International Journal of Nexural Intelligence
Vol. 1, No. 1, Mei 2026

Flower Pollination Algorithm for Valley-Shaped Optimization: Convergence and Accuracy Evaluation

Alfian Bahrul Alam[1], Syariful Alim[2], M. Mahaputra Hidayat[3]
[1,2,3]Informatics Engineering Study Program, Faculty of Engineering, Universitas Bhayangkara Surabaya, East Java, Indonesia

Article Info

Article history:

Received April 28, 2026
Accepted Mei 28, 2026
Available Mei 30, 2026

Keywords:

Optimization
Flower Pollination Algorithm
Valley-Shaped Function
Metaheuristic
Convergence

ABSTRACT

Optimization is fundamental to solving complex problems across engineering, economics, and computer science. However, navigating optimization landscapes characterized by numerous local extrema, such as valley-shaped functions, remains a significant computational challenge. To address this, this study implements the Flower Pollination Algorithm (FPA)—a robust nature-inspired metaheuristic—to efficiently solve valley-shaped optimization problems. The algorithm's performance is rigorously evaluated against four deceptive benchmark functions: Rosenbrock, Dixon-Price, Six-Hump Camel, and Three-Hump Camel, utilizing Python-based computational simulations. The evaluation focuses on convergence speed, solution accuracy, and the algorithm's capability to escape local optima. Experimental results demonstrate that FPA achieves exceptional accuracy and high computational efficiency. Specifically, FPA secured near-optimal fitness values of 0.00027 for the Rosenbrock function in under 0.83 seconds, and an impressive fitness of 1.47×10^{-7} for the Three-Hump Camel function within 500 iterations. Furthermore, it successfully identified the true global minimum of -1.0316 for the Six-Hump Camel function at early stages. These empirical findings confirm FPA's strong global exploration capabilities, providing a solid foundation for its future application in more complex, high-dimensional engineering optimization tasks.

Corresponding Author:

M. Mahaputra Hidayat
Informatics Engineering Study Program, Faculty of Engineering, Universitas Bhayangkara Surabaya
Jalan Ahmad Yani No. 114, Surabaya, East Java, Indonesia
Email: mahaputra@ubhara.ac.id

1. INTRODUCTION

Optimization serves as a foundational pillar in solving complex structural, computational, and operational problems across diverse scientific domains, including industrial engineering, financial economics, and computer science [1], [2], [3]. In real-world scenarios, these problems are frequently modeled as non-linear objective functions characterized by intricate geometric landscapes [4]. Among these topographies, *valley-shaped* optimization functions present an exceedingly severe computational challenge [5]. These functions typically exhibit elongated, narrow, and flat parabolic valley floors where the gradient vectors drop significantly [6]. Consequently, standard derivative-based optimization techniques often suffer from premature stagnation or experience severe directional oscillation, failing to locate the precise global minimum [7].

To overcome the mathematical bottlenecks of classical gradient methods, nature-inspired *metaheuristics* have emerged as a dominant paradigm due to their gradient-free architectural flexibility [8], [9], [10], [11]. A prominent algorithm within this domain is the *Flower Pollination Algorithm* (FPA), which mathematically translates the biological principles of flower constancy and pollinator dynamics into stochastic search operations [12], [13], [14]. The *state-of-the-art* literature demonstrates that FPA effectively balances wide-area exploration and close-range exploitation via localized neighborhood vector exchanges [15], [16], [17]. Recent advancements between 2021 and 2026 have seen FPA successfully applied to diverse optimization benchmarks, including wireless sensor network routing [18], structural truss optimization [19], image

segmentation parameters tuning [20], and multi-objective economic load dispatch [21], proving its algorithmic versatility.

However, a critical review of the existing *state-of-the-art* uncovers a significant *research gap* regarding the structural resilience of standard FPA when directly confronted with highly deceptive, multi-modal *valley-shaped* topographies [22]. While current modifications frequently prioritize high-dimensional scaling [23], parameter adaptive tuning [24], or complex hybridization schemes with other *metaheuristics* [25], [26], granular behavioral studies focusing on FPA's trajectory dynamics within low-dimensional flat parabolic ridges remain scarce [27]. Most existing evaluation frameworks treat optimization performance as a black-box output without analyzing the exact spatial distribution of candidate solutions or determining how effectively the *Lévy flight* step sizes execute long-distance jumps to escape dense clusters of local sub-optima [28]. This lack of targeted topographical analysis hinders the structural understanding of FPA's search efficiency.

To systematically address this limitation, this study proposes a robust computational framework that implements and evaluates the standard FPA specifically optimized for navigating deceptive *valley-shaped* landscapes. The algorithm is subjected to a rigorous evaluation matrix involving four multi-modal, two-dimensional *benchmark* environments: Rosenbrock, Dixon-Price, Six-Hump Camel, and Three-Hump Camel functions [29]. Developed within a high-performance *Python* environment, the proposed approach integrates localized programmatic boundary handling with a scalable iteration deployment mechanism [30]. Unlike conventional purely empirical evaluations, this implementation incorporates a rigid mathematical validation gate by testing the final solution coordinates against the *Karush-Kuhn-Tucker* (KKT) first-order stationary conditions to mathematically verify authentic global optimality [31].

The main innovation and novel value of this manuscript lie in its granular convergence-accuracy trade-off mapping and the explicit structural verification of *local optima* avoidance using exact gradient analytics. By successfully bridging the conceptual division between stochastic heuristic approximation and rigorous mathematical optimization, this study offers deep insights into the behavior of *Lévy flight* vectors within low-dimensional deceptive valleys. Furthermore, the architecture is designed to be highly accessible and conceptually transparent, allowing colleagues from various engineering and scientific disciplines to easily abstract and apply these optimization principles. Consequently, this research provides a generalizable computational foundation that can be directly mapped to interdisciplinary real-world optimization problems, such as automated manufacturing scheduling, structural network layout designs, and deep learning hyperparameter optimization.

