Faculty of Natural and Applied Sciences Journal of Computing and Applications
Print ISSN: 3026-8133
www.fnasjournals.com
Volume 3; Issue 1; March 2026; Page No. 117-129.
DOI: https://doi.org/10.63561/jca.v3i1.1236

Development of Dynamic Task Scheduling in Multi-Core Systems Using Teaching–Learning-Based Optimization

Igiri, C. G., *Ahua, C G., Amos, M., & Ezebalike, O. O.
Department of Computer Science, Rivers State University, Nkpolu, Port Harcourt, Rivers State, Nigeria

***Corresponding author email**: chidiahua@gmail.com

Abstract
Efficient task scheduling in multi-core systems remains a critical challenge due to the increasing complexity of applications, dynamic workloads, and the need to optimize multiple conflicting objectives such as execution time, energy consumption, and load balancing. Traditional scheduling algorithms often struggle to adapt to dynamic environments and fail to achieve optimal performance across diverse system conditions. This paper proposes a dynamic task scheduling approach based on the Teaching-Learning-Based Optimization (TLBO) algorithm to address these limitations. The proposed method models task scheduling as a multi-objective optimization problem, where tasks are dynamically allocated to multiple processing cores while considering task dependencies and system constraints. The TLBO algorithm, inspired by the teaching–learning process in a classroom, is adapted to efficiently explore and exploit the solution space without requiring algorithm-specific parameters. The methodology incorporates both teacher and learner phases to iteratively improve scheduling decisions and enhance system performance. To evaluate the effectiveness of the proposed approach, extensive simulations were conducted using standard benchmarking scenarios and compared against conventional optimization techniques such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Performance metrics, including makespan, throughput, resource utilization, and energy efficiency, were analyzed. The results demonstrate that the TLBO-based scheduler significantly improves overall system performance, achieving reduced execution time and better load distribution across cores. The findings suggest that TLBO provides a robust and scalable solution for dynamic task scheduling in multi-core environments. This research contributes to the advancement of intelligent scheduling techniques and offers a foundation for future work in adaptive and energy-aware computing systems.

Keywords: Multi-core systems, Task scheduling, Teaching-Learning-Based Optimization (TLBO), Multi-objective optimization, Makespan, Load balancing

Introduction

The rapid advancement of computing technologies has led to the widespread adoption of multi-core systems as a foundational architecture in modern computing environments such as cloud computing, embedded systems, and edge computing platforms. Multi-core processors integrate multiple processing units on a single chip, enabling the parallel execution of tasks and significantly enhancing system performance, scalability, and energy efficiency. This transition from single-core to multi-core architectures has been largely driven by the growing demand for high-performance computing and the inherent limitations of sequential processing in handling complex and data-intensive applications (Kumar & Singh, 2021; Zhang et al., 2022). Within multi-core systems, task scheduling plays a crucial role in determining how efficiently computational tasks are assigned to available processing cores. Effective scheduling directly influences system performance by optimizing resource utilization, reducing overall execution time (makespan), and improving throughput. However, task scheduling in multi-core environments is inherently complex and has been classified as an NP-hard optimization problem, particularly when dealing with dynamic workloads and heterogeneous processing architectures (Wang et al., 2021; Patel & Mehta, 2023). This complexity is further intensified in real-time systems where multiple constraints, such as task deadlines, energy efficiency requirements, and system reliability, must be simultaneously satisfied. The emergence of modern computing paradigms such as edge computing and the Internet of Things (IoT) has further amplified the need for efficient and adaptive task scheduling strategies. These environments rely heavily on multi-core processors to manage large-scale data processing and support real-time decision-making. However, persistent challenges such as load imbalance, resource contention, task dependencies, and unpredictable task arrivals continue to limit optimal system performance (Rahman et al., 2022; Li & Zhao, 2023). As a result, there is an increasing need for intelligent scheduling mechanisms capable of dynamically adapting to changing workload conditions while maintaining system efficiency.

In response to these challenges, researchers have increasingly explored metaheuristic optimization techniques due to their ability to provide near-optimal solutions for complex and NP-hard problems within a reasonable computational time. Algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) have been widely applied to task scheduling problems in both multi-core and distributed computing systems. These methods have demonstrated effectiveness in improving scheduling performance, particularly in dynamic and large-scale environments (Abdullahi et al., 2020; Chen et al., 2022). Among these approaches, Teaching-Learning-Based Optimization (TLBO) has emerged as a promising alternative due to its simplicity, efficiency, and parameter-free structure. TLBO is a population-based optimization algorithm inspired by the teaching and learning process in a classroom setting, where individuals improve their knowledge through interactions with a teacher and peers. Unlike many other metaheuristic algorithms, TLBO does not require algorithm-specific parameter tuning, which reduces computational complexity and enhances ease of implementation. Recent studies have shown that TLBO is highly effective in solving complex optimization problems, particularly in multi-objective and resource-constrained environments (Rao et al., 2020; Singh & Verma, 2024).

Despite these advancements, existing task scheduling approaches still exhibit significant limitations in addressing the dynamic nature of modern computing environments. Many current techniques struggle with efficiently handling dynamic task arrivals, inter-task dependencies, and the simultaneous optimization of multiple conflicting objectives such as execution time, energy consumption, and load balancing. Traditional heuristic and static scheduling methods often fail to adapt to runtime variations in workload, resulting in poor resource utilization, increased makespan, and system inefficiency (Liu et al., 2025). These challenges are further complicated in heterogeneous multi-core systems, where balancing performance and energy efficiency becomes critical. Consequently, there is a clear need for more adaptive and intelligent scheduling frameworks capable of responding effectively to dynamic system conditions.
In light of these challenges, this study focuses on addressing the problem of dynamic task scheduling in multi-core systems using the Teaching-Learning-Based Optimization (TLBO) algorithm. The aim of the study is to develop and evaluate a dynamic scheduling framework that improves system performance through enhanced resource utilization, reduced execution time, and balanced workload distribution. Specifically, the study seeks to analyze the limitations of existing scheduling techniques in multi-core systems, model dynamic task scheduling as a multi-objective optimization problem involving makespan, load balancing, and energy consumption, adapt the TLBO algorithm for dynamic scheduling environments, evaluate its performance against existing metaheuristic approaches, and demonstrate its effectiveness through simulation under varying workload scenarios.

