# Performance Dynamics of Bio-Inspired Metaheuristics: A Comparative Study

**OPEN ACCESS**

Volume: 13

Special Issue: 2

Month: January

Year: 2026

E-ISSN: 2582-0397

P-ISSN: 2321-788X

Citation:
Karthiga, K., and B. Rajdeepa. "Performance Dynamics of Bio-Inspired Metaheuristics: A Comparative Study." *Shanlax International Journal of Arts, Science and Humanities*, vol. 13, no. 2, 2026, pp. 67–75.

DOI:
https://doi.org/10.34293/sijash.v13iS2-i3-Jan.10552

**K. Karthiga**
*Research Scholar, Department of Computer Science*
*PSG College of Arts & Science, Coimbatore, Tamil Nadu, India*

**Dr. B. Rajdeepa**
*Associate Professor and Head, Department of Information Technology*
*PSG College of Arts & Science, Coimbatore, Tamil Nadu, India*

**Abstract**
*Bio-inspired optimisation leverages the principles of biological systems to address complex computational challenges. This paper provides an analytical study of the field, categorising major techniques based on their evolution, underlying mechanisms, and performance. By comparing these modern approaches with traditional legacy methods, the study highlights the unique strengths and limitations of nature-inspired algorithms. This review serves as a foundational resource for researchers seeking to identify current trends and future directions in optimisation research. Furthermore, the paper synthesises recent empirical data to evaluate the scalability and convergence speed of these algorithms across diverse datasets. Ultimately, this survey serves as a strategic roadmap, identifying current research lacunae and proposing future directions for the development of hybrid and hyper-heuristic bio-inspired frameworks.*

**Keywords: Application, Bio-Inspired Optimisation, Development, Intent, Optimisation, Performance**

## Introduction

Historically, the distinction between biological entities and computational systems was considered absolute. In a traditional context, biological organisms (such as ants, bees, or birds) and technical components (such as algorithms or operating systems) represent disparate domains with no functional overlap. Indeed, the physical incompatibility of these worlds is often underscored by the fact that biological elements, such as water, are typically detrimental to hardware integrity.

The rapid evolution of technology has significantly transformed the perceived divide between biological and computational systems. Contemporary computer science increasingly incorporates biological concepts, integrating them into computational frameworks and thereby blurring the distinction between the two domains. Biology now serves as a crucial foundation for evaluating algorithmic complexity and improving optimisation outcomes. This convergence highlights the adaptability of computational methods, with natural processes offering models for addressing complex engineering problems [1]. Rather than promoting any particular bio-inspired technique, this study delivers a rigorous and impartial analysis of the current landscape. The principal objective is to provide a critical survey of bio-inspired optimisation techniques.

The conceptual flow and organisational structure of this survey are illustrated in Figure 1.

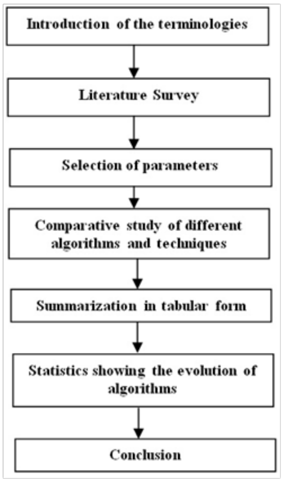


**Figure 1 Conceptual Flow and Organisational Structure of the Survey**

### Defining Optimisation Techniques

Optimisation is the systematic process of identifying the most effective solution from a set of available alternatives. In any given field, the primary objective is to achieve superior results while minimising resource consumption. Mathematically, this involves identifying the optimal parameters that maximise or minimise a specific objective function. In a typical optimisation problem, the goal is to determine the variables that govern system behaviour to maximise productivity or minimise waste and inefficiency.

### Conceptualising Bio-Inspired Optimisation

Recent advancements have marked a transition from traditional, gradient-based methods to a more versatile category of bio-inspired optimisation techniques. These methodologies are modelled after natural phenomena, such as the evolutionary selection of species and the collective intelligence observed in biological societies. By mimicking these adaptive behaviours, bio-inspired algorithms offer robust solutions to complex problems, effectively overcoming the limitations—such as premature convergence—associated with classical techniques. Consequently, these methods have gained significant traction within both academic research and industrial applications [2].

### The Framework of a Critical Survey

A critical survey is defined as an exhaustive, qualitative, and quantitative investigation that transcends a basic summary. It involves a granular examination of specific methodologies to uncover underlying complexities and performance nuances.