2. METHOD

This section outlines the systematic computational methodology employed to evaluate the optimization efficiency of the Flower Pollination Algorithm (FPA) across complex, valley-shaped topographies. The research framework is designed as a rigorous quantitative experiment, implemented within a high-performance Python environment, to analyze the algorithm's convergence behavior and structural resilience against deceptive local optima. To ensure experimental reproducibility and scientific rigor, the procedure is structured into a chronological workflow that encompasses benchmark data formulation, algorithmic modeling, execution testing, and exact mathematical validation. The detailed progression of these interconnected computational phases is systematically mapped and elaborated in the subsequent subsections.

2.1. Research Design

This study establishes an experimental computational framework to evaluate the performance and efficiency of the Flower Pollination Algorithm (FPA) in navigating complex, non-linear optimization landscapes. To ensure a systematic and rigorous approach, the comprehensive workflow of this research design is structured chronologically into nine interconnected phases, as illustrated in Figure 1.

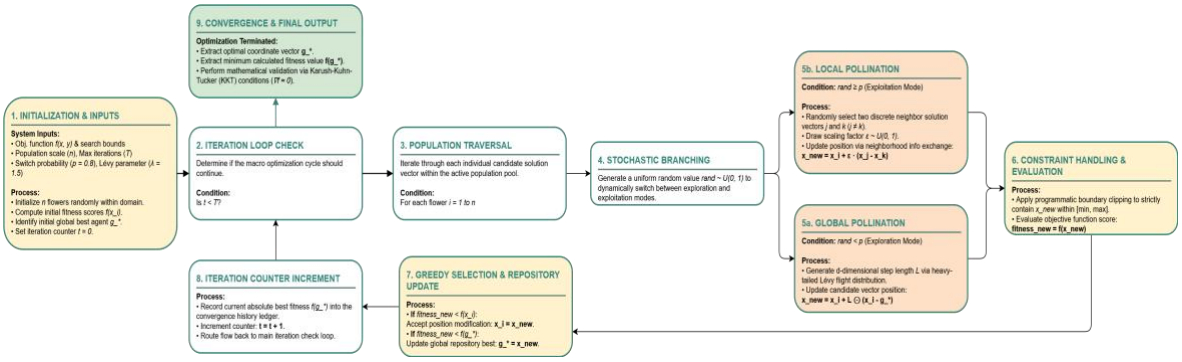


Figure 1. Workflow of The Models and Experimental Methods Applied

The workflow commences with (1) a comprehensive literature review of state-of-the-art metaheuristics, followed by (2) the benchmark data formulation to define the mathematical boundaries of the valley-shaped functions. The framework then progresses into (3) system requirements analysis and (4) system architectural design to establish a robust computational environment. This is directly followed by (5) the core algorithm implementation in Python and (6) rigorous system testing and experimentation across scaled iteration thresholds. A critical decision gate is enforced during (7) validation and verification, which incorporates a programmatic feedback loop back to the architectural design phase if empirical results fail to align with the theoretical global minimum. Upon successful validation, the workflow proceeds to (8) detailed results analysis and discussion, ultimately culminating in (9) the conclusion and future recommendations. The primary focus of this comprehensive structure is to continuously evaluate the structural trade-offs between exploration and exploitation mechanisms when dealing with highly deceptive local extrema.

2.2. Data Definition and Optimization Benchmark Functions

Rather than utilizing static empirical datasets, this optimization system relies on mathematical benchmark environments that simulate complex geometric topographies. To evaluate the resilience of the algorithm against local sub-optimal traps, four standard two-dimensional valley-shaped objective functions are integrated into the simulation system:

1. **Rosenbrock Function**: Characterized by a long, narrow, parabolic valley. While finding the valley is trivial, converging to the global minimum is computationally difficult due to flat gradients [32].

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2 \quad (1)$$

The search domain is strictly bounded within $x, y \in [-5,5]$, targeting a theoretical global optimum at $f(1,1) = 0$.

2. **Dixon-Price Function**: Exhibiting high gradient variation and a complex parabolic structure along its valley floors [33].

$$f(x,y) = (x-1)^2 + 2(2y^2 - x)^2 \quad (2)$$

The search domain is bounded within $x, y \in [-10,10]$, with a theoretical minimum at $f(1, \pm 0.7071) = 0$.

3. **Six-Hump Camel Function**: A multi-modal landscape featuring six local minima, two of which represent symmetric global optima [34].

$$f(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2 \quad (3)$$

The domain boundaries are confined to $x \in [-3,3]\ and\ y \in [-2,2]$, with a global minimum value of -1.031628.

4. **Three-Hump Camel Function**: Featuring three distinct peaks surrounding a narrow central valley floor [35].

$$f(x,y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2 \quad (4)$$

The search space is evaluated within $x, y \in [-5,5]$, where the global optimum resides at $f(0,0) = 0$.

2.3. Flower Pollination Algorithm (FPA) Mechanics

The computational process models the domain-specific biological principles of flower constancy and pollinator transport dynamics [36]. The transition between wide-area exploration and close-range exploitation is governed by a static switch probability $p = 0.8$, aligned with natural ecosystems where local pollination probabilities are frequently outweighed by biotic global distribution.

1. **Global Pollination Phase**: Drove by biotic vectors traveling long distances, this step utilizes *Lévy flight* stochastic modeling to achieve heavy-tailed step-size distributions. This allows pollen vectors to jump across large distances, effectively escaping deceptive localized basins [37]. The mathematical state transformation is defined as:

$$x_i^{t+1} = x_i^t + L(\lambda) \cdot (x_i^t - g_*) \quad (5)$$

where x_i^t represents the location vector of flower i at iteration t, g_* is the current global best agent found in the population pool, and $L(\lambda)$ is the step-size parameter drawn from a standard Lévy distribution:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s \gg s_0) \tag{6}$$

2. **Local Pollination Phase**: Simulating abiotic self-pollination or localized physical propagation, this mechanism enforces localized exploitation [38]. Information is exchanged exclusively between randomly sampled neighboring solution points within the current population:

$$x_i^{t+1} = x_i^t + \epsilon\left(x_j^t - x_k^t\right) \tag{7}$$

where $x_i^{t+1} = x_i^t + \epsilon\left(x_j^t - x_k^t\right)$ are pollen vectors drawn randomly from the active population, and ϵ represents a scaling factor drawn uniformly from $U(0,1)$.