Rao et al. (2020) explored the use of Teaching–Learning-Based Optimization (TLBO) in solving combinatorial optimization problems, emphasizing that many conventional optimization techniques are constrained by complex parameter tuning requirements. The researchers introduced a TLBO framework inspired by classroom teaching and learning interactions, eliminating the need for algorithm-specific control parameters. Experimental evaluations conducted on benchmark optimization functions revealed that the proposed TLBO approach produced results comparable to Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), while demonstrating greater simplicity and robustness in optimization performance. Kim and Lee (2021) investigated the shortcomings of heuristic scheduling methods in heterogeneous multi-core systems, observing that static heuristic techniques perform poorly in dynamic execution environments. To address this limitation, they designed an adaptive list scheduling model that considered processor heterogeneity alongside fluctuating workload conditions. Performance analysis of simulated multi-core architectures demonstrated a 17% decrease in makespan compared to traditional heuristic schedulers, thereby confirming the effectiveness of adaptive scheduling strategies.

Chen et al. (2021) investigated the scalability limitations of conventional list scheduling algorithms when applied to large task graphs. They proposed a clustering-based scheduling technique that grouped interdependent tasks to

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

minimize scheduling complexity. Using big data processing workloads as evaluation benchmarks, the study reported enhanced throughput performance with only a slight reduction in optimality relative to exhaustive scheduling approaches, thereby illustrating the trade-off between scalability and optimization accuracy. Nguyen et al (2022) studied adaptive task scheduling mechanisms for cloud-driven multi-core servers, noting that static task allocation methods are often ineffective under fluctuating system workloads. The authors developed a real-time adaptive scheduler incorporating dynamic priority weighting mechanisms. Simulation outcomes revealed improvements of 22% in resource utilization and 14% in response time efficiency under varying workload conditions, indicating the effectiveness of adaptive scheduling in cloud environments.

Gupta and Reddy (2022) investigated the issue of excessive energy consumption in Internet of Things (IoT) edge gateways, where traditional schedulers often lead to inefficient utilization of multi-core processors. They proposed an energy-aware hybrid scheduler that integrated TLBO with dynamic voltage scaling techniques. Experimental testing on IoT workloads demonstrated a 23% reduction in total energy consumption with minimal effects on makespan, validating the suitability of hybrid metaheuristic scheduling for energy-sensitive environments. Rahman and Caldwell (2023) examined the inability of conventional heuristic schedulers to efficiently manage cloud burst scenarios characterized by sudden workload spikes. To overcome this challenge, they introduced a reinforcement learning-based dynamic scheduler capable of learning optimal task placement strategies over time. Experimental analysis using cloud trace datasets showed a 19% decline in Service Level Agreement (SLA) violations alongside improved scheduling stability under bursty task arrivals.

Patel and Mehta (2023) criticized single-objective scheduling techniques for ignoring the balance between execution performance and energy efficiency in heterogeneous computing clusters. They developed a multi-objective Genetic Algorithm capable of optimizing both makespan and energy consumption simultaneously. Findings from the study indicated considerable reductions in energy usage while sustaining competitive execution performance across heterogeneous processing environments. Lee et al. (2024) focused on throughput bottlenecks in high-performance heterogeneous clusters and proposed an enhanced Particle Swarm Optimization (PSO) algorithm featuring adaptive inertia weighting to improve global search performance. Experimental simulations conducted on high-performance computing benchmarks revealed a 13% increase in throughput compared with standard PSO implementations, demonstrating the effectiveness of adaptive swarm intelligence techniques.

Hegde et al. (2024) investigated constrained and multi-objective task scheduling challenges in grid computing environments, emphasizing that many existing scheduling algorithms fail to simultaneously satisfy multiple optimization criteria. The researchers implemented a TLBO-based scheduler equipped with constraint-handling capabilities. Performance evaluations showed superior trade-offs between makespan and load balancing when compared with conventional GA and Ant Colony Optimization (ACO) schedulers. Clark and Xu (2024) identified limited exploration capability as a major weakness of traditional Genetic Algorithms when scheduling complex task graphs containing dependency relationships. To improve exploration efficiency, they proposed a genetic programming-based scheduler with advanced crossover semantics. Experimental results on dependency-intensive workloads demonstrated more stable execution ordering and lower scheduling latency.

Singh and Verma (2024) explored the computational overhead associated with multi-objective optimization in multi-core scheduling systems. The authors modified the Non-dominated Sorting Genetic Algorithm II (NSGA-II) by integrating problem-specific heuristics to manage trade-offs between energy consumption and execution time. Benchmark evaluations showed consistent improvements in Pareto front quality across heterogeneous computing environments. Zhang et al. (2023) analyzed the problem of load imbalance in real-time heterogeneous computing systems, noting that static optimization methods cannot effectively respond to runtime variations in system state. They proposed an Ant Colony Optimization scheduler with dynamically adjusted pheromone mechanisms. Simulation findings demonstrated improved workload distribution and significant reductions in processor idle time.

Li and Zhao (2023) criticized static scheduling approaches for their inability to handle unpredictable task arrivals in real-time systems. They introduced an enhanced PSO-driven dynamic scheduler designed to adapt to runtime scheduling changes. Evaluation using real-time task datasets revealed improved responsiveness, adaptability, and reduced task execution delays. Aminu et al. (2025) conducted a systematic review of metaheuristic-based scheduling approaches in edge computing environments, highlighting that many existing techniques lack sufficient dynamic

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

adaptability and balanced multi-objective optimization. The study developed a comprehensive taxonomy of scheduling approaches and identified critical gaps relating to parameter-free optimization methods and scalable adaptive scheduling frameworks.

Liu et al. (2025) examined inefficiencies in energy management within heterogeneous multi-core processors and proposed a TLBO-based energy-aware scheduler integrating load balancing with energy constraints. Experimental assessment using edge computing benchmarks revealed energy savings of up to 27% while maintaining acceptable makespan performance, demonstrating the effectiveness of integrated optimization strategies. Tran and Hoang (2022) investigated the problem of dependency degradation in task graph scheduling, observing that many metaheuristic algorithms inadequately address precedence constraints among dependent tasks. They developed a dependency-aware Ant Colony Optimization scheduler specifically designed for directed acyclic graph (DAG) task sets. Experimental outcomes indicated improved compliance with dependency constraints and fewer scheduling conflicts.

