

An Improved Ant Colony Optimization with Levy Flight and Gaussian Mutation for Global Optimization

Ibrahim Musa Conteh
Department of Computer Science
Ernest Bai Koroma University of
Science and Technology
Magburaka, Sierra Leone.
School of Information Engineering
Wuhan University of Technology.
iconteh@ebkustsl.edu.sl

Aminu Onimisi Abdulsalami
Department of Computer Science
Ahmadu Bello University, Zaria.
School of Computer Science & AI
Wuhan University of Technology.
aoabdulsalami@abu.edu.ng

Abstract—This study proposes an Enhanced Ant Colony Optimization (EACO) algorithm that integrates Levy Flight and Gaussian Mutation to strengthen global exploration and accelerate convergence. Traditional Ant Colony Optimization (ACO) methods often experience premature convergence and an imbalance between exploration and exploitation. To address these limitations, Levy Flight introduces probabilistic long-distance movements that enable the algorithm to escape local optima, while Gaussian Mutation enhances local exploitation through adaptive fine-tuning. The proposed methodology was chosen for its adaptive balance between diversification and intensification, an improvement over previous ACO hybrid variants that typically employ static or single-type perturbation mechanisms. The EACO algorithm was implemented in MATLAB and tested on 12 benchmark functions from the CEC2022 suite under 10-dimensional (D) and 20-dimensional (D) scenarios. Performance was evaluated using convergence rate, mean fitness value, and standard deviation metrics. Comparative results with Particle Swarm Optimization (PSO), Linear Success-History based Adaptive Differential Evolution (LSHADE), Arithmetic Optimization Algorithm (AOA), and the classical Ant Colony Optimization (ACO) demonstrated that EACO achieves faster convergence, higher accuracy, and greater robustness. Statistical validation using the Wilcoxon signed-rank test confirmed its superior overall performance as an effective high-dimensional global optimizer.

Keywords— Ant Colony Optimization, Levy Flight, Gaussian Mutation, Exploration and Exploitation balance, Global Optimization, Metaheuristic Algorithms.

I. INTRODUCTION

The Ant Colony Optimization (ACO) algorithm, introduced by Dorigo and Stützle [1], is one of the most influential swarm intelligence techniques inspired by the pheromone-based foraging behavior of real ants. In this algorithm, artificial ants simulate the process of path construction and pheromone deposition to identify optimal or near-optimal solutions. Due to its simplicity, flexibility, and strong problem-solving ability, ACO has been successfully applied to a wide range of optimization tasks, including scheduling, routing, clustering, and feature selection spaces [2]. However, the conventional ACO still faces significant challenges that limit its performance

in complex and high-dimensional search spaces. The major issues include premature convergence, low population diversity, and the tendency to get trapped in local optima, primarily caused by excessive pheromone accumulation and lack of exploration mechanisms. These shortcomings often result in suboptimal solutions and slow convergence, particularly in multimodal optimization problems. Motivated by these limitations, recent research has explored hybridization techniques that integrate ACO with advanced strategies to enhance its search dynamics and strike a balance between exploration and exploitation. Among these, Lévy flight and Gaussian mutation have gained attention for their complementary roles. Lévy flight enhances global exploration through probabilistic long-distance jumps. In contrast, Gaussian mutation refines local exploitation by introducing adaptive perturbations. Building on this motivation, the present study proposes an Enhanced Ant Colony Optimization (EACO) algorithm that combines these two mechanisms to overcome ACO's inherent weaknesses. The main contributions of this work include developing a more balanced exploration-exploitation process, achieving faster convergence, and demonstrating superior performance across benchmark optimization problems.

A. Literature Review

Many researchers have proposed various approaches to improve ACO by combining adaptive control, hybrid learning, and stochastic search to address these weaknesses. For instance, Qi et al.[3] proposed an adaptive pheromone updating strategy, where the evaporation rates were dynamically updated to balance exploration and exploitation. Similarly, Chen et al.[4] introduced chaotic maps in ACO to improve the population diversity and global search efficiency. Although such strategies enhance ACO to a certain extent, they often cannot escape a deep local minimum or adjust well to non-convex search landscapes.

Levy Flight (LF), a random movement characterized by heavy-tailed step lengths, has proven to be an efficient tool for exploring processes when employed in metaheuristics [5]. Levy flights enable algorithms to undertake a long-range random walk, thereby increasing the probability of escaping local optima. Liu et al. [6] suggested a Greedy-Levy ACO with

ϵ -greedy and Levy Flight, showing improved convergence rates and solution quality over several benchmark optimization problems. Similarly, El Asri et al. [7] employed a Levy Flight-based ACO for the PTSP, which effectively integrated the global-local search balance, yielding better solutions. Moreover, Bansal et al. [8] proposed an Improved ACO with Levy distribution, achieving faster convergence and better stability in large-scale combinatorial optimization problems. These observations verify that the Levy Flight enhances ACO's exploration power by stimulating more random and varied movements in the search space.

