International Journal of Advanced Engineering, Management and Science (IJAEMS)
Peer-Reviewed Journal
ISSN: 2454-1311 | Vol-12, Issue-1; Jan-Feb, 2026
Journal Home Page: https://ijaems.com/
DOI: https://dx.doi.org/10.22161/ijaems.121.14

Energy-efficient task scheduling algorithms in hybrid CPU-GPU systems

Khushboo Kumari Yadav[1], Ankur Yadav[2]

[1]Software Enabling and Optimization Engineer, Santa Clara, USA
[2]Software engineer, Sunnyvale, USA

Received: 14 Jan 2026; Received in revised form: 16 Feb 2026; Accepted: 19 Feb 2026; Available online: 24 Feb 2026

Abstract *— This article presents a systematic analysis of energy-efficient task scheduling algorithms in hybrid CPU–GPU systems, where heterogeneous computing resources exhibit fundamentally different performance and energy characteristics. Such systems are widely used in high-performance computing, data centers, and AI-driven workloads, where increasing computational demand and energy constraints limit the effectiveness of scheduling strategies focused solely on execution time. The study is conducted as a review-and-analytical synthesis of peer-reviewed publications published between 2022 and 2025, without quantitative aggregation of results due to heterogeneity of experimental setups, metrics, and energy models. Particular attention is paid to the role of optimization metrics, quality of input energy data, and integration of power management mechanisms in shaping the observed effectiveness of scheduling algorithms. The analysis shows that reported improvements strongly depend on the choice and aggregation level of performance and energy metrics, as well as on the accuracy of task energy characterization, rather than on the algorithmic class alone. It is demonstrated that single-objective formulations fail to capture the behavior of hybrid CPU–GPU systems, while composite time–energy criteria and explicit device-selection policies provide a more consistent basis for evaluation. The study establishes that meaningful comparison of energy-efficient scheduling approaches requires a clear separation between architectural scheduling frameworks and concrete algorithms, along with explicit fixation of the energy management context. The article is intended for researchers and practitioners working on scheduling, power-aware computing, and heterogeneous system design in high-performance and data-intensive environments.*

Keywords *— **energy-efficient scheduling, hybrid CPU–GPU systems, heterogeneous computing, power-aware scheduling, composite metrics, task profiling, high-performance computing, data centers.***

I. INTRODUCTION

The growth of computational workloads in HPC, machine learning, and AI services intensifies energy constraints in the operation of computing infrastructure, particularly in hybrid CPU–GPU systems characterized by diverse device performance and energy profiles. While modern platforms increasingly utilize such architectures, their actual effectiveness is determined by task scheduling strategies; meanwhile, a focus solely on minimizing execution time fails to account for the relationship between power consumption, resource utilization, and computation type, leading to increased energy costs and reduced stability under dynamic loads [3]. The absence of a unified methodological framework ensuring metric consistency, result comparability, and the integration of power management mechanisms limits the reproducibility and portability of energy-efficient scheduling solutions in hybrid CPU–GPU systems.

The aim of this study is to identify the methodological conditions for the correct evaluation

and comparison of energy-efficient task scheduling algorithms in hybrid CPU–GPU systems, taking into account the metrics, energy models, and power management mechanisms employed. To achieve this goal, the work addresses the following research tasks:

– classify modern energy-efficient scheduling algorithms applied in hybrid CPU–GPU systems from the perspective of abstraction level and architectural binding;

– analyze metrics and optimization criteria used in the literature and determine their impact on the interpretation of energy efficiency results;

– summarize methods for integrating energy management mechanisms into the task scheduling loop and assess their influence on observed effects;

– determine methodological limitations of result comparability and conditions for the practical applicability of algorithms in real-world computing environments.

The research hypothesis states that differences in the reported energy efficiency of scheduling algorithms in hybrid CPU–GPU systems are largely conditioned by methodological factors—the choice of metrics, energy models, and the architectural context of the experiment—rather than the algorithm's belonging to a specific class. It is assumed that when these conditions are fixed, the differences between algorithmic approaches are substantially reduced.