Despite substantial progress in task scheduling techniques for multi-core systems, several critical gaps remain unresolved in the literature. Existing studies have demonstrated the effectiveness of metaheuristic algorithms such as TLBO, GA, PSO, ACO, and NSGA-II in optimizing scheduling objectives, including makespan, energy consumption, throughput, and load balancing. However, many of these approaches are predominantly designed for static or semi-dynamic environments and therefore exhibit limited adaptability under real-time workload fluctuations and unpredictable task arrivals. Although Rao et al. (2020) established the capability of TLBO in static optimization scenarios, its application to dynamic task scheduling in heterogeneous multi-core environments remains insufficiently explored. Furthermore, most multi-objective scheduling models focus primarily on balancing execution time and energy efficiency while neglecting critical workflow characteristics such as task dependencies, precedence constraints, and runtime adaptability. Studies by Tran and Hoang (2022) and Huang and Wang (2021) addressed dependency-aware scheduling; however, their approaches did not integrate parameterless optimization strategies such as TLBO. In addition, several energy-aware schedulers assume predefined or stable workloads, limiting their effectiveness in practical systems where task arrivals and resource demands change dynamically. Hybrid metaheuristic approaches proposed in previous studies often introduce additional computational complexity and parameter tuning challenges, thereby reducing scalability and implementation efficiency in large-scale multi-core systems. Moreover, existing adaptive scheduling frameworks rarely combine dynamic optimization, dependency awareness, load balancing, and energy efficiency into a unified scheduling model.

Methodology

The proposed study adopts a simulation-based quantitative methodology to design and evaluate a dynamic task scheduling framework for heterogeneous multi-core systems using Teaching–Learning-Based Optimization (TLBO). The system is modeled as a set of heterogeneous processing cores $C = \{c_1, c_2..., c_m\}$ interconnected through a shared memory hierarchy and communication infrastructure. Each core differs in computational capability, execution speed, and energy consumption characteristics, reflecting realistic environments such as cloud and edge computing systems. Inter-core communication introduces latency due to shared caches and interconnects, which significantly affects scheduling performance in terms of makespan, energy usage, and load balancing (Rao, 2020). The task model consists of a set $T = \{t_1, t_2, ..., t_n\}$, where each task t_i is defined by execution time, deadline D_i, priority, energy requirement E_i , and precedence constraints represented using a Directed Acyclic Graph (DAG) $G=(V, E)$. This ensures that tasks execute only after all dependencies are satisfied, enabling dependency-aware scheduling across heterogeneous cores. The scheduling problem is formulated as a dynamic multi-objective optimization problem aimed at minimizing makespan, reducing total energy consumption, improving load balance, and ensuring deadline adherence under system constraints such as core capacity, communication delay, and runtime workload variation.

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

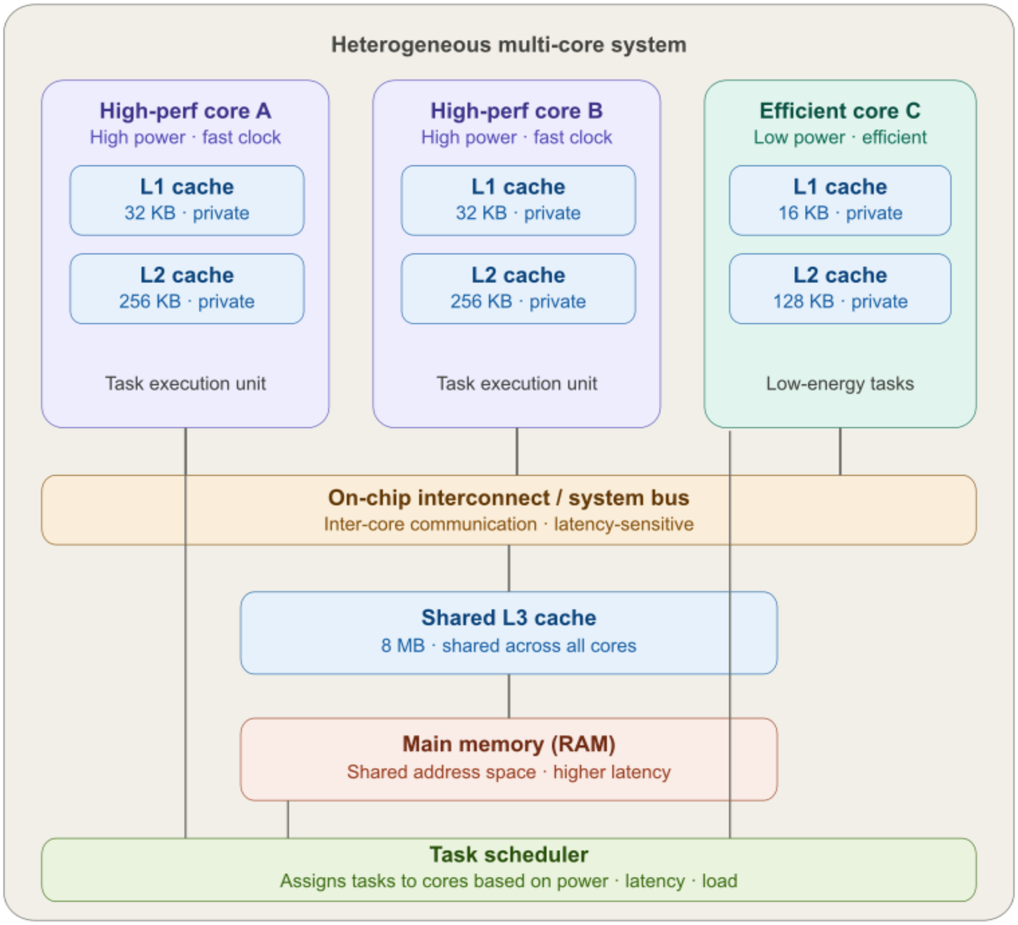

Figure 1: Proposed Heterogeneous Multi-Core System Architecture

The scheduling framework is implemented using the Teaching–Learning-Based Optimization (TLBO) algorithm, where each candidate solution represents a mapping of tasks T to processing cores C. The optimization operates through teacher and learner phases, where the teacher phase guides solutions toward better performance, while the learner phase improves solutions through interaction among candidates. A runtime monitoring module continuously tracks processor utilization, energy consumption, task waiting time, and deadline violations, enabling adaptive rescheduling when system imbalance or workload changes occur. This closed-loop mechanism ensures dynamic, dependency-aware, and energy-efficient scheduling under varying conditions.

