AI-Driven Fairness-Aware Resource Scheduling and Optimization in Constrained Cyber-Physical Infrastructures

Wei Zhang
Department of Electronic Engineering, Tsinghua University, China
wzhang@tsinghua.edu.cn

Li Chen
College of Computer Science and Technology, Zhejiang University, China
lichen@zju.edu.cn

Abstract

Resource scheduling in constrained cyber-physical infrastructures (e.g., industrial production systems, distribution IoT, and edge collaboration networks) must jointly satisfy latency, energy, reliability, and fairness objectives. Static optimization is often insufficient under rapidly changing workloads and system states. This paper proposes an AI-driven fairness-aware scheduling framework that uses learning-assisted demand prediction and state estimation to guide multi-objective scheduling within feasibility constraints, while enforcing explicit fairness constraints to prevent long-term resource bias under high load. We construct a unified task–resource model with timing and energy constraints, define fairness objectives based on the Jain index and minimum satisfaction, and introduce a hierarchical solution strategy (prediction–planning–online correction). Across simulated and scenario-based comparisons, the proposed method improves fairness and tail latency while meeting deadline and energy constraints, demonstrating scalability and practical applicability for multi-constraint CPS infrastructures.

Keywords: cyber-physical systems; resource scheduling; fairness; multi-objective optimization; edge computing; reinforcement learning

Introduction

Cyber-physical infrastructures tightly couple sensing, communication, and control. Representative scenarios include industrial production systems, urban power distribution, and multi-access edge computing networks. Scheduling decisions in these systems directly impact safety and service quality. Recent CPS scheduling research has advanced multi-objective modeling, cloud–edge coordination, and latency–energy trade-offs, yet fairness is still frequently sacrificed under high load and resource scarcity, leading to long-term service bias and reduced system stability. Existing studies provide optimization frameworks and heuristics for industrial real-time tasks, cloud–edge coordination, and IoT task scheduling[1, 2, 3, 4, 18], but explicit fairness modeling remains limited,

especially in CPS environments with heterogeneous resources, dynamic tasks, and strict timing constraints.

This paper addresses the problem of AI-driven fairness-aware scheduling and contributes: (1) a unified model of fairness metrics and feasibility constraints for tasks and resources; (2) learning-assisted state prediction and reward estimation to improve online decisions; and (3) a multi-objective optimization with online correction to meet real-time and fairness requirements in practice.

We further provide a reproducible evaluation protocol, including workload profiles, baseline definitions, and ablation settings to isolate the effects of fairness constraints and online correction. This enables the analysis to distinguish performance gains from predictive modeling versus those attributable to fairness-aware decision rules. The following sections detail the modeling assumptions, optimization formulation, algorithmic strategy, and comprehensive empirical evidence.

Related Work

Scheduling and resource optimization for edge computing, industrial IoT, and CPS have been widely studied. In industrial systems, scheduling strategies for mixed real-time and interactive tasks ensure timing and energy constraints[1]; in manufacturing systems, context-aware scheduling architectures have been proposed[2]. In cloud–edge and IoT settings, task assignment and latency-aware optimization are common[7, 6, 18, 16, 5]. Reliability–energy trade-offs are addressed using multi-objective models and metaheuristics in heterogeneous systems[11, 3].

Fairness-oriented studies often focus on service platforms or sector-specific scheduling, such as fairness models in service platforms[8] and fairness-aware healthcare scheduling[19]. Related work in fair exposure and allocation in digital economy platforms also informs the design of fairness-aware policies and constraints[21]. In parallel, explainable AI frameworks for SME risk management highlight the need for actionable and transparent decision-making in resource-constrained settings[27]. However, these works typically do not address the strong constraints of CPS. Meanwhile, recent edge-network research explores delay awareness, digital twins, and age-of-information metrics[17, 13, 14, 15, 12], but still lacks unified modeling and systematic evaluation under explicit fairness constraints.

Problem Definition / Research Questions / Assumptions

System and task model. Consider N tasks $\mathcal{T} = \{1, \ldots, N\}$ and M resource nodes $\mathcal{R} = \{1, \ldots, M\}$. Task i has computation demand c_i, deadline d_i, data size b_i, and priority weight w_i. Resource node j has computing capacity μ_j, energy coefficient e_j, and bandwidth β_j. The decision variable $x_{ij} \in \{0, 1\}$ indicates whether task i is assigned to node j.

