Journal of Engineering and Sustainable Development
Vol. 30, No. 06, November 2026
https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd
ISSN 2520-0917
Research Article
https://doi.org/10.31272/jeasd.3886

Night Jasmine Blooming Optimization (NJBO): A Novel Nature-Inspired Algorithm for Electrical Engineering Applications

Hussein M. Hathal[1]*, Riyadh A. Abdulhussein[2], Takele Ferede Agajie[3], Alfian Ma'arif[4]

[1,2] Electrical Engineering Department, College of Engineering, Mustansiriyah University, Baghdad, Iraq.
[3]Department of Electrical and Computer Engineering, Faculty of Technology, Debre Markos University, Debre Markos, Ethiopia
[4]Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.

*Email: hussein.m.hathal@uomustansiriyah.edu.iq

Article Info

Received	**18/01/2026**
Revised	**18/09/2026**
Accepted	**27/09/2026**

Abstract

Optimization algorithms have become vital to the solution of complex optimization problems that are difficult to solve using conventional methods. In this paper, we introduce a novel nature-inspired metaheuristic algorithm called the Night Jasmine Blooming Optimization (NJBO) algorithm, which is inspired by the blooming of the night jasmine plant (*Cestrum nocturnum*). The suggested algorithm combines adaptive hormonal regulation, Gibberellin-based disturbance, scent-guided exploitation, elite preservation, and Nelder–Mead local refinement, which allows us to have a balanced exploration–exploitation process. It was tested with thirteen classical benchmark functions and the CEC2017 benchmark suites and benchmarked with several well-known and recent metaheuristic methods. As a practical application, NJBO was used for identification of an IIR system and optimal design of a five-stage Cockcroft–Walton voltage multiplier circuit. Additionally, the scalability analysis, runtime evaluation, component contribution analysis, parameter sensitivity analysis, and statistical validation with the Friedman ranking test and Wilcoxon signed-rank test were performed to assess the system completely. The results show that NJBO is a very effective general-purpose optimization algorithm with competitive convergence performance, good optimization accuracy, and stability for the benchmark and engineering optimization problems that were investigated.

Keywords: Benchmark Functions, Cockcroft–Walton voltage multiplier (CW), IIR system identification, Night Jasmine Blooming Optimization (NJBO).

1. Introduction

Optimization approaches were introduced due to the inability of traditional solutions to solve some complex problems. The result has been the adoption of algorithms that solve many of these problems. Technological developments have led to the emergence of a number of complex optimization challenges in many fields of modern engineering and science. Optimization approaches have been effectively applied to a variety of essential current disciplines, including communication applications [1], image processing applications [2], mobile robot navigation [3], and satellite observation planning [4]. The following is a review of some of the algorithms that were the primary basis for these algorithms.

In 1995, Eberhart and Kennedy proposed the renowned Particle Swarm Optimization (PSO) methodology, grounded in behavioral principles observed in natural swarms and flocks of animals during collective movement and action. This represents the inaugural population-based approach to nature-inspired intelligent algorithms documented in the literature, as it relies on the collective intelligence of several agents rather than the enhancement of a singular solution. Two years later, Storn and Price 1997) suggested Differential Evolution, drawing inspiration from Holland's (1992) work and Genetic Algorithms. Although NII algorithms are regarded as meta-heuristics, the researchers have stated that their approach is heuristic. All of these approaches were developed in the Artificial Intelligence (AI) literature around the year 2000. Since then, the variety of reported novel algorithms has consistently increased each year.

Table 1 lists major nature-inspired metaheuristics developed or frequently used between 2022 and 2024 [5]-[15], including

Work of This Research is Licensed under CC BY

their biological inspiration, main operators, advantages, limitations, and publication details.

A common worry among researchers is whether there is a real need for so many nature-inspired approaches in the literature. Are all of these developing algorithms required for efficiently handling various existing problems? These questions are answered in the explanation of the No Free Lunch Theory [16]. This theory states that all algorithms will produce the same results if given infinite data and infinite time. In other words: if algorithm A outperforms algorithm B in finding a solution to a particular problem, then algorithm B will outperform algorithm A in solving any problem other than this one. Based on this, this paper proposes developing a new algorithm inspired by the flowering mechanism of the night-blooming jasmine plant, then testing it, and finally applying it to one of the complex problems in the field of electronics: the application of filter design.

Table 1. Most important modern algorithms.

Algorithm	Year & Source	Nature Inspiration	Typical Strengths	Common Limitations and Notes
Red Fox Optimizer (RFO) [5]	**2022 – Sensors**	Red-fox searching and hunting behavior	Effective exploration and exploitation; adaptable to multi-objective search	Ref. [5] presents a multi-objective RFO variant for WSNs rather than the original RFO formulation; performance can depend on search-control settings
Pelican Optimization Algorithm (POA) [6]	**2022 – Sensors**	Pelican hunting and prey-catching behavior	Simple structure; effective exploration–exploitation balance; few control parameters	May lose population diversity and converge prematurely on complex multimodal problems
Snake Optimizer (SO) [7]	**2022 – Knowledge-Based Systems**	Snake foraging and mating behavior	Strong exploitation and adaptive transition between search behaviors	May become trapped in local optima when population diversity decreases
Reptile Search Algorithm (RSA) [8]	**2022 – Expert Systems with Applications**	Crocodile encircling and hunting behavior	Multiple search strategies provide effective exploration and exploitation	Search balance and convergence behavior may vary with problem complexity and dimensionality
Smart Flower Optimization Algorithm (SFOA) [9]	**2023 – Applied Machine Learning and Data Analytics (Springer)**	Flower-inspired search behavior	Demonstrated capability for constrained and non-convex optimization problems	Ref. [9] evaluates SFOA on industrial non-convex constrained problems rather than introducing the original algorithm
Golden Jackal Optimization (GJO) [10]	**2022 – Expert Systems with Applications**	Cooperative hunting behavior of golden jackals	Effective global exploration and exploitation around promising solutions	Population diversity may decrease during later iterations, increasing susceptibility to local optima
Beluga Whale Optimization (BWO) [11]	**2022 – Knowledge-Based Systems**	Beluga-whale swimming, foraging, and whale-fall behavior	Adaptive exploration–exploitation mechanisms; effective global-search capability	Search performance may deteriorate when population diversity declines
White Shark Optimizer (WSO) [12]	**2022 – Knowledge-Based Systems**	White-shark movement, prey tracking, and hunting behavior	Competitive global-search ability with progressive exploitation	Can experience premature convergence on difficult multimodal or high-dimensional landscapes
Hybrid Reptile Search Algorithm (HRSA) [13]	**2022 – Symmetry**	Hybridization of Reptile Search Algorithm (RSA) and Remora Optimization Algorithm (ROA)	Improves exploration–exploitation balance and helps alleviate premature convergence	Hybridization introduces additional search mechanisms and consequently greater algorithmic complexity
Coati Optimization Algorithm (COA) [14]	**2023 – Knowledge-Based Systems**	Coati hunting/foraging and predator-escape behavior	Simple population-based structure with effective exploration and exploitation	Performance and computational burden can become problem- and dimension-dependent
Levy Pelican Optimization Algorithm (LPOA) [15]	**2024 – Applied Artificial Intelligence**	Pelican search enhanced by Levy flight and hunter–prey mechanisms	Enhanced exploration and ability to escape local optima; improved exploration–exploitation balance	Additional mechanisms increase algorithmic complexity; Ref. [15] develops LPOA specifically for UAV path-planning applications

Despite significant advances in swarm and plant-based metaheuristics (e.g., Coati Optimization Algorithm (COA), Beluga Whale Optimization (BWO), and PSO), current models suffer from limited precision and inconsistent convergence in nonlinear system identification tasks.

In order to close this gap, the Night Jasmine Bloom Optimization (NJBO) technique for tackling complex optimization problems is proposed.

This work's novelty is found in the biologically based structure of NJBO, which enhances the harmony between exploitation and discovery by simulating the nighttime blooming actions of night jasmine through hormone-dependent and contextual interaction mechanisms.

This work's primary contribution is the demonstration of NJBO's superior solution quality, robust convergence, and competitive strength in the optimization of IIR coefficients for filters when compared to well-known metaheuristic approaches under the same simulated conditions.

2. Research Methodology

The research approach is divided into four major sections that discuss the development and use of the proposed Night Jasmine Bloom Optimization (NJBO) algorithm.

(2.1) Discussing the biological inspiration for the NJBO algorithm to establish the natural principles directing search behavior.

(2.2) The mathematical modeling step is presented, which expresses the algorithmic mechanisms using formal equations and update rules.

(2.3) A benchmark testing framework is developed utilizing a set of common test functions to create a consistent and fair evaluation environment.

(2.4) The IIR filter design problem is described as an optimization task, allowing the proposed approach to be applied under specified stability and performance restrictions.

This section strictly covers methodological development and experimental setup. All performance results, comparative analyses, and discussions are presented in the Results and Discussion section.

2.1. Algorithm Inspiration

In this paper, a novel optimization algorithm is proposed that mimics the blooming behavior of the night jasmine plant (Cestrum nocturnum). Light is one of the most important factors that determines plant growth and development [17]. The night jasmine flowers in a unique way that ensures that they are pollinated efficiently during the night, when light levels are lower. Changes in hormones as the day nears its end trigger a slow, gradual petal expansion that allows the flower to open and be fully enjoyed at the time when nocturnal pollinators are at work. The intensity and timing of blooming are determined by the environmental factors known as the photoperiod, humidity, and temperature.

In the blooming stage, the flower emits very volatile aromatic substances that spread in the environment according to concentration gradients. Pollinators gravitate toward areas of higher concentration, which provides a natural feedback system to strengthen successful blooming directions. Scent diffusion and intensity of blooming are further controlled by environmental conditions, allowing the plant to explore a broad environment while being attracted to a specific region of favorable environmental conditions. The proposed NJBO algorithm applies an adaptive behavior, the exploration–exploitation mechanism, inspired from this behavior of the bees.

Gibberellins are one of several plant hormones that are important in the process of flowering and in controlling plant growth. Gibberellins were first found in the fungus Gibberella fujikuroi, and it was then realized that gibberellins form a very large group of plant hormones collectively called gibberellic acid (GA1, GA3, GA4, etc.) [18]. The chemical structures of typical gibberellins are shown in Fig. 1.

Several developmental processes are regulated by gibberellins, such as the elongation of the stem, germination of seeds, and flowering [18]. It has been shown experimentally that they play a part in promoting flowering, when gibberellin GA3 was applied to Guinea impatiens, where flowering was increased greatly as illustrated in Fig. 2 [19].

