. *Discover Computing*, *28*(1), *173*. https://doi.org/10.1007/s10791-025-09671-6.

BahraniPour, F., Farshi, M., & Ebrahimi Mood, S. (2025). Enhanced multi-objective cuckoo search with migration operator for benchmark optimization and IoT task scheduling in cloud-fog computing. *Journal of Supercomputing*, *81*(8), Article 1024. https://doi.org/10.1007/s11227-025-07531-0.

Begum, F., & Valan, J. A. (2026). Enhancing heart disease diagnosis with Meta-Heuristic Algorithms: A combined HHO and PSO Approach. *Biomedical Materials & Devices*, *4*(1), 574-589. https://doi.org/10.1007/s44174-025-00288-3.

Cai, M. (2025). Using Cuckoo search algorithm to predict corporate financial risks and alleviate economic uncertainty. *International Journal of Computational Intelligence Systems*, *18*(1), 216. https://doi.org/10.1007/s44196-025-00950-0.

Chandrasekaran, K., & Simon, S. P. (2012). Multi-objective scheduling problem: Hybrid approach using fuzzy assisted cuckoo search algorithm. *Swarm and Evolutionary Computation*, *5*(0), 1-16. https://doi.org/http://dx.doi.org/10.1016/j.swevo.2012.01.001.

Chauhan, D., Shivani, & Suganthan, P. N. (2025). Learning strategies for particle swarm optimizer: A critical review and performance analysis. *Swarm and Evolutionary Computation*, *98*, 102048. https://doi.org/https://doi.org/10.1016/j.swevo.2025.102048.

https://doi.org/10.24191/jcrinn.v11i2.609

Chen, J., Cai, Z., Chen, H., Chen, X., Escorcia-Gutierrez, J., Mansour, R. F., & Ragab, M. (2023a). Renal Pathology Images Segmentation Based on Improved Cuckoo Search with Diffusion Mechanism and Adaptive Beta-Hill Climbing. *Journal of Bionic Engineering*, *20*(5), 2240-2275. https://doi.org/10.1007/s42235-023-00365-7.

Chen, Y., Tan, B., & Zeng, L. (2023b). Inspection path planning of free-form surfaces based on improved Cuckoo Search Algorithm. *Measurement and Control*, *56*(7-8), 1321-1332. https://doi.org/10.1177/00202940231157422.

Choudhary, A., & Rajak, R. (2026). EMMCA: Enhancing modified Min-Min using Cuckoo search algorithm in cloud computing. *Evolving Systems*, *17*(1), 15. https://doi.org/10.1007/s12530-025-09768-9.

Dar, T. H., Singh, S., & Duru, K. K. (2025). Lithium-ion battery parameter estimation based on variational and logistic map cuckoo search algorithm. *Electrical Engineering*, *107*(2), 1427-1440. https://doi.org/10.1007/s00202-024-02580-9.

Dorigo, M., Maniezzo, V., & Colorni, A. (1991). The Ant System: An autocatalytic optimizing process. *TR91-016*, *Politecnico Di Milano*, 1-21.

Ekinci, S., Izci, D., Kayri, M., Elsayed, F., & Salman, M. (2025). A novel hybridization of birds of prey-based optimization with differential evolution mutation and crossover for chaotic dynamics identification. *Scientific Reports*, *15*(1), 43451. https://doi.org/10.1038/s41598-025-27220-5.

Gaborit, R., van der Hurk, E., Nielsen, O. A., & Jiang, Y. (2026). Optimisation of bus timetables: An adaptive large neighbourhood search-based matheuristic with a novel operator weight. *European Journal of Operational Research*, *333*(1), 38-52. https://doi.org/https://doi.org/10.1016/j.ejor.2026.01.032.

Garg, P., Gautam, M., & Sharma, V. (2025). GCSO: Grey-Cuckoo search based optimization for security of medical images through watermarking. *Circuits, Systems, and Signal Processing*, *44*(8), 6027-6055. https://doi.org/10.1007/s00034-025-03083-z.

Higashi, N., & Iba, H. (2003). Particle swarm optimization with Gaussian mutation. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium* (pp. 72-79). https://doi.org/10.1109/SIS.2003.1202250.

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press.