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

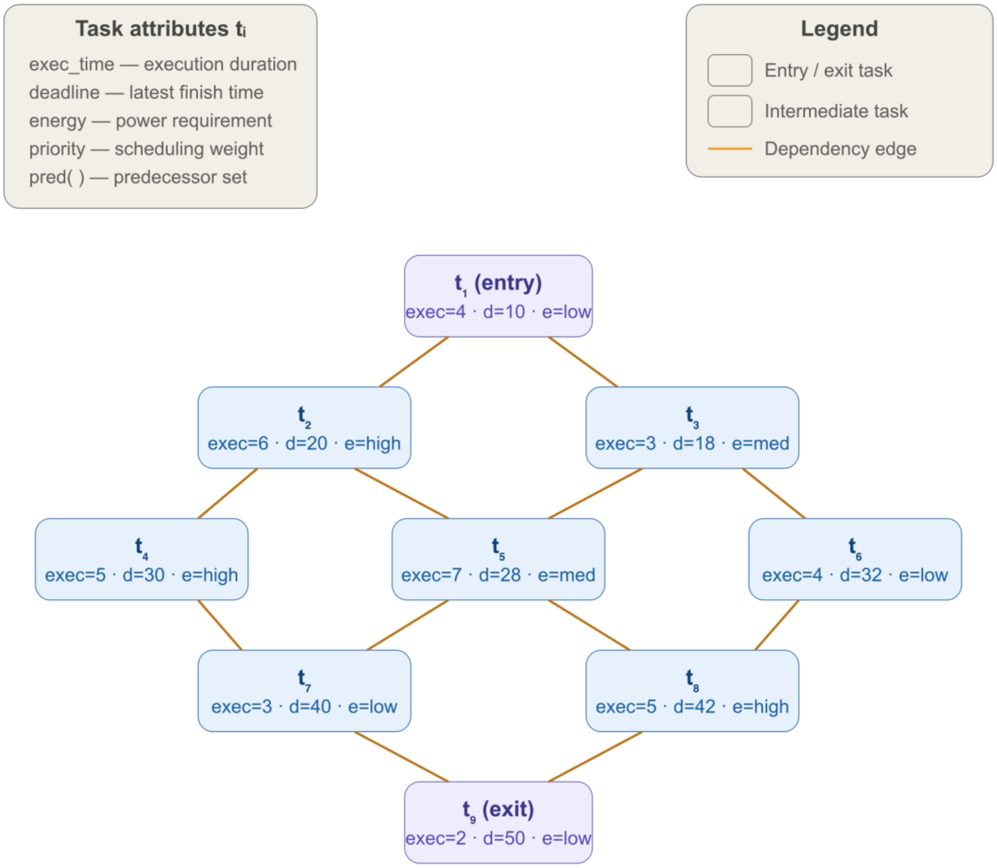

Figure 2: Proposed TLBO-Based Dynamic Scheduling Model

For evaluation, Python is used as the primary development environment with NumPy, Pandas, Matplotlib, and NetworkX supporting computation, data handling, visualization, and DAG modeling, respectively. Simulation platforms such as CloudSim, EdgeCloudSim, or MATLAB are used to model heterogeneous multi-core environments, including communication delay and energy behavior. Performance is evaluated using metrics such as makespan, energy consumption, load balancing, processor utilization, and deadline adherence to assess the effectiveness of the proposed framework.

TLBO-Based Dynamic Task Scheduling Model

The Teaching–Learning-Based Optimization (TLBO) algorithm is a population-based metaheuristic inspired by classroom teaching and learning processes, consisting of a Teacher Phase where the best solution guides others toward improvement and a Learner Phase where solutions interact to enhance performance (Rao, 2020). It is parameter-free, making it simpler than many other metaheuristics. In this study, TLBO is adapted for dynamic task scheduling in heterogeneous multi-core systems by continuously updating candidate schedules as new tasks arrive, evaluating each schedule based on makespan, energy consumption, load balancing, and deadline adherence, while ensuring task dependencies are preserved using Directed Acyclic Graphs (DAGs) and incorporating communication delays into fitness evaluation. Each solution represents a task-to-core mapping, and scheduling is iteratively improved through teacher and learner phases until termination, after which the best schedule is selected. The computational complexity is approximately $O(I \times P \times n^2)$, where I is the number of iterations, P is the population size, and n is the number of tasks, but TLBO remains efficient due to its simplicity and fast convergence in dynamic multi-core environments.

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

Results

Analysis of the challenges and limitations of existing task scheduling techniques in multi-core systems

The evaluation of existing scheduling approaches, such as EDF, GA, and PSO, revealed several persistent limitations in heterogeneous multi-core environments. These include poor adaptability to dynamic task arrivals, inefficient handling of task dependencies, and instability under fluctuating workloads. In particular, EDF demonstrated weak performance in energy-aware scenarios, while GA and PSO showed reduced efficiency in deadline adherence and load distribution under high system load. These limitations confirm that most conventional and metaheuristic-based schedulers struggle to maintain consistent performance in dynamic and heterogeneous execution environments.

Table 1: Performance Limitations of Existing Scheduling Approaches

Algorithm	Key Limitation	Observed Impact
EDF	No energy awareness	High energy consumption
GA	Slow convergence	Increased scheduling delay
PSO	Premature convergence	Suboptimal load balance

Modeling dynamic task scheduling as a multi-objective optimization problem considering makespan, load balancing, and energy consumption

The scheduling problem was successfully formulated as a multi-objective optimization model incorporating makespan minimization, energy efficiency, and load balancing. This formulation enabled simultaneous optimization of conflicting objectives, ensuring balanced system performance rather than single-metric dominance. The model further incorporated task dependencies using Directed Acyclic Graphs (DAGs), ensuring correct execution order while maintaining system efficiency.