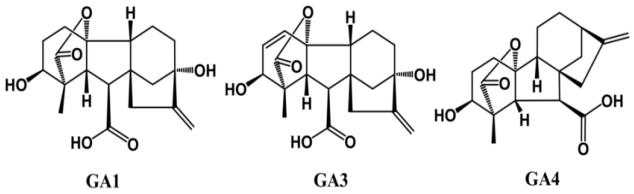


Figure 1. Gibberellin's chemical formula.



Figure 2. Effect of Gibberellin on the plants. A) Plantlets before spraying; B) after first spraying; C) after second spraying; D) plants flowering.

The gibberellin signal pathway is involved in flowering at the molecular level by the activation of gibberellin-dependent transcription factors upon degradation of DELLA proteins. Through this, the transcription factors can trigger the expression of the growth- and flowering-related genes of plants, as shown in Fig. 3 [18].

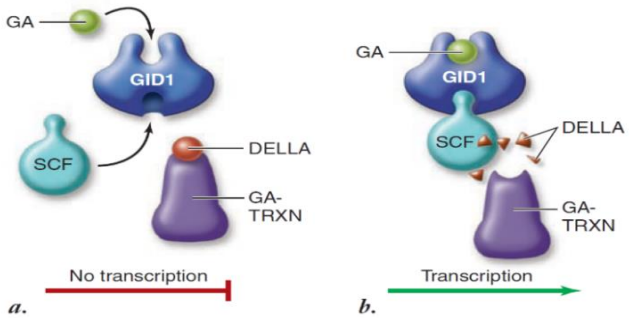


Figure 3. Gibberellins activate transcription factors called GA-TRXN. a. GA-TRXN cannot bind to promoters when attached to DELLA proteins. b. GA activates a protein complex that destroys DELLA proteins, allowing GA-TRXN to connect to a promoter and initiate gene transcription.

2.2. Novel Mathematical Operators

The biological inspiration described in the preceding section was mapped into a set of complementary mathematical operators and used to model the blooming behavior of Cestrum nocturnum. The blooming behavior of Cestrum nocturnum was modeled from its biological inspiration described in the previous section by a set of complementary mathematical operators. Each operator was constructed to represent a different biological phenomenon and carry out a unique optimization function, rather than to be based on traditional search stages. These operators constitute an adaptive optimization framework that combines exploration/ exploitation, enhances population diversity, improves

convergence stability, and increases local search accuracy. The proposed mathematical operators and their optimization roles are summarized in Table 2.

Table 2. Mathematical Operators of the Proposed Algorithm.

Operator	Purpose	Mathematical Role
Adaptive Hormonal Dynamics	Adaptive parameter regulation	Controls α and β
Bloom Expansion	Exploration	Global search
Gibberellin Disturbance	Diversity enhancement	Adaptive perturbation
Scent-Guided Motion	Exploitation	Elite-guided search
Merge & Boundary Enforcement	Feasibility	Constraint handling
Survival Selection	Selection	Greedy replacement
Elite Preservation	Memory	Preserve best solution
Nelder–Mead Refinement	Local search	Fine tuning

The combination of the proposed mathematical operators offers some unique features that make NJBO different from traditional flower-inspired and swarm-based optimization algorithms. This is a summary of the main features:

- Adaptive hormonal regulation: The exploration and exploitation coefficients (α, β) are not fixed throughout the optimization process; rather, they are updated based on a hormone-inspired mechanism.
- Biologically inspired diversity enhancement: An adaptive perturbation using a Gibberellin disturbance operator is created to preserve population diversity and avoid premature convergence.
- Elite-guided exploitation: NJBO uses both the global best and elite-group information to give more stable exploitation, not only based on the best found on the global level.
- Balanced search transition: Exploration and exploitation solutions are combined before the selection phase, so as to have a balanced transition process from global exploration to local intensification.
- Optimizer solution preservation mechanism: Elite preservation ensures that the top solution produced in the optimization process is never thrown away, which increases the reliability of the optimization convergence.
- Local refinement (Nelder–Mead): The Nelder–Mead local search procedure is embedded into the optimization scheme without compromising the global search ability.

2.3. Mathematical Modeling

Night jasmine is classified as a short-day and long-day plant. This means that they flower in spring and fall. Three-night jasmine seedlings were planted in different locations (with varying hours and durations of direct sunlight exposure) to determine the factors affecting flowering. It was found that the number of daylight hours (biological time) has a direct impact on the flowering period. Another factor, temperature, was also found to influence this plant. These two factors directly influence the production of the gibberellin hormone G3, which is responsible for flowering in this plant family. This behavior was translated into a curve representing the relationship between light energy and temperature on the one hand, and flowering on the other. The mathematical structure of NJBO directly maps the biological stages of Cestrum nocturnum. The hormone-driven blooming rate corresponds to the exploration factor α, while the scent diffusion intensity controls β. The environmental effects, modeled by Gibberellin randomness, simulate the variability in hormone distribution.

The stages of flowering of the plant are shown in Fig. 4. The equations for the algorithm were then arranged as follows [20]:

In the proposed model, the parameter α represents the exploration capability of the algorithm, simulating early bloom expansion, while β represents exploitation, corresponding to directed scent-based blooming.

These parameters are dynamically adjusted to mimic the hormonal behavior of gibberellin, which regulates plant growth and flowering.

This adaptive mechanism ensures a balance between diversification and intensification during the optimization process.

The equations for the algorithm were then arranged as follows:

1) Initialization: Each solution (flower) is initialized uniformly within bounds. Represents random seed dispersion before blooming, influenced by environmental uncertainty.

$$x_i = LB + rand(0,1) * (UB - LB) \quad (1)$$

Where:

- x_i = candidate solution
- LB, UB = lower/upper bounds
- rand (0,1) = uniform random numbers

2) Adaptive Hormonal Dynamics (Gibberellin-regulated α & β): Exploration and exploitation evolve gradually. Biologically, α and β mimic gibberellin Hormone (GH) dynamics. GH increases at night, promoting petal elongation, bloom expansion, and scent release.

$$\alpha_i(t+1) = \alpha_i(t) \cdot r\alpha \quad (2)$$

$$\beta_i(t+1) = \beta_i(t) \cdot r\beta \quad (3)$$

Where:

- α_i controls exploratory search (early bloom expansion)
- β_i controls exploitation (directed scent blooming)
- $r\alpha < 1$ decays exploration over time
- $r\beta > 1$ strengthens exploitation



Figure 4. Blooming stages of Night–Jasmine plants.

3) Bloom Expansion (Exploration): Exploratory update: Simulates initial rapid petal expansion triggered by rising GA concentration during early nighttime.

$$x_bloom = x_i + \alpha_i \cdot rand(0,1) * (UB - LB) \quad (4)$$

In order to explore new areas of the search space, each potential solution is permitted to stray from its existing location during this phase. The initial rapid growth of night jasmine petals during the start of evening blooming serves as the biological inspiration for this behavior. Mathematically, the search amplitude is determined by the value of the parameter α_i, which controls the exploration step. While a smaller number limits the search to a smaller neighborhood surrounding the current point, a bigger value of α_i permits the potential solution to explore a wider region. In order to avoid all potential solutions from going in the same direction and to maintain population diversity in the early stages of the search, the stochastic term $rand(0,1)$ is employed to add randomness to the process.

4) Gibberellin- Disturbance Parameter: Gibberellin disturbance that represents natural $G.H$ secretion fluctuations influenced by humidity, temperature, and the biological clock is given by the term:

$$G.H = d_{max} * e^{(T*B.C)-\beta} + 10 * d_{min} \quad (5)$$

Where, d_{max} and d_{min} are the maximum and minimum flower diameters (0.5 and 0.1, respectively). Temperature ($T\ between 30℃\ and\ 35℃$), biological clock effect ($B.C\ between\ 7\ PM\ and\ 6\ AM$) are arranged in order of increasing diversity. Night-blooming jasmine flowers close during the day and begin to bloom at the onset of the early evening hours, continuing until sunrise the following day. Night-blooming jasmine flowers can be modified as a subsequent solution based on this cycle.

To simulate the biological and environmental variations influencing the blooming behavior of night jasmine, the term GH is introduced in (5). In the suggested optimization framework, this term serves as a biologically motivated disturbance element that modifies the search dynamics rather than representing an exact botanical hormone concentration. This factor's goal is to add controlled variation to the candidate updating process, which reflects how the surrounding environment affects night jasmine blooming intensity. Temperature, humidity, and the biological clock are the three main variables linked to the disruption. Because warmer evening conditions typically encourage stronger smell release and more vigorous blooming behavior, temperature is used as a primary regulator among them. Because of this, the suggested model chooses the temperature range between 30℃ and 35℃ as a good blooming range. Rather than being a rigid, all-encompassing botanical limit, this range should be viewed as a biologically inspired operating interval utilized for algorithmic abstraction.

In the same manner, the biological clock element represents the night jasmine's natural nocturnal blooming cycle, which starts in the early hours of the night and lasts until morning. This periodic behavior is separated into a search-control process in the algorithm, which progressively changes the population behavior from widespread exploration in the early iterations to more targeted exploitation in the subsequent iterations.

As a result, the term of disturbance GH functions as an adjustable disruption component designed to help the algorithm avoid early convergence, particularly in complicated multimodal search landscapes, and enhances the diversity of the search process.

This can be achieved by altering the range of factors that influence gibberellin secretion over the course of a 24-hour day/night cycle, as illustrated in (Fig. 5).

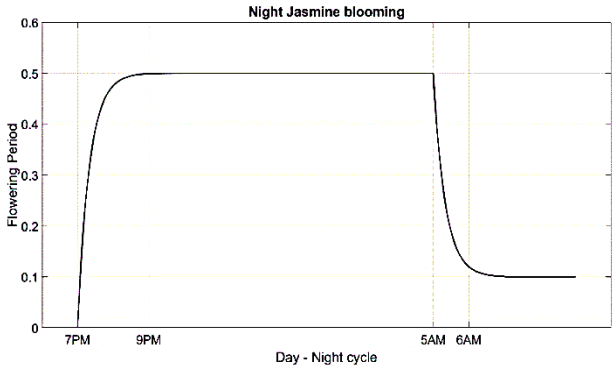


Figure 5. Night Jasmine blooming period during the night.