Where Levy Flight is designed to enhance global exploration, Gaussian Mutation (GM) is a locally inspired mutation activation technique that applies a controlled random perturbation around candidate solutions. Gaussian mutation has been commonly employed in evolutionary algorithms, such as Genetic Algorithms (GA) and Differential Evolution (DE), to exploit promising solutions and overcome premature convergence [9]. Yu et al. [10] recently incorporated a Gaussian mutation into Particle Swarm Optimization (PSO) and presented improved local exploitation properties without compromising diversity. However, its integration with ACO remains limited. The combination of LF and GM presents a promising hybridization approach that simultaneously strengthens exploration and exploitation, two key components of search balance in swarm intelligence.

Current directions in studying hybrid metaheuristics tend towards the use of multiple operators and adaptive learning strategies, thereby reinforcing robustness. For example, Zhang et al. [11] introduced an adaptive hybrid ACO based on dynamic search operators. Lin et al. [12] included Gaussian-based local refinements into an evolutionary ACO variant for continuous optimization. These results demonstrate that integrating the global and local search strategies can significantly enhance convergence robustness and accuracy.

Hussain et al. [13] also proposed hybridization trends for swarm intelligence, concluding that combined randomization strategies, such as Levy Flight or Gaussian mutation, improved robustness in uncertain optimization scenarios. Likewise, Saka and Kose [14] introduced a hybrid PSO–ACO approach that combines ACO's local intensification with PSO's global exploration to provide better convergence without stagnation. Wang et al. [15] presented a balance between evaporation rate and solution quality as an adaptive technique for setting the pheromone decay model. This approach ensures that no exploration stops, as it guarantees stable convergence. Similarly, Liu et al. [16] proposed an adaptive version of ACO with reinforcement learning. The pheromone control is based on search performance feedback, allowing the algorithm to adjust its behavior accordingly. According to Li et al. [17], population diversity is significantly enhanced, particularly at the initial stages of optimization, by employing the Levy distribution in swarm fly algorithms. Such random walking can lead to superior global searching and avoid local convergence. Moreover, Bui et al. developed a hybrid ACO-Bat Algorithm model that employed the crossover operation to solve the problem of the continuous bidding price of indivisible goods [18], showing that adding Levy flight movements increased the escape from local minima and reduced the convergence time on

a set of engineering design problems. Likewise, Panwar and Agrawal [19] also incorporated Levy Flight in a modified Genetic Algorithm. They experimentally demonstrated improved robustness and better global optimization over benchmark functions and real-life problems. These results support the importance of levy-based exploration in addressing stagnation problems in swarm intelligence-based optimizers.

Gaussian mutation remains one of the most successful and widely used mutation strategies for local search refinement. It supplies tiny, stochastic perturbations that can make fine-tuned movements toward good approximations of the solutions. According to Sun et al. [20], the Gaussian mutation improves the balance between convergence accuracy and population diversity in continuous optimization. Moreover, a study by Kumar and Singh [21] found that a Gaussian-mutated DE variant outperformed classical DE and PSO in benchmark functions due to its adaptive exploitation characteristics. Similarly, Lin et al. [22] introduced a Gaussian-based local search into a multi-strategy optimization, obtaining better fine-tuning and solution stability. These results indicate that Gaussian mutation is particularly successful when combined with global search strategies such as Levy Flight or pheromone-based learning in ACO.

Building upon these insights, the EACO algorithm proposed in this study integrates both Levy Flight and Gaussian Mutation mechanisms. The Levy Flight component increases exploration by guiding ants toward distant and diverse regions of the search space, while Gaussian Mutation improves exploitation by finetuning high-quality solutions. Together, these enhancements aim to overcome the ACO's inherent weaknesses, including limited diversity and premature convergence, by striking a balance between exploration and exploitation. EACO's hybrid structure is expected to deliver faster convergence, improved robustness, and adaptability to complex global optimization challenges, such as those in the CEC2022 benchmark suite.

II. METHODOLOGY

A. Overview of the Proposed EACO Framework

The proposed Enhanced Ant Colony Optimization (EACO) algorithm highlighted in Figure 1, integrates two powerful stochastic mechanisms, Levy Flight and Gaussian Mutation, into the traditional Ant Colony Optimization (ACO) framework to address its main limitations, including poor population diversity, premature convergence, and slow search in high-dimensional spaces. The hybridization aims to establish a balanced trade-off between exploration and exploitation throughout the search process. Levy Flight enables long-distance jumps in the solution space, thereby increasing the exploration capacity of the ants during early iterations. Conversely, Gaussian Mutation performs fine-tuning around promising solutions to enhance exploitation in the later optimization stages. The proposed method optimizes continuous and combinatorial problems by adaptively switching between global and local search behaviors.

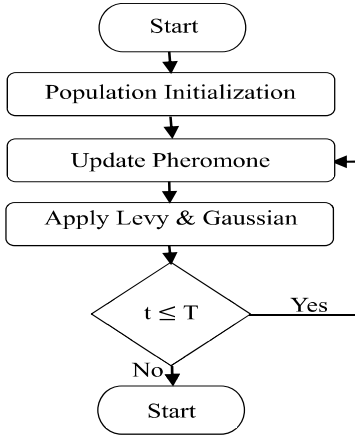


Figure 1: The EACO structure and workflow

B. Standard Ant Colony Optimization (ACO)

ACO is a bio-inspired metaheuristic algorithm that simulates the foraging behavior of real ants seeking the shortest path between their colony and food sources. Artificial ants build candidate solutions by probabilistically selecting paths based on pheromone trails and heuristic information. The pheromone level on each path is updated dynamically to reflect the quality of previously discovered solutions.