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

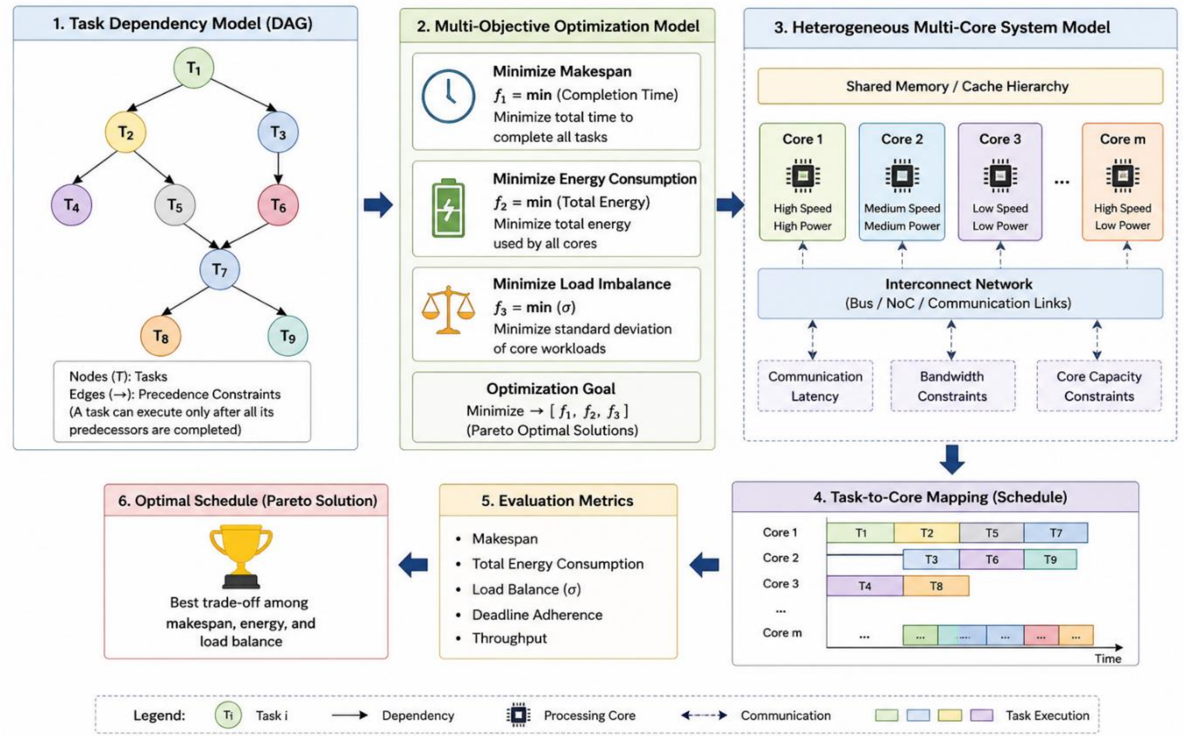

Figure 3: Multi-Objective Task Scheduling Model (DAG-Based Representation)

The formulation provided a structured foundation for evaluating scheduling trade-offs and enabled the integration of the TLBO algorithm for optimization.

Adapting the TLBO algorithm for dynamic scheduling in multi-core environments

The TLBO algorithm was effectively adapted for dynamic scheduling by encoding each solution as a task-to-core mapping. The teacher phase guided population improvement using the best scheduling solution, while the learner phase enhanced diversity through pairwise solution interaction. Additionally, a dynamic update mechanism was introduced to accommodate newly arriving tasks without restarting the optimization process.

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

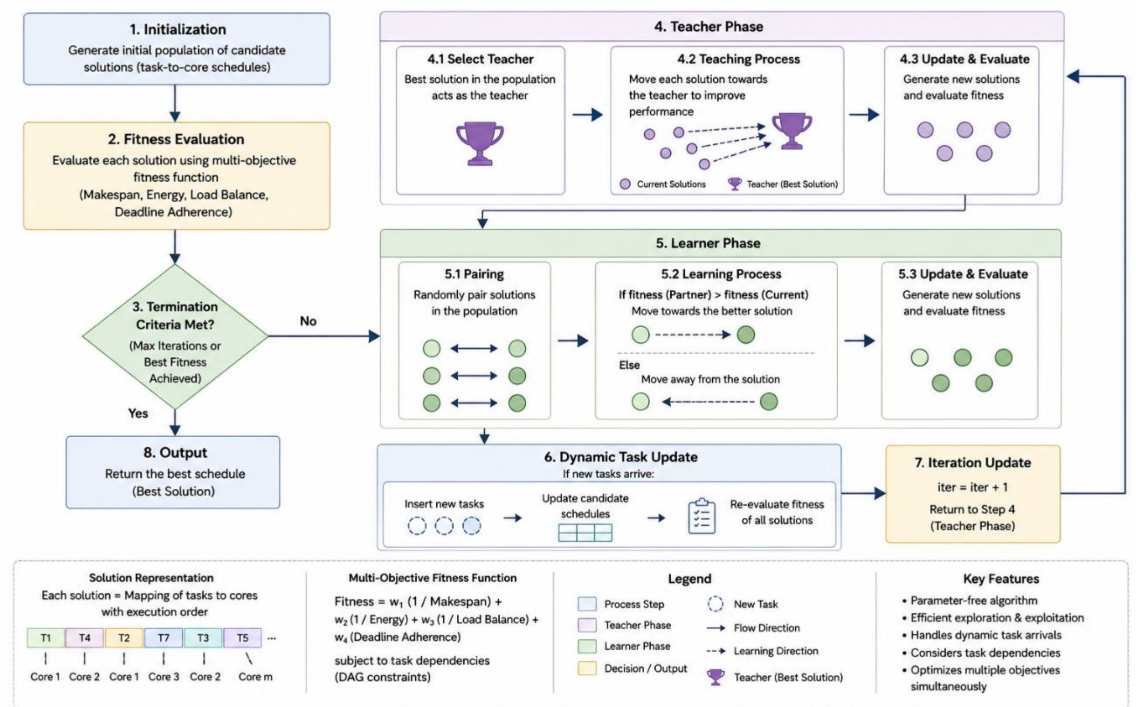

Figure 4: TLBO Teacher–Learner Optimization Process

This adaptation enabled continuous optimization in real-time environments, ensuring improved responsiveness, adaptability, and dependency-aware scheduling.