The scope of the study is limited to the analysis of algorithmic, methodological, and operational aspects of energy-efficient scheduling in hybrid CPU–GPU systems. HPC clusters, server nodes, and heterogeneous platforms with shared CPU and GPU usage are considered, whereas questions of hardware design and economic assessment are touched upon only in the context of interpreting energy metrics and the practical applicability of algorithms.

II. MATERIALS AND METHODS

The methodological basis of the study is formed on the systematization of theoretical and applied models of energy-efficient task scheduling in heterogeneous computing systems with shared CPU and GPU usage. The source corpus includes publications from 2022–2025 dedicated to scheduling algorithms, energy metrics, power management mechanisms, and architectural features of hybrid computing nodes in data centers, high-performance, and edge computing environments.

In the study by Hou and Ismail [1], energy-efficient scheduling is formalized as a trade-off problem between power consumption and response time at the software scheduler level. The review by Kocot et al. [2] is used to systematize energy efficiency metrics and algorithm classes in high-performance computing. The approach of Li et al. [3] is applied to analyze energy-efficient scheduling of containerized tasks in a dynamic environment. The work of Liu et al. [4] is used to account for the specifics of heterogeneous multi-core processors and edge computing. The classification of scheduling methods for heterogeneous chips is borrowed from the review by Miao et al. [5]. The architectural scheme of two-phase task flow scheduling is used according to Peng and Wang [6]. Features of joint task execution on CPU and GPU are analyzed based on Raúl and Bosque [7]. The method for obtaining task energy characteristics without full profiling is borrowed from Salinas-Hilburg et al. [8]. The operational loop of energy-efficient computing system management is considered according to Suarez et al. [9]. The formalization of computing device selection considering latency and energy is used according to Xie and Fang [10].

The research methodology is based on a combination of structural analysis of scheduling algorithms, a comparative review of used energy metrics, and generalization of power management mechanisms in hybrid computing systems. Such an approach allows for viewing energy-efficient task scheduling in CPU–GPU environments as an integration of algorithmic, architectural, and operational solutions determining result reproducibility and practical method applicability in real-world computing systems.

III. RESULTS

Energy-efficient scheduling in hybrid CPU–GPU systems is determined by the coordinated choice of efficiency metrics and power management mechanisms, rather than solely by the applied

algorithm, since with unchanged task distribution logic, changing the optimization indicator leads to different efficiency interpretations in heterogeneous nodes with diverse device energy profiles [2]. The use of single-axis metrics oriented only toward execution time or power consumption distorts the efficiency assessment, as acceleration via GPU may be accompanied by an increase in CPU background power consumption, requiring the application of composite indicators linking time and energy [7]. Simultaneously, power management mechanisms form an extended solution space for the scheduler, as dynamic frequency and voltage scaling, power capping, and shutdown modes directly alter the task execution energy profile and derived metric values, reducing result comparability when the energy context changes [4]. Figure 1 shows the distribution of reported improvements in key metrics used in energy-aware HPC-level formulations.