In classical ACO, the probability $P_{ij}^k(t)$ that ant k moves from node i to node j at iteration t is defined as:

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in N_i} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta} \quad (1)$$

where $\tau_{ij}(t)$ denotes the pheromone level, $\eta_{ij}(t) = 1/d_{ij}$ is the heuristic visibility, and α, β Control the relative influence of pheromone intensity and visibility. After all ants complete a solution, pheromone trails are updated as:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (2)$$

where ρ is the evaporation rate and $\Delta\tau_{ij}^k(t)$ is the pheromone deposited by ant k , typically proportional to the solution quality.

Although effective, ACO faces challenges in continuous optimization due to its limited exploration range, premature convergence, and dependence on initial pheromone settings. To mitigate these, EACO introduces Levy-based exploration and Gaussian-based mutation.

C. Levy Flight-Based Exploration Mechanism

After the solution construction phase, a Levy Flight mechanism is introduced to improve ACO's global exploration capacity. Levy Flight is a random walk process with step lengths drawn from a heavy-tailed probability distribution, allowing small local moves and occasional long jumps. It

prevents ants from being trapped in local minima by exploring new regions of the search space.

The step size s in Levy Flight follows:

$$s = \frac{u}{v^{1/\beta}}, u, v \sim N(0, \sigma^2) \quad (3)$$

where $\beta \in [1,3]$ controls the distribution's tail index, and σ is calculated as:

$$\sigma = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\beta 2^{(\beta-1)/2}} \right)^{1/\beta} \quad (4)$$

The new candidate position for each ant is updated as:

$$X_{\text{new}} = X_{\text{current}} + \alpha \times \text{Levy}(\beta) \quad (5)$$

where α is a scaling parameter that controls the step length. In EACO, Levy Flight is applied selectively to the top-performing ants during the early and mid-phases of optimization, enhancing global search diversity without disrupting convergence stability.

D. Gaussian Mutation for Local Exploitation

After global exploration through Levy Flight, Gaussian Mutation is applied in later iterations to fine-tune the best solutions. This mechanism introduces small perturbations based on a Gaussian distribution around the current solution, enabling local exploitation near optimal regions.

For a given ant solution X_i , the mutation process is defined as:

$$X'_i = X_i + N(0, \sigma_m^2) \quad (6)$$

where $N(0, \sigma_m^2)$ denotes a normal distribution with mean 0 and variance σ_m^2 . The variance σ_m decreases dynamically over iterations as:

$$\sigma_m = \sigma_{\text{max}} \left(1 - \frac{t}{T_{\text{max}}} \right) \quad (7)$$

This adaptive variance ensures wide search movements at the beginning and more focused refinements as the algorithm progresses, improving convergence accuracy. The Gaussian mutation enhances exploitation efficiency, ensuring EACO converges precisely toward global optima.

E. Adaptive Integration Strategy

The Enhanced Ant Colony Optimization (EACO) employs an adaptive integration strategy to balance exploration and exploitation dynamically throughout the optimization process. In the early stage (0–40% of iterations), the algorithm prioritizes exploration by applying Levy Flights to 40% of the ant population, which has a high step variance, thereby encouraging the discovery of diverse and promising search regions. During the middle stage (40–80%), a gradual transition occurs in which both enhancement mechanisms are adaptively

utilized ants nearer to the global best solution undergo Gaussian Mutation for fine-tuning. In contrast, others continue exploring via Levy-based random jumps. In the late stage (80–100%), the focus shifts toward exploitation, with Gaussian Mutation taking precedence to refine solutions locally around the best-so-far individuals. At the same time, pheromone updates are intensified to strengthen convergence. This adaptive strategy enables EACO to maintain diversity, prevent premature stagnation, and ensure robust convergence toward global optima.

F. Pheromone Update Strategy

The pheromone updating process in EACO is refined to incorporate solution improvement dynamics. The pheromone increment is defined as:

$$\Delta\tau_{ij}^k(t) = \frac{Q}{f(X_k) - f_{best} + \epsilon} \quad (8)$$

where Q is a constant, $f(X_k)$ is the fitness of ant k , f_{best} is the best solution's fitness, and ϵ avoids division by zero. The pheromone evaporation rate ρ is also adaptive, given by:

$$\rho(t) = \rho_{min} + (\rho_{max} - \rho_{min}) \times e^{-t/T_{max}} \quad (9)$$

This ensures that pheromone evaporation is more substantial in early stages (promoting exploration) and weaker in later stages (preserving exploitation).