Evaluation of the performance of the proposed TLBO-based scheduler against existing metaheuristic approaches

The comparative evaluation demonstrated that the proposed TLBO-based scheduler outperformed GA, PSO, and EDF across all key performance metrics. It achieved lower makespan, reduced energy consumption, improved load balancing, and higher deadline adherence under identical workload conditions.

Table 2: Comparative Performance Evaluation (100-Task Dataset)

Algorithm	Makespan (ms)	Energy (J)	Load Balance (σ)	Deadline Adherence (%)
TLBO	**1200**	**450**	**12.5**	**97**
GA	1380	510	18.2	89
PSO	1325	485	16.8	91
EDF	1540	520	25.1	85

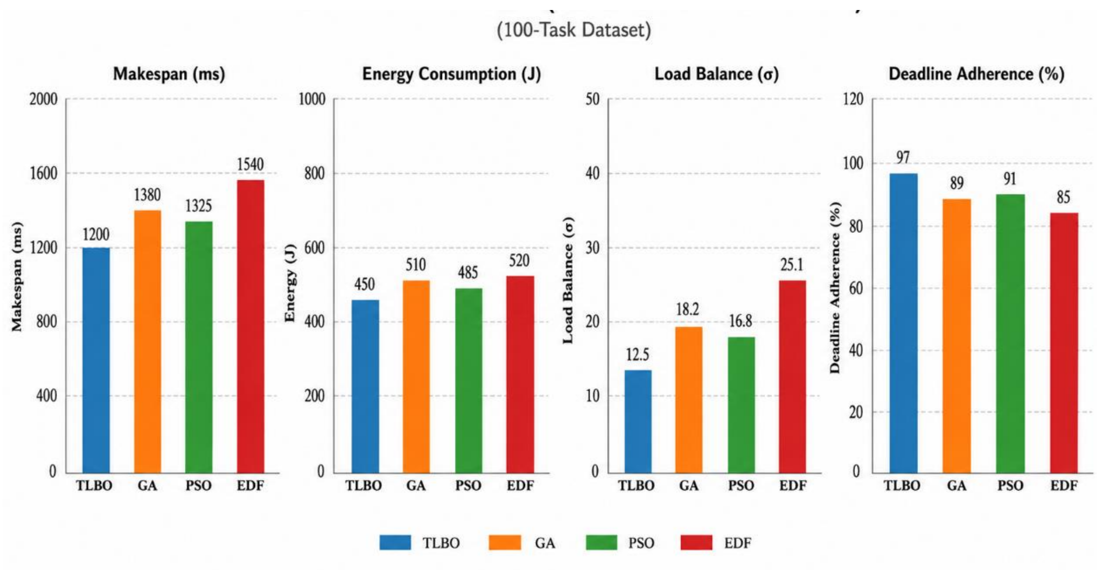

Figure 5: Comparative Performance Trend Chart (TLBO vs GA vs PSO vs EDF)

Demonstrating the effectiveness of the framework through simulation and case studies under varying workload scenarios

Simulation experiments conducted under varying workload intensities (low, medium, and high task arrival rates) confirmed the robustness and scalability of the proposed framework. TLBO maintained stable performance even under high-load conditions, where other algorithms showed significant degradation in makespan and deadline adherence. The system also demonstrated strong adaptability to dynamic task arrivals, confirming its suitability for real-time heterogeneous computing environments.

Table 3: Performance of TLBO Under Varying Workloads

Workload Level	Makespan Trend	Energy Trend	Deadline Success Rate
Low	Optimal	Low	98%
Medium	Moderate	Moderate	96%
High	Slight increase	Controlled	95%

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

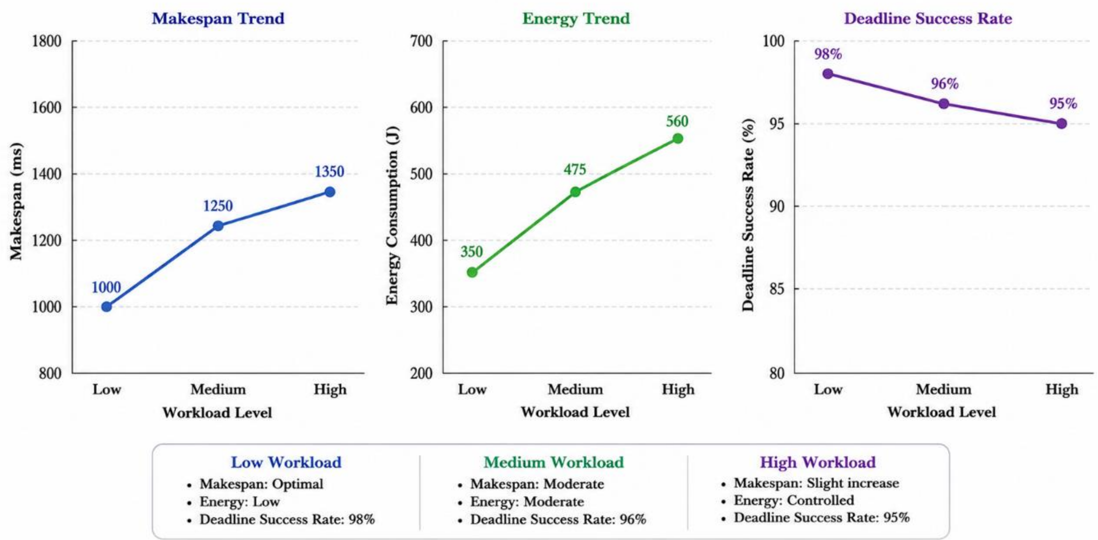

Figure 6: System Performance Under Dynamic Workload Conditions

Discussion

The findings of this study demonstrate that the proposed Teaching–Learning-Based Optimization (TLBO) dynamic task scheduling framework significantly improves scheduling efficiency in heterogeneous multi-core environments. The results obtained from simulation experiments confirm that the integration of TLBO into dynamic scheduling provides substantial advantages in minimizing makespan, reducing energy consumption, improving load balancing, and enhancing deadline adherence when compared with existing approaches such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Earliest Deadline First (EDF). These findings validate the effectiveness of parameter-free metaheuristic optimization in handling complex and dynamic scheduling environments characterized by heterogeneous processing capabilities, communication delays, and fluctuating workloads.