### A Critical Review of Bio-Inspired Techniques

This study provides a comprehensive investigation into the landscape of bio-inspired optimisation. The objective is to evaluate diverse algorithms within a unified framework, analysing them through standardised parameters such as convergence rate, computational complexity, and robustness. By navigating these techniques on a singular comparative platform, this survey seeks to provide a definitive roadmap for their application and future development. In the contemporary era, the rapid pace of technological innovation has introduced significant computational complexities. Optimisation remains one of the most critical methodologies for addressing these challenges.

## Literature Survey

### The Classical Era of Optimisation

The mathematical foundations of optimisation can be traced back to the pioneering work of Newton, Leibniz, Lagrange, and Bernoulli. The advent of differential calculus by Newton and Leibniz provided the essential tools for early optimisation, while Lagrange introduced transformative methods for solving constrained problems. The mid-20th century marked a significant turning point with the introduction of high-speed computing, which allowed for the implementation of sophisticated algorithms. Key milestones include the linear programming simplex method, developed by Dantzig in 1947; Bellman's formalisation of the optimality principle for dynamic programming; the emergence of Non-linear Programming (Zovtendijk and Rosen) in 1960; as well as Geometric, Integer, and Stochastic programming (Duffin, Gomory, and Cooper). Subsequently, Multi-Objective Programming was developed to address scenarios requiring the simultaneous optimisation of multiple, often conflicting, objectives under specific constraints.

### The Biological Paradigm Shift

The integration of biological principles into optimisation represents a significant advancement in computer science. Bio-inspired optimisation serves as an umbrella term for a diverse array of metaheuristic approaches modelled after the efficiency and adaptability of natural systems. Notable examples include Swarm Intelligence: Ant Colony Optimisation (ACO), Bee Colony Optimisation (BCO), Bat Algorithm (BA), and Krill Herd (KH); Evolutionary Computation: Genetic Algorithms (GA) and broader Evolutionary Algorithms (EA); and other metaheuristics such as Simulated Annealing (SA) and Neural Networks (NN).

Bio-inspired optimisation was classified into three categories: Evolutionary algorithms, Swarm intelligence, and Bacterial foraging, as shown in Table 1.

### Motivation for Bio-Inspired Research

The burgeoning interest in this field stems from the increasing prevalence of large-scale, ill-structured, and ambiguous problems that traditional mathematical methods struggle to solve. In contrast, nature demonstrates that seemingly simple biological organisations are capable of performing highly complex tasks with remarkable efficiency. This survey provides a comprehensive synthesis of recent developments in the bio-inspired domain. By offering a "state-of-the-art" overview of theoretical and empirical results, this work serves as a foundational resource for researchers aiming to advance the frontiers of optimisation science.

**Table 1 Classification of Bio-Inspired Optimisation Algorithms**

| Algorithm | Definition | Classifications |
|---|---|---|
| Evolutionary | Inspired by the genetic evolution process. Population-based stochastic search algorithms working with best-to-survive criteria. | GA (Genetic Algorithm); GP (Genetic Programming); ES (Evolutionary Strategy); DE (Differential Evolution) |

| | | |
|---|---|---|
| Swarm Intelligence | Based on the collective social behaviour of organisms. Implements collective intelligence of groups of simple agents based on the behaviour of real-world insect swarms as a problem-solving tool. | ACO (Ant Colony Optimisation); PSO (Particle Swarm Optimisation); RSO (Reactive Search Optimisation); GSA (Gravitational Search Algorithm); KH (Krill Herd); SDS (Stochastic Diffusion Search); BA (Bat Algorithm); IWD (Intelligent Water Drop); FSA (Fish Swarm Algorithm); FA (Firefly Algorithm); AIS (Artificial Immune System) |
| Bacterial Foraging | Bacterial Foraging behaviour serves as the source for developing a bio-inspired optimisation approach called the bacterial foraging algorithm. | COSMIC (Computing System of Microbial Interactions and Communications); RUBAM (Rule-Based Bacterial Modelling) |

**Critical Comparative Study**

This section compares key bio-inspired optimisation algorithms based on development, intent, performance, and applications. Development traces the evolution of algorithms over time, including key researchers and milestones. Intent outlines the core purpose, optimisation goals, and mechanisms employed. Performance evaluates overall effectiveness, achievement methods, and improvement potential. Applications identifies practical domains where algorithms are applied.