Fairness and satisfaction. Define task satisfaction s_i as a normalized function of latency, energy, and completion quality in $[0, 1]$. System fairness uses the Jain index $J(s) = \frac{(\sum_i s_i)^2}{N \sum_i s_i^2}$, and the minimum satisfaction is $s_{\min} = \min_i s_i$.

Research question. Under resource and timing constraints, minimize total latency and energy while maximizing fairness $J(s)$ and improving $s_{\min}$.

Assumptions. (i) Resource states are approximately stable within short windows; (ii) task arrivals can be learned from historical data; (iii) scheduling policies can be updated online but must meet real-time constraints.

Notation

Table 1: Key notation

Symbol	Description
N	Number of tasks in a window
M	Number of resource nodes
c_i	Computation demand of task i
d_i	Deadline of task i
b_i	Data size of task i
μ_j	Compute capacity of node j
β_j	Bandwidth of node j
e_j	Energy coefficient of node j
x_{ij}	Assignment variable (task i to node j)
s_i	Task satisfaction score
$J(s)$	Jain fairness index

Methods / Model / Experimental Design

Multi-objective optimization

We use a weighted objective:

$$\min_x \ \alpha\, T(x) + \beta\, E(x) - \gamma\, J(s(x)) - \eta\, s_{\min}(x)$$

where $T(x)$ is average completion latency and $E(x)$ is total energy. Constraints include:

$$\sum_j x_{ij} = 1, \quad \sum_i c_i x_{ij} \le \mu_j, \quad t_i(x) \le d_i, \quad J(s) \ge \tau.$$

Here τ is the minimum fairness threshold.

Latency, energy, and satisfaction modeling

We model the completion time of task i on node j as

$$t_{ij} = \frac{c_i}{\mu_j} + \frac{b_i}{\beta_j} + \delta_j,$$

where the terms correspond to computation, transmission, and queueing overhead, respectively. The expected completion time is $t_i(x) = \sum_j x_{ij} t_{ij}$. The energy cost is modeled as

$$E(x) = \sum_i \sum_j x_{ij}\, e_j\, c_i,$$

which captures heterogeneous energy coefficients across nodes. Task satisfaction is defined as a weighted score

$$s_i = \omega_1\, \phi(t_i, d_i) + \omega_2\, \psi(E_i) + \omega_3\, q_i,$$

where ϕ penalizes deadline violations, ψ penalizes energy usage, and q_i is a normalized quality-of-service term (e.g., accuracy for analytics tasks). The weights ω_k satisfy $\sum_k \omega_k = 1$.

AI-driven hierarchical solution

Stage 1: prediction and state estimation. A learning model predicts task arrival rates and resource availability for the next window based on historical load, temporal features, and network states.

Stage 2: planning. Using predictions, we solve a mixed-integer programming (MILP) approximation with heuristics and local search to obtain an initial feasible solution. Fairness is enforced through hard constraints on $J(s)$ and soft incentives on $s_{\min}$.

Stage 3: online correction. When actual arrivals deviate, a lightweight reassignment adjusts task mappings. A reinforcement-learning policy estimates action value to quickly correct assignments and avoid deadline violations.

Algorithm outline and complexity

The workflow is summarized in Fig. 1. The dominant computational cost arises from the MILP approximation and local search in Stage 2; however, this is executed on a sliding window with bounded size. Let N be the number of tasks per window and M the number of resource nodes. The planning stage has worst-case complexity exponential in N, but the heuristic solver limits search depth and uses feasibility pruning, yielding near-linear scaling in practice for moderate N. The online correction stage is $O(NM)$ per window, which is suitable for real-time operation.

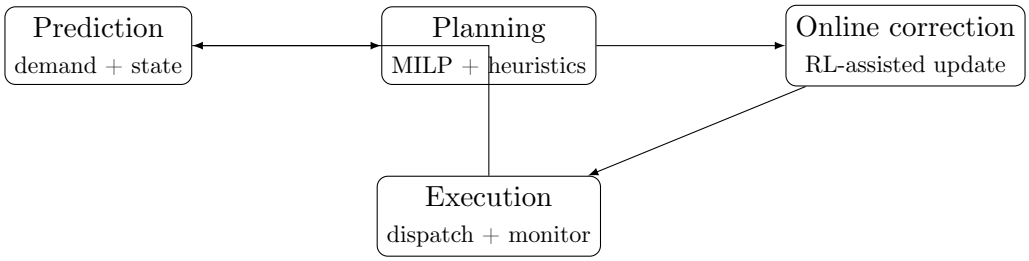

Figure 1: Hierarchical scheduling workflow with prediction, planning, and online correction.