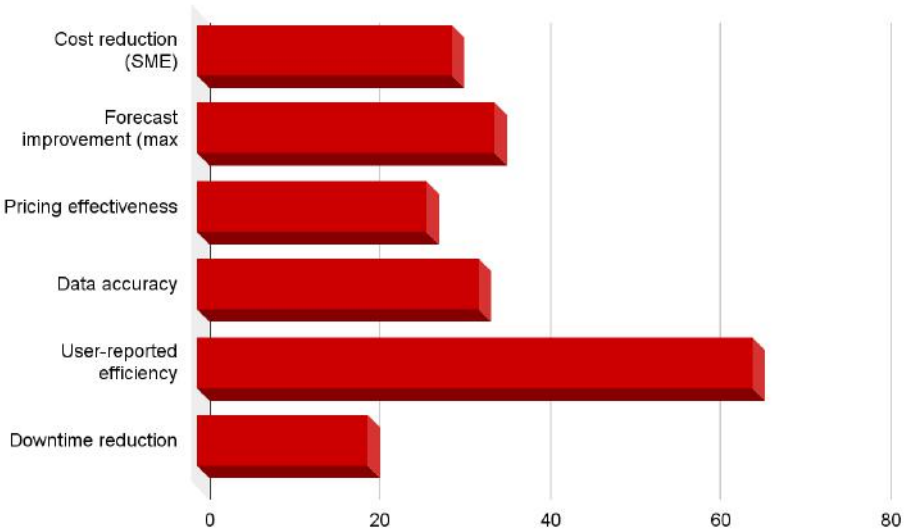

Fig.1 – Comparison of reported improvements in key metrics under energy-aware formulations (Composed by the author based on source [2])

The data presented in the diagram demonstrate the heterogeneity of reported improvements across various energy-aware scheduling metrics. The greatest relative effect is recorded for the efficiency indicator evaluated by users, reaching 62.5%, whereas metrics related to forecasting and data quality show improvements at the level of 33–35%. Indicators directly related to operational costs and physical processes, such as operating cost reduction (30%) and downtime reduction (20%), are characterized by a more limited improvement range. Such a distribution confirms that the magnitude of the reported effect is determined by the selected metric and its aggregation level, which complicates result comparability in the absence of explicit indicator fixation.

Energy-efficient task scheduling in hybrid CPU–GPU systems is viewed as a function of input energy data quality, not merely as a consequence of algorithmic class selection. With identical task distribution strategies, differences in the availability and accuracy of energy characteristics lead to changes in the observed effect comparable to the transition between heuristic and optimization methods [8]. This indicates the independent role of task energy parameters in forming the scheduling result.

Full profiling ensures maximum scheduler awareness; however, its computational cost makes this approach impractical for large-scale and long-running workloads [4]. In hybrid systems, this limitation is intensified by the need to account for diverse CPU and GPU energy profiles and their interaction during joint task execution [2], making approximate methods for obtaining energy characteristics a necessary element of scalable scheduling.

Comparing methods under different profiling scenarios shows that replacing full profiling with

reduced energy representations leads to a moderate decrease in target indicator improvement while preserving result structure [3]. Figure 2 shows the distribution of improvements for optimization, metaheuristic, and heuristic methods under various scenarios for obtaining energy characteristics.

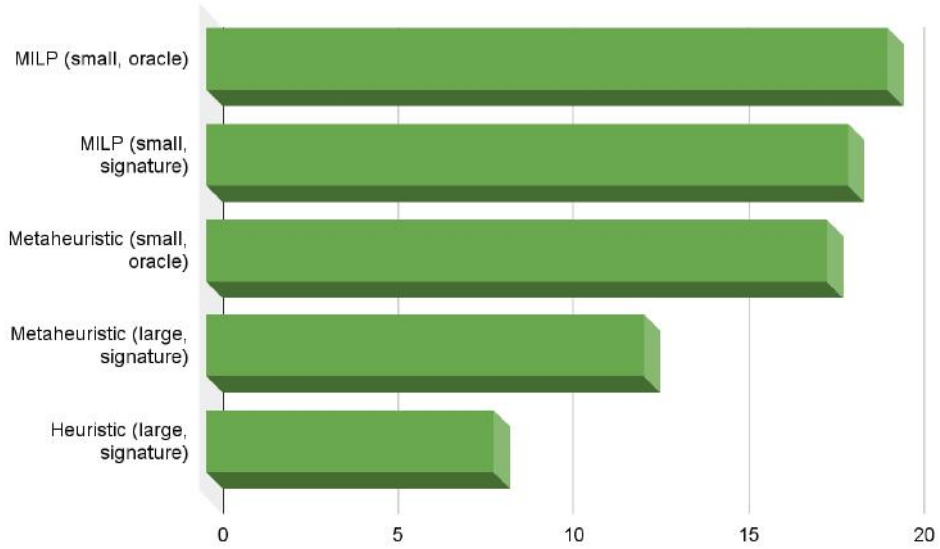

Fig.2 – Comparison of scheduling methods in terms of target metric improvement under oracle and signature scenarios (Composed by the author based on source [8])

Based on the data presented in the diagram, it is evident that using reduced profiling leads to a decrease in absolute values of target indicator improvement while maintaining the relative order of scheduling methods. The optimization method demonstrates the highest value with full profiling (19.4) and a moderate decrease when using reduced data (18.3), whereas metaheuristic and heuristic approaches are characterized by more pronounced sensitivity to input information accuracy. The obtained values show that reduced profiling influences the effect magnitude but does not distort result comparability between method classes.