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimed Tools Appl*, *80*(5), 8091-8126. https://doi.org/10.1007/s11042-020-10139-6.

Maddaiah, P. N., & Narayanan, P. P. (2023). An improved Cuckoo search algorithm for optimization of artificial neural network training. *Neural Processing Letters*, *55*(9), 12093-12120. https://doi.org/10.1007/s11063-023-11411-0.

Mazari, A., Laroussi, K., Fergani, O., Abbas, H. A., & Rezk, H. (2025). A hybrid whale optimization—Cuckoo search algorithm for maximum power point tracking in PMSG-Based wind turbine systems. *International Transactions on Electrical Energy Systems*, *2025*(1), 7411272. https://doi.org/https://doi.org/10.1155/etep/7411272.

https://doi.org/10.24191/jcrinn.v11i2.609

Mechaacha, A., Belkaid, F., & Brahimi, N. (2026). Multi-objective multi-product process planning with reconfigurable machines: Exact and metaheuristic approaches. *Expert Systems with Applications*, *298*, 129591. https://doi.org/https://doi.org/10.1016/j.eswa.2025.129591.

Mimansha, & Kumar, A. (2025). Reliability optimization of complex system under failure dependencies: a dynamic method in adaptive cuckoo optimization. *International Journal of System Assurance Engineering and Management*, *16*(6), 2166-2175. https://doi.org/10.1007/s13198-025-02754-z.

Ong, K. M., Ong, P., & Sia, C. K. (2021). A carnivorous plant algorithm for solving global optimization problems. *Applied Soft Computing*, *98*, 106833. https://dx.doi.org/10.1016/j.asoc.2020.106833.

Ouyang, C., Liu, X., Zhu, D., Li, Y., Mao, J., Zhou, C., & Xue, J. (2025). Hierarchical adaptive cuckoo search algorithm for global optimization. *Cluster Computing*, *28*(5), 321. https://doi.org/10.1007/s10586-024-04924-3.

Pavithra, L., & Rekha, D. (2025). Real time broadcast scheduling in tsch for smart healthcare using Cuckoo search algorithm. *Results in Engineering*, 105018. https://dx.doi.org/10.1016/j.rineng.2025.105018.

Safdar, K., Rani, K. N. A., Rosli, S. J., Jamlos, M. A., & Younus, M. U. (2025). Modified Cuckoo search algorithm using sigmoid decreasing inertia weight for global optimization. *Pertanika Journal of Science and Technology*, *33*(5), 2069-2095. https://doi.org/10.47836/pjst.33.5.03.

Saleh, B. J., Omar, Z., As'ari, M. A., Bhateja, V., & Izhar, L. I. (2025). A novel hybrid deep learning model based on simulated annealing and Cuckoo Search algorithms for automatic radiomics-based COVID-19 diagnosis. *Applied Computational Intelligence and Soft Computing*, *2025*(1), 8829294. https://doi.org/https://doi.org/10.1155/acis/8829294.

Salgotra, R., Singh, S., Verma, P., Abualigah, L., & Gandomi, A. H. (2025). Mutation adaptive cuckoo search hybridized naked mole rat algorithm for industrial engineering problems. *Scientific Reports*, *15*(1), 19655. https://doi.org/10.1038/s41598-025-01033-y.

Salinas-Gutiérrez, R., & Muñoz Zavala, A. E. (2023). An explicit exploration strategy for evolutionary algorithms. *Applied Soft Computing*, *140*, 110230. https://doi.org/https://doi.org/10.1016/j.asoc.2023.110230.

Shi, X., Javidan, S. M., Ampatzidis, Y., & Zhang, Z. (2025). AI-driven identification of grapevine fungal spores via microscopic imaging and feature optimization with cuckoo search algorithm: ClassifierSVM with linear kernelDTk-NNAccuracy (%) 97.5076. 4280.01. *Smart Agricultural Technology*, *11*, 101029. https://dx.doi.org/10.1016/j.atech.2025.101029.

Subha, S., & Kumaran. (2025). Adaptive cuckoo search algorithm based fuzzy C means clustering with random walker algorithm for liver segmentation using CT images. *Multimedia Tools and Applications*, *84*(8), 5051-5068. https://dx.doi.org/10.1007/s11042-024-18708-9.