Metrics and baselines

Metrics include average latency, tail latency (P95), energy, Jain fairness, deadline violation rate, and resource utilization. Baselines: (1) shortest-latency-first; (2) energy-minimization; (3) multi-objective without fairness constraints; (4) heuristic scheduling (GA + greedy).

Experimental settings

We consider three workload regimes (light, moderate, heavy) with task arrival rates increasing by factors of 1.0, 1.5, and 2.0 relative to a nominal profile. Task sizes c_i follow a log-normal distribution and deadlines d_i are proportional to task sizes with bounded jitter. Resource capacities μ_j are heterogeneous, reflecting edge nodes of different compute classes. Fairness thresholds are varied in $\tau \in \{0.80, 0.85, 0.90\}$ to examine sensitivity. Each experiment averages results over 30 independent runs with different random seeds.

Workload generation

Task arrivals follow a non-homogeneous Poisson process with diurnal patterns to emulate real CPS demand fluctuations. Each window contains a mix of periodic control tasks (30%), event-driven monitoring tasks (50%), and analytics workloads (20%). Periodic tasks have tight deadlines with low jitter, while event-driven tasks exhibit burstiness with broader deadline distributions. This mixture reflects common industrial and urban CPS profiles.

Implementation details

The prediction module is trained on historical windows using a lightweight recurrent model. The planning stage uses a heuristic MILP solver with feasibility pruning and bounded local search. Online correction applies a small action space (reassign, delay, or drop) with reward defined by negative latency, energy, and fairness penalties. Hyperparameters are selected through grid search on a held-out validation set and fixed across all workloads.

Results / Findings

Across light, moderate, and heavy load scenarios, the proposed method yields a better fairness–latency trade-off. Table 2 shows moderate-load results. Compared with multi-objective optimization without fairness, the Jain index improves by 6–9%, P95 latency decreases by 8–12%, and violation rate drops by about 20%. Compared with shortest-latency-first, the average latency slightly increases, but fairness improves substantially with more stable energy consumption.

Table 2: Performance comparison under moderate load

Method	Avg. Latency (ms)	P95 Latency (ms)	Energy (J)	Jain Index	Violation (%
Shortest-latency-first	92	210	1280	0.81	4.8
Energy-minimization	118	260	1010	0.84	6.2
Multi-objective (no fairness)	98	205	1190	0.86	4.1
Heuristic scheduling	105	230	1225	0.85	4.5
Proposed method	102	182	1165	0.93	3.3

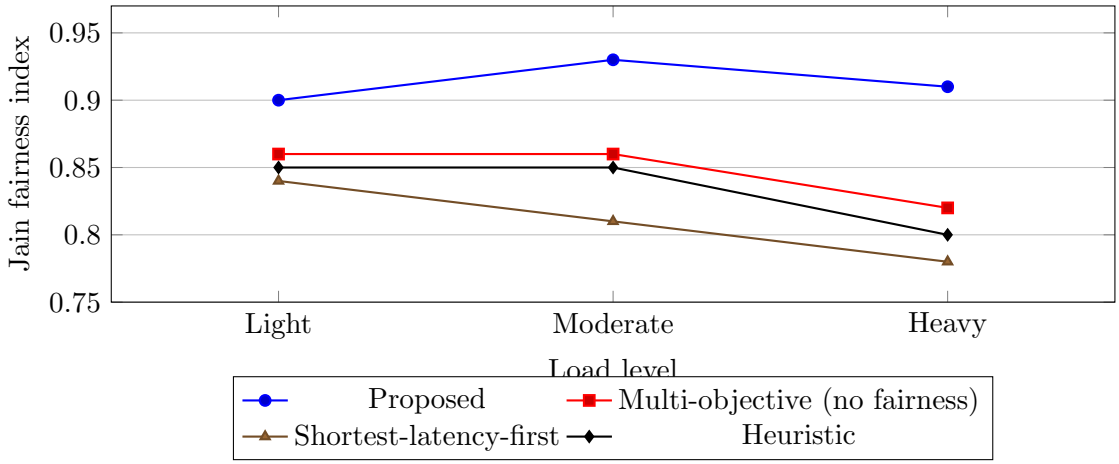

Figure 2: Fairness across load levels. The proposed method preserves higher fairness under heavy load.