G. Integration of Levy and Gaussian Mutation with ACO

The proposed Levy and Gaussian mutation-based Ant Colony Optimization, presented in Algorithm 1, retains the core structure of classical ACO but enhances its search dynamics through the use of Levy Flights and Gaussian Mutations. Instead of using a uniform random number for site selection, Levy Flight modifies the selection probability, allowing for adaptive long-distance jumps that enhance global exploration and prevent premature convergence. After each site is selected, Gaussian Mutation introduces small, normally distributed perturbations for local refinement, enhancing exploitation around promising regions. This hybrid integration achieves a balanced exploration and exploitation, resulting in improved convergence precision and robustness compared to the conventional ACO approach.

Algorithm 1: Levy ACO Candidate Selection Mechanism

- 1 Sort candidate positions by selection probability
- 2 Generate a uniform random number $P_{now} \in [0,1]$
- 3 Generate other uniform random number $P_{Levy} \in [0,1]$
- 4 **If** $P_{Levy} \geq P_{threshold}$ **then**
- 5 $P_{new} = 1 - A * \frac{1 - P_{Levy}}{1 - P_{threshold}} * P_{now}$
- 6 **Else ---** $P_{new} = P_{now}$
- 7 **End if**

- 8 Select the next position by P_{new} from the candidate list
- 9 Apply Gaussian mutation $X'_i = X_i + N(0, \sigma_m^2)$
- 1 **If** the mutated position X'_i gives a better fitness, **then**
- 0 Replace the selected position with X'_i
- 1 **Else ---** return the original position X_i
- 2
- 1 **End if**
- 3
- 1 **Return** the final position X'_i (or X_i if no improvement)
- 4

III. EXPERIMENTAL SETUP AND PERFORMANCE ANALYSIS

A CEC2022 benchmark suite was used to comprehensively compare the performance of the proposed EACO algorithm with that of four other optimization algorithms. It serves as a toolbox for testing the robustness, convergence, and generalization capabilities of new algorithms across multiple Unimodal (F1), Basic (F2-F5), Hybrid (F6-F8), and Composition (F9-F12) functions [23]. The experiments are implemented in MATLAB R2020b for consistency and sustainability. All algorithms were compared under identical conditions, such as 10-D and 20-D problem sizes, and 30 independent tests were run for 1000 iterations.

Table 1: COMPARISON OF EACO AND COMPETING ALGORITHMS ON CEC2022 (10-D)

Fn.	Metric	PSO	LSHADE	AOA	AOC	EACO (Ours)
F1	Mean	1.06E-03	6.82E-14	1.17E+04	3.49E-03	1.10E-10
	Std	1.67E-03	5.87E-14	4.04E+03	2.49E-03	1.34E-10
F2	Mean	7.33E+00	5.65E+00	8.35E+02	1.42E+01	1.02E-02
	Std	3.36E+00	3.73E+00	5.78E+02	2.40E+01	7.09E-02
F3	Mean	2.56E-04	1.17E-03	4.23E+01	2.14E-04	2.14E-04
	Std	5.48E-04	2.00E-03	4.62E+00	1.66E-04	1.66E-04
F4	Mean	2.15E+01	4.98E+00	3.28E+01	2.40E+01	1.10E+01
	Std	8.05E+00	2.97E+00	5.88E+00	1.08E+01	5.01E+00
F5	Mean	2.26E-01	8.12E-02	3.96E+02	8.12E-02	1.09E-01
	Std	4.20E-01	1.68E-01	1.05E+02	1.68E-01	1.86E-01
F6	Mean	7.47E+02	1.58E+01	2.73E+03	2.57E+03	1.34E+00
	Std	6.63E+02	6.33E+00	1.29E+03	1.88E+03	4.97E-01
F7	Mean	1.86E+01	2.70E+00	1.02E+02	1.93E+01	2.71E+00
	Std	6.34E+00	6.12E+00	2.96E+01	5.92E+00	4.59E+00
F8	Mean	1.94E+01	1.46E+01	1.29E+02	1.24E+01	1.24E+01
	Std	6.48E+00	9.70E+00	1.04E+02	6.15E+00	6.15E+00
F9	Mean	2.33E+02	2.29E+02	3.87E+02	2.29E+02	4.76E+02
	Std	4.34E-01	0.00E+00	5.29E+01	0.00E+00	7.24E+01
F10	Mean	1.46E+02	1.00E+02	2.48E+02	1.26E+02	1.00E+02
	Std	5.93E+01	9.25E-02	8.87E+01	5.31E+01	9.25E-02

F11	Mean	4.50E+01	4.51E+01	7.50E+02	7.52E+01	4.00E+01
	Std	1.01E+02	7.27E+01	2.26E+02	1.06E+02	1.26E+02
F12	Mean	1.68E+02	1.65E+02	3.15E+02	1.63E+02	1.62E+02
	Std	9.86E-01	9.64E-01	5.21E+01	1.01E+00	1.40E+00

	Std	1.15E+02	1.22E+02	8.61E+02	4.19E+01	1.20E+02
F12	Mean	3.09E+02	2.49E+02	1.14E+03	2.54E+02	2.40E+02
	Std	2.72E+01	3.39E+00	2.17E+02	1.08E+01	5.80E+00

