STM JOURNALS

Journal of
Operating Systems
Development & Trends

ISSN: 2454-9355
Volume 12, Issue 2, 2025
May–August
DOI (Journal): 10.37591/JoOSDT

https://journals.stmjournals.com/joosdt

Review JoOSDT

Systematic Review of Application of Nature-Inspired Algorithms for Resource Optimization in Multi-Programmed Operating Systems

Yamuna Mundru[1], Atti Manga Devi[2], Manas Kumar Yogi[3,*]

Abstract

Multi-programmed operating systems are increasingly confronted with complex challenges in efficiently managing system resources, primarily due to the need to handle numerous concurrent processes with diverse and often conflicting resource demands. As these systems evolve, ensuring optimal performance across various dimensions, such as CPU scheduling, memory allocation, and load balancing, has become crucial. In this context, nature-inspired algorithms have emerged as promising solutions for enhancing resource optimization. These algorithms, which draw inspiration from natural processes such as evolution, swarm behaviour, and biological systems, offer flexible and adaptive mechanisms for tackling optimization problems in dynamic computing environments. This systematic review delves into the characteristics and functionalities of various nature-inspired algorithms, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), assessing their efficiency in resource management within multi-programmed operating systems. The review highlights that these algorithms can outperform traditional approaches by offering better adaptability and higher optimization accuracy. Notably, significant improvements are observed in areas like CPU task scheduling, effective memory usage, and balancing system load. However, the implementation of these algorithms is not without challenges. Issues such as high computational overhead, slower convergence rates, and difficulties in adapting to real-time system changes continue to pose limitations. Future research should focus on hybrid approaches and real-time adaptability to enhance system performance further.

***Author for Correspondence**
Manas Kumar Yogi
E-mail: manas.yogi@gmail.com

[1]Assistant Professor, Department of Computer Science and Engineering - Artificial Intelligence and Machine Learning (CSE-AI&ML), Pragati Engineering College (A), Surampalem, Andhra Pradesh, India
[2]Assistant Professor, Information Technology Department, Pragati Engineering College (A), Surampalem, Andhra Pradesh, India
[3]Assistant Professor, Department of Computer Science and Engineering, Pragati Engineering College (A), Surampalem, Andhra Pradesh, India

Received Date: May 27, 2025
Accepted Date: May 30, 2025
Published Date: July 23, 2025

Citation: Yamuna Mundru, Atti Manga Devi, Manas Kumar Yogi. Systematic Review of Application of Nature-Inspired Algorithms for Resource Optimization in Multi-Programmed Operating Systems. Journal of Operating Systems Development & Trends. 2025; 12(2): 8–14p.

Keywords: Nature-inspired algorithms, multi-programmed operating systems, resource optimization, CPU scheduling, memory management, load balancing

INTRODUCTION

Multi-programmed operating systems represent the foundation of modern computing environments, enabling multiple programs to execute concurrently while efficiently managing system resources. The primary challenge in such systems lies in optimizing resource allocation to maximize system throughput, minimize response time, and ensure fair resource distribution among competing processes [1]. Traditional optimization techniques often fall short when dealing with the dynamic and complex nature of resource allocation problems in multi-programmed environments.

Nature-inspired algorithms have gained significant attention in recent years as effective solutions for complex optimization problems. These algorithms draw inspiration from natural phenomena such as evolution, swarm behaviour, and biological processes to develop computational techniques that can solve intricate optimization challenges [2]. The application of these algorithms to operating system resource optimization has shown promising results, offering adaptive and robust solutions to traditional scheduling and resource management problems.

The significance of this research area stems from the ever-increasing demands on computing systems, driven by cloud computing, distributed systems, and resource-intensive applications. As system complexity grows, the need for intelligent and adaptive resource optimization becomes critical for maintaining system performance and user satisfaction [3]. This systematic review aims to provide a comprehensive analysis of the current state of research in applying nature-inspired algorithms to multi-programmed operating system resource optimization.

RESEARCH OBJECTIVES

The primary objectives of this systematic review are to:

1. Identify and analyse the most applied nature-inspired algorithms in multi-programmed operating system resource optimization.
2. Evaluate the effectiveness of these algorithms in addressing specific resource optimization challenges.
3. Compare the performance of nature-inspired algorithms with traditional optimization techniques.
4. Identify current research gaps and future research directions.
5. Provide recommendations for practitioners and researchers in the field.

Scope and Limitations

This review focuses specifically on nature-inspired algorithms applied to resource optimization in multi-programmed operating systems. The scope includes CPU scheduling, memory management, load balancing, and process synchronization. The review is limited to peer-reviewed research published between 2015 and 2024, ensuring relevance to current technological contexts while maintaining a manageable scope for comprehensive analysis.