Tian, X., Yang, B., Na, X., Cheng, S., Ba, L., & Yuan, Y. (2025). An improved cuckoo search algorithm for inverse heat conduction and heat convection. *Chinese Journal of Physics*, *95*, 765-789. https://doi.org/https://doi.org/10.1016/j.cjph.2025.03.039.

Tian, Y., Zhang, D., Zhang, H., Zhu, J., & Yue, X. (2024). An improved cuckoo search algorithm for global optimization. *Cluster Computing*, *27*(6), 8595-8619. https://doi.org/10.1007/s10586-024-04410-w.

https://doi.org/10.24191/jcrinn.v11i2.609

Torres-Cardenas, F. A., Lozano Suarez, L. M., & Díaz Bohórquez, C. E. (2026). A multi-objective black widow spider algorithm for solving the flexible job shop problem with energy efficiency and transport time. *Soft Computing*, *30*, 3503-3523. https://doi.org/10.1007/s00500-026-11297-9.

Walton, S., Hassan, O., Morgan, K., & Brown, M. R. (2011). Modified cuckoo search: A new gradient free optimisation algorithm. *Chaos, Solitons & Fractals*, *44*(9), 710-718. https://doi.org/http://dx.doi.org/10.1016/j.chaos.2011.06.004.

Wang, X., & Xu, H. (2026). Multi-strategy enhanced human evolution optimization algorithm and its engineering applications. *Ain Shams Engineering Journal*, *17*(1), 103818. https://doi.org/https://doi.org/10.1016/j.asej.2025.103818.

Wu, L., Li, Z., Dong, W., Ye, H., & Jiang, M. (2026). Integrated scheduling method for heavy-haul transportation with virtual coupling based on two-phase optimization strategy. *Expert Systems with Applications*, *297*, 129326. https://doi.org/https://doi.org/10.1016/j.eswa.2025.129326.

Xin-She, Y., & Deb, S. (2009). Cuckoo Search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (pp. 210-214). https://doi.org/10.1109/NABIC.2009.5393690.

Yajid, M. S. A., Bhosle, N., Sudhamsu, G., Khatibi, A., Sharma, S., Jeet, R.,…Santhosh, A. J. (2025). Hybrid big bang-big crunch with cuckoo search for feature selection in credit card fraud detection. *Scientific Reports*, *15*(1), 23925. https://doi.org/10.1038/s41598-025-97149-2.

Yang, J., Liu, J., & Liu, J. (2025). A hybrid evolution Jaya algorithm for meteorological drone trajectory planning. *Applied Mathematical Modelling*, *137*, 115655. https://doi.org/https://doi.org/10.1016/j.apm.2024.115655.

Yang, X.-S., & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, *1*(4), 330-343. https://doi.org/10.1504/IJMMNO.2010.03543.

Yang, X., Hao, X., Yang, T., Li, Y., Zhang, Y., & Wang, J. (2023). Elite-guided multi-objective cuckoo search algorithm based on crossover operation and information enhancement. *Soft Computing*, *27*(8), 4761-4778. https://doi.org/10.1007/s00500-022-07605-8.

Yin, S., Tu, J., & Chen, X. (2025). A new tree-based data aggregation method in the wireless sensor networks using Ant Colony Optimization and Cuckoo search algorithms. *Journal of Engineering and Applied Science*, *72*(1), 83. https://doi.org/10.1186/s44147-025-00652-6.

Yu, X., & Luo, W. (2023). Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning. *Expert Systems with Applications*, *223*, 119910. https://doi.org/https://doi.org/10.1016/j.eswa.2023.119910.

Yu, X., Xu, P., Wang, X., & Zhang, W. (2026). Optimizing islanded microgrid scheduling with reinforcement learning enhanced multi-strategy cuckoo search algorithm. *Applied Soft Computing*, *186*, 114251. https://doi.org/https://doi.org/10.1016/j.asoc.2025.114251.

Zhou, N., Xu, K., & Ma, L. (2026). An adaptive decision sand cat swarm optimization algorithm combined with Q-learning for solving engineering problems. *Journal of Engineering and Applied Science*, *73*(1), 46. https://doi.org/10.1186/s44147-026-00885-z.