In the 10-D experiments in Table I, the proposed EACO algorithm demonstrated outstanding performance on seven of the twelve benchmark test functions, including F2, F3, F6, F8, and F10–F12. This result reflects the generalization abilities of EACO over complex and straightforward problems. The outstanding performance of this EACO algorithm is mainly due to its Levy Flight and Gaussian mutation strategies. Therefore, the balance between exploration and exploitation can be achieved to prevent falling into local optima or premature convergence. LSHADE, however, performed well primarily in unimodal and multimodal functions F1, F4, F5, F7, F9, and F10, but lacked the flexibility to adapt as EACO did on highly complex problems. On the other hand, the classic ACO performs well on functions such as F3, F5, F8, and F9, but underperforms in cases of higher complexity, as AOA and PSO do not outperform it in any of these functions. The high-quality performance of the EACO in all CEC2022 functions in 10-D validates an improved balance between exploration and exploitation of its search pattern, as well as its better optimization capability compared to other competing algorithms.

Table II: COMPARISON OF EACO AND COMPETING ALGORITHMS ON CEC2022 (20-D)

Fn.	Metric	PSO	LSHADE	AOA	ACO	EACO (Ours)
F1	Mean	1.49E-01	2.61E+04	3.45E+04	1.49E-01	5.50E+01
	Std	2.58E-01	1.25E+04	9.62E+03	2.58E-01	5.32E+00
F2	Mean	7.09E+01	4.51E+01	1.83E+03	3.07E+01	3.07E+01
	Std	1.50E+01	5.68E+00	5.09E+02	5.58E+00	5.58E+00
F3	Mean	1.62E-01	9.51E-06	6.16E+01	9.51E-06	1.35E-02
	Std	1.20E-01	9.48E-06	8.13E+00	9.48E-06	6.33E-03
F4	Mean	8.25E+01	4.50E+01	1.44E+02	4.31E+01	4.31E+01
	Std	3.80E+01	5.32E+00	1.10E+01	1.35E+01	1.35E+01
F5	Mean	7.87E+01	8.94E+02	1.97E+03	1.04E+03	1.02E+00
	Std	7.72E+01	1.71E+02	3.78E+02	9.72E+02	1.15E+00
F6	Mean	5.68E+03	8.60E+01	8.29E+08	1.43E+04	8.60E+01
	Std	4.88E+03	5.92E+01	6.53E+08	9.12E+03	5.92E+01
F7	Mean	3.30E+01	3.07E+01	1.84E+02	7.70E+01	3.07E+01
	Std	7.72E+00	5.58E+00	3.14E+01	4.41E+01	5.58E+00
F8	Mean	2.91E+01	2.23E+01	3.57E+02	2.56E+01	2.23E+01
	Std	5.70E+00	5.68E-01	1.75E+02	5.63E+00	5.68E-01
F9	Mean	1.86E+02	1.81E+02	8.85E+02	1.81E+02	1.81E+02
	Std	9.16E-01	2.84E-01	2.65E+02	2.84E-01	1.08E-07
F10	Mean	3.25E+02	4.11E+01	3.50E+03	7.04E+02	4.11E+01
	Std	1.93E+02	5.15E+01	5.08E+02	3.22E+02	5.15E+01
F11	Mean	3.56E+02	3.38E+02	5.73E+03	3.21E+02	3.10E+02

The CEC2022 benchmark comparison on 20-D problems outlined in Table II presents the EACO algorithm as a leader, achieving better results than other compared algorithms for ten out of ten benchmark functions, F2, F4–F12. The high-quality solutions obtained by the proposed EACO are due to its developed ability to efficiently handle various types of functions, including unimodal, multimodal, and more complex hybrid functions. This capability is attributed to the integration of Levy Flight and Gaussian mutation, which enhances the exploration and exploitation of the proposed EACO algorithms. The EACO prevents premature convergence by enhancing exploration and exploitation, thereby effectively searching for a global optimum from early iterations. In comparison, LSHADE also achieves competitive results by ranking first on six functions, including F3 and F6–F10, and exhibits outstanding performance in solving complex and hybrid multimodal problems. However, despite the ability to exploit by using differential evolution strategies, LSHADE fails to explore broadly and adapt flexibly as EACO does. For some selected functions, such as F1–F4 and F9, it is observed that while classical ACO performs well, PSO and AOA are stable but cannot dominate any of the test cases. Notably, the AOA performs the weakest in all functions, indicating that it is limited to high-dimensional optimization problems.

In general, as observed in the 10-D and 20-D experiments, the proposed EACO is a competitive and flexible algorithm that outperforms others on various benchmarks. Its good performance is mainly due to the integration of Levy Flight and Gaussian mutation strategies, which improve this balance between exploration and exploitation. Although LSHADE can still be considered a robust and effective optimizer for various complex multimodal and hybrid problems, it is more domain-sensitive than EACO. On the other hand, EACO remains strong in terms of generalization and stability, even across various problem landscapes, demonstrating its flexibility in real-world high-dimensional optimization. While the performance of algorithms such as PSO and AOA is suitable for more straightforward scenarios, they become less effective when problems are complex, highlighting the benefits of the optimization approach provided by EACO.