METHODOLOGY

Search Strategy

A systematic literature search was conducted across multiple academic databases, including IEEE Xplore, ACM Digital Library, ScienceDirect, and Google Scholar. The search strategy employed a combination of keywords related to nature-inspired algorithms and operating system resource optimization. The primary search terms included: “nature-inspired algorithms”, “bio-inspired algorithms”, “evolutionary algorithms”, “swarm intelligence”, “multi-programmed operating systems”, “resource optimization”, “CPU scheduling”, and “memory management”.

Inclusion and Exclusion Criteria

Inclusion Criteria

- Peer-reviewed research articles published between 2015 and 2024.
- Studies focusing on nature-inspired algorithms for operating system resource optimization.
- Research conducted in multi-programmed or multi-tasking environments.
- Articles written in English.

Exclusion Criteria

- Non-peer-reviewed publications.
- Studies focusing solely on single-threaded or single-process systems.
- Research not directly related to operating system resource optimization.
- Duplicate publications or conference abstracts without full implementations.

Study Selection and Quality Assessment

The initial search yielded 247 potential articles. After applying inclusion and exclusion criteria, 19 studies were selected for comprehensive analysis. Each selected study underwent quality assessment based on research methodology, experimental design, result validation, and contribution to the field.

LITERATURE REVIEW

Theoretical Foundation of Nature-Inspired Algorithms

Nature-inspired algorithms represent a class of optimization techniques that mimic natural processes to solve complex computational problems. These algorithms can be broadly categorized into evolutionary algorithms, swarm intelligence algorithms, and physics-based algorithms [4]. The fundamental principle underlying these algorithms is the simulation of natural selection, collective behaviour, or physical phenomena to explore solution spaces efficiently.

Evolutionary algorithms, including genetic algorithms (GA) and differential evolution (DE), simulate the process of natural selection and genetic inheritance. These algorithms maintain a population of candidate solutions and iteratively improve them through selection, crossover, and mutation operations [5]. Swarm intelligence algorithms, such as particle swarm optimization (PSO) and ant colony optimization (ACO), model the collective behaviour of social organisms to find optimal solutions through cooperation and information sharing [6].

Application Domains in Operating System Resource Optimization

CPU Scheduling Optimization

CPU scheduling represents one of the most critical aspects of operating system resource management. Traditional scheduling algorithms such as First-Come-First-Served (FCFS), Shortest Job First (SJF), and Round Robin (RR) often fail to adapt to dynamic system conditions and varying process characteristics [7]. Nature-inspired algorithms have been successfully applied to develop adaptive scheduling policies that optimize multiple objectives simultaneously.

The effectiveness of genetic algorithms in multi-objective CPU scheduling has been demonstrated, achieving a 23% improvement in average turnaround time compared to traditional algorithms [8]. The study employed chromosome representation of scheduling sequences and fitness functions based on turnaround time, waiting time, and response time metrics.

Particle swarm optimization has also shown significant promise in CPU scheduling applications. A PSO-based scheduling algorithm has been developed that dynamically adjusts scheduling parameters based on system load and process characteristics, resulting in improved system throughput and reduced response time [9].

Memory Management Optimization

Memory management in multi-programmed systems involves complex decisions regarding memory allocation, page replacement, and memory compaction. Nature-inspired algorithms have been applied to optimize these processes, particularly in virtual memory systems where page replacement policies significantly impact system performance [10].

Ant colony optimization has been successfully employed for memory allocation problems, with studies showing improved memory utilization and reduced page fault rates. The algorithm models memory allocation as a pathfinding problem, where ants represent allocation requests and pheromone trails indicate optimal allocation patterns [11].

Load Balancing and Process Distribution

Load balancing in distributed and multi-core systems requires intelligent distribution of processes across available resources. Nature-inspired algorithms have proven effective in this domain by considering multiple factors such as processor utilization, communication overhead, and process dependencies [12].

Table 1. Comparison of nature-inspired algorithms in resource optimization.

Algorithm	Primary application	Key advantages	Limitations	Performance improvement
Genetic algorithm	CPU scheduling, load balancing	Multi-objective optimization, adaptability	High computational overhead	15–25% improvement in turnaround time
Particle swarm optimization	CPU scheduling, memory allocation	Fast convergence, simple implementation	Premature convergence risk	12–20% improvement in response time
Ant colony optimization	Memory management, path optimization	Distributed computation, pheromone memory	Slow convergence	10–18% improvement in memory utilization
Simulated annealing	Resource allocation, Scheduling	Global optimization capability	Parameter sensitivity	8–15% improvement in system throughput
Artificial bee colony	Load balancing, task scheduling	Self-organization, robustness	Limited exploration capability	10–22% improvement in load distribution

Genetic algorithms have been extensively used for load balancing problems, with chromosome representations encoding process-to-processor mappings. Fitness functions typically consider load distribution metrics, communication costs, and execution time estimates [13].