5) Scent-Guided Motion (Exploitation)

Directed movement toward strongest bloom: Biologically, Night jasmine flowers intensify bloom toward dominant scent sources, attracting nocturnal pollinators.

$$x_track = x_i + \beta_i [\, \rho(x_best - x_i) + (1-\rho)(\bar{x}_E - x_i) \,] + GH \quad (6)$$

Where:

• x_best = global best solution

• $\bar{x}_E$ = mean of elite group

• ρ ∈ [0,1] = elite influence factor

The exploitation phase of the suggested NJBO algorithm is represented by (6), where each potential solution is directed toward the most promising area found throughout the search. The ability of night-blooming jasmine to accelerate blooming in an environment of stronger scent attraction served as the biological inspiration for this operator. This behavior is abstracted into a targeted motion toward the elite-group mean and the global best solution within the optimization framework. The phrase " x_best " indicates the most powerful blooming vector in the present population and the best solution discovered thus far. The elite group's mean position is denoted by the word $\bar{x}_E$, which offers collective direction from the top-performing candidate solutions. By ensuring that the exploitation process is not dependent on just one leader, the usage of both terms improves search stability and lowers the possibility of premature convergence.

The proportional influence of the elite-group mean and the global best solution is controlled by the coefficient ρ ∈ [0,1]. The update is more strongly drawn to x-best when ρ increases. On the other hand, the elite-group mean contributes more to the movement when ρ is less, which helps sustain directed convergence while preserving diversity. During this stage, adaptive variability is maintained by the additive disturbance term GH, while the exploitation intensity is controlled by the coefficient β_i. As a result, this operator maintains enough flexibility to prevent undue concentration around a single candidate solution while improving local search efficiency around high-quality regions.

6) Merge & Boundary Enforcement

Combine candidates:

$$x_mid = 0.5\,(x_bloom + x_track) \quad (7)$$

$$x_cand = clip(x_mid, LB, UB) \quad (8)$$

$$u = clip(u, LB, UB) \quad (9)$$

Biological Interpretation: Petal expansion is biologically bounded by structural and environmental constraints.

The suggested algorithm integrates both search inclinations through the intermediary solution x_mid after producing the preliminary option x_bloom and the exploitation candidate x_track. The freshly created candidate can benefit from both directed local improvement and broad-space exploration at the same time thanks to this merging stage, which offers an equilibrium transition between intensification and diversity. The clipping operator, which enforces the predetermined lower and upper constraints of the search space, is then applied to obtain the candidate solution x_cand. The additional candidate u is similarly subjected to a comparable border modification. Because both discovery and exploitation procedures may produce infeasible values outside of the allowed search interval, this boundary-control method is crucial.

As a result, the merge-and-boundary stage has two crucial functions: first, it maintains the balance between local and global search behaviors; and second, it ensures that all updated solutions remain viable and numerically stable throughout the optimization process.

7) Survival Selection

Select best: Reflects natural selection; only the strongest bloom patterns persist.

$$x_i \leftarrow argmin\{J(x_i), J(x_cand), J(u)\} \quad (10)$$

Which potential solution is kept for the following iteration is decided at the survival selection stage. The solution with the most effective objective value is chosen after the objective function values of the present solution (x_i), the merged candidate solution (x_cand), and the auxiliary candidate (u) are assessed.

Only the most powerful blooming pattern remains and passes on to the next generation, according to a natural-selection principle reflected in this mechanism. By consistently maintaining the best option among the competing alternatives produced within the same iteration, this selection rule raises convergence pressure from an optimization standpoint.

Therefore, without compromising search efficiency, the existence selection stage enhances robustness and guarantees that the population develops in a way that preserves quality.

8) Elite Preservation

Worst solution replaced by global best. Biologically, Optimal blooming traits propagate across the population, preserving scent efficiency.

9) Local Refinement (Nelder-Mead)

x_best ← Local Refine(x_best)

Biological Interpretation: Fine-tuning petal curvature and orientation to maximize scent dispersion efficiency.

The flowchart of the algorithm is presented in Fig. 6.

2.4. Benchmark Test Functions

In order to evaluate the accuracy of the proposed algorithm, it must be tested using benchmark functions. These functions have known results and known initial conditions, but they represent real problems that are difficult to solve or quickly identify the best possible solution. These functions have various forms and variables. Some contain a single variable and a single solution point (unimodal), while others up to four variables and more than one local solution point, but with a single global solution point (multimodal). The benchmark functions were selected to cover both unimodal and multimodal landscapes, ensuring comprehensive evaluation of the exploration–exploitation balance.

Hathal et al. [21] has mentioned some of these properties. Thirteen functions presented in Table 3 (showing the properties of each function) are tested using the proposed algorithm to evaluate the algorithm's behavior, and then compared with other algorithms.

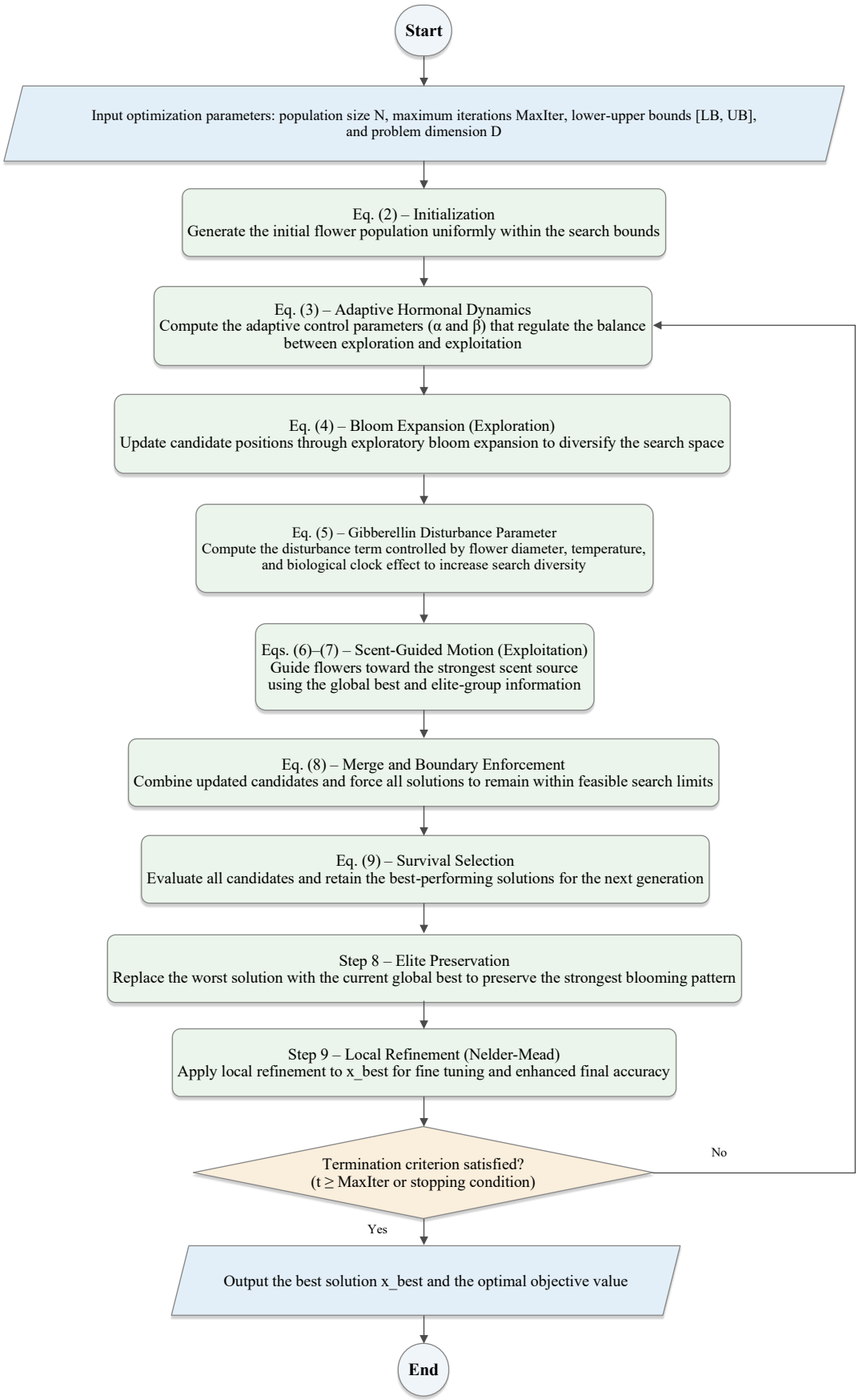


Figure 6. Flowchart of Night Jasmine Optimization Algorithm (NJBO).

Table 3. The properties of thirteen benchmark functions.

Type	Fun. no.	Name	Specification	Range of search	Global minimum
Unimodal	**Fn. 1.**	Acklay2 Function	Continuous, Differentiable, non–separable, non–Scalable.	$-32 \le x_i \le 32.$	$f(x^*) = -200$ at $x^* = (0, 0)$
	Fn. 2.	Beale Function	Continuous, Differentiable, non–Separable, non–Scalable.	$-4.5 \le x_i \le 4.5$	$f(x^*) = 0$ at $x^* = (3, 0.5)$
	Fn. 3.	Leon Function	Continuous, Differentiable, non–Separable, non–Scalable.	$-1.2 \le x_i \le 1.2$	$f(x^*) = 0$ at $x^* = (1, 1)$
	Fn. 4.	Matyas Function	Continuous, Differentiable, non–Separable, non–Scalable.	$-10 \le x_i \le 10$	$f(x^*) = 0$ at $x^* = (0, 0)$
	Fn. 5.	Rosenbrock Function	Continuous, Differentiable, non–Separable, Scalable.	$-30 \le x_i \le 30$	$f(x^*) = 0$ at $x^* = (1, \cdots, 1)$
	Fn. 6.	Colville Function	Continuous, Differentiable, non–Separable, non–Scalable.	$-10 \le x_i \le 10$	$f(x^*) = 0$ at $x^* = (1, \ldots, 1)$
	Fn. 7.	Easom Function	Continuous, Differentiable, Separable, non–Scalable.	$-100 \le x_i \le 100$	$f(x^*) = -1$ at $x^* = (\pi, \ldots, \pi)$
	Fn. 8.	Quartic Function	Continuous, Differentiable, Separable, Scalable	$-1.28 \le x_i \le 1.28$	$f(x^*) = 0$ at $x^* = (0, \ldots, 0)$
Multimodal	**Fn. 9.**	Quartic with noise Function	Continuous, Differentiable, Separable, Scalable	$-1.28 \le x_i \le 1.28$	$f(x^*) = 0$ at $x^* = (0, \ldots, 0)$
	Fn. 10.	Schwefel 2.36 Function	Continuous, Differentiable, Separable, Scalable	$0 \le x_i \le 500$	$f(x^*) = -3456$ at $x^* = (12\ldots 12)$
	Fn. 11.	Bohachevsky Function 1	Continuous, Differentiable, non–separable, non–scalable, Multimodal	$-100 \le x_i \le 100$	$f(x^*) = 0$ at $x^* = (0, \ldots,0)$
	Fn. 12.	Bohachevsky Function 3	Continuous, Differentiable, Separable, non-Scalable, Multimodal	$-100 \le x_i \le 100$	$f(x^*) = 0$ at $x^* = (0, \ldots,0)$
	Fn. 13.	Camel Function – three Hump	Continuous, Differentiable, non-Separable, non-Scalable, Multimodal	$-5 \le x_i \le 5.$	$f(x^*) = 0$ at $x^* = (0, 0)$

2.5. Infinite Impulse Response (IIR) designing

IIR filter design is one of electrical engineering's most essential applications. Second-order filter design will be used as an example. Adaptive infinite impulse response (IIR) filters have gained popularity in recent years due to its use in a variety of real-world applications, including modeling difficulties and system identification in signal processing [22], [23].