Table III: WILCOXON P-VALUE COMPARISON OF EACO VS. FOUR ALGORITHMS ON CEC2022

Function	PSO	LSHADE	AOA	ACO
F1	2.89E-06	1.32E-07	2.89E-06	2.91E-06
F2	2.42E-06	7.09E-05	2.42E-06	2.91E-06
F3	1.18E-02	4.62E-01	2.91E-06	2.91E-06
F4	3.83E-04	8.59E-05	2.91E-06	2.67E-04
F5	4.33E-01	7.39E-02	2.91E-06	1.49E-03
F6	1.69E-05	2.91E-06	2.91E-06	2.89E-06
F7	8.59E-05	6.50E-01	2.91E-06	5.00E-2
F8	6.79E-03	5.84E-01	2.91E-06	1.07E-03
F9	2.35E-06	5.00E-02	2.42E-06	2.42E-06
F10	9.38E-01	5.00E-02	8.59E-05	1.00E+00
F11	1.07E-03	7.32E-03	4.55E-06	3.21E-02
F12	2.88E-06	2.49E-06	2.88E-06	2.28E-01
+/-	10/1/2	6/2/4	12/1/1	9/1/2

Table IV: WILCOXON P-VALUE COMPARISON OF EACO VS. FOUR ALGORITHMS ON CEC2022 (20-D)

Fn.	PSO	LSHADE	AOA	ACO
F1	2.8915e-06 2	.8915e-06 2	.8915e-06 7	.1016e-06 2
F2	2.42E-06	2.42E-06	2.91E-06	2.42E-06
F3	2.89E-06	2.91E-06	2.91E-06	2.89E-06
F4	3.21E-02	6.50E-01	2.91E-06	2.67E-04
F5	2.89E-06	2.91E-06	2.91E-06	2.91E-06
F6	2.57E-05	2.91E-06	2.91E-06	1.72E-05
F7	1.07E-03	2.57E-05	2.91E-06	3.63E-04
F8	9.41E-02	2.89E-06	2.91E-06	4.04E-02
F9	5.00E-2	2.89E-06	2.91E-06	5.00E-2
F10	3.87E-05	2.28E-01	2.91E-06	4.55E-06
F11	1.37E-01	3.80E-03	2.91E-06	9.41E-02
F12	2.89E-06	5.44E-04	2.91E-06	8.59E-05
+/-/	9/1/2	10/1/2	12/1/1	10/1/1

The Wilcoxon signed-rank test was used to statistically compare the performance of EACO with that of four other approaches on the CEC2022 10-D and 20-D benchmark problems. Since the results on a numerical scale alone are insufficient for this comparison, we used the Wilcoxon rank-sum test at a 5% specification level. The p-values of the comparisons also using EACO, listed in Table III (10-D) and Table IV (20-D), are represented by ‘+’, ‘=’, and ‘-’, which refer to better, equivalent, or worse performance than the corresponding algorithms. The proposed EACO achieved better performance than LSHADE, the classical ACO, and other genetic algorithms in optimizing test functions across different dimensions, confirming its robustness, competitiveness, and superior optimization capability across various problem dimensions.