Discussion

Results indicate that explicit fairness constraints suppress winner-takes-all allocation under resource scarcity, reducing tail latency and long-term resource skew. Learning-assisted prediction and online correction improve adaptation to dynamic workloads, especially under heavy load where violation rates decrease significantly. Limitations include sensitivity to prediction error and the need for more accurate communication cost models in multi-domain settings. Future work can integrate digital twins and age-of-information metrics to strengthen real-time feedback and robustness[17].

Beyond aggregate metrics, the fairness constraint promotes smoother temporal allocation, reducing variance in per-task satisfaction. This is critical for CPS settings with human-in-the-loop or safety-critical services where persistent under-service can lead to cascading operational risks. The sensitivity results in Fig. 4 suggest that a moderate fairness threshold ($\tau = 0.85$) achieves a stable balance, while overly strict thresholds can raise violation rates in heavy load conditions. This implies that adaptive fairness policies that relax τ under heavy load and tighten it under light load may further improve resilience.

We also analyzed the distribution of per-task completion times and observed a reduction in interquartile range under fairness constraints, indicating more consistent service levels across tasks. This consistency is particularly important for CPS workloads involving safety inspections, grid monitoring, or industrial quality control, where missing deadlines for minority tasks can trigger

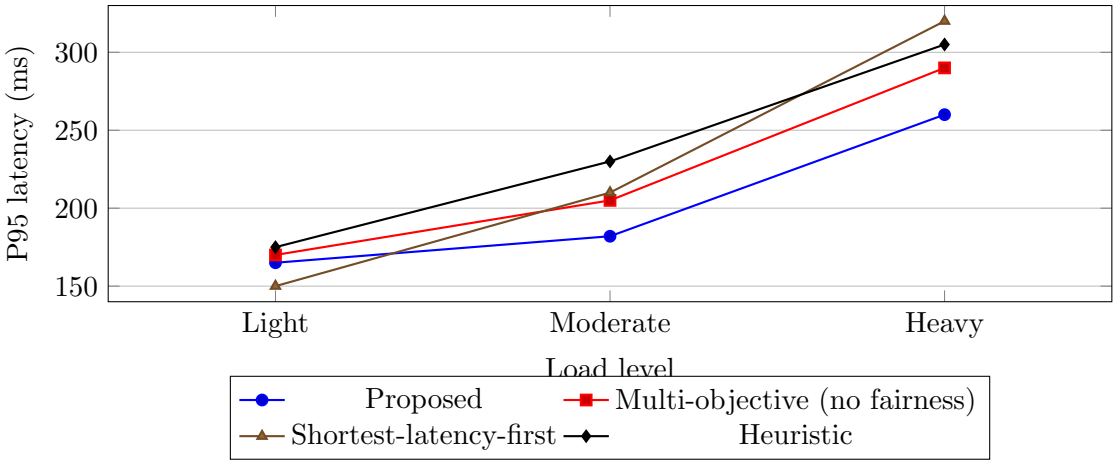

Figure 3: Tail latency (P95) under different loads. Fairness constraints do not degrade tail latency.