IIR filters are commonly employed for system identification in real-world applications as they are more accurate and require fewer system coefficients than Finite Impulse Response (FIR) filters [24].

Adaptive IIR filters use a mean square error function to estimate the output from the desired response. This function is typically gradient-free, non-linear, and multimodal. Metaheuristic algorithms have been effective for designing adaptive IIR filters [22].

Although IIR filters are linear systems, the optimization problem associated with coefficient estimation is inherently nonlinear due to the complexity of the search space and the error minimization process.

By changing the input-output coefficients of an unknown plant using a metaheuristic algorithm, system identification using an IIR filter is a potent method for obtaining its mathematical model. Until the error between the adaptive IIR system's output and the unknown system output is as small as possible, the NJBO tries to adjust the IIR system's coefficients.

The adaptive IIR system identification problem is shown through the use of NJBO in Fig. 7. In the Z-plane, the IIR filter's transfer function is provided by:

$$\frac{O_p(z)}{I_p(z)} = \frac{(p^0 + p^1 z^{-1} + \ldots + p_L z^{-L})}{(1 + q^1 z^{-1} + \ldots + q_K z^{-K})} \quad (11)$$

Where $O_p(z)$ represents the output signal in Z-domain, $I_p(z)$ represent the input signal in Z-domain, p_0, p_1, ..., p_L represent the numerator coefficients (feedforward), and q_1, q_2, ..., q_K are the denominator coefficients (feedback)

The time-domain difference equation for the previous equation is given by:

$$O_p(t) + q^1 O_p(t-1) + \ldots + q_K O_p(t-K) = p^0 I_p(t) + \ldots + p_L I_p(t-L) \quad (12)$$

Rewriting the recursive difference equation (10) can be done as follows:

$$O_p(t) = \sum_{i=0}^{L} [p_i \, i_p(t-i)] - \sum_{j=1}^{K} [q_j \, O_p(t-j)] \quad (13)$$

The output signal of the unknown plant is represented by p(t), whereas the output signal of the adaptive IIR filter is represented by $O_p(t)$. These outputs are illustrated in Fig. 8

The error $e_f\ (t) = p(t) - o_p(t)$ is the result of the difference between the output of the adaptive IIR filter and the output of the unknown plant. Thus, it is possible to introduce the IIR model identification problem as a minimization problem of the function $U(\theta)$, which is defined as follows:

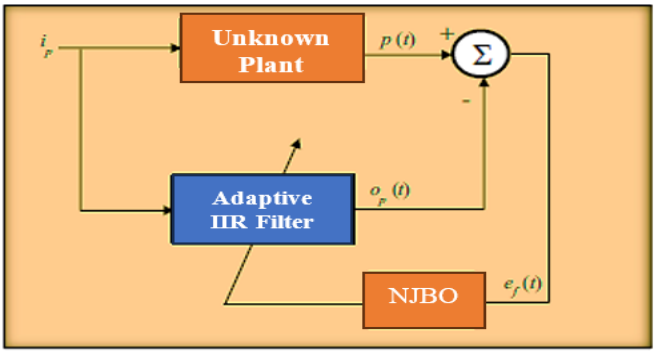


Figure 7. Systematic identification of the IIR system.

$$U(\theta) = \frac{1}{R}\sum_{t=1}^{R}(p(t) - O_{\mathrm{p}}(t))^2 \qquad (14)$$

R is the sample value needed to compute the function $U(\theta)$. The primary goal of this research is to reduce the value of the function $U(\theta)$ through improving θ. The optimal solution for θ is obtained when the error function $U(\theta)$ reaches its smallest value, as shown below:

$$arg_{\theta \in S} \min(U(\theta)) \qquad (15)$$

2.6. Cockcroft-Walton Circuit Design

A Cockcroft–Walton (CW) voltage multiplier is a diode–capacitor cascade circuit that produces a high Direct Current (DC) output voltage by inputting a relatively small Alternating Current (AC) voltage, in the absence of a step-up transformer. Due to its simplicity, scalability, and high voltage capability, the CW multiplier has been widely used in various particle accelerators, X-ray generators, electrostatic precipitators, neutron generators, plasma systems, and other applications which require high voltages and low currents. Thaker and Vora [25] show a detailed study of modern CW topologies, their working and practical design considerations, with an emphasis on the fact that the output performance is largely dependent on the number of stages, operating frequency, load current, and characteristics of the capacitor. The second engineering optimization problem is the classical five-stage half-wave Cockcroft–Walton voltage multiplier shown in Fig. 9.

The circuit is made of ten diodes, five pumping capacitors (Cp1 −Cp5), and five smoothing capacitors connected across the rectifier (Cs1−Cs5).

If the output voltage is to be approximated for an ideal N-stage half-wave Cockcroft–Walton multiplier under no-load conditions, then it will be:

$$Vo = 2NVp \qquad (16)$$

where $\boldsymbol{N}$ is the number of stages and $\boldsymbol{V_p}$ is the peak value of the AC input voltage.

If the output voltage needs to be lowered due to a load current, the output voltage will be lower due to the reduction in the charge on the capacitors and losses in the diodes, resulting in:

$$Vout = Vo - \Delta V \qquad (17)$$

where $\boldsymbol{\Delta V}$ is the overall voltage drop due to the loading effect.

The output ripple voltage can be approximately given by

$$Vripple \propto \frac{I_L\, N(N+1)}{2fC} \qquad (18)$$

It is to be noted that the load current $\boldsymbol{I_L}$, $\boldsymbol{f}$ is the input frequency and $\boldsymbol{C}$ is the capacitance of the stage.

From these equations, it is seen that the more stages that are added, the higher the output voltage that can be achieved but the more the voltage drop and ripple are worsened under load conditions. Thus, choosing the right capacitor is one of the most critical design parameters of the whole multiplier.

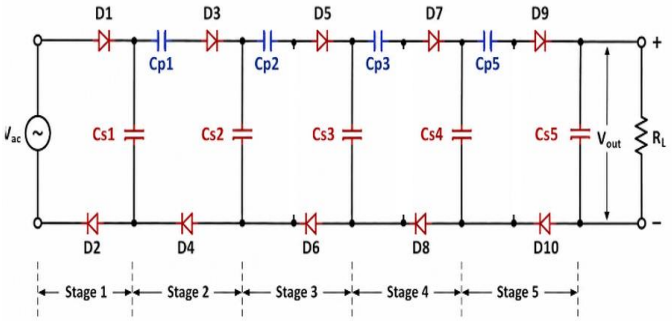


Figure 8. A (5 stages) of the CW Circuit.

3. Results and Discussion

This section describes and evaluates the performance of the proposed Night Jasmine Bloom Optimization (NJBO) algorithm through two major components. First, the algorithm is assessed using a series of standard benchmark test functions to determine its ability to optimize in terms of accuracy, convergence behavior, and robustness under unified experimental settings. Second, the proposed algorithm's validation is demonstrated by applying it to the IIR filter coefficient design problem and comparing the results with those of well-established optimization strategies.

The results are presented in terms of quantitative performance indicators, such as solution quality, convergence characteristics, statistical consistency, and computational efficiency, to provide a thorough evaluation of the suggested approach.

3.1. Benchmark Functions Results

These functions are applied to several algorithms for comparison. The compared algorithms were Particle Swarm Optimization (PSO) [26], Biogeography-Based Optimization (BBO) [27], and Sine Cosine Algorithm (SCA) [28], Whale Optimization Algorithm (WOA) [29], Sunflower Optimization Algorithm (SFOA) [30], Secretary Bird Optimization

Algorithm (SBOA) [31], and Artificial Protozoa Optimizer (APO) [32].

Twenty iterations were performed to ensure the accuracy of the obtained results. The average of the best results was taken from these iterations. This average of the best result was adopted as a benchmark for comparing the results of the algorithms. The proposed algorithm demonstrated its superiority over other algorithms in terms of the accuracy of the results and the best average of the best solutions with the lowest percentage of standard deviation compared to the others.

The results of the benchmark functions with these algorithms are explained in Table 4.