**Stud Genetic Algorithm (SGA)**

**Development:** Evolutionary computing pioneers like Fogel (1995) highlighted algorithm strengths and weaknesses. Multidisciplinary optimisation (MDO) needs spurred SGA's creation.

**Intent:** Pairs the population's best individual with others for offspring, skipping stochastic selection. Relies on crossover (two-point or uniform) and low-probability bit mutation.

**Performance:** Tested on challenging functions from Keane (1996), Fogel (1995), and De Falco et al. (1996). Uses discrete bit representation without stochastic selection, excelling in multimodal spaces.

**Applications:** Power systems, data mining rule extraction, robot path planning, scheduling, portfolio optimisation, pattern recognition, flight control, and multi-objective vehicle routing.

**Ant Colony Optimisation (ACO)**

**Development:** Inspired by Grasse's 1959 Stigmergy theory on nest-building. Dorigo introduced Ant System (1991) and Ant Colony System (1997).

**Intent:** Mimics ants' pheromone-based shortest-path finding for synchronised, local solution searches.

**Performance:** Enhanced via ACS with modified transition/pheromone rules, local updates, candidate lists, and pheromone-table crossovers to boost exploration.

**Applications:** Scheduling (project, open shop); routing (vehicle, TSP, sequential); assignment (quadratic, graph colouring); set problems; data mining, image processing, intelligent testing.

**Particle Swarm Optimisation (PSO)**

**Development:** Debuted by Kennedy and Eberhart (1995) in "Particle Swarm Intelligence," followed by modifications (Shi & Eberhart, 1998; Kennedy & Eberhart, 2001). Recent advances include Roy et al. (2012) for multi-objective combinatorial problems.

**Intent:** Stochastic swarm-based method using social interaction; particles adjust positions based on personal and neighbourhood best fitness experiences.

**Performance:** Simple, parameter-light heuristic outperforms others in ease and effectiveness.

**Applications:** System design, image classification, pattern recognition, biological modelling, scheduling, signal processing, robotics, decision-making, simulation, gesture recognition, and target detection.

### Reactive Search Optimisation (RSO)

**Development:** Came into existence through the book Reactive Search and Intelligent Optimisation (Springer Verlag, 2000), authored by Roberto Battiti, Mauro Brunato, and Franco Mascia. Local search heuristics are the reason behind the evolution of RSO.

**Intent:** RSO aims to bring maximum ease to the final user of the optimisation through characteristics such as learning on the job, rapid generation and analysis of many alternatives, flexible decision support, diversity of solutions, and anytime solutions. It aims to provide automation of the complete optimisation procedure, including the fine-tuning phase; dynamic adjustment of search parameters, possibly at every search step; and enhanced reproducibility of results [23].

**Performance:** Intelligent optimisation, being the superset of the basic search method, provides enhanced solutions based on both offline and online schemes using incremental models, intelligent tuning, adaptation, and memory.

**Applications:** Quadratic assignment, training neural nets, control problems, vehicle-routing, structural acoustic control, VLSI realisations, graph partitioning, electric power distribution, maximum satisfiability, constraint satisfaction, optimisation of continuous functions, traffic grooming in optical networks, maximum clique, time dispatch of trams in storage yards, roof truss design, increasing internet capacity, improving vehicle safety, and aerial reconnaissance simulations.

### Gravitational Search Algorithm (GSA)

**Development:** GSA was introduced in 2009 by Rashedi et al. as a modern heuristic approach based on the metaphor of gravitational interaction between masses, inspired by Newton's laws of physics.

**Intent:** GSA is a population-based model that utilises the laws of gravity and motion. Agents are treated as objects with varying masses that "perceive" the positions of others. Information is transferred via gravitational force; heavier masses represent superior solutions. Over time, smaller masses are attracted to the heaviest mass, eventually identifying the optimum point in the search space.

**Performance:** GSA is computationally efficient and requires significantly less execution time compared to Genetic Algorithms (GA). Benchmarks show that GSA consistently outperforms GA in terms of processing speed.

**Applications:** As a multidisciplinary tool, GSA is used for optimal reactive power dispatch, data clustering, and bioinformatics, specifically in gene clustering.

### Krill Herd (KH) Algorithm

**Development:** Proposed by Gandomi and Alavi in 2012, this algorithm was inspired by the herding and schooling behaviours of Antarctic krill (Euphausia superba). These small crustaceans form dense groups in response to ocean circulations, which served as the blueprint for this computational model.

**Intent:** The algorithm focuses on the movement of individual krill toward food sources and the highest density of the herd. Movement is determined by three factors: interactions with other individuals, foraging activity, and random diffusion. To increase precision, genetic operators like mutation and crossover are integrated into the model.

**Performance:** When tested against benchmark problems, KH was evaluated in four versions (varying the use of genetic operators). Comparative studies against GA, PSO, and Differential Evolution (DE) demonstrate that KH generally provides superior results.

**Applications:** As part of the swarm intelligence family, KH is used in crowd simulation, combinatorial optimisation, and specialised fields like controlling nanobots for cancer detection.

**Stochastic Diffusion Search (SDS)**

**Development:** Introduced by J.M. Bishop in 1989, SDS is a population-based pattern-matching algorithm. It is a member of the swarm intelligence family and is modelled after natural search and optimisation processes.

**Intent:** The primary goal of SDS is to locate a specific data pattern within a search space. Agents maintain "hypotheses" about potential solutions and evaluate them partially. Successful agents communicate directly to recruit unsuccessful ones, creating a positive feedback loop. This mechanism balances the trade-off between exploring the entire search space and exploiting specific, promising areas.

**Performance:** Unlike many metaheuristics, SDS is supported by a robust mathematical framework. Research has confirmed its strengths in global optimality, linear time complexity, system robustness, and efficient resource allocation.

**Applications:** SDS has been successfully applied to text search, object recognition, and feature tracking. It is also used in mobile robot self-localisation, wireless network site selection, and medical tracking technologies like eye and lip movement analysis.

**Bat Algorithm (BA)**

**Development:** Introduced by Xin-She Yang in 2010, the Bat Algorithm is a prominent metaheuristic optimisation method. The framework was later refined in 2012 by A.H. Gandomi et al. to specifically address complex constrained optimisation tasks.

**Intent:** The BA algorithm is inspired by the echolocation capabilities of microbats, which allow them to navigate and locate prey in total darkness. In this model, "bats" (agents) fly through the search space with specific velocities, frequencies, and loudness levels. As they approach a target solution (prey), they dynamically adjust their pulse emission rates and loudness. The optimisation process is refined through local random walks until the specified stopping criteria are met.

**Performance:** BA effectively integrates the advantages of various metaheuristic strategies. Benchmark studies demonstrate that the Bat Algorithm consistently outperforms other heuristic methods in terms of both accuracy and computational efficiency, making it a highly robust tool for complex searches.

**Applications:** The algorithm is widely applied in engineering design and classification problems. Specifically, bat clustering techniques have been successfully utilised to solve ergonomic workplace optimisation challenges.

**Computing System of Microbial Interactions and Communication (COSMIC)**

**Development:** The COSMIC framework was established by Gregory, Paton, and Saunders in 2004. It was further enhanced in 2006 with the introduction of "COSMIC-Rules," a rule-based bacterial simulation system developed by R. Gregory, V.A. Saunders, and J.R. Saunders.

**Intent:** COSMIC models cellular evolution using discrete time and pseudo-continuous space. The system simulates genomes within cells that interact with each other and their external environment. This approach combines the original COSMIC principles with Rule-based Bacterial Modelling (RUBAM) across three distinct levels: the genome, the individual cell, and the shared environment.

**Performance & Applications:** By applying rules derived from the principles of bacterial genetics, COSMIC-Rules reproduces biologically realistic conditions. This model is primarily used to study bacterial evolution and adaptation, providing researchers with the ability to predict the behaviour and growth patterns of pathogenic bacteria.

**Rule-Based Bacterial Modelling (RUBAM)**

**Development:** Developed in 2006 by Paton, Vlachos, Wu, and Saunders, RUBAM represents a distinct paradigm from Genetic Algorithms or Genetic Programming. It functions as a streamlined version of the

COSMIC framework, utilising discrete time, simplified genomes, and pseudo-continuous space. Its logic is derived from the complex interactions between organisms and their environment within natural bacterial ecologies.

**Intent:** In this model, bacteria are treated as information-processing units that map environmental "messages" to specific "actions." This mapping is dynamic; through mutation, the system evolves continuously, allowing organisms to adapt to environmental shifts, extend their functional lifespan, and optimise reproductive processes. Rather than acting as a literal simulation of real-world ecology, RUBAM aims to extract and apply the adaptive features and evolutionary patterns of bacteria to computational problem-solving.

**Performance & Applications:** The design of RUBAM focuses on achieving a balance between biological complexity and computational efficiency. The architecture comprises an artificial environment, a collection of artificial organisms, and a suite of evolutionary operators. Knowledge is not pre-programmed but emerges through the evolutionary process within the organisms. In multi-modal search spaces, RUBAM is particularly effective at generating multiple optimal solutions simultaneously. Unlike traditional methods that optimise toward an external objective function, RUBAM measures fitness by allowing individual organisms to interact with the environment until their energy reserves are depleted. Furthermore, the solutions generated are distributed across multiple processors, making it suitable for parallel computing environments [33].

| Algorithms | Year of Development | Researcher | Description |
|---|---|---|---|
| SGA | 1995 | Fogel | Crossover is the heart of this algorithm. Mate with the best individual for producing new offspring. Uses two types of crossover. |
| ACO | 1991 | M. Dorigo | Ants are able to find the shortest route to their food. Uses their antenna and pheromones for communication. This capability of monitoring and synchronization helps in finding solutions for local problems. |
| PSO | 1995 | J. Kennedy and R. Eberhart | Based on movement and intelligence of swarms. Each particle uses its own as well as the experience of other particles. Used to work out the most challenging problems. |
| RSO | 2000 | Roberto Battiti, Mauro Brunato and Franco Mascia | Brings maximum ease to the final user of the optimization. Has characteristics like learning, rapid generation, flexible decision and diversity. Provides advantages like automation, dynamic adjustment etc. |
| GSA | 2009 | Rashedi et al. | Based on the laws of gravity and mass interaction. Gravitational forces act as the way of transferring information between masses. Masses are attracted by heaviest mass which produces the optimum solution. |
| KH | 2012 | Gandomi and Alavi | Based on herding behavior of krill. Minimum distance from the food and from the highest density of herd is the objective function. Mutation and crossover are used for precise modeling. |
| SDS | 1989 | J.M. Bishop | Locate predefined data pattern or its best instantiation within a given search space. Uses population where each agent poses a hypothesis about the possible solution. |
| BA | 2010 | Xin-She Yang | Based on echolocation behavior of micro bats. Each bat flies randomly with a velocity, varying frequency and loudness. Selection of best continues with the intensification of local random walk. |
| COSMIC | 2004 | Gregory, Paton and Saunders | Rule based computing system. Have three levels: genome, cell and environment populated with cells. Study the evolution and adaptation of bacteria and predict the behavior of pathogenic bacteria. |
| RUBAM | 2006 | Paton, Vlachos, Wu and Saunders | Simplification of COSMIC. It grasps the features of ecology and exhibits adaptation and evolvable patterns. Generates multiple optimal solutions which are shared among different processors. |

**Figure 2 Comparative Performance Diagram Of Rubam And Related Algorithms.**

## Conclusion

This paper concludes that by establishing a rigorous comparative framework, it has provided a clear roadmap of the current bio-inspired optimisation landscape. The findings highlight a significant opportunity to transition these nature-centric techniques into software engineering for tasks like automated debugging and resource allocation. Ultimately, the study suggests that the future of the field lies in developing hybrid algorithms to overcome current limitations, ensuring these frameworks remain vital for solving increasingly complex, large-scale computational challenges in both academic and industrial sectors.

## References

1. Simon, D. (2008). Biogeography-based optimisation. IEEE Transactions on Evolutionary Computation, 12(6).
2. Junchen & Mahfouf, M. (2009). Artificial immune systems: A bio-inspired optimisation technique and its engineering application. IGI Global.
3. Ma, X., El-Keib, A. A., Smith, R. E., & Ma, H. (1995). A genetic algorithm-based approach to thermal unit commitment of electric power systems. Electric Power Systems Research, 34(1), 29–36.
4. Greco, F. (2008). Travelling salesman problem. In-Tech.
5. Neumann, F., & Witt, C. (2006). Runtime analysis of a simple ant colony optimisation algorithm. Technical Report CI-200/06, Department of Computer Science, University of Dortmund, Germany.
6. Zhou, Y. (2009). Runtime analysis of an ACO algorithm for TSP instances. IEEE Transactions on Evolutionary Computation, 13(5).
7. Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimisation for resource constrained project scheduling. IEEE Transactions on Evolutionary Computation, 6(4), 333–346.
8. Blum, C. (2005). Beam-ACO—hybridising ant colony optimisation with beam search: An application to open shop scheduling. Computers & Operations Research, 32(6), 1565–1591.
9. Gambardella, L. M., Taillard, E. D., & Agazzi, G. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne et al. (Eds.), New ideas in optimisation (pp. 63–76). McGraw Hill.
10. Reimann, M., Doerner, K., & Hartl, R. F. (2004). D-ants: Savings-based ants divide and conquer the vehicle routing problem. Computers & Operations Research, 31(4), 563–591.
11. Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1), 53–66.
12. Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. Future Generation Computer Systems, 16(8), 889–914.
13. Stützle, T., & Hoos, H. H. (1997). The MAX–MIN ant system and local search for the travelling salesman problem. In Proc. 1997 IEEE International Conference on Evolutionary Computation (pp. 309–314). IEEE Press.
14. Gambardella, L. M., & Dorigo, M. (2000). Ant Colony System hybridised with a new local search for the sequential ordering problem. INFORMS Journal on Computing, 12(3), 237–255.
15. Costa, D., & Hertz, A. (1997). Ants can colour graphs. Journal of the Operational Research Society, 48, 295–305.
16. Lessing, L., Dumitrescu, I., & Stützle, T. (2004). A comparison between ACO algorithms for the set covering problem. In M. Dorigo et al. (Eds.), Proc. ANTS'2004, LNCS (vol. 3172, pp. 1–12). Springer-Verlag.
17. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimisation. In Proc. IEEE International Conference on Neural Networks (pp. 1942–1948). IEEE Service Centre.
18. Bai, Q. (2010). Analysis of particle swarm optimisation algorithm. Computer and Information Science,

3(1).
19. Omran, M., Salman, A., & Engelbrecht, A. P. (2002). Image classification using particle swarm optimisation. In Proc. 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002) (pp. 370–374).
20. Hardin, T., Cui, X., Ragade, R. K., Graham, J. H., & Elmaghrabey, A. S. (2004). A modified particle swarm algorithm for robotic mapping of hazardous environments. In Proc. 2004 World Automation Congress.
21. Wang, L., Cui, Z., & Zeng, J. (2009). Particle swarm optimisation with group decision making. In Hybrid Intelligent Systems (HIS'09), 9th International Conference (vol. 1, pp. 388–393).
22. Zhang, T., & Lu, J. (2006). A PSO-based multivariable fuzzy decision-making predictive controller for a once-through 300-MW power plant. Cybernetics and Systems, 37(5).
23. Battiti, R., Brunato, M., & Mascia, F. (2008). Reactive search and intelligent optimisation. Operations Research/Computer Science Interfaces Series, vol. 45. Springer.
24. Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. Information Sciences, 179(13), 2232–2248.
25. Hassanzadeh, H. R. (2010). A multi-objective gravitational search algorithm. IEEE Second International Conference on CICSyN.
26. Roy, P. K., Mandal, B., & Bhattacharya, K. (2012). Gravitational search algorithm based optimal reactive power dispatch for voltage stability enhancement. Electric Power Components and Systems, 40(9), 956–976.
27. Duman, S., Guvenc, U., Sonmez, Y., & Yorukeren, N. (2012). Optimal power flow using gravitational search algorithm. Energy Conservation and Management, 59, 86–95.
28. Gandomi, A. H., & Alavi, A. H. (2012). Krill Herd algorithm: A new bio-inspired optimisation algorithm. Communications in Nonlinear Science and Numerical Simulation.
29. Omran, M. G. H., et al. (2011). Stochastic diffusion search for continuous global optimisation. International Conference on Swarm Intelligence, id-13.
30. Yang, X. S., et al. (2012). Bat algorithm for constrained optimisation tasks. Neural Computing and Applications. Springer-Verlag.
31. Yang, X. S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimisation. Engineering Computations, 29(5), 464–483.
32. Saunders, V. A., et al. (2009). Individual-based modelling of bacterial genetic elements. University of Liverpool, U.K. IGI Global, pp. 1930–1932.
33. Gheorghe, M. (2005). Molecular computation models: Unconditional approaches (pp. 204–218). Idea Group Inc. (IGI).
34. Karthiga, K., & Rajdeepa, B. (2026). Rhythm-guided convolutional neural network segmental transition entropy-modulated lesion discrimination cotton leaf disease identification. Journal of Theoretical and Applied Information Technology, 104(3).
35. Karthiga, K., & Rajdeepa, B. (2025, February). Performance evaluation of Faster R-CNN models with ResNet-50 and ResNet-101 backbones for cotton plant image processing. In National Symposium on Recent Advances in Electronic, Optical and Magnetic Materials (pp. 263–279). Springer Nature Switzerland.