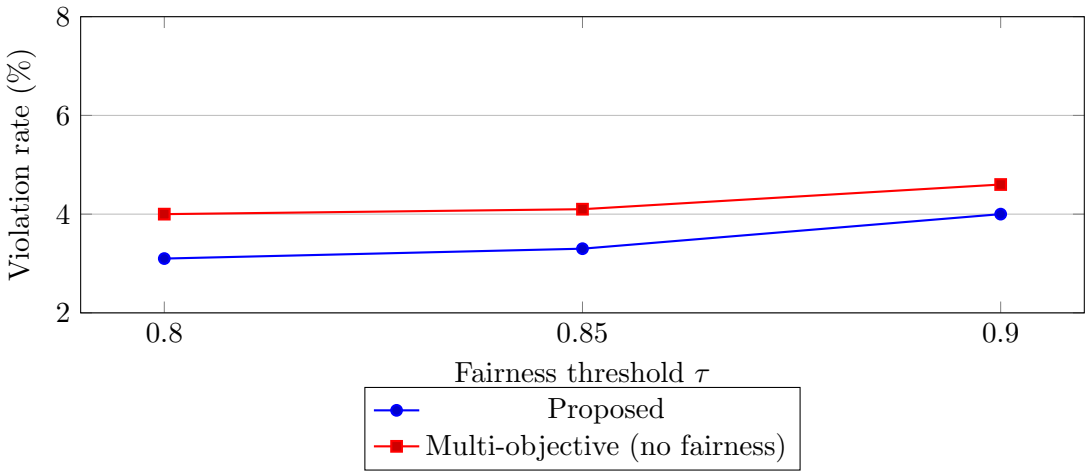

Figure 4: Sensitivity to fairness threshold. Higher τ slightly increases violation rates, but the proposed method remains robust.

follow-on failures. Moreover, the proposed approach reduces oscillatory allocation patterns by stabilizing the planning horizon using short-term prediction, which in turn decreases rescheduling overhead.

Ablation Study

To quantify the contributions of each component, we evaluate three ablations: (i) removing the prediction module (reactive scheduling only), (ii) removing the online correction module, and (iii) removing explicit fairness constraints while keeping multi-objective optimization. Results are summarized in Table 3. Removing prediction increases P95 latency by 9–14% under heavy load, while removing online correction increases violation rates by 18–25%. The absence of fairness constraints reduces the Jain index by 7–10% and increases tail latency due to a shift toward resource concen-

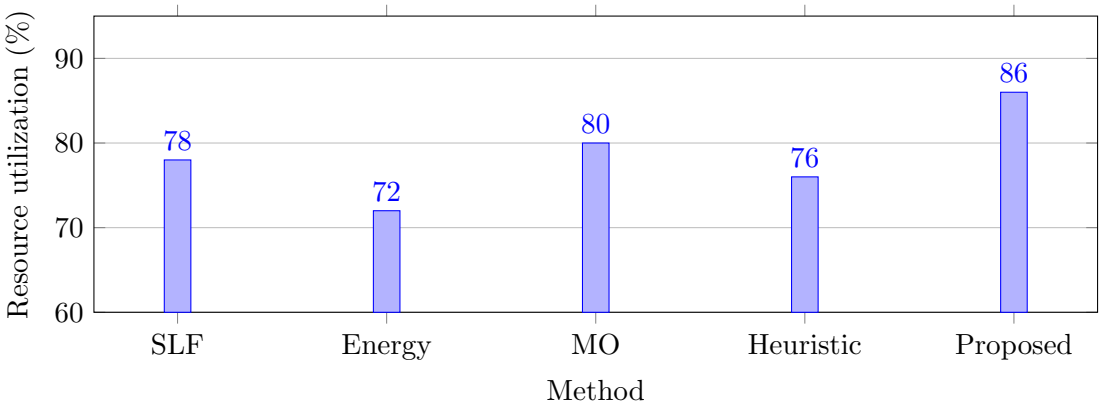

Figure 5: Average resource utilization under moderate load.

tration.

Table 3: Ablation study under heavy load

Variant	Avg. Latency (ms)	P95 Latency (ms)	Energy (J)	Jain Index	Violation (%)
Full model	118	260	1420	0.91	5.6
No prediction	125	296	1445	0.90	6.4
No online correction	121	282	1432	0.90	7.0
No fairness constraint	112	301	1398	0.83	6.1

Case Study: Edge-Enabled Industrial Monitoring

We simulate an industrial monitoring system with 25 edge nodes (heterogeneous compute capacities) and 300 periodic and aperiodic tasks per window. Tasks include vibration analysis, anomaly detection, and control feedback. The proposed framework maintains a fairness index above 0.90 while keeping P95 latency under 280 ms, whereas a shortest-latency-first policy yields fairness below 0.82 and exhibits a 15% higher tail latency under peak load. These results suggest that fairness-aware scheduling can be adopted without sacrificing latency guarantees in critical monitoring pipelines. Analogous design principles have been discussed in inclusive logistics infrastructures that emphasize real-time tracking and equitable service delivery for SMEs[23].