Table 4. The results of the benchmark functions with the compared algorithms

Fn. No.			NJB	PSO	BBO	SCA
Unimodal	Fn.1	Best	-200	–199.42	-199.9961	-199.9981
		Mean	**-200**$^{(1)}$	–194.2915$^{(4)}$	-199.9878$^{(2)}$	-199.9489$^{(3)}$
		Std.	0	2.991608043	0.005827	0.038617
	Fn.2	Best	3.8998e-17	0	2.5905e-07	2.8926e-07
		Mean	**5.78715e-08**$^{(1)}$	0.049739907$^{(4)}$	7.9672e-06$^{(2)}$	8.3029e-06$^{(3)}$
		Std.	1.35295e-07	0.100935338	5.9104e-06	8.7002e-06
	Fn.3	Best	0	2.1952E–16	7.747e-09	6.4977e-08
		Mean	**1.21393e-07**$^{(1)}$	0.008226761$^{(4)}$	3.4868e-06$^{(2)}$	1.5178e-05$^{(3)}$
		Std.	1.99572e-07	0.021144602	4.0018e-06	2.1034e-05
	Fn.4	Best	1.4688e-51	5.5254E–18	3.5511e-08	3.1712e-08
		Mean	**1.36127e-09**$^{(1)}$	0.000349969$^{(4)}$	1.428e-06$^{(3)}$	7.2898e-07$^{(2)}$
		Std.	3.41800e-09	0.001343955	1.2527e-06	7.2123e-07
	Fn.5	Best	1.0700e-09	0	7.4157e-05	4.6452e-06
		Mean	**5.2600e-08**$^{(1)}$	0.000824868$^{(3)}$	0.0015798$^{(4)}$	9.3286e-05$^{(2)}$
		Std.	3.75758e-08	0.003229872	0.0016646	8.4808e-05
	Fn.6	Best	1.2815e-08	0.000019144	0.80019	0.047584
		Mean	**6.16188e-08**$^{(1)}$	0.112682992$^{(2)}$	6.9133$^{(4)}$	0.89624$^{(3)}$
		Std.	2.91979e-08	0.191159812	3.4697	0.55579
	Fn.7	Best	-1	–1	-0.99997	-0.99998
		Mean	**-1** $^{(1)}$	–0.937501$^{(4)}$	-0.99854$^{(3)}$	-0.9997$^{(2)}$
		Std.	0	0.130961483	0.0011853	0.00021772
Multimodal	Fn.8	Best	0	0.00094364	6.4898e-27	1.0111e-13
		Mean	**0**$^{(1)}$	0.006156197$^{(4)}$	4.9596e-22$^{(2)}$	4.5427e-09$^{(3)}$
		Std.	0	0.006189817	2.13e-21	1.1421e-08
	Fn.9	Best	2.3946e-08	0.000086509	9.4085e-06	1.4278e-05
		Mean	**9.70536e-07**$^{(1)}$	0.002846577$^{(4)}$	0.00047517$^{(3)}$	0.00012285$^{(2)}$
		Std.	9.42779e-07	0.011295313	2.359e-04	8.3e-05
	Fn.10	Best	-3456	–3456	-3455.9973	-3455.7443
		Mean	**-3456.15**$^{(1)}$	–1.3532E+11$^{(4)}$	-3455.8781$^{(2)}$	-2254.5323$^{(3)}$
		Std.	0.81982	3.54737E+11	0.085978	1196.199
	Fn.11	Best	0	0	0.00055391	0.47047
		Mean	**0**$^{(1)}$	0.000122$^{(2)}$	0.021317$^{(3)}$	9.2194$^{(4)}$
		Std.	0	0.000547	0.019519	6.713
	Fn.12	Best	0	0	0.00012594	1.4391
		Mean	**0**$^{(1)}$	1.40782E-06$^{(2)}$	0.0060082$^{(3)}$	11.1655$^{(4)}$
		Std.	0	5.60868E-06	0.0048965	9.7919
	Fn.13	Best	6.3735e-48	0	8.5371e-09	0.0020713
		Mean	**1.5787e-13**$^{(1)}$	0.009099$^{(3)}$	2.6834e-06$^{(2)}$	0.022976$^{(4)}$
		Std.	7.06016e-13	0.031647	1.9622e-06	0.018956

The results of Table 5 demonstrate that NJBO has the best overall performance over the 13 benchmark functions. It achieved the best average mean absolute error value of 0.00126 and the first place in the Friedman test, with high stability in the results of all optimization problems. APO and PSO were 2nd and 3rd, respectively, but their differences between average and median errors were large, which indicates that they were not very stable on some of the functions. Computation cost: WOA required the lowest cost, and NJBO required relatively high cost. However, the extra time required was still not significant, and this was outweighed by the improved accuracy and consistency of the results. Thus, NJBO offers a good performance balance on the quality of the solution and the computational cost.

The Wilcoxon signed-rank test results are summarized in Table 6, which has undergone Holm correction. The NJBO results were statistically significantly better than the GA, BBO, and SFOA results on all 13 test functions. NJBO outperformed

significantly on 6 functions and similarly on 7 functions in comparison to SCA. The differences were not statistically significant for most of the algorithms compared to IPSO, WOA, SBOA, and APO, meaning that their performance was similar to NJBO on the majority of benchmark functions. Importantly, NJBO did not record any statistically significant losses when compared to any competitor algorithm, further highlighting the robustness and competitive optimization performance of NJBO.

Table 5. Overall Performance Summary of the compared algorithms.

Algorithm	Average Mean Absolute Error	Median Mean Absolute Error	Average Runtime Per Run (s)	Total Mean Runtime Across 13 Functions (s)	Friedman Mean Rank
PSO	186.0952567	9.61616E-60	0.032480401	0.42224521	3
BBO	2.445474914	0.242286784	0.066382172	0.86296823	8
SCA	26.59122545	8.84499E-07	0.024012732	0.31216552	4
WOA	119.7936529	0.000254342	0.022895524	0.29764181	7
SFOA	0.005889748	0.000175819	0.046323635	0.60220726	6
SBOA	93.08187606	4.97346E-14	0.024447468	0.31781708	5
APO	0.009835803	4.08E-113	0.056445502	0.73379152	2
NJBO	0.001262818	6.64513E-31	0.076025568	0.98833238	1

Table 6. Summary of Wilcoxon Signed-Rank Test Results after Holm Correction.

Algorithm	Significant Wins	Non-significant	Significant Losses
GA	13	0	0
PSO	1	12	0
BBO	13	0	0
SCA	6	7	0
WOA	1	12	0
SFOA	13	0	0
SBOA	2	11	0
APO	2	11	0

The success rate test was carried out to test the reliability of the proposed algorithm. Table 6 displays the success rate of the NJBO algorithm and the compared algorithms for all the benchmark functions when the stopping criterion is set to $\beta=10^{-5}$. The results show that NJBO had a good success rate on most of the benchmark functions and was competitive with the compared algorithms. The success rate of the reported results in Table 7 was derived from being able to obtain the intended solution within the set tolerable error.

The NJBO method is not necessarily optimal, even if it has a 100% success rate in many common functions. However, it has demonstrated strength and robustness in the assessed functions. It will therefore be employed in an engineering application to evaluate its ability to resolve some difficult issues.

Table 7. Reliability results of four comparative algorithms on thirteen benchmark functions.

Fun. no.	NJBA	PSO	BBO	SCA
Fn. 1.	100%	0%	0%	0%
Fn. 2.	100%	60%	95%	5%
Fn. 3.	100%	25%	100%	0%
Fn. 4.	100%	95%	100%	0%
Fn. 5.	100%	90%	10%	0%
Fn. 6.	100%	15%	0%	0%
Fn. 7.	100%	65%	5%	0%
Fn. 8.	100%	0%	100%	100%
Fn. 9.	100%	55%	95%	30%
Fn. 10.	100%	5%	0%	0%
Fn. 11.	100%	95%	0%	0%
Fn. 12.	100%	100%	0%	0%
Fn. 13.	100%	90%	100%	0%

Table 8 shows the scalability performance of NJBO for three benchmark functions of increasing dimensionality (10, 20, 30, and 50 variables). The resulting rankings show that the proposed algorithm keeps up with the competition as the problem size grows. Rosenbrock is a popular problem with a small curved valley and also has a growing optimization difficulty, but NJBO was one of the best-performing algorithms for all dimensions tested. For Quartic and Quartic with Noise, NJBO showed excellent scalability and robustness, being the best-performing algorithm even with the high complexity of the search space. From these results, it can be concluded that the proposed adaptive search mechanism provides the ability to maintain convergence stability while avoiding performance degradation in higher-dimensional optimization problems.

Table 8. Scalability Ranking of NJBO across Different Problem Dimensions.

Function	D=10	D=20	D=30	D=50	Overall Trend
Rosenbrock	Rank 3	Rank 3	Rank 3	Rank 3	Stable top-tier performance
Quartic	Rank 2	Rank 3	Rank 2	Rank 2	Excellent scalability
Quartic with Noise	Rank 2	Rank 1	Rank 1	Rank 1	Highly robust

The convergence process to reach the best possible result takes place after a number of iterations. This number of iterations is not constant in all applications, but rather it is not constant for the application itself due to the randomness present in choosing the conditions specific to the algorithm. As an example of this, the Rosenbrock function was chosen (not specifically) to demonstrate how to converge, as shown in Fig. 9.

Table 9 shows the computational scalability of the algorithms investigated by comparing the average time of execution of all algorithms at both dimensions 10 and 50, as well as the ratio of the growth of the execution time at dimension 50 to that at dimension 10, and the estimated dimension exponent. It is noted that the proposed NJBO does not possess the shortest execution time, but as the problem dimension increases, its execution time increases in a controlled manner. In particular, the growth ratio of the run-time for NJBO is 2.39, and the estimated dimension exponent is 0.586, which means that the computational cost grows moderately with the size of the search space.

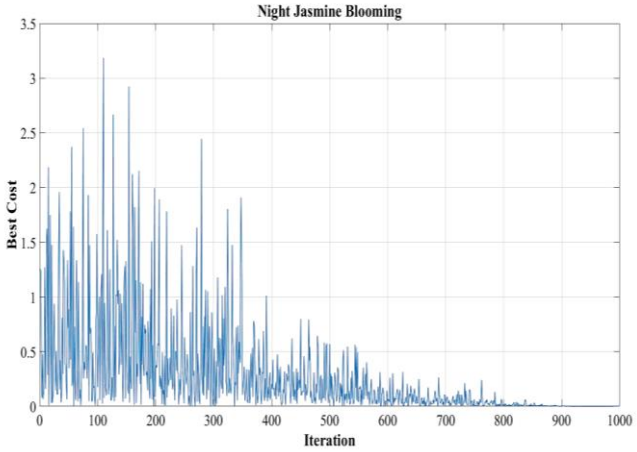


Figure 9. Best solution for the Rosenbrock Function with the NJBO Algorithm.

Compared with other algorithms, NJBO demonstrates better scalability than BBO (growth ratio = 6.13, exponent = 1.18), SCA (3.57, 0.834), SBOA (2.92, 0.707), PSO (2.82, 0.688), and WOA (2.79, 0.685). The growth exponents of APO (2.34, 0.578) are slightly lower, but the gap is not significant, whereas the optimization accuracy and solution stability of NJBO are better, as seen in the preceding benchmark results. Thus, it can be inferred that the proposed algorithm provides a good performance balance between scale and optimization, and the search operators introduced into the algorithm successfully enhance the quality of the solutions without causing excessive computational overhead with the growth in problem size.

Table 9. Comparison of Runtime Growth Ratio and Estimated Dimension Exponent for the Investigated Algorithms.

Algorithm	Mean Runtime at D10 (s)	Mean Runtime at D50 (s)	Runtime Growth Ratio	Estimated Dimension Exponent
PSO	0.13958281	0.393487827	2.819027835	0.688053736
BBO	0.461974513	2.830048417	6.125983869	1.180436212
SCA	0.137094117	0.488938623	3.566444974	0.833780893
WOA	0.11319925	0.315849757	2.790210683	0.684640473
SFOA	0.18420386	0.457069473	2.481324079	0.606318404
SBOA	0.11699183	0.34199603	2.923247119	0.707460319
APO	0.237532547	0.555740767	2.339640502	0.57806136
NJBO	0.339222003	0.81173969	2.392945275	0.58602277

An ablation study was performed with the benchmark function Quartic with Noise to explore the main contributions of NJBO. The function was intentionally chosen as it retains a nice convex search landscape in combination with a stochastic noise term in every evaluation. Thus, the optimizer should not only be converging to the correct solution, but also be robust to random perturbations. The Quartic with Noise is made especially interesting for the study of the success of adaptive search mechanisms, diversity enhancement, and local refinement strategies in uncertain optimization conditions because of these properties.