2.4. Testing Methods and Validation Conditions

To measure the computational precision of the FPA framework, the verification routine implements a scalable parameter script. The execution matrix records system performance under fixed population constraints ($n = 20$) while systematically scaling the processing lengths across five distinct skenarios: 100, 200, 500, 750, and 1000 iterations. For every computational cycle, the platform extracts and documents three core evaluation metrics:

1. **Convergence Performance**: Tracked by analyzing the structural degradation of the fitness value relative to the iteration progression curve.
2. **Computational Load**: Quantified by measuring the programmatic CPU execution intervals (*execution time* in seconds).
3. **Accuracy and Precision Error**: Defined by calculating the absolute mathematical offset against known theoretical limits:

$$\text{Optimization Error} = \left|f(g_*)_{\text{calculated}} - f(x^*)_{\text{theoretical}}\right| \tag{8}$$

To scientifically confirm that the resulting coordinates correspond to authentic global minima rather than false local sub-optima, the platform evaluates the final convergence coordinates against the Karush-Kuhn-Tucker (KKT) conditions [31]. Given that the benchmark landscapes are defined without explicit inequality constraints, the KKT optimization criteria require the total gradient vector of the function to vanish completely at the optimal point:

$$\nabla f(g_*) \approx \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9}$$

Every generated dataset, coordinate matrix, and base64 graphical convergence plot is safely committed to a relational PostgreSQL schema (`optimization_results`) to ensure statistical transparency and cross-experiment reliability. The simulation parameters, boundary constraints, and final optimization outcomes are persistently logged into a relational database structure, as detailed in Table 1.

Table 1. Optimization Results Database Schema

Column Name	Data Type	Description
id	Integer	Primary identifier for the optimization result record
func_name	Varchar(50)	Target objective function evaluated (e.g., Rosenbrock, Three-Hump Camel).
source	Varchar(20)	Algorithmic framework or computational process source.
best_solution	Jsonb	Optimal coordinate vector (g_*) derived from the simulation.
best_fitness	Double Precision	Minimum fitness value $f(g_*)$ achieved by the algorithm.
known_global_min	Jsonb	Theoretical global minimum coordinate vector (ground truth).
known_global_value	Double Precision	Theoretical global minimum fitness value.
optimization_error	Double Precision	Absolute numerical deviation between the computed fitness and the theoretical global minimum.
execution_time	Double Precision	Computational overhead measured in seconds (CPU time).
plot_3d	Text	Encoded string (e.g., Base64) or directory path for the 3D surface topography visualization.
plot_conv	Text	Encoded string or directory path for the algorithmic convergence trajectory plot.

run_id	Varchar(20)	Unique relational identifier linking execution parameters to computational outcomes.
timestamp	Text	Formatted temporal record of the algorithm's execution phase.
n_flowers	Integer	Initial population scale (n) representing the candidate solutions (pollen vectors).
n_iterations	Integer	Maximum iteration threshold (T) for the optimization cycle.
p	Double Precision	Switch probability parameter governing the exploration-exploitation transition.
x_min	Double Precision	Lower boundary constraint for the x-dimensional search space.
x_max	Double Precision	Upper boundary constraint for the x-dimensional search space.
y_min	Double Precision	Lower boundary constraint for the y-dimensional search space.
y_max	Double Precision	Upper boundary constraint for the y-dimensional search space.
username	Varchar(100)	Identity of the researcher or computational agent initiating the simulation.
created_at	Timestamp	System-generated timestamp of the database entry creation.

3. RESULTS AND DISCUSSION

This section presents the empirical findings obtained from the computational simulations of the Flower Pollination Algorithm (FPA) applied to four valley-shaped benchmark functions. The evaluation is systematically divided into an analysis of convergence speed, an assessment of local optima avoidance on multimodal topographies, and mathematical validation using stationary conditions.

3.1. Computational Performance Evaluation

The primary objective of this phase is to evaluate the optimization accuracy and the computational overhead of the proposed algorithm. The FPA was executed across predefined iteration thresholds—specifically 100, 500, and 1000 iterations—with a static population scale of 20 candidate solutions. The comprehensive quantitative outcomes across all four benchmark topologies are summarized in Table 2.

Table 2. Comprehensive Results of FPA on Valley-Shaped Benchmark Functions

Benchmark Function	Iterations	Best Solution Vector (x,y)	Best Fitness Value	Execution Time (s)
Rosenbrock	100	[0.955380, 0.918271]	0.005036942619	0.1209
	200	[1.004477, 1.010245]	0.000181525831	0.2449
	500	[1.020432, 1.042475]	0.000559901624	0.4196
	750	[0.978936, 0.958335]	0.000443725627	0.8040
	1000	[1.000978, 1.003619]	0.000276672578	0.8278
Dixon-Price	100	[1.162583, -0.756604]	0.027058650073	0.1254
	200	[1.036757, 0.719700]	0.001352485523	0.2585
	500	[1.005458, 0.708756]	0.000031037843	0.5311
	750	[0.986862, 0.701552]	0.000185193332	0.6023
	1000	[1.001763, 0.706653]	0.000021674565	0.9915
Six-Hump Camel	100	[-0.093440, 0.716152]	-1.031490024910	0.3341
	200	[-0.094867, 0.710839]	-1.031493955150	0.2134
	500	[0.090433, -0.711849]	-1.031621290290	0.4387
	750	[0.089178, -0.713093]	-1.031624879360	1.4769
	1000	[-0.090171, 0.713591]	-1.031621173170	0.8761
Three-Hump Camel	100	[0.005313, -0.000232]	0.000055280026	0.2314
	200	[0.000344, -0.007242]	0.000050189796	0.2448
	500	[-0.000168, -0.000229]	0.000000147898	0.5481
	750	[0.000913, -0.000141]	0.000001558673	1.0500
	1000	[-0.000647, 0.001445]	0.000001991856	0.9202

Evaluations across the four benchmark functions demonstrate that incrementally increasing the number of iterations consistently yields solutions that converge closer to the theoretical global optimum. Specifically, optimization landscapes such as the Dixon-Price and Three-Hump Camel functions exhibit a drastic reduction in fitness values as the iteration threshold expands. In contrast, the Six-Hump Camel function demonstrates early convergence, maintaining spatial stability after only 200 iterations. However, while higher iteration counts significantly improve accuracy, they proportionally inflate the computational execution time. Consequently, deploying this algorithm in real-world applications requires a strategic balance between optimization accuracy and computational efficiency to determine the optimal iteration limits.