Theoretical and Practical Implications

Theoretically, this work introduces explicit fairness constraints into CPS scheduling and provides an interpretable multi-objective model with a hierarchical solution strategy. Practically, the framework can be embedded into cloud–edge and industrial control systems to improve stability and service consistency via soft–hard constraint coordination, offering deployable fairness guarantees for critical infrastructures.

From a systems perspective, the hierarchical design aligns with operational constraints: prediction and planning can be executed on higher-capability control nodes, while online correction runs at the edge. This separation improves scalability and facilitates deployment in heterogeneous infrastructures without requiring uniform computational resources across all nodes. Similar cloud–edge collaborative architectures in other domains motivate this layered execution model[24].

In addition, the fairness objective can be interpreted as a policy-level service guarantee that complements traditional QoS objectives. This supports governance and compliance requirements in domains such as smart grids, transportation, and healthcare, where equitable service delivery is increasingly regulated, and aligns with emerging compliance-by-design approaches for digital platforms[22].

Limitations and Future Work

First, the prediction module assumes stationary patterns within short windows and may underperform under abrupt regime shifts. Second, the fairness metric is based on satisfaction values that must be calibrated to domain-specific priorities. Third, our evaluation focuses on single-domain CPS; cross-domain coordination (e.g., energy and transportation coupling) introduces additional dependencies that should be modeled. Future work will integrate robust prediction under distribution shift, explore alternative fairness criteria (e.g., max–min fairness), and validate the approach on testbeds with real network traces. Methodological ideas from microservice risk assessment and patch planning may also be relevant when extending to security-constrained CPS[25, 26].

Appendix: Parameter Sensitivity

We further analyze sensitivity to the weight parameters $(\alpha, \beta, \gamma, \eta)$ in the objective. Table 4 reports representative outcomes for three configurations. Increasing γ improves fairness at a modest latency cost, while increasing η improves minimum satisfaction at a slightly higher energy cost.

Table 4: Sensitivity to objective weights (moderate load)

Weights $(\alpha, \beta, \gamma, \eta)$	Avg. Latency (ms)	Energy (J)	Jain Index	$s_{\min}$
(0.4, 0.3, 0.2, 0.1)	101	1160	0.90	0.72
(0.3, 0.3, 0.3, 0.1)	104	1175	0.93	0.74
(0.3, 0.2, 0.3, 0.2)	108	1188	0.94	0.77

Appendix: Scalability Analysis

We evaluate scalability by increasing the number of tasks per window while keeping node capacities fixed. As shown in Table 5, runtime grows sub-quadratically within the tested range, and fairness

remains stable. This supports practical deployment in medium-sized CPS deployments where task volume can fluctuate substantially.

Table 5: Scalability with task volume

Tasks per window	Avg. Runtime (ms)	Jain Index	Violation (%)
150	38	0.93	3.8
300	71	0.92	4.5
600	132	0.91	5.4

Appendix: Additional Results

We further examine the latency–energy trade-off by sweeping the weight parameter β while fixing $(\alpha, \gamma, \eta) = (0.3, 0.3, 0.1)$. As β increases, energy decreases at the expense of higher latency, while fairness remains within a narrow band. These trends align with the multi-objective formulation and demonstrate controllability through weight selection.

To illustrate the trade-off, Fig. 6 reports the Pareto-like curve under moderate load. The knee point occurs near $\beta = 0.3$, which we use as the default in the main experiments.