To quantify the contribution of the principal components of the proposed NJBO algorithm, an ablation study is performed on the Quartic with Noise benchmark function, as shown in Table 10. The best exact error (1.35×10^{-4}) and the first place in the ranking were obtained by the complete NJBO. The mean error increased by over an order of magnitude when adaptive α–β regulation was removed from the system, highlighting the key role adaptive parameter control has in a successful tradeoff between exploration/exploitation. Likewise, the accuracy of the optimization process was significantly reduced when the Gibberellin disturbance mechanism was turned off, reflecting its importance in maintaining population diversity and robustness against noise in the evaluations.

Removal of the Nelder–Mead refinement stage resulted in the most dramatic increase in the mean exact error of 5.35×10^{-3}, a factor of about forty larger than that of the full NJBO. This result clearly indicates that the local refinement mechanism is crucial in achieving the optimum solution after the global search phase. The results of the ablation study validate the proposed NJBO, showing that the main components work well together and that each operator plays a part in optimizing the performance of the process.

The parameter sensitivity analysis of the main control parameters of the proposed NJBO algorithm is shown in Table 11 for the benchmark function Quartic with Noise. This benchmark was chosen because it is a stochastic noise function, making it difficult to test the robustness of the parameters in the face of uncertain function evaluations. The results show that moderate changes in the control parameters of NJBO led to only slight changes in the mean exact error over the ranges of the parameters investigated.

Of the tested values, a slight improvement was seen at an α scale of 1.15, a β scale of 0.95, and an elite fraction of 0.25. However, the improvements from the obtained parameters to the default parameters were small, suggesting that the default parameters used in this study are near the optimum operating region. The results show that the proposed NJBO algorithm is suitable for optimizing the problem without extensive parameter tuning, which is a significant advantage for real-world engineering optimization problems

Table 10. Contribution Analysis of the Main NJBO Operators on the Quartic with Noise Function.

Metric	Exact	Full NJBO	No Alpha Beta	No Gibberellin Disturbance	No Nelder Mead
Best Exact Error	0	1.00327E-05	1.00616E-05	1.53679E-05	0.000306889
Mean Exact Error	0	0.000134621	0.001638855	0.000961898	0.005347778
Std Exact Error	0	0.000161407	0.00224	0.000887632	0.005937445
Worst Exact Error	0	0.000718958	0.009283398	0.002883756	0.02660556
Mean Final Noisy Cost		0.503114951	0.503407593	0.502730635	0.507116515
Std Final Noisy Cost		0.012153436	0.012658073	0.012626282	0.014264074
Mean Runtime (s)		0.241934015	0.23192879	0.1670823	0.22874074
Rank By Mean Exact Error	0	1	3	2	4

Table 11. Parameter Sensitivity Analysis of the Main NJBO Control Parameters Using the Quartic with Noise.

Parameter	Tested Values	Default Value	Best Value	Mean Exact Error
α Scale	0.70, 0.85, 1.00, 1.15, 1.30	1.00	1.015	7.67×10^{-4}
β Scale	0.70, 0.85, 1.00, 1.15, 1.30	1.00	0.95	7.92×10^{-4}
Elite Fraction	0.10, 0.15, 0.20, 0.25, 0.30	0.20	0.25	8.77×10^{-4}

3.2. NJBO With IIR Filter Results

The overall search behavior of NJBO was evaluated using appropriate algorithms for benchmark-function validation, while comparison algorithms were chosen for the IIR engineering application based on their applicability and consistent use in related system identification research.

To assess the effectiveness and applicability of the NJBO for system identification, one benchmark IIR system was utilized. In this study, the unknown system is represented using a system of the same order as the benchmark. For the system input ip(t), a white sequence of 1000 samples was used.

Six well-known algorithms are used to compare the results with NJBO. These algorithms are Genetic Algorithm (GA) [33], Particle Swarm Optimization (PSO), Biogeography-Based Optimization (BBO), and Sine Cosine Algorithm (SCA). Wale Optimization Algorithm (WOA) and Sunflower Optimization Algorithm (SFOA). All methods have a population size of 100 and a maximum of 1000 iterations. Additionally, each technique was repeated 20 independent runs.

Example: Serruto [23] describes a second-order plant. This system's transfer function is as follows:

$$H_{\mathrm{p}}(z) = \frac{(1.25z^{-1} - 0.25z^{-2})}{(1 - 0.3z^{-1} + 0.4z^{-2})} \tag{19}$$

This second-order system was modeled using a second-order unknown system with the transfer function.

$$H_{IIR}(z) = \frac{(P_1z^{-1} + P_2z^{-2})}{(1 + q_1z^{-1} + q_2z^{-2})} \tag{20}$$

Table 12 shows the estimated coefficient values for several algorithms. NJBO outperforms other algorithms in estimating coefficients, as seen in the table. These results prove the accuracy of algorithm in finding the optimum solutions of coefficients.

Table 13 illustrates that the proposed algorithm with PSO was more accurate because of the effective results of the average rate of error and average standard deviation.

Table 14 shows statistical findings for the $U(\theta)$function, including best, worst, mean, and standard deviation. Table 14 shows that the proposed NJBO obtained the minimum mean error and performed competitively when compared to the other algorithms.

The convergence of the optimum solution for this application is illustrated in Fig. 10. The convergence behavior observed in the IIR filter design experiment demonstrates that NJBO, PSO, SBOA, and APO exhibit rapid movement toward the optimal solution within the early stages of the iterative process. However, PSO achieves this through a strong social cognitive component that drives aggressive exploitation, enabling it to reach the vicinity of the optimum in fewer iterations. In contrast, NJBO follows a more stable and adaptive trajectory; its learning phase balances exploration and exploitation through dynamic α–β regulation and a local memetic refinement stage.

The convergence curve of NJBO shows a smooth and monotonic decrease in error, indicating superior stability and consistency across runs. While PSO converged slightly faster in runtime due to its simpler update structure, NJBO achieved a lower final error (zero mean squared error with zero variance over 20 runs), confirming its higher precision and robustness. This behavior reflects NJBO's ability to avoid premature convergence and maintain exploitation efficiency until the global optimum is precisely reached. Therefore, NJBO and PSO can be considered the fastest algorithms in convergence, but NJBO distinguishes itself by combining high convergence speed with absolute accuracy and reliability, making it ideal for sensitive signal-processing and adaptive filtering tasks.

The results show that the NJBO algorithm achieves superior accuracy and stability across benchmark functions. Compared to PSO, BBO, APO, SFOA, WOA, SBOA, and SCA, the proposed algorithm shows improved convergence behavior and lower standard deviation.

In the IIR filter application, NJBO achieves highly accurate coefficient estimation with consistent performance across multiple runs.

Although NJBO requires slightly higher computational time, this is justified by its enhanced exploration mechanism and improved solution quality.

The convergence curves indicate that NJBO maintains a stable and monotonic improvement, avoiding premature convergence and ensuring robust performance.

3.3. NJBO With CW circuit results

The overall performance of the investigated algorithms is summarized in Table 15, where several statistical indicators such as the best, mean, median, worst objective values, the standard deviation of the objective function, the average time of execution, the mean number of function evaluations, and the overall ranking according to the mean objective value are reported. All algorithms were run over the same number of objective function evaluations (12,040), and all results were feasible solutions, making the comparison mainly between their optimization capability and not based on computational effort. Among all investigated algorithms, the proposed NJBO algorithm in this article has the lowest mean objective value (0.172872756), which gives this algorithm the first rank. Furthermore, the standard deviation of NJBO is the lowest (1.31696×10^{-9}) in the optimization process, which shows a very stable optimization process with highly repeatable results in independent runs. While the best objective value was approximately the same for both GA and APO, the standard deviations indicate poorer consistency of solutions compared to NJBO. As far as computational efficiency is concerned, all the algorithms have relatively similar average execution times ranging from around 1.96 s to 2.05 s, with NJBO showing a good improvement by optimizing the computation without a significant increase in computational overhead. Thus, NJBO's superiority is not due to more computing power, but to its search strategy.

Table 12. Estimated coefficient values for algorithms.

Coefficients	Actual value	PSO	BBO	SCA	WOA	SFOA	SBOA	APO	NJBO
a1	-0.3	-0.3794	-0.2642	-0.2985	-0.3006	-0.3201	-0.3	-0.3	-0.3
a2	0.4	0.4201	0.4093	0.3802	0.4095	0.4064	0.4	0.4	0.4
b1	1.25	1.2566	1.2440	1.2628	1.2496	1.2005	1.25	1.25	1.25
b2	0.25	0.3778	0.1750	0.2561	0.2506	0.2573	0.25	0.25	0.25

Table 13. Average error, standard deviation values, and average execution time

Algorithm	Avg %Error	Std %Error	Avg Time (s)	Std Time (s)
NJBO	0	0	0.47684062	0.004873823
APO	0	0	0.13214821	0.001894549
PSO	0	0	0.11252186	0.001771494
SBOA	0.018529801	0.00143558	0.10245874	0.000725317
BBO	1.287465451	0.69661088	0.15136140	0.088059386
SCA	3.959390818	3.441904901	0.117159265	0.000864421
SFOA	2.852667191	1.115909003	0.120986345	0.000922094

Table 14. Statistical findings for the $U(\theta)$function with second-order IIR filter

Results	PSO	BBO	SCA	WOA	SFOA	SBOA	APO	NJBO
Best	5.221E-06	2.65E-06	8.494E-07	5.347E-10	3.269E-13	4.129E-14	5.651E-15	1.651E-15
Worst	4.579E-05	1.851E-05	2.304E-05	1.055E-04	7.506E-06	2.107E-09	8.829E-11	5.865E-12
Mean	2.015E-05	9.609E-06	1.606E-06	2.317E-05	4.578E-07	2.824E-07	1.310E-11	1.150E-12
Std.	1.092E-05	4.572E-06	4.252E-06	3.187E-05	1.740E-06	6.519E-08	2.481E-12	1.651E-12

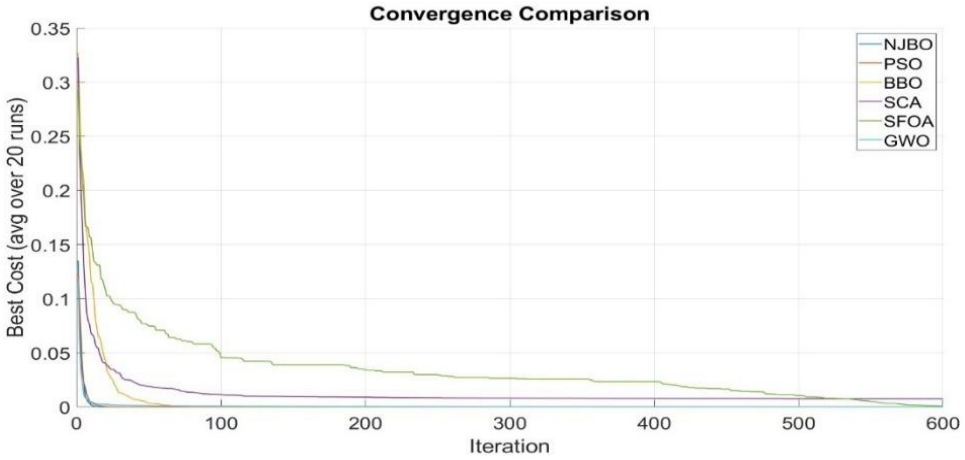


Figure 10. Convergence comparison of the IIR system.