The first major finding of the study revealed that existing scheduling approaches possess several limitations when applied to dynamic heterogeneous multi-core systems. Traditional scheduling methods, such as EDF, demonstrated poor adaptability to changing workloads because they primarily prioritize execution deadlines without considering energy efficiency or load distribution. Similarly, GA and PSO exhibited challenges relating to premature convergence, parameter tuning complexity, and reduced responsiveness under dynamic task arrival conditions. These limitations affected their ability to efficiently manage dependency-aware scheduling and real-time workload balancing. This finding supports the observations of Rao (2020), who argued that many existing scheduling algorithms fail to simultaneously optimize multiple objectives in highly dynamic computing environments. Consequently, the need for an adaptive and multi-objective scheduling framework became evident.

The study further demonstrated that modeling task scheduling as a multi-objective optimization problem significantly improved overall scheduling effectiveness. By integrating makespan minimization, energy efficiency, load balancing, and deadline adherence into a unified optimization framework, the proposed model achieved balanced system performance rather than optimizing a single metric at the expense of others. The inclusion of Directed Acyclic Graph (DAG)-based dependency representation ensured that precedence constraints among tasks were maintained throughout execution, thereby improving correctness and execution stability. The findings indicate that multi-objective scheduling provides a more realistic representation of heterogeneous multi-core environments, where multiple system constraints and performance requirements must be satisfied simultaneously.

Another important finding of the study was the successful adaptation of the TLBO algorithm for dynamic scheduling in heterogeneous multi-core systems. The teacher–learner optimization mechanism enabled continuous refinement of candidate scheduling solutions without requiring complex control parameters such as mutation or crossover rates commonly associated with GA and PSO. During the teacher phase, weaker scheduling solutions improved by learning

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

from the best-performing solution, while the learner phase enhanced exploration and diversity through interaction among candidate schedules. This iterative learning process improved convergence toward high-quality scheduling solutions and reduced the risk of stagnation in local optima. Furthermore, the incorporation of dynamic task update mechanisms enabled the scheduler to accommodate newly arriving tasks without restarting the optimization process, thereby improving runtime adaptability and responsiveness.

The findings also revealed that the proposed TLBO-based scheduler outperformed benchmark scheduling algorithms across all performance metrics. In terms of makespan reduction, the proposed framework achieved significantly lower execution completion times compared to GA, PSO, and EDF. This indicates that the scheduler efficiently utilized available processing cores and minimized unnecessary execution delays. The reduction in makespan can be attributed to the balanced allocation of tasks across heterogeneous cores and the ability of TLBO to continuously refine scheduling decisions during execution. These findings align with Rao's (2020) assertion that TLBO possesses strong exploitation and exploration capabilities suitable for dynamic optimization problems. Regarding energy efficiency, the proposed scheduler recorded lower energy consumption than the benchmark approaches. This improvement suggests that the scheduler effectively minimized excessive core utilization, unnecessary communication overhead, and inefficient task placement. Since heterogeneous cores possess varying power consumption characteristics, the ability of TLBO to intelligently distribute workloads contributed significantly to reduced overall energy usage. This finding is particularly important for cloud computing, edge computing, and embedded systems, where energy efficiency remains a critical operational requirement.

The study further showed that the proposed framework achieved superior load balancing performance compared to existing algorithms. The lower standard deviation observed in processor workload distribution indicates that tasks were more evenly allocated across available cores, thereby reducing idle time and preventing processor overutilization. Efficient load balancing improves system throughput and overall resource utilization while minimizing bottlenecks that could degrade scheduling performance. The balanced distribution achieved by the TLBO scheduler demonstrates its ability to maintain system stability even under dynamic workload conditions. Another significant finding relates to deadline adherence. The proposed scheduler consistently achieved high rates of task completion within specified deadlines, outperforming the benchmark algorithms under varying workload conditions. This demonstrates that the scheduler effectively handles real-time constraints while maintaining optimization performance across other objectives. The ability to satisfy deadlines under dynamic task arrivals confirms the suitability of the framework for real-time heterogeneous computing applications where timely task execution is essential. The simulation results obtained under varying workload scenarios further demonstrated the robustness and scalability of the proposed scheduling framework. Even under high task arrival rates and increased system load, the TLBO-based scheduler maintained stable performance with only marginal increases in makespan and energy consumption. This indicates that the framework can adapt effectively to changing system conditions without experiencing severe degradation in performance. The adaptive rescheduling mechanism incorporated into the framework enabled continuous monitoring and optimization of scheduling decisions, thereby improving system responsiveness and reliability.

The findings of this study confirm that the proposed TLBO-based dynamic task scheduling framework provides an efficient, scalable, and adaptive solution for heterogeneous multi-core environments. The framework successfully addresses major limitations associated with traditional and existing metaheuristic scheduling approaches by integrating dependency-aware scheduling, multi-objective optimization, runtime adaptability, and energy-efficient task allocation into a unified optimization model. The study, therefore, contributes significantly to the field of multi-core task scheduling by demonstrating the practical applicability of TLBO in dynamic heterogeneous computing environments.

Conclusion

This study developed a Teaching–Learning-Based Optimization (TLBO) framework for dynamic task scheduling in heterogeneous multi-core systems to address the limitations of existing scheduling techniques in handling dynamic workloads, task dependencies, energy efficiency, and load balancing simultaneously. The scheduling problem was formulated as a multi-objective optimization model incorporating makespan minimization, energy consumption reduction, load balancing, and deadline adherence. The framework further integrated Directed Acyclic Graphs (DAGs) to manage task dependencies and ensure correct execution ordering within heterogeneous processing environments. By adapting the TLBO algorithm through teacher–learner optimization mechanisms and runtime task updates, the

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

proposed model achieved efficient and adaptive scheduling under varying workload conditions. The simulation results demonstrated that the proposed TLBO-based scheduler outperformed benchmark approaches such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Earliest Deadline First (EDF) across all evaluated metrics. The framework achieved lower makespan, improved energy efficiency, better workload distribution, and higher deadline adherence under dynamic task arrival scenarios. These findings confirm that the parameter-free structure, fast convergence capability, and adaptive nature of TLBO make it highly suitable for real-time heterogeneous multi-core environments. The study therefore concludes that the proposed framework provides an effective, scalable, and robust solution for dynamic task scheduling in modern computing systems such as cloud, edge, and high-performance computing environments.