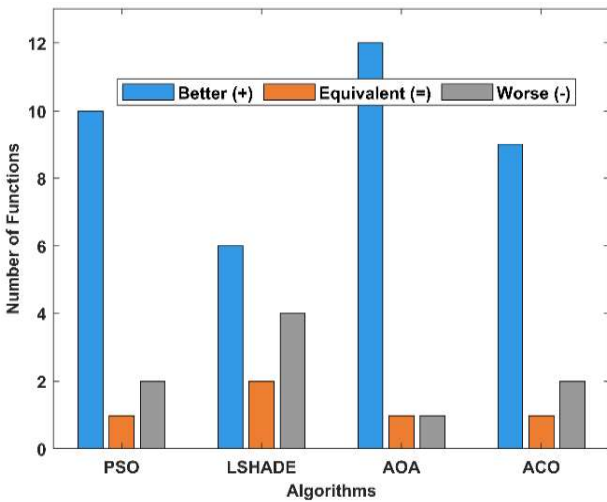


Figure 2: Wilcoxon signed-rank test 10-D

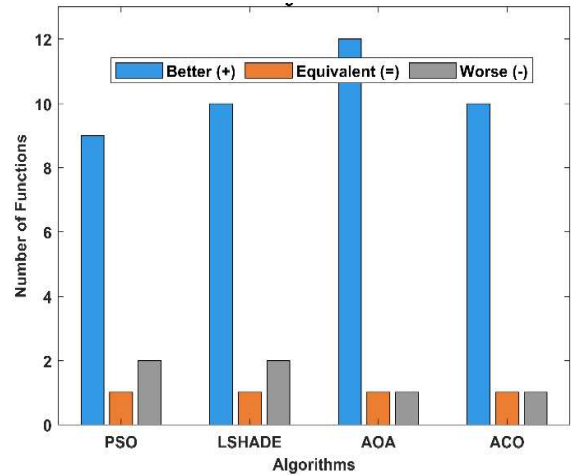


Figure 3: Wilcoxon signed-rank test 20-D

The Wilcoxon signed-rank test results presented in Figure 2 and Figure 3 provide strong evidence of the superior performance of the proposed Enhanced Ant Colony Optimization (EACO) algorithm over the compared approaches, Particle Swarm Optimization (PSO), Linear Success-History based Adaptive Differential Evolution (LSHADE), Arithmetic Optimization Algorithm (AOA), and the classical Ant Colony Optimization (ACO). Across both 10-dimensional (10-D) and 20-dimensional (20-D) benchmark problems from the CEC2022 suite, EACO consistently achieved higher counts of “+” outcomes, signifying statistically better performance. This improvement is particularly evident when compared with AOA and PSO, where EACO demonstrated notable enhancements in convergence rate, solution precision, and robustness. The relatively few “=” and “-” outcomes indicate that EACO maintained a consistently strong performance without significant degradation across different problem landscapes and dimensions. These results confirm that the incorporation of Lévy Flight and Gaussian Mutation effectively enhances ACO’s exploration and exploitation balance, allowing it to avoid premature convergence and improve overall search diversity. The statistical validation through the Wilcoxon signed-rank test further reinforces that EACO’s improvements are not by chance but are significant and consistent. Overall, the findings highlight EACO’s reliability and adaptability as a powerful optimization framework for tackling complex, high-dimensional global optimization problems.

IV. CONCLUSION

This work proposes an improved algorithm, called Lévy Flight and Gaussian Mutation Enhanced Ant Colony Optimization (EACO), to overcome the drawbacks of classical ACO, including premature convergence and the exploration-exploitation balance. Using the Lévy Flight mechanism increases exploration of the whole search space, avoiding getting stuck within local optima and promoting a more random spread across the solution space. In the meantime, some local exploitation processes optimize the promising solution, leveraging similar ideas to Gaussian Mutation, which

improve search speed and precision. Lévy Flight functions at the beginning of iterations to promote exploration, whereas Gaussian Mutation gradually becomes dominant in the latter for intensive local search. Numerical results on the CEC2022 benchmark set indicated that EACO achieved a more stable and efficient convergence process, superior solution quality, even on large-scale problems, and better robustness across different optimization landscapes compared with conventional ACO and other state-of-the-art algorithms. The Proposed EACO is a promising development in the swarm intelligence optimization technique, offering strong adaptability and scalability for high-dimensional, complex, and practical problems. Future research will focus on extending EACO to multi-objective optimization, dynamic environments, and large-scale science and engineering problems to validate its generalization and practical effectiveness.

ACKNOWLEDGMENT

We sincerely acknowledge the support of our supervisors, colleagues, and fellow researchers, whose guidance and encouragement contributed immensely to this work. We also thank our university and academic mentors for providing resources and inspiration. We appreciate the patience, motivation, and continuous support of our family and friends.