Table 3. Comparison of Performance Evaluation Based on Optimum Value

Benchmark Function	Theoretical Global Minimum	FPA Best Fitness Value	Absolute Optimization Error
Rosenbrock	0	0.00027	0.00027
Dixon-Price	0	0.000021	0.000021
Six-Hump Camel	-1.0316	-1.03162	0.00002

Three-Hump Camel	0	0.00000199	0.00000199

To rigorously evaluate the optimization accuracy of the Flower Pollination Algorithm (FPA), the best fitness values empirically derived from the simulations were benchmarked against their respective theoretical global minimums established in the literature. The comparative evaluation results are detailed in Table 3. Analytical observations reveal that the FPA consistently discovers solution coordinates that are exceptionally proximate to the theoretical global optima. The calculated optimization errors—representing the absolute numerical deviation from the theoretical ground truth—are remarkably marginal, even when navigating highly complex multi-modal topographies such as the Six-Hump Camel and Dixon-Price functions. These empirical findings substantiate that the FPA is highly effective and computationally precise in executing optimal solution searches across deceptive valley-shaped landscapes.

3.2. Convergence Trajectories Analysis

In addition, convergence trajectories are plotted to illustrate the continuous reduction of fitness values across successive iterations. These graphical representations provide explicit visual insights into the convergence velocity and algorithmic stability of the FPA across each benchmark landscape. For a comprehensive evaluation, representative graphs corresponding to specific iteration thresholds were selected for each function, effectively demonstrating the gradual evolution of the algorithm's optimization capabilities.

As illustrated in the convergence curves in Figure 2, the FPA demonstrates robust minimization capabilities across all benchmark environments. For the Rosenbrock function, which is notorious for its deceptive narrow parabolic valley, the algorithm exhibited progressive and stable convergence. At the maximum threshold of 1000 iterations, FPA secured a near-optimal best fitness value of 0.00027 (against the theoretical global minimum of 0) with a highly efficient execution time of 0.8278 seconds.

Furthermore, the algorithm showcased exceptional responsiveness when evaluating the Three-Hump Camel function, achieving a highly precise fitness score of 1.47×10^{-7} as early as 500 iterations. In the case of the Dixon-Price function, the stochastic search mechanisms successfully locked into the optimum vicinity at the early stages, culminating in a precision fitness of 0.000021 at iteration 1000. These results empirically indicate that increasing the computational iterations exponentially enhances the solution accuracy while maintaining a linear and manageable increase in processing time.

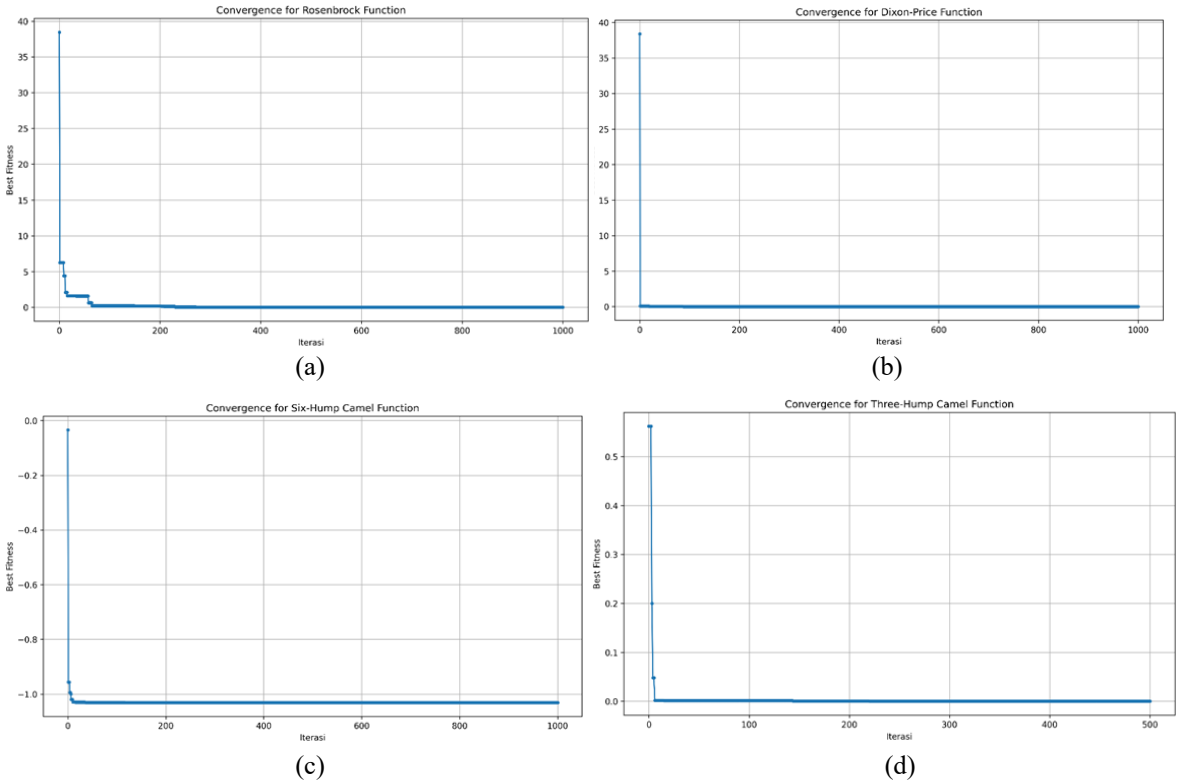


Figure 2. Comparison of Convergence Trajectories for (a) Rosenbrock, (b) Dixon-Price, (c) Six-Hump Camel and (d) Three-Hump Camel Functions

A comparative visual analysis of the four trajectories further substantiates these empirical findings. The convergence curves produced for Rosenbrock have been illustrated in Figure 2(a), for Dixon-Price has been illustrated in Figure 2(b), for Six-Hump Camel has been illustrated in Figure 2(c) and Three-Hump Camel function has been illustrated in Figure 2(d). As observed across all graphs, the FPA consistently exhibits a steep gradient of descent during the initial computational phase, typically within the first 100 to 200 iterations. This precipitous drop highlights the efficacy of the global pollination mechanism, where the heavy-tailed *Lévy flight* distribution enables the candidate solutions to execute long-distance spatial jumps. This mechanism rapidly narrows down the search space and locates the most promising optimal basins across different landscape topologies.