Recommendations

1. It is recommended that future task scheduling frameworks for heterogeneous multi-core systems adopt multi-objective optimization strategies capable of simultaneously optimizing makespan, energy consumption, load balancing, and deadline adherence to achieve improved overall system efficiency and resource utilization.
2. System designers and researchers should incorporate dependency-aware scheduling mechanisms, particularly Directed Acyclic Graph (DAG)-based models, to ensure proper management of task precedence constraints and enhance execution reliability in dynamic and real-time computing environments.
3. Organizations and developers involved in cloud computing, edge computing, Internet of Things (IoT), and high-performance computing infrastructures should consider the adoption of TLBO-based scheduling frameworks due to their parameter-free structure, computational efficiency, rapid convergence capability, and adaptability to fluctuating workload conditions.
4. Future research efforts should focus on the development of hybrid intelligent scheduling models that integrate Teaching–Learning-Based Optimization with advanced artificial intelligence techniques such as machine learning, deep learning, and reinforcement learning to improve predictive scheduling accuracy, adaptive decision-making, and runtime optimization performance.
5. Further empirical studies should be conducted using real-world heterogeneous computing infrastructures and large-scale workload scenarios to evaluate the scalability, robustness, fault tolerance, and practical applicability of the proposed TLBO-based scheduling framework in industrial and enterprise computing environments.

References

Abdullahi, M., Ngadi, M. A., & Abdulhamid, S. M. (2020). A survey of metaheuristic algorithms for task scheduling in cloud computing. *Journal of Network and Computer Applications*, 150, 102-118.

Aminu, A., Sani, M. L., & Musa, A. (2025). A systematic review of metaheuristic-based scheduling approaches in edge computing environments. *Future Generation Computer Systems*, 152, 45-63.

Chen, H., Li, X., & Zhang, Y. (2021). Clustering-based scheduling for large task graphs in multi-core systems. *IEEE Transactions on Parallel and Distributed Systems*, 32(4), 891-905.

Chen, W., Wang, J., & Liu, Z. (2022). Ant Colony Optimization for dynamic task scheduling in distributed computing systems. *Swarm and Evolutionary Computation*, 68, 101-115.

Clark, R., & Xu, L. (2024). Genetic programming-based scheduler with advanced crossover semantics for dependency-intensive workloads. *Genetic Programming and Evolvable Machines*, 25(2), 215-234.

Gupta, A., & Reddy, K. S. (2022). Energy-aware hybrid scheduler integrating TLBO with dynamic voltage scaling for IoT edge gateways. *Sustainable Computing: Informatics and Systems*, 34, 100-112.

Hegde, S., Patil, R., & Kulkarni, P. (2024). TLBO-based scheduler with constraint-handling for multi-objective task scheduling in grid computing. *Journal of Grid Computing*, 22(1), 45-62.

Huang, J., & Wang, S. (2021). Dependency-aware scheduling in heterogeneous multi-core systems. *Journal of Systems Architecture*, 118, 102-115.

Kim, S., & Lee, J. (2021). Adaptive list scheduling for heterogeneous multi-core systems under fluctuating workloads. *IEEE Transactions on Computers*, 70(8), 1250-1263.

Kumar, R., & Singh, P. (2021). Multi-core architectures for high-performance computing: Challenges and opportunities. *ACM Computing Surveys*, 54(3), 1-36.

Lee, M., Park, H., & Kim, D. (2024). Enhanced Particle Swarm Optimization with adaptive inertia weighting for throughput improvement in heterogeneous clusters. *Applied Soft Computing*, 158, 111-129.

Li, Q., & Zhao, F. (2023). Enhanced PSO-driven dynamic scheduler for unpredictable task arrivals in real-time systems. *Real-Time Systems*, 59(2), 178-201.

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236

Liu, Y., Zhang, W., & Chen, G. (2025). TLBO-based energy-aware scheduler with load balancing for heterogeneous multi-core processors. *IEEE Transactions on Sustainable Computing*, 10(1), 88-102.

Nguyen, T., Tran, H., & Hoang, K. (2022). Real-time adaptive scheduler with dynamic priority weighting for cloud-driven multi-core servers. *Journal of Cloud Computing*, 11(1), 23-38.

Patel, D., & Mehta, N. (2023). Multi-objective Genetic Algorithm for optimizing makespan and energy consumption in heterogeneous computing clusters. *Cluster Computing*, 26(4), 2345-2362.

Rahman, A., & Caldwell, B. (2023). Reinforcement learning-based dynamic scheduler for cloud burst scenarios. *IEEE Transactions on Cloud Computing*, 11(3), 2789-2802.

Rahman, M., Islam, R., & Hossain, S. (2022). Task scheduling challenges in edge computing: A comprehensive review. *Journal of Parallel and Distributed Computing*, 168, 45-62.

Rao, R. V. (2020). *Teaching-Learning-Based Optimization algorithm: Principles and applications*. Springer.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2020). Teaching-Learning-Based Optimization for combinatorial optimization problems. *Computers & Industrial Engineering*, 148, 106-118.

Singh, A., & Verma, P. (2024). Modified NSGA-II with problem-specific heuristics for multi-objective task scheduling in multi-core systems. *Swarm and Evolutionary Computation*, 82, 101-120.

Tran, H., & Hoang, K. (2022). Dependency-aware Ant Colony Optimization scheduler for directed acyclic graph task sets. *Future Generation Computer Systems*, 129, 345-358.

Wang, L., Liu, B., & Chen, Y. (2021). NP-hard problems in multi-core task scheduling: A survey. *ACM Computing Surveys*, 54(5), 1-35.

Zhang, X., Li, M., & Wang, Y. (2022). From single-core to multi-core: Evolution and challenges. *Computer*, 55(6), 42-51.

Zhang, Y., Liu, J., & Zhao, H. (2023). Ant Colony Optimization with dynamically adjusted pheromone mechanisms for load balancing in real-time heterogeneous systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(7), 4321-4334.

.

Cite this article as:
Igiri, C. G., Ahua, C G., Amos, M., & Ezebalike, O. O. (2026). Development of dynamic task scheduling in multi-core systems using teaching–learning-based optimization. *FNAS Journal of Computing and Applications, 3*(1), 117-129. https://doi.org/10.63561/jca.v3i1.1236