REFERENCES

- [1] "M. Dorigo and T. Stützle, Ant Colony Optimization.... - Google Scholar." Accessed: Oct. 15, 2025. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=M.+Dorigo+and+T.+St%2C3%BCtzle%2C+Ant+Colony+Optimization.+MIT+Pres%2C+2004.&btnG=
- [2] "X.-S. Yang, Nature-Inspired Optimization Algorithms.... - Google Scholar." Accessed: Oct. 15, 2025. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=X.-S.+Yang%2C+Nature-Inspired+Optimization+Algorithms.+Elsevier%2C+2014.&btnG=
- [3] S. Chusanapiputt, ... D. N.... on P. S., and undefined 2006, "Relativity pheromone updating strategy in ant colony optimization for constrained unit commitment problem," *ieeexplore. iee.org* S Chusanapiputt, D Nualhong, S Jantarang, S Phoomvuthisarn 2006 International Conference on Power System Technology, 2006•*ieeexplore.ieee.org*, Accessed: Oct. 15, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/4116341/>
- [4] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Syst Appl*, vol. 37, no. 8, pp. 5682–5687, Aug. 2010, doi: 10.1016/J.ESWA.2010.02.042.
- [5] X. Yang, ... S. D. Nature & Biologically Inspired Computing, and undefined 2009, "Cuckoo search via Lévy flights," *IEEEExplore. iee.org* XS Yang, S Deb 2009 World congress on nature & biologically inspired computing, 2009•*ieeexplore.ieee.org*, pp. 210–214, 2009, Accessed: Oct. 15, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5393690/>
- [6] Y. Liu, B. Cao, and H. Li, "Improving ant colony optimization algorithm with epsilon greedy and Levy flight," *Complex and Intelligent Systems*, vol. 7, no. 4, pp. 1711–1722, Aug. 2021, doi: 10.1007/S40747-020-00138-3/TABLES/6.
- [7] F. El Asri, C. Tajani, H. F.-M., Modeling and undefined 2023, "Investigation of ant colony optimization with Levy flight technique for a class of stochastic combinatorial optimization problem," *Science, LPNU.ua* F El Asri, C Tajani, H Fakhouri. Mathematical modeling and computing, 2023•*science.lpnu.ua*, vol. 10, no. 4, pp. 1132–1142, 2023, doi: 10.23939/mmc2023.04.1132.
- [8] S. Bansal, R. Goel, R. M.-Adv. Math. Sci. J, and undefined 2020, "An improved ant colony algorithm based on Levy flight distribution," *research-publication.com* S Bansal, R Goel, R Maini Adv. Math. Sci. J, 2020•*research-publication.com*, vol. 9, no. 6, pp. 1857–8438, 2020, doi: 10.37418/amj. 9.6.70.
- [9] R. Storn, K. P.-J. of global optimization, and undefined 1997, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Springer R Storn, K Price* Journal of global optimization, 1997•*Springer*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.
- [10] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation," 2003 IEEE Swarm Intelligence Symposium, SIS 2003 - Proceedings, pp. 72–79, 2003, doi: 10.1109/SIS.2003.1202250.
- [11] A. Ma, X. Zhang, C. Zhang, B. Z.-I. T. and Control, and undefined 2019, "An adaptive hybrid ant colony optimization algorithm for the classification problem," *itc.ktu.lt* A Ma, X Zhang, C Zhang, B Zhang Information Technology and Control, 2019•*itc.ktu.lt*, vol. 48, no. 4, pp. 590–601, 2019, doi: 10.5755/j01.itc.48.4.22330.
- [12] A. Taranto, B. P. Nunes, and R. Addie, "Survey of Continuous Ant Colony Optimization: Theory, Applications and Algorithms," *Authorea Preprints*, Sep. 2025, doi: 10.36227/TECHRXIV.175693561.14761595/V1.
- [13] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artif Intell Rev*, vol. 56, no. 11, pp. 13187–13257, Nov. 2023, doi: 10.1007/S10462-023-10470-Y/TABLES/3.
- [14] A. Kaveh and S. Talatahar, "A hybrid particle swarm and ant colony optimization for design of truss structures," 2008, Accessed: Oct. 15, 2025. [Online]. Available: <https://www.sid.ir/paper/298786/en>
- [15] H. Zhao and C. Zhang, "An ant colony optimization algorithm with evolutionary experience-guided pheromone updating strategies for multi-objective optimization," *Expert Syst Appl*, vol. 201, p. 117151, Sep. 2022, doi: 10.1016/J.ESWA.2022.117151.
- [16] S. Nimmala, M. Ramchander, M. Mahendar, P. Manasa, D. D. Bhavani, and K. Raghavendar, "Dynamic RL-ACO: Reinforcement Learning-based Ant Colony Optimization for Load Balancing in Cloud Networks," *Proceedings of the 5th International Conference on Smart Electronics and Communication, ICOSec 2024*, pp. 475–480, 2024, doi: 10.1109/ICOSec61587.2024.10722410.
- [17] H. Hakli and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Appl Soft Comput*, vol. 23, pp. 333–345, Oct. 2014, doi: 10.1016/J.ASOC.2014.06.034.
- [18] R. Sagban, H. Marhoon, ... R. A. C. E., and undefined 2020, "Hybrid bat-ant colony optimization algorithm for rule-based feature selection in health care," *alayan.edu.iq* R Sagban, HA Marhoon, R Alubady International Journal of Electrical and Computer Engineering (IJECE), 2020•*alayan.edu.iq*, vol. 10, no. 6, pp. 6655–6663, 2020, doi: 10.11591/ijece.v10i6.pp6655-6663.
- [19] R. Jensi and G. W. Jiji, "An enhanced particle swarm optimization with Levy flight for global optimization," *Appl Soft Comput*, vol. 43, pp. 248–261, Jun. 2016, doi: 10.1016/J.ASOC.2016.02.018.
- [20] K. Zhang, B. Z.-I. J. of P. Research, and undefined 2024, "A self-adaptive Gaussian mutation-based arithmetic optimiser algorithm for integrated production scheduling and vehicle routing problem in the distributed," *Taylor & Francis K Zhang, B Zhou* International Journal of Production Research, 2024•*Taylor & Francis*, vol. 62, no. 21, pp. 7952–7980, 2024, doi: 10.1080/00207543.2024.2334416.
- [21] G. Sun, Y. Lan, and R. Zhao, "Differential evolution with Gaussian mutation and dynamic parameter adjustment," *Soft Comput*, vol. 23, no. 5, pp. 1615–1642, Mar. 2019, doi: 10.1007/S00500-017-2885-Z/FIGURES/12.
- [22] W. Chu, X. Gao, and S. Sorooshian, "A new evolutionary search strategy for global optimization of high-dimensional problems," *Inf Sci (N Y)*, vol. 181, no. 22, pp. 4909–4927, Nov. 2011, doi: 10.1016/J.INS.2011.06.024.
- [23] "IEEE CEC 2022 Competition on Dynamic Optimization Problems Generated by Generalized Moving Peaks Benchmark | Request PDF." Accessed: Oct. 15, 2025. [Online]. Available: https://www.researchgate.net/publication/357575875_IEEE_CEC_2022_Competition_on_Dynamic_Optimization_Problems_Generated_by_Generalized_Moving_Peaks_Benchmark