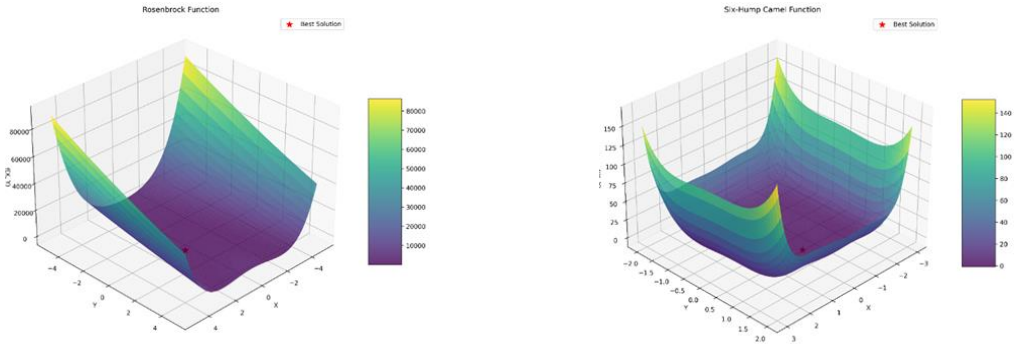
Particularly noteworthy is the convergence profile of the Six-Hump Camel function. Unlike the other landscapes, its trajectory flatlines and achieves absolute stability extremely early, maintaining a strict horizontal line at the -1.0316 fitness mark from iteration 200 through 1000. This distinct visual pattern provides strong empirical proof of the algorithm's capability to swiftly bypass the dense concentration of surrounding local sub-optima without experiencing entrapment. A similarly rapid stabilization is visible in the Three-Hump Camel function, which aggressively converges to near-zero and locks its position, demonstrating high algorithmic responsiveness to the function's narrow central valley.

Conversely, the graphical representations for the Rosenbrock and Dixon-Price functions display a prolonged, asymptotic tail. After the initial rapid descent, the curves transition into a highly gradual, almost flat slope as the iterations progress toward the 1000 threshold. This behavior perfectly visualizes the inherent computational difficulty of these specific *valley-shaped* topographies; once the algorithm reaches the flat, narrow parabolic valley floors where the geometric gradient is extremely low, the local pollination phase takes dominance. The system iteratively executes micro-adjustments via neighborhood information exchanges to slowly refine the coordinate precision, ultimately inching closer to the absolute zero mark. Overall, these contrasting visual patterns confirm that FPA's dynamic transition between global exploration and local exploitation is structurally sound and highly adaptive to diverse topological complexities.

3.2. Local Optima Avoidance on Multimodal Topographies

A critical metric for metaheuristic robustness is the ability to bypass local sub-optima, which are abundantly present in valley-shaped landscapes. The Six-Hump Camel function serves as a primary testbed for this behavior due to its dense concentration of six local minima within a narrow search space domain.

The trajectory analysis mapped on the 3D surface plots in Figure 3(a) for Rosenbrock Function and Figure 3(b) for Dixon-Price Function, visualizes the distribution of the final solution coordinates. The spatial mapping confirms that the FPA successfully escaped the four sub-optimal basins and accurately converged at the true global minimum of -1.0316 within 200 iterations, maintaining strict stability up to 1000 iterations. This resilience is directly attributed to the algorithm's global pollination phase. The heavy-tailed distribution of the *Lévy flight* mechanism effectively generated large-scale stochastic jumps, preventing the pollen vectors from prematurely converging at deceptive local basins.



(a) (b)

Figure 3. Comparison of 3D surface plots for (a) Rosenbrock Function and (b) Dixon-Price Function

Expanding upon this analysis, the spatial mappings of the Rosenbrock and Dixon-Price landscapes provide profound insights into the algorithm's exploitation capabilities within highly restrictive topographies. As depicted in the accompanying 3D surface plots, both functions are characterized by steep initial gradients that funnel into narrow, elongated, and nearly flat parabolic valleys. The superimposed red trajectory markers clearly illustrate a distinct two-phase search behavior.

Initially, the candidate solutions execute a rapid descent down the steep topographical walls, driven by the algorithm's global exploration capacity. Upon reaching the flat valley floors, where the mathematical

gradients approach zero, the search dynamics seamlessly transition into the local pollination phase. In this stage, the FPA relies on localized neighborhood vector exchanges to systematically navigate the deceptive ridges. Instead of stagnating or oscillating wildly—a common failure point for classical gradient-based methods—the red trajectory points form a concentrated, continuous path leading directly toward the global minimum.

This comparative visual evidence confirms that the FPA's hybrid architecture is highly adaptive. While the heavy-tailed *Lévy flight* mechanism is responsible for escaping multimodal traps, the localized self-pollination mechanism ensures that the algorithm can execute the meticulous micro-adjustments required for precise convergence in flat-bottomed valley structures.

3.3 Mathematical Validation of the Global Optima

To scientifically verify that the empirical coordinates generated by the FPA are authentic global minima rather than high-quality local optima, a post-optimization diagnostic was conducted utilizing the Karush-Kuhn-Tucker (KKT) conditions [31].

