Journal of Applied Automation Technologies

Research Paper

Adaptive Job Shop Scheduling Based on Proximal Policy Optimization

Edward Mazur[1,*] and Stefan Kołodziej[1]

[1] Faculty of Mechanical Engineering, Casimir Pulaski University of Radom, 26-600 Radom, Poland
*Corresponding author: edward.m@uniwersytetradom.pl

Abstract. Computer-based adaptive scheduling is currently being used in production to meet Industry 4.0 objectives for flexibility, resilience, and high efficiency. This research addresses the dynamic job shop scheduling issue (JSSP) using a reinforcement learning framework based on the Proximal Policy Optimisation (PPO) method. Its objective is to create a comprehensive, self-sufficient intelligent schedule-optimization system that can adapt to changes in the real working environment. The first kind employs neural policy networks for continuous optimisation under uncertainty and reformulates JSSP as a Markov Decision Process (MDP). To mimic real-world