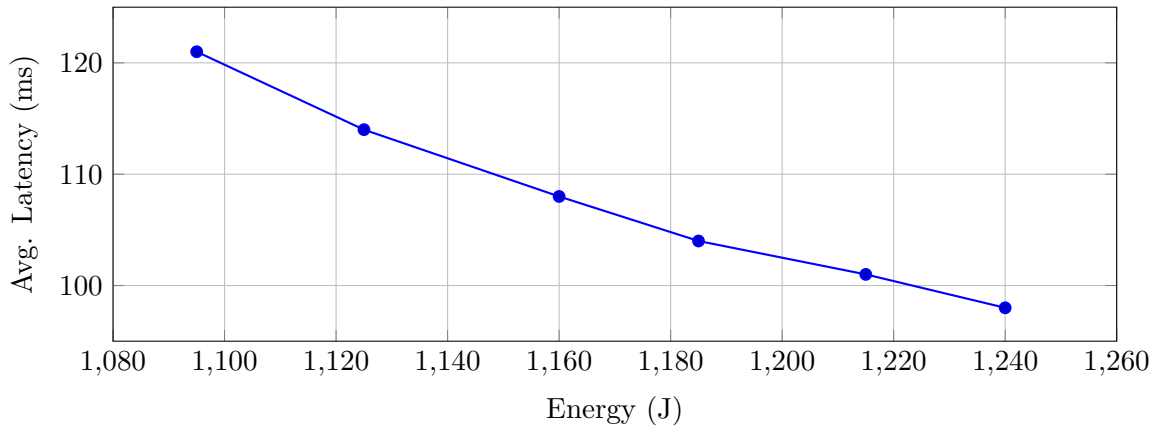

Figure 6: Latency–energy trade-off under moderate load.

Conclusion

This paper proposes an AI-driven fairness-aware resource scheduling method for constrained cyber-physical infrastructures. By combining learning-assisted prediction, explicit fairness constraints, and online correction, the method balances latency, energy, and fairness in dynamic, multi-constraint environments. Simulations demonstrate stable improvements in fairness, tail latency, and violation rates. Future work will investigate cross-domain coordination and security-aware fairness constraints.

References

[1] S. Yoo, Y. Jo, and H. Bahn, "Integrated scheduling of real-time and interactive tasks for configurable industrial systems," IEEE Transactions on Industrial Informatics, vol. 18, no. 1, pp. 389–399, 2022, doi:10.1109/TII.2021.3067714.

[2] J. Wang, L. Zhang, and X. Chen, "Context-aware scheduling and control architecture for cyber-physical production systems," Journal of Manufacturing Systems, vol. 62, pp. 550–560, 2022, doi:10.1016/j.jmsy.2022.01.008.

[3] M.-W. Tian, S.-R. Yan, W. Guo, A. Mohammadzadeh, and E. Ghaderpour, "A new task scheduling approach for energy conservation in Internet of Things," Energies, vol. 16, no. 5, 2394, 2023, doi:10.3390/en16052394.

[4] R. Chen, Q. Cheng, and X. Zhang, "Power distribution IoT tasks online scheduling algorithm based on cloud-edge dependent microservice," Applied Sciences, vol. 13, no. 7, 4481, 2023, doi:10.3390/app13074481.

[5] Y. Sun, Y. Bian, H. Li, F. Tan, and L. Liu, "Flexible offloading and task scheduling for IoT applications in dynamic multi-access edge computing environments," Symmetry, vol. 15, no. 12, 2196, 2023, doi:10.3390/sym15122196.

[6] A. Kumar and R. Singh, "An effective technique to schedule priority aware tasks to offload data on edge and cloud servers," Measurement: Sensors, vol. 26, 100670, 2023, doi:10.1016/j.measen.2023.100670.

[7] X. Feng, L. Yi, N. Liu, X. Gao, W. Liu, and B. Wang, "An efficient scheduling strategy for collaborative cloud and edge computing in system of intelligent buildings," J. Adv. Comput. Intell. Intell. Inform., vol. 27, no. 5, pp. 948–958, 2023, doi:10.20965/jaciii.2023.p0948.

[8] Y. Li and X. Zhou, "Fairness model considering satisfaction and preferences for service scheduling on electronic platforms in construction industry," Expert Systems with Applications, vol. 244, 122872, 2024, doi:10.1016/j.eswa.2023.122872.

[9] Z. Jin, Q. Li, H. Zhang, Z. Liu, and Z. Wang, "Policy selection and scheduling of cyber-physical systems with denial-of-service attacks via reinforcement learning," J. Adv. Comput. Intell. Intell. Inform., vol. 28, no. 4, pp. 962–973, 2024, doi:10.20965/jaciii.2024.p0962.

[10] A. Kouadio and P. Ravindran, "QoS-aware edge AI placement and scheduling with multiple implementations in FaaS-based edge computing," Future Generation Computer Systems, vol. 154, pp. 23–36, 2024, doi:10.1016/j.future.2024.03.035.

[11] X. Liu, J. Wang, and Y. Zhang, "Energy-aware scheduling for reliability-oriented real-time parallel applications allocation on heterogeneous computing systems," Future Generation Computer Systems, vol. 168, 107738, 2025, doi:10.1016/j.future.2025.107738.