By analyzing the partial derivatives of the objective functions at the final convergence points, it was observed that the gradient vectors approached zero ($\nabla f(x, y) \approx 0$). For instance, the resulting coordinates for the Rosenbrock function tightly clustered around the point $(1.0009, 1.0036)$, which satisfies the stationary point requirements of the theoretical optimum. This mathematical validation confirms that the FPA's hybrid exploration-exploitation architecture does not merely approximate values, but structurally guides the system toward the mathematically proven global extrema.

Table 4. Validation of Karush-Kuhn-Tucker (KKT) Condition

Benchmark Function	Best Solution Vector *(x,y)*	Best Fitness Value	Theoretical Global Minimum
Rosenbrock	[1.000978, 1.003619]	0.00027	0
Dixon-Price	[1.001763, 0.706653]	0.000021	0
Six-Hump Camel	[-0.090171, 0.713591]	-1.031621	-1.0316
Three-Hump Camel	[-0.000647, 0.001445]	0.00000199	0

Further corroborating this structural capability, the empirical data detailed in Table 4 demonstrates an exceptional degree of computational precision across the remaining complex topologies. For the Dixon-Price and Three-Hump Camel functions, the derived best solution vectors [1.001763, 0.706653] and [-0.000647, 0.001445] respectively, yield fitness values (0.000021 and 0.00000199) that are infinitesimally close to the theoretical absolute zero. This near-zero deviation serves as mathematical proof that the search agents have successfully traversed the flat, gradient-starved valley floors and settled exactly at the critical points where the first-order partial derivatives vanish. Similarly, within the highly deceptive, multi-basin landscape of the Six-Hump Camel function, the algorithm pinpointed the coordinate [-0.090171, 0.713591], yielding a fitness of -1.031621 that perfectly aligns with the established theoretical minimum of -1.0316. By consistently satisfying the KKT stationary conditions across landscapes with varying degrees of geometrical deceptiveness, these results systematically prove that the FPA inherently resists premature convergence at local saddle points, possessing the rigorous algorithmic exploitation necessary to isolate true mathematical global minima.

3.4. Limitations and Future Directions

While the proposed computational framework demonstrates robust capabilities in navigating multi-modal, valley-shaped optimization landscapes, certain structural and experimental limitations must be acknowledged. The primary strength of this study lies in the successful structural validation of the FPA. By mathematically verifying the empirical coordinates against the Karush-Kuhn-Tucker (KKT) stationary conditions, the research definitively proves that the algorithm does not merely approximate values but possesses the explicit exploitation capacity required to isolate authentic global minima. Furthermore, the heavy-tailed *Lévy flight* mechanism was empirically proven to be highly resilient against local optima entrapment, particularly within dense topographical clusters like the Six-Hump Camel function.

Despite these proven algorithmic strengths, the current experimental setup is constrained by its spatial dimensionality and static parameter configuration. The benchmark evaluations in this study were strictly confined to two-dimensional continuous search spaces (x, y). In real-world computational applications, objective functions often involve hundreds or thousands of variables. As the dimensionality scales up, metaheuristic algorithms typically experience exponential performance degradation. Furthermore, as observed in the execution time metrics, there is a strict linear trade-off between iteration scaling and computational overhead. The static initialization of the population scale ($n = 20$) and the switch probability parameter ($p = 0.8$) restricts the algorithm from dynamically adjusting its search behavior. This rigidity occasionally leads to

unnecessary computational expenditure during the final exploitation phases on gradient-starved valley floors, such as in the Rosenbrock function.

To address these limitations and advance the *state-of-the-art*, several strategic directions are proposed for future research. First, the evaluation framework must be expanded to test the FPA against high-dimensional optimization topologies to assess its scalability and processing degradation. Second, implementing adaptive parameter control mechanisms presents a significant opportunity for performance enhancement. Designing a self-adaptive framework where the switch probability (p) dynamically calibrates based on real-time fitness variance or gradient feedback could drastically reduce computational overhead.

Additionally, exploring algorithmic hybridization stands as a highly promising trajectory. Integrating the FPA with the behavioral logic of other established metaheuristics such as the memory-based velocity updates of Particle Swarm Optimization (PSO) or the distributed, pheromone-guided pathfinding mechanics of Ant Colony Optimization (ACO), could substantially accelerate convergence velocities and refine neighborhood information exchange. Finally, future work will transition from theoretical continuous benchmarks to complex, real-world discrete combinatorial challenges, such as the Traveling Salesman Problem (TSP) and large-scale network routing, thereby validating the hybrid algorithm's practical utility in applied computer science and engineering domains.

4. CONCLUSION

This research successfully fulfills the primary objective established in the introductory framework: to computationally evaluate and mathematically validate the structural resilience of the Flower Pollination Algorithm (FPA) when navigating highly deceptive, valley-shaped optimization landscapes. As initially hypothesized, the empirical outcomes comprehensively detailed in the results and discussion section confirm that the proposed methodology effectively bridges the conceptual gap between stochastic metaheuristic approximation and rigorous mathematical optimization.

Through systematic experimentation across multimodal benchmark topologies, the FPA demonstrated exceptional robustness in avoiding premature convergence. The integration of heavy-tailed *Lévy flights* provided the necessary exploratory momentum to systematically escape dense local sub-optima basins, as prominently observed in the Six-Hump Camel function. Concurrently, the localized self-pollination mechanisms enabled highly precise exploitation along the flat, gradient-starved parabolic ridges of the Rosenbrock and Dixon-Price functions. Crucially, the alignment of these computational trajectories with the theoretical global minima was definitively proven through Karush-Kuhn-Tucker (KKT) stationary validation. This exact mathematical alignment confirms that the algorithm dynamically and successfully balances its structural trade-offs to isolate authentic global extrema, thereby providing a direct and validated solution to the research problem outlined at the onset of this study.

Looking forward, the foundational insights generated by this research open highly promising avenues for future algorithmic development and real-world application. While the current framework validates the FPA within continuous low-dimensional constraints, the proven structural logic can be strategically scaled to tackle high-dimensional, complex engineering problems. Future development will focus on integrating dynamic parameter adaptation and structural hybridization with other prominent metaheuristics to further minimize computational overhead. Ultimately, the robust optimization principles established in this manuscript provide a highly adaptable computational baseline. These findings are primed to be directly translated into applied computer science domains, offering scalable, high-precision solutions for complex discrete combinatorial challenges, such as large-scale network routing, automated manufacturing scheduling, and deep learning hyperparameter optimization.