When using reduced profiling, the hierarchy of methods by efficiency is preserved, despite the decrease in absolute indicator values [8]. In the context of hybrid CPU–GPU systems, this indicates that scheduling result stability is determined by the consistency of the energy model and algorithmic logic, rather than by maximum input data accuracy [7]. A similar problem formulation, where resource selection is based on normalized energy and time estimates, is used in other heterogeneous environments as well.

Thus, the analysis results show that the quality of energy-efficient scheduling in hybrid CPU–GPU systems is determined by algorithm selection and the method of obtaining and representing task energy characteristics, while reduced profiling can be considered a methodologically admissible substitute for full measurement while preserving result comparability.

IV. DISCUSSION

The correctness of comparing energy-efficient scheduling algorithms in hybrid CPU–GPU systems is determined by the abstraction level at which the analysis is performed. Comparing results from various studies proves methodologically justified only with a clear separation between task management architecture and the specific distribution and ordering algorithm, as mixing these levels leads to incorrect interpretation of observed effects. In hybrid systems, this aspect acquires particular significance because architectural solutions often include energy monitoring and management mechanisms that directly influence measured indicators independently of algorithmic logic.

Analysis shows that a substantial part of the differences between reported improvements in the

literature is conditioned specifically by architectural premises rather than algorithm features. The use of centralized or distributed frameworks, differences in queue models, and methods for accounting for energy characteristics form different solution spaces for the scheduler, making direct comparison of algorithms without fixing the architectural context methodologically incorrect. In hybrid CPU–GPU systems, this manifests in varying degrees of CPU participation during GPU-oriented workloads and in unequal system reaction to power caps and energy management modes. Table 1 examines the methodological breakdown of aspects necessary for correct solution positioning in hybrid CPU–GPU systems, allowing for the separation of effects conditioned by architecture and experimental environment from effects related to scheduling algorithms.

Table 1 – Methodological aspects of scheduler comparison for heterogeneous systems (Compiled by the author based on source [5])

Analytical aspect	Core finding	Relevance for this study
Scheduling levels	Clear separation between framework and algorithm	Used as the base structure for architectural description
Algorithm types	Five classes with systemic trade-offs	Provides a common language for method comparison
Experimental basis	Lack of standards, dominance of SIL/MIL	Explains limits of result comparability
AI perspective	RL is promising but methodologically complex	Enables careful positioning of ML-based approaches

The interpretation of energy efficiency in hybrid CPU–GPU systems cannot be reduced to minimizing power consumption or improving a single aggregated indicator. In such architectures, scenarios are often observed where gains in execution time and resource utilization are achieved with ambiguous changes in energy efficiency indicators, including the absence of improvement in EDP compared to GPU-only mode. This indicates the necessity of an explicit choice of optimization criterion when formulating the scheduling problem.

A practically applicable scheduling formulation in a heterogeneous environment implies including execution device selection within the optimization model itself, rather than outsourcing it as a fixed decision [10]. Energy and latency in such models are included in the objective function with explicit weighting coefficients, which allows for managing the trade-off between performance and power consumption depending on operational priorities. For hybrid CPU–GPU systems, this approach is fundamental, since CPUs and GPUs possess different energy profiles and react differently to load changes and power management mechanisms.

Thus, the analysis shows that correct interpretation of energy-efficient scheduling in hybrid CPU–GPU systems is determined by separating abstraction levels, consciously choosing optimization criteria, and fixing the architectural context within which observed effects are evaluated.