Comparative Analysis of Nature-Inspired Algorithms

Table 1 depicts the various factors based on which the comparison of the popular nature inspired algorithms is used for resource optimization.

Performance Evaluation Metrics

The evaluation of nature-inspired algorithms in operating system resource optimization typically employs multiple performance metrics to assess effectiveness across different dimensions [14]. Common metrics include:

1. *Temporal metrics:* Average turnaround time, waiting time, response time, and execution time.
2. *Utilization metrics:* CPU utilization, memory utilization, and resource efficiency.
3. *Throughput metrics:* System throughput, process completion rate, and job processing capacity.
4. *Quality metrics:* Fairness index, load balancing ratio, and resource allocation equality.

RESULTS AND ANALYSIS

Algorithm Effectiveness Analysis

The analysis of the 19 selected studies reveals significant variations in algorithm effectiveness across different application domains. Genetic algorithms demonstrate superior performance in multi-objective optimization scenarios, particularly in CPU scheduling applications where multiple conflicting objectives must be balanced simultaneously [15].

Particle swarm optimization exhibits excellent performance in dynamic environments where rapid adaptation to changing conditions is required. The algorithm's ability to maintain solution diversity while converging quickly makes it particularly suitable for real-time scheduling applications [16].

Implementation Challenges and Solutions

The implementation of nature-inspired algorithms in operating system environments presents several challenges that must be addressed for practical deployment:

1. *Computational overhead:* Nature-inspired algorithms typically require additional computational resources for population maintenance, fitness evaluation, and iteration processing. Studies indicate that the overhead ranges from 5–15% of total system resources, depending on algorithm complexity and population size [17].
2. *Parameter tuning:* The effectiveness of nature-inspired algorithms heavily depends on parameter settings such as population size, mutation rate, crossover probability, and convergence criteria. Adaptive parameter control mechanisms have been developed to address this challenge, with studies showing improved performance through dynamic parameter adjustment [18].

Table 2. Hybrid algorithm performance analysis.

Hybrid approach	Component algorithms	Application domain	Performance gain	Complexity
GA-PSO	Genetic algorithm + PSO	CPU scheduling	28% improvement	Medium
ACO-SA	Ant colony + simulated annealing	Memory management	22% improvement	High
PSO-ABC	PSO + artificial bee colony	Load balancing	25% improvement	Medium
GA-Heuristic	Genetic algorithm + traditional	Multi-resource	30% improvement	Low

3. *Real-time constraints:* Operating systems must make scheduling and resource allocation decisions within strict time constraints. Hybrid approaches combining nature-inspired algorithms with traditional heuristics have been proposed to meet real-time requirements while maintaining optimization benefits.

Hybrid Algorithm Approaches

Recent research trends indicate a shift toward hybrid algorithms that combine multiple nature-inspired techniques or integrate them with traditional optimization methods [19]. These hybrid approaches aim to leverage the strengths of different algorithms while mitigating their individual limitations as shown in Table 2.

DISCUSSION

Key Findings and Implications

The systematic review reveals several key findings regarding the application of nature-inspired algorithms in multi-programmed operating system resource optimization:

1. *Algorithm suitability:* Different nature-inspired algorithms demonstrate varying degrees of effectiveness depending on the specific optimization problem. Genetic algorithms excel in multi-objective scenarios, while PSO performs well in dynamic environments.
2. *Performance improvements:* Nature-inspired algorithms consistently outperform traditional optimization techniques, with improvements ranging from 8 to 30% across different performance metrics.
3. *Implementation feasibility:* While computational overhead remains a concern, advances in algorithm efficiency and hardware capabilities make practical implementation increasingly viable.
4. *Hybrid approaches:* Combining multiple algorithms or integrating nature-inspired techniques with traditional methods yields superior results compared to single-algorithm approaches.

Research Gaps and Future Directions

Despite significant progress in this field, several research gaps remain:

1. *Scalability analysis:* Limited research exists on the scalability of nature-inspired algorithms in large-scale multi-programmed systems with thousands of concurrent processes.
2. *Real-time performance:* More research is needed to develop nature-inspired algorithms that can meet strict real-time constraints without compromising optimization quality.
3. *Dynamic adaptation:* Current algorithms often struggle with rapidly changing system conditions, indicating a need for more adaptive and responsive optimization techniques.
4. *Energy efficiency:* With growing concerns about energy consumption, future research should focus on nature-inspired algorithms that optimize both performance and energy efficiency.

Practical Implementation Considerations

For practitioners considering the implementation of nature-inspired algorithms in operating system resource optimization, several factors must be considered:

1. *System requirements:* The computational overhead of nature-inspired algorithms must be balanced against the expected performance improvements.
2. *Algorithm selection:* The choice of algorithm should be based on specific optimization objectives and system characteristics.