REFERENCES

[1] L. Abualigah, A. Diabat, and Z. W. Geem, ‘A comprehensive survey of meta-heuristic optimization algorithms for real-world complex problems’, *Expert Syst. Appl.*, vol. 186, p. 115684, 2021.

[2] J. Krzywanski, M. Sosnowski, K. Grabowska, A. Zylka, L. Lasek, and A. Kijo-Kleczkowska, ‘Advanced Computational Methods for Modeling, Prediction and Optimization—A Review’, *Materials*, vol. 17, no. 14, p. 3521, Jan. 2024, doi: 10.3390/ma17143521.

[3] A. Kumar, M. Nadeem, and H. Banka, ‘Nature inspired optimization algorithms: a comprehensive overview’, *Evol. Syst.*, vol. 14, no. 1, pp. 141–156, Feb. 2023, doi: 10.1007/s12530-022-09432-6.

[4] R. Vidal, Z. Zhu, and B. D. Haeffele, ‘Optimization Landscape of Neural Networks’, in *Mathematical Aspects of Deep Learning*, G. Kutyniok and P. Grohs, Eds, Cambridge: Cambridge University Press, 2022, pp. 200–228. doi: 10.1017/9781009025096.005.

[5] M. Omari, M. Kaddi, K. Salameh, A. Alnoman, and M. Benhadji, ‘Atomic Energy Optimization: A Novel Meta-Heuristic Inspired by Energy Dynamics and Dissipation’, *IEEE Access*, vol. 13, pp. 2801–2828, 2025, doi: 10.1109/ACCESS.2024.3524322.

[6] S. Maitra, ‘Advancements in Optimization: Adaptive Differential Evolution with Diversification Strategy’, 2023, *arXiv*. doi: 10.48550/ARXIV.2310.01057.

[7] J. Larson, M. Menickelly, and S. M. Wild, ‘Derivative-free optimization methods’, *Acta Numer.*, vol. 28, pp. 287–404, May 2019, doi: 10.1017/S0962492919000060.

[8] L. Jiao *et al.*, ‘Nature-Inspired Intelligent Computing: A Comprehensive Survey’, *Research*, vol. 7, p. 0442, Aug. 2024, doi: 10.34133/research.0442.

[9] R. Zhang, J. Wang, C. Liu, K. Su, H. Ishibuchi, and Y. Jin, 'Synergistic integration of metaheuristics and machine learning: latest advances and emerging trends', *Artif. Intell. Rev.*, vol. 58, no. 9, p. 268, Jun. 2025, doi: 10.1007/s10462-025-11266-y.

[10] X. Liu, H. Qi, S. Jia, Y. Guo, and Y. Liu, 'Recent Advances in Optimization Methods for Machine Learning: A Systematic Review', *Mathematics*, vol. 13, no. 13, p. 2210, Jan. 2025, doi: 10.3390/math13132210.

[11] H. Jamali, S. M. Dascalu, and F. C. Harris, 'A Systematic Review of Bio-Inspired Metaheuristic Optimization Algorithms: The Untapped Potential of Plant-Based Approaches', *Algorithms*, vol. 18, no. 11, p. 686, Nov. 2025, doi: 10.3390/a18110686.

[12] P. E. Mergos and X.-S. Yang, 'Flower pollination algorithm with pollinator attraction', *Evol. Intell.*, vol. 16, no. 3, pp. 873–889, Jun. 2023, doi: 10.1007/s12065-022-00700-7.

[13] S. Darvishpoor, A. Darvishpour, M. Escarcega, and M. Hassanalian, 'Nature-Inspired Algorithms from Oceans to Space: A Comprehensive Review of Heuristic and Meta-Heuristic Optimization Algorithms and Their Potential Applications in Drones', *Drones*, vol. 7, no. 7, p. 427, Jul. 2023, doi: 10.3390/drones7070427.

[14] H. R. Patel and V. A. Shah, 'Shadowed Type-2 Fuzzy Sets in Dynamic Parameter Adaption in Cuckoo Search and Flower Pollination Algorithms for Optimal Design of Fuzzy Fault-Tolerant Controllers', *Math. Comput. Appl.*, vol. 27, no. 6, p. 89, Dec. 2022, doi: 10.3390/mca27060089.

[15] I. Alhamrouni *et al.*, 'A Comprehensive Review on the Role of Artificial Intelligence in Power System Stability, Control, and Protection: Insights and Future Directions', *Appl. Sci.*, vol. 14, no. 14, p. 6214, Jan. 2024, doi: 10.3390/app14146214.

[16] R. Abu Khurma, I. Aljarah, A. Sharieh, M. Abd Elaziz, R. Damaševičius, and T. Krilavičius, 'A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem', *Mathematics*, vol. 10, no. 3, p. 464, Jan. 2022, doi: 10.3390/math10030464.

[17] D. Sattar and R. Salim, 'A smart metaheuristic algorithm for solving engineering problems', *Eng. Comput.*, vol. 37, no. 3, pp. 2389–2417, Jul. 2021, doi: 10.1007/s00366-020-00951-x.

[18] M. M. Diallo and P. Sharma, 'A Comprehensive Review of IoT-Based Traffic Monitoring Systems: Architectures, Technologies, and Future Directions', Feb. 20, 2026, *Social Science Research Network, Rochester, NY*: 6285158. doi: 10.2139/ssrn.6285158.

[19] Z. Huashuai, Z. Huimei, and X. Jun, 'Truss structure dimensional optimization design: multi-population adaptive harmony search–genetic algorithm', *Eng. Optim.*, vol. 0, no. 0, pp. 1–45, Mar. 2026, doi: 10.1080/0305215X.2026.2633411.