V. CONCLUSION

The conducted study has shown that energy-efficient task scheduling in hybrid CPU–GPU systems cannot be viewed as a consequence of selecting a single algorithm and requires a coordinated problem formulation including correct selection of metrics, energy models, and power management mechanisms. Under conditions of heterogeneous computing nodes, it is the methodological premises that determine the interpretation and comparability of reported effects.

It is established that the use of single-axis criteria oriented exclusively toward execution time or power consumption does not reflect the real behavior of hybrid CPU–GPU systems and leads to a distorted efficiency assessment. A practically applicable assessment requires the use of composite indicators accounting for the trade-off between latency, energy,

and resource utilization, and the explicit inclusion of computing device selection in the optimization loop.

It is shown that the quality of input energy data is an independent factor of scheduling efficiency. Reduced task profiling lowers the absolute values of target indicators but preserves relative method comparability, making it an admissible tool for scalable computing environments given correct energy model formalization.

The research results confirm that correct comparison of energy-efficient scheduling algorithms is possible only when separating control architecture levels and algorithmic logic, with explicit fixation of the experimental and energy context. This approach allows for transitioning from local improvement of individual indicators to a reproducible and interpretable assessment of energy efficiency in hybrid CPU–GPU systems. The obtained conclusions can be used as a methodological basis for building reproducible experimental protocols and developing comparable energy-efficient schedulers for hybrid CPU–GPU computing platforms.

REFERENCES

[1] Hou, H., & Ismail, A. (2024). EETS: An energy-efficient task scheduler in cloud computing based on improved DQN algorithm. Journal of King Saud University - Computer and Information Sciences, 36(8), 102177. https://doi.org/10.1016/j.jksuci.2024.102177

[2] Kocot, B., Czarnul, P., & Proficz, J. (2023). Energy-aware scheduling for high-performance computing systems: A survey. Energies, 16(2), 890. https://doi.org/10.3390/en16020890

[3] Li, Z., Zhang, S., Li, Y., Liu, X., Huang, J., & Hu, J. (2025). Energy-efficient container scheduling based on deep reinforcement learning in data centers. Computers, 14(12), 560. https://doi.org/10.3390/computers14120560

[4] Liu, Y., Qu, H., Chen, S., Zhang, T., & Wang, J. (2025). Energy efficient task scheduling for heterogeneous multicore processors in edge computing. Scientific Reports, 15, 11819. https://doi.org/10.1038/s41598-025-92604-6

[5] Miao, Z., Shao, C., Li, H., & Tang, Z. (2025). Review of task-scheduling methods for heterogeneous chips. Electronics, 14(6), 1191. https://doi.org/10.3390/electronics14061191

[6] Peng, Q., & Wang, S. (2023). MASA: Multi-application scheduling algorithm for heterogeneous resource platform. Electronics, 12(19), 4056. https://doi.org/10.3390/electronics12194056

[7] Raúl, N., & Bosque, J. L. (2025). CPU-GPU co-execution through the exploitation of hybrid technologies via SYCL. The Journal of Supercomputing, 81, 452. https://doi.org/10.1007/s11227-025-06963-y

[8] Salinas-Hilburg, J. C., Zapater, M., Moya, J. M., & Ayala, J. L. (2022). Energy-aware task scheduling in data centers using an application signature. Computers & Electrical Engineering, 97, 107630. https://doi.org/10.1016/j.compeleceng.2021.107630

[9] Suarez, E., Bockelmann, H., Eicker, N., Eitzinger, J., El Sayed, S., Fieseler, T., Frank, M., Frech, P., Giesselmann, P., Hackenberg, D., Hager, G., Herten, A., Ilsche, T., Koller, B., Laure, E., Manzano, C., Oeste, S., Ott, M., Reuter, K., … von St. Vieth, B. (2025). Energy-aware operation of HPC systems in Germany. Frontiers in High Performance Computing, 3. https://doi.org/10.3389/fhpcp.2025.1520207

[10] Xie, Y., & Fang, Q. (2025). An energy-aware generative AI edge inference framework for low-power IoT devices. Electronics, 14(20), 4086. https://doi.org/10.3390/electronics14204086