In Table 16. The non-parametric Wilcoxon signed-rank test was carried out with NJBO as the reference algorithm to further confirm the statistical significance of the results obtained. The NJBO showed statistically significant improvements compared to GA, IPSO, BBO, SCA, WOA, SFOA, and SBOA, as presented in Table 16, where the Holm-adjusted p-values are below 0.05. NJBO has more wins (13) than losses (7), and the only exception (APO) had an adjusted p-value (0.6012), which means that the difference in performance is not statistically significant. This observation indicates that APO is the best competitor for this engineering optimization problem, while NJBO still has the best overall numerical performance.

Friedman Ranking Analysis: Overall comparison of all competing algorithms is given in the Friedman ranking results shown in Table 17. The proposed NJBO performed best with the Friedman mean rank (1.65), and ranked first of all nine optimization methods. The second overall ranking was achieved by APO with a mean rank of 2.35, followed by GA with a mean rank of 2.75, and the lowest overall ranking was obtained by IPSO with a mean rank of 8.625. As opposed to pairwise statistical testing, the Friedman test compares all algorithms against the whole set of experimental data. So, getting the first Friedman rank means that NJBO has been able to consistently outperform other optimizations, not just in a few instances. Overall, these results demonstrate the robustness, stability and competitive optimization ability of the proposed algorithm.

The optimal values of design variables and the electrical performance of each optimization method for the five-stage Cockcroft–Walton voltage multiplier are shown in Table 18. While all algorithms yielded feasible solutions, the suggested NJBO algorithm produced the best objective value, and it has the highest solution consistency. In addition, the optimal capacitor distribution designed by NJBO results in a desired relationship of various parameters (output voltage, ripple voltage, voltage drop, efficiency, regulation, and total capacitance), which shows its efficiency in solving real engineering optimization problems without any additional computational complexity.

Table 15. Overall performance of the investigated algorithms.

Metric	GA	IPSO	BBO	SCA	WOA	SFOA	SBOA	APO	NJBO
Best	0.172872791	0.172881001	0.172875661	0.172883206	0.17287282	0.172873863	0.172878426	0.172872755	0.172872755
Mean	0.172873097	0.173075242	0.172887757	0.172904485	0.172876585	0.172876338	0.172935252	0.172876861	0.172872756
Std	3.09012E-07	0.000138919	9.13779E-06	9.07156E-06	3.44689E-06	1.43051E-06	7.4816E-05	8.40965E-06	1.31696E-09
Median	0.172872995	0.173064534	0.172884863	0.172906223	0.172875364	0.17287618	0.172908336	0.172872755	0.172872756
Worst	0.172873803	0.173215062	0.172904314	0.172917993	0.172883555	0.172879324	0.173123647	0.172895971	0.172872759
Mean Time (s)	1.983917255	1.958616505	2.04631782	1.97642299	1.968573015	1.966777955	1.97472827	2.03249509	1.98059527
Mean Rank	2	9	6	7	4	3	8	5	1

Table 16. The non-parametric Wilcoxon signed-rank test.

Reference Algorithm	Compared Algorithm	Raw P Value	Holm Adjusted P Value	Signed Rank Statistic	NJBO Wins	Ties	NJBO Losses	Significant at 0.05
NJBO	GA	8.85746E-05	0.000708597	0	20	0	0	TRUE
NJBO	IPSO	8.85746E-05	0.000708597	0	20	0	0	TRUE
NJBO	BBO	8.85746E-05	0.000708597	0	20	0	0	TRUE
NJBO	SCA	8.85746E-05	0.000708597	0	20	0	0	TRUE
NJBO	WOA	8.85746E-05	0.000708597	0	20	0	0	TRUE
NJBO	SFOA	8.85746E-05	0.000708597	0	20	0	0	TRUE
NJBO	SBOA	8.85746E-05	0.000708597	0	20	0	0	TRUE
NJBO	APO	0.601212706	0.601212706	91	13	0	7	FALSE

Table 17. Friedman Ranking Analysis: Overall comparison of all competing algorithms.

Algorithm	Friedman Mean Rank	Rank Position
GA	2.75	3
IPSO	8.625	9
BBO	6.25	6
SCA	7.35	7
WOA	4.25	4
SFOA	4.4	5
SBOA	7.375	8
APO	2.35	2
NJBO	1.65	1

Table 18. The optimal values of design variables.

Field	GA	PSO	BBO	SCA	WOA	SFOA	SBOA	APO	NJBO
Best Run	6	13	2	6	14	13	5	3	12
Best Cost	0.17287279	0.17288100	0.17287566	0.17288320	0.17287282	0.17287386	0.17287842	0.17287275	0.17287275
Cp1 ($\boldsymbol{uF}$)	0.47023175	0.47	0.48477575	0.51796076	0.47	0.47	0.47	0.47	0.47
Cp2 ($\boldsymbol{uF}$)	0.47013823	0.47	0.51595860	0.48706934	0.47373225	0.47	0.47	0.47	0.47
Cp3 ($\boldsymbol{uF}$)	0.49734217	0.47	0.48770343	0.50635823	0.50508907	0.49985075	0.52896240	0.50170227	0.50195987
Cp4 ($\boldsymbol{uF}$)	0.54535932	0.47	0.56244658	0.49806548	0.53745466	0.54040709	0.47	0.54317105	0.54306796
Cp5 ($\boldsymbol{uF}$)	0.58934150	0.66875818	0.57245408	0.63731080	0.59032697	0.58892282	0.63687462	0.58804365	0.58801280
Cs1 ($\boldsymbol{uF}$)	3.26448836	3.26439414	3.21224833	3.26117824	3.26354962	3.22453541	3.26252554	3.26298423	3.26270631
Cs2 ($\boldsymbol{uF}$)	4.29160514	4.28991494	4.27590295	4.25638877	4.28662134	4.26608501	4.28774884	4.28805932	4.28781838
Cs3 ($\boldsymbol{uF}$)	5.17172783	5.17821142	5.16355197	5.21951337	5.17049786	5.15951507	5.17555262	5.17563755	5.17597885
Cs4 ($\boldsymbol{uF}$)	6.14517279	6.15238755	6.17844255	6.23154709	6.15039934	6.15556595	6.14905312	6.14888621	6.14861183
Cs5 ($\boldsymbol{uF}$)	6.78708272	6.79432392	6.80231251	6.72347571	6.79214001	6.83864307	6.79057037	6.79011131	6.78984446
$\boldsymbol{Vout}$ V	781.266653	781.194847	781.278577	781.389880	781.265030	781.247475	781.220986	781.261149	781.261273
Ripple $\boldsymbol{Vpp}$	0.27400407	0.27379583	0.27424978	0.27368439	0.27397069	0.27413436	0.27394263	0.27394073	0.27394794
Voltage Drop V	218.733346	218.805152	218.721422	218.610119	218.734969	218.752524	218.779013	218.738850	218.738726
Efficiency	78.9717844	78.9702431	78.9720701	78.9744459	78.9717528	78.9713767	78.9708138	78.9716681	78.9716708
Regulation	21.8733346	21.8805152	21.8721422	21.8610119	21.8734969	21.8752524	21.8779013	21.8738850	21.8738726
Total C($\boldsymbol{uF}$)	28.232489	28.227990	28.255796	28.3388678	28.2398111	28.2135252	28.2412875	28.2385956	28.2380005
Feasible	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

In addition to the numerical results shown for the benchmark functions and for the two engineering applications, namely, optimization of a Cockcroft–Walton voltage multiplier and optimization of an IIR digital filter, there are several engineering features that are shown that may be useful in practical optimization applications. The adaptive mechanism of balancing exploration and exploitation during the search process maintains the population diversity to the greatest extent; the elite-preservation mechanism ensures the consistency of the solutions found across different independent runs; the gibberellin-inspired perturbation mechanism enables the algorithm to escape from local optima. Such mechanisms ensure that NJBO preserves stable search behavior and is a good optimization framework for complex engineering design problems.

4. Limitations of NJBO

Despite its strong performance, the NJBO algorithm has certain limitations.

The computational time is relatively higher compared to simpler algorithms such as PSO due to its adaptive exploration mechanisms.

Additionally, the performance of NJBO may depend on parameter tuning, which requires further investigation in future work.

5. Conclusion and Future Work

In this study, the authors proposed a new nature-inspired metaheuristic called the Night Jasmine Blooming Optimization (NJBO) algorithm, which is inspired by the blooming property of a night jasmine plant (Cestrum nocturnum). The proposed algorithm combines adaptive hormonal regulation, gibberellin-based disturbance, scent-guided exploitation, elite preservation, and Nelder–Mead local refinement in order to provide a balance between exploration and exploitation that is good enough.

The performance of NJBO was tested by thirteen classical benchmark functions and the CEC2017 benchmark suite, then compared to seven popular and recent optimization algorithms: Particle Swarm Optimization (IPSO), Biogeography-based Optimization (BBO), Whale Optimization Algorithm (WOA), Smart Flower Optimization Algorithm (SFOA), Secretary Bird Optimization Algorithm (SBOA), and Artificial Protozoa Optimizer (APO). The results obtained showed that NJBO is comparable in terms of optimization accuracy, convergence stability, and solution variance in a broad spectrum of optimization problems. Other analyses, such as the scalability evaluation, running time assessment, Wilcoxon signed-rank test, Friedman ranking test, component contribution analysis, and parameter sensitivity analysis, also verified the robustness and effectiveness of the proposed optimization strategy.