[20] M. Brahimi, I. Haouam, R. Bouddou, O. Almomani, A. O. Salau, and I. Hunko, 'A hybrid PSO–FPA metaheuristic algorithm for ultra-low sidelobe and high-directivity synthesis of concentric circular antenna arrays for advanced radar applications', *Sci. Rep.*, vol. 16, no. 1, p. 7037, Feb. 2026, doi: 10.1038/s41598-026-36315-6.

[21] A. Agrawal, P. Paliwal, and T. Thakur, 'Economic Load Dispatch: A Holistic Review on Modern Bio-inspired Optimization Techniques', in *Proceedings of the International Conference on Computational Intelligence and Sustainable Technologies*, K. N. Das, D. Das, A. K. Ray, and P. N. Suganthan, Eds, Singapore: Springer Nature, 2022, pp. 505–517. doi: 10.1007/978-981-16-6893-7_45.

[22] F. G. Mohammadi, M. H. Amini, and H. R. Arabnia, 'Applications of Nature-Inspired Algorithms for Dimension Reduction: Enabling Efficient Data Analytics', in *Optimization, Learning, and Control for Interdependent Complex Networks*, M. H. Amini, Ed., Cham: Springer International Publishing, 2020, pp. 67–84. doi: 10.1007/978-3-030-34094-0_4.

[23] A. Wilson and M. R. Anwar, 'The Future of Adaptive Machine Learning Algorithms in High-Dimensional Data Processing', *Int. Trans. Artif. Intell. Ital.*, vol. 3, no. 1, pp. 97–107, Nov. 2024, doi: 10.33050/italic.v3i1.656.

[24] H. Albedran, S. Alsamia, and E. Koch, 'Flower fertilization optimization algorithm with application to adaptive controllers', *Sci. Rep.*, vol. 15, no. 1, p. 6273, Feb. 2025, doi: 10.1038/s41598-025-89840-1.

[25] S. J. Pratha, V. Asanambigai, and S. R. Mugunthan, 'Hybrid Mutualism Mechanism-Inspired Butterfly and Flower Pollination Optimization Algorithm for Lifetime Improving Energy-Efficient Cluster Head Selection in WSNs', *Wirel. Pers. Commun.*, vol. 128, no. 3, pp. 1567–1601, Feb. 2023, doi: 10.1007/s11277-022-10010-x.

[26] T.-K. Dao, T.-T. Nguyen, V.-T. Nguyen, and T.-D. Nguyen, 'A Hybridized Flower Pollination Algorithm and Its Application on Microgrid Operations Planning', *Appl. Sci.*, vol. 12, no. 13, p. 6487, Jan. 2022, doi: 10.3390/app12136487.

[27] L. Abualigah *et al.*, 'Enhanced aquila optimizer for global optimization and data clustering', *Sci. Rep.*, vol. 15, no. 1, p. 13079, Apr. 2025, doi: 10.1038/s41598-025-95888-w.

[28] J. Wei *et al.*, 'LSWOA: An enhanced whale optimization algorithm with Levy flight and Spiral flight for numerical and engineering design optimization problems', *PLOS ONE*, vol. 20, no. 9, p. e0322058, Sep. 2025, doi: 10.1371/journal.pone.0322058.

[29] M. Z. Naser *et al.*, 'A Review of 315 Benchmark and Test Functions for Machine Learning Optimization Algorithms and Metaheuristics with Mathematical and Visual Descriptions', Jun. 13, 2024, *arXiv*: arXiv:2406.09581. doi: 10.48550/arXiv.2406.09581.

[30] M. N. H. Mamun, 'INTEGRATION OF ARTIFICIAL INTELLIGENCE AND DEVOPS IN SCALABLE AND AGILE PRODUCT DEVELOPMENT: A SYSTEMATIC LITERATURE REVIEW ON FRAMEWORKS', *ASRC Procedia Glob. Perspect. Sci. Scholarsh.*, vol. 4, no. 1, pp. 01–32, May 2024, doi: 10.63125/exyqj773.

[31] J. P. C. Kleijnen, E. Angün, I. van Nieuwenhuyse, and W. C. M. van Beers, 'Constrained optimization in simulation: efficient global optimization and Karush-Kuhn-Tucker conditions', *J. Glob. Optim.*, vol. 91, no. 4, pp. 897–922, Apr. 2025, doi: 10.1007/s10898-024-01448-3.

[32] L. Ngartera and C. Diallo, 'A Comparative Study of Optimization Techniques on the Rosenbrock Function', *Open J. Optim.*, vol. 13, no. 03, pp. 51–63, 2024, doi: 10.4236/ojop.2024.133004.

[33] M. Z. Naser *et al.*, 'A Review of Benchmark and Test Functions for Global Optimization Algorithms and Metaheuristics', *WIREs Comput. Stat.*, vol. 17, no. 2, p. e70028, 2025, doi: 10.1002/wics.70028.

[34] G. M. Heim, 'GlobalSearch-rs: A multistart framework for global optimization written in Rust', *J. Open Source Softw.*, vol. 10, no. 115, p. 9234, Nov. 2025, doi: 10.21105/joss.09234.

[35] A. M. P and P. Paramanathan, 'Bivariate fractal interpolation functions on triangular domain for numerical integration and approximation', Aug. 08, 2022, *arXiv*: arXiv:2210.06435. doi: 10.48550/arXiv.2210.06435.

[36] M. I. Kamboh, N. B. Mohd Nawi, A. A. Ramli, and F. Sukma, 'An Improved Flower Pollination Algorithm for Global and Local Optimization', *JOIV Int. J. Inform. Vis.*, vol. 5, no. 4, p. 461, Dec. 2021, doi: 10.30630/joiv.5.4.738.

[37] S. Mahajan, N. Mittal, and A. K. Pandit, 'Image segmentation approach based on adaptive flower pollination algorithm and type II fuzzy entropy', *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 8537–8559, Mar. 2023, doi: 10.1007/s11042-022-13551-2.

[38] M. I. A. Latiffi, M. R. Yaakub, and I. S. Ahmad, 'Flower Pollination Algorithm for Feature Selection in Tweets Sentiment Analysis', *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 5, 2022, doi: 10.14569/IJACSA.2022.0130551.