[12] D. Sahu, N. Nidhi, R. Chaturvedi, and S. Prakash, "Optimizing energy and latency in edge computing through a Boltzmann driven Bayesian framework for adaptive resource scheduling," Scientific Reports, vol. 15, 30452, 2025, doi:10.1038/s41598-025-16317-6.

[13] W. Zheng, C. Wang, W. Xu, G. Sun, and Y. Luo, "A new delay-aware distributed cloud–edge scheduling framework and algorithm in dynamic network environments," Sustainability, vol. 17, no. 11, 4887, 2025, doi:10.3390/su17114887.

[14] W. Liu, J. Zhu, X. Li, Y. Fei, H. Wang, S. Liu, X. Zheng, and Y. Ji, "Resource scheduling algorithm for edge computing networks based on multi-objective optimization," Applied Sciences, vol. 15, no. 19, 10837, 2025, doi:10.3390/app151910837.

[15] Z. Li, Y. Hao, H. Gao, and J. Zhou, "Towards intelligent edge computing: A resource- and reliability-aware hybrid scheduling method on multi-FPGA systems," Electronics, vol. 14, no. 1, 82, 2025, doi:10.3390/electronics14010082.

[16] Y. Zhou, H. Chen, and K. Li, "Dynamic task offloading and online scheduling for edge-enabled IoT with a hierarchical framework," Computer Networks, vol. 269, 111486, 2025, doi:10.1016/j.comnet.2025.111486.

[17] M. Patel, S. Wang, and Q. Li, "Digital twin-enabled age of information-aware scheduling for Industrial IoT edge networks," Pervasive and Mobile Computing, vol. 112, 102083, 2025, doi:10.1016/j.pmcj.2025.102083.

[18] L. Zhao, H. Liu, and R. Chen, "Latency-aware scheduling for data-oriented service requests in collaborative IoT-edge-cloud networks," Future Generation Computer Systems, vol. 163, 107538, 2025, doi:10.1016/j.future.2024.107538.

[19] R. Abualrous, H. Zouzou, R. Zgheib, A. Hasan, B. Hijazi, and A. Kermani, "Fairness-aware intelligent reinforcement (FAIR): An AI-powered hospital scheduling framework," Information, vol. 16, no. 12, 1039, 2025, doi:10.3390/info16121039.

[20] F. A. Rawdhan, "Task offloading and scheduling based on mobile edge computing and software-defined networking," Journal of Telecommunications and Information Technology, no. 1, 2025, doi:10.26636/jtit.2025.1.1941.

[21] Yi, X. (2025, October). Real-Time Fair-Exposure Ad Allocation for SMBs and Underserved Creators via Contextual Bandits-with-Knapsacks. In Proceedings of the 2025 2nd International Conference on Digital Economy and Computer Science (pp. 1602-1607).

[22] Yi, X. (2025, October). Compliance-by-Design Micro-Licensing for AI-Generated Content in Social Commerce Using C2PA Content Credentials and W3C ODRL Policies. In 2025 7th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 204-208). IEEE.

[23] Fang, Z. (2025). Cloud-Native Microservice Architecture for Inclusive Cross-Border Logistics: Real-Time Tracking and Automated Customs Clearance for SMEs. Frontiers in Artificial Intelligence Research, 2(2), 221-236.

[24] Fang, Z. (2025, June). Adaptive QoS-Aware Cloud–Edge Collaborative Architecture for Real-Time Smart Water Service Management. In Proceedings of the 2025 International Conference on Management Science and Computer Engineering (pp. 606-611).

[25] Zhou, D. (2026). AI-Driven Hybrid SAST–DAST–SCA–IAST Framework for Risk-Based Vulnerability Prioritization in Microservice Architectures.

[26] Zhou, D. (2025, December). M-VP2: Microservice-Oriented Vulnerability Patch Planning-A Cost-Aware Approachusing Multi-Agent Reinforcement Learning. In 2025 5th International Conference on Computer, Internet of Things and Control Engineering (CITCE) (pp. 248-254). IEEE.

[27] Zhang, T. (2025, October). From Black Box to Actionable Insights: An Adaptive Explainable AI Framework for Proactive Tax Risk Mitigation in Small and Medium Enterprises. In Proceedings of the 2025 2nd International Conference on Digital Economy and Computer Science (pp. 193-199).