The proposed algorithm was also tested practically in two engineering optimization problems, that is, Infinite Impulse Response (IIR) system identification and optimum design of a Cockcroft–Walton voltage multiplier. For both uses, NJBO delivered very competitive solutions, with consistent statistical performance.

NJBO has a number of adaptive features that improve the accuracy of optimization and stability of the convergence process; however, these features add to the overall computational time of the NJBO compared to less sophisticated optimization algorithms.

Future work will concentrate on minimizing the computational expense of NJBO by parallel implementation, implementation on GPUs, adaptive population sizing, and surrogate-assisted optimization. Further, using adaptive termination criteria can make even further fitness evaluations unnecessary without compromising the quality of the solutions. These enhancements should improve NJBO's ability to efficiently solve large-scale and computationally intensive engineering optimization problems.

Future research will look into combining NJBO with other algorithms and expanding its applications, such as the application of [34] for high-dimensional optimization issues and real-time signal processing as in [21].

Acknowledgments

The authors would like to express their appreciation to Mustansiriyah University (www.uomustansiriyah.edu.iq), Baghdad, Iraq for its assistance and academic advice in the writing of this work.

Conflict of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Use of LLM and AI

The authors declare that no AI tools were used in the preparation of this article.

Author Contributions Statement Authors

Hussein M. Hathal and Riyadh A. Abdulhussein proposed the research problem.

Hussein M. Hathal and Riyadh A. Abdulhussein developed the theory and carried out the calculations.

Hussein M. Hathal, and Riyadh A. Abdulhussein verified the analytical methods and investigated the performance comparisons of algorithms and managed the results of this article.

All authors analyzed the findings and participated in the final article.

References

[1] X. Zhou, T. Fang, and Y. Mao, “Rate-splitting multiple access for green communications: A survey and robust beamforming design”. *IEEE Internet of Things Journal,* vol. 12, no. 10, pp. 14469–14483. 2025. doi: https://doi.org/10.1109/JIOT.2025.3526250.

[2] D. Costarelli and A. R. Sambucini, “A comparison among a fuzzy algorithm for image rescaling with other methods of digital image processing,” *Constructive Mathematical Analysis*, vol. 7, no. 2, pp. 45–68, Jun. 2024. doi: https://doi.org/10.33205/cma.1467369.

[3] S. P. Mishra, D. R. Parhi, and A. K. Sahoo, “Integration of white shark algorithm with PID in U-ROV navigation for enhanced obstacle avoidance”. *Ocean Engineering*, vol. 341, Art. no. 122501, 2025. doi: https://doi.org/10.1016/j.oceaneng.2025.122501.

[4] J. G. Wang, W. T. Qin, D. C. Ran, P. Wu, K. Wang, and Y. Liu, “Research on multi-satellite collaborative task planning technology,” *Journal of Physics: Conference Series*, vol. 3022, no. 1, Art. no. 012011, 2025. doi: https://doi.org/10.1088/1742-6596/3022/1/012011.

[5] R. Natarajan, G. Megharaj, A. Marchewka, P. B. Divakarachari, and M. R. Hans, “Energy and distance based multi-objective red fox optimization algorithm in wireless sensor network,” *Sensors*, vol. 22, no. 10, Art. no. 3761, May 2022. doi: https://doi.org/10.3390/s22103761.

[6] P. Trojovský and M. Dehghani, “Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications”. Sensors, vol. 22, no. 3, Art. no. 855, Jan. 2022. doi: https://doi.org/10.3390/s22030855.

[7] F. A. Hashim and A. G. Hussien, “Snake optimizer: A novel meta-heuristic optimization algorithm,” Knowledge-Based Systems, vol. 242, Art. no. 108320, Apr. 2022. doi: https://doi.org/10.1016/j.knosys.2022.108320.

[8] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, “Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer,” *Expert Systems with Applications*, vol. 191, Art. no. 116158, Apr. 2022. doi: https://doi.org/10.1016/j.eswa.2021.116158.

[9] A. K. Singh and S. Mehta, “Performance evaluation of smart flower optimization algorithm over industrial non-convex constrained optimization problems,” in: Jabbar, M.A., Ortiz-Rodríguez, F., Tiwari, S., Siarry, P. (eds) Applied Machine Learning and Data Analytics. AMLDA 2022. Communications in Computer and Information Science, vol 1818.

Springer, Cham, 2023, pp. 231–239. doi: https://doi.org/10.1007/978-3-031-34222-6_20.

[10] N. Chopra and M. M. Ansari,"Golden jackal optimization: A novel nature-inspired optimizer for engineering applications ". *Expert Systems with Applications,* vol. 198, Art. no. 116924, Jul. 2022. doi: https://doi.org/10.1016/j.eswa.2022.116924.

[11] C. Zhong, G. Li, and Z. Meng, "Beluga whale optimization: A novel nature-inspired metaheuristic algorithm". *Knowledge-Based System*s, vol. 251, Art. no. 109215, 2022. doi: https://doi.org/10.1016/j.knosys.2022.109215.

[12] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, "White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems," *Knowledge-Based Systems*, vol. 243, Art. no. 108457, 2022. doi: https://doi.org/10.1016/j.knosys.2022.108457.

[13] K. H. Almotairi and L. Abualigah, "Hybrid reptile search algorithm and remora optimization algorithm for optimization tasks and data clustering," *Symmetry*, vol. 14, no. 3, Art. no. 458, Mar. 2022. doi: https://doi.org/10.3390/sym14030458.

[14] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, Art. no. 110011, 2023. doi: https://doi.org/10.1016/j.knosys.2022.110011.

[15] G. Zhou, S. Lü, L. Mao, K. Xu, T. Bao, and X. Bao, "Path planning of UAV using Levy Pelican Optimization Algorithm in mountain environment," *Applied Artificial Intelligence*, vol. 38, no. 1, Art. no. 2368343, 2024. doi: https://doi.org/10.1080/08839514.2024.2368343.

[16] X.-S. Yang, Nature-Inspired Optimization Algorithms, 1st ed. Amsterdam, The Netherlands: Elsevier, 2014. doi: https://doi.org/10.1016/C2013-0-01368-0.

[17] X. Li, B. Hu, Z. Chen, and B. Zeng, "The effects of light quality on growth and physiological responses of *Aquilaria crassna* tissue-cultured plantlets," *Horticulturae*, vol. 12, no. 3, Art. no. 394, Mar. 2026. doi: https://doi.org/10.3390/horticulturae12030394.

[18] P. H. Raven, G. B. Johnson, K. A. Mason, J. B. Losos, and S. R. Singer, *Biology*, 11th ed. New York, NY, USA: McGraw-Hill Education, 2017.

[19] L. G. Verdolin, B. L. Mariz, and L. L. C. Dias, "Gibberellin and polyamines effects in growth and flowering of New Guinea impatiens," Ornamental Horticulture, vol. 27, no. 2, pp. 247–254, 2021. doi: https://doi.org/10.1590/2447-536X.v27i2.2255.

[20] S. Qiao et al., "Unraveling the mechanism of fragrance release in Cestrum nocturnum through transcriptome and volatile compound profiling," *Scientific Reports*, vol. 15, Art. no. 15376, May 2025. doi: https://doi.org/10.1038/s41598-025-99542-3.

[21] H. M. Hathal, R. S. Ali, and A. S. Abdullah, "A novel metaheuristic Moss-Rose-inspired algorithm with engineering applications," *Electronics*, vol. 10, no. 16, Art. no. 1877, Aug. 2021. doi: https://doi.org/10.3390/electronics10161877.

[22] L. Abualigah, D. Izci, M. Jabari, S. Ekinci, K. Saleem, H. Migdady, and A. Smerat, "Adaptive Gbest-guided atom search optimization for designing stable digital IIR filters," *Circuits, Systems, and Signal Processing*, vol. 44, no. 6, pp. 4059–4081, Jun. 2025. doi: https://doi.org/10.1007/s00034-025-02997-y.

[23] W. F. Serruto, "Optimization of IIR digital filters using differential evolution: A comparative analysis of FDDE and AMECoDEs algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 3, pp. 168–180, 2025. doi: https://doi.org/10.14569/IJACSA.2025.0160317.

[24] Z. Liang, T. Shu, and Z. Ding, "A novel improved whale optimization algorithm for global optimization and engineering applications," *Mathematics*, vol. 12, no. 5, Art. no. 636, Feb. 2024. doi: https://doi.org/10.3390/math12050636.

[25] U. Thaker and S. C. Vora, "Review of Cockcroft-Walton high voltage low current DC power supplies," *Power Research—A Journal of CPRI*, vol. 19, no. 2, pp. 217–224, Dec. 2023. doi: https://doi.org/10.33686/pwj.v19i2.1129.

[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks (ICNN'95)*, Perth, WA, Australia, Nov. 27–Dec. 1, 1995, vol. 4, pp. 1942–1948. doi: https://doi.org/10.1109/ICNN.1995.488968.

[27] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, Dec. 2008. doi: https://doi.org/10.1109/TEVC.2008.919004.

[28] X. Liang, Z. Cai, M. Wang, X. Zhao, H. Chen, and C. Li, "Chaotic oppositional sine–cosine method for solving global optimization problems," *Engineering with Computers*, vol. 38, no. 2, pp. 1223–1239, Mar. 2022. doi: https://doi.org/10.1007/s00366-020-01083-y.

[29] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016. doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[30] E. G. Magacho, A. B. Jorge, and G. F. Gomes, "Inverse problem based multiobjective sunflower optimization for structural health monitoring of three-dimensional trusses," *Evolutionary Intelligence*, vol. 16, no. 1, pp. 247–267, Feb. 2023. doi: https://doi.org/10.1007/s12065-021-00652-4.

[31] Y. Fu, D. Liu, J. Chen, and L. He, "Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems," *Artificial Intelligence Review*, vol. 57, no. 5, Art. no. 123, 2024. doi: https://doi.org/10.1007/s10462-024-10729-y.

[32] X. Wang, V. Snášel, S. Mirjalili, J.-S. Pan, L. Kong, and H. A. Shehadeh, "Artificial Protozoa Optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization," *Knowledge-Based Systems*, vol. 295, Art. no. 111737, Jul. 2024. doi: https://doi.org/10.1016/j.knosys.2024.111737.

[33] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, 2nd ed. Cambridge, MA, USA: MIT Press, 1992. doi: https://doi.org/10.7551/mitpress/1090.001.0001.

[34] A. Al-Gizi, A. Miry, H. M. Hathal, and A. Craciunescu, "Fuzzy maximum power point tracking controllers for photovoltaic systems: A comparative analysis," *Journal of Engineering and Sustainable Development*, vol. 28, no. 3, pp. 364–374, May 2024. doi: https://doi.org/10.31272/jeasd.28.3.6.