3. *Parameter configuration:* Proper parameter tuning is crucial for algorithm effectiveness, and adaptive parameter control mechanisms should be employed when possible.
4. *Integration strategy:* Hybrid approaches often provide better results than single-algorithm implementations, but they require careful design and integration.

CONCLUSION

The analysis of 19 research studies reveals consistent performance improvements across various optimization metrics, showing particular promise in different application domains. The review identifies several key success factors for implementing nature-inspired algorithms in operating system environments, including appropriate algorithm selection, effective parameter tuning, and consideration of hybrid approaches. While challenges such as computational overhead and real-time constraints remain, on-going research and technological advances continue to address the inherent constraints.

In future work can be initiated which can handle the increasing complexity of modern computing environments while maintaining real-time performance requirements. The integration of energy efficiency considerations and the development of algorithms specifically designed for cloud and distributed computing environments represent particularly promising research directions.

The findings of this review suggest that nature-inspired algorithms have moved beyond theoretical interest to practical implementation, offering operating system designers and developers powerful tools for addressing resource optimization challenges in multi-programmed environments. As computing systems continue to evolve in complexity and scale, these algorithms will significantly improve the efficiency and effective resource utilization.

REFERENCES

1. Tanenbaum AS, Bos H. Modern Operating Systems. London: Pearson Education, Incorporated; 2015.
2. Yang XS. Nature-Inspired Optimization Algorithms. Cambridge (MA): Academic Press; 2020.
3. Kapur R. Review of nature inspired algorithms in cloud computing. In IEEE International Conference on Computing, Communication & Automation. 2015 May 15; 589–594.
4. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput Intell Mag. 2007 Mar 19; 1(4): 28–39.
5. Goldberg DE. Genetic and evolutionary algorithms come of age. Commun ACM. 1994 Mar 1; 37(3): 113–9.
6. Kennedy J, Eberhart R. Particle swarm optimization. In IEEE Proceedings of ICNN'95: International Conference on Neural Networks. 1995 Nov 27; 4: 1942–1948.
7. Silberschatz A, Galvin PB, Gagne G. Operating System Concepts, Windows XP Update. Hoboken (NJ): John Wiley & Sons; 2006.
8. Mukundan J, Martinez JF. MORSE: Multi-objective reconfigurable self-optimizing memory scheduler. In IEEE International Symposium on High-Performance Comp Architecture. 2012 Feb 25; 1–12.
9. Pacini E, Mateos C, García Garino C. Dynamic scheduling based on particle swarm optimization for cloud-based scientific experiments. CLEI Electron J. 2014 Apr; 17(1): 1–14.
10. Ullah I, Khitab Z, Khan MN, Hussain S. An efficient energy management in office using bio-inspired energy optimization algorithms. Processes. 2019 Mar 7; 7(3): 142.
11. Liu XF, Zhan ZH, Deng JD, Li Y, Gu T, Zhang J. An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Trans Evol Comput. 2016 Nov 21; 22(1): 113–28.
12. Marella ST, Gunasekhar T. An empirical survey on load balancing: a nature-inspired approach. In: Righi RR, editor. Scheduling Problems: New Applications and Trends. London: IntechOpen; 2020. doi: 10.5772/intechopen.87002.
13. Pillai AS, Singh K, Saravanan V, Anpalagan A, Woungang I, Barolli L. A genetic algorithm-based method for optimizing the energy consumption and performance of multiprocessor systems. Soft Comput. 2018 May; 22: 3271–85.

14. Wedde HF, Farooq M. A performance evaluation framework for nature inspired routing algorithms. In: Workshops on Applications of Evolutionary Computation. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005 Mar 30; 136–146.
15. Rai D, Tyagi K. Bio-inspired optimization techniques: a critical comparative study. ACM SIGSOFT Softw Eng Notes. 2013 Jul 12; 38(4): 1–7.
16. Kumar R. Simulative Analysis and Performance Evaluation for Data Variety Aware Power Optimization Technique Using Big Data. Wirel Pers Commun. 2023 Dec; 133(3): 1987–2002.
17. Kaul S, Kumar Y, Ghosh U, Alnumay W. Nature-inspired optimization algorithms for different computing systems: novel perspective and systematic review. Multimed Tools Appl. 2022 Aug; 81(19): 26779–801.
18. Domanal SG, Guddeti RM, Buyya R. A hybrid bio-inspired algorithm for scheduling and resource management in cloud environment. IEEE Trans Serv Comput. 2017 Mar 8; 13(1): 3–15.
19. Mikhalevich VS, Sergienko IV, Zadiraka VK, Babich MD. Optimization of computations. Cybern Syst Anal. 1994;30:213–35. doi: 10.1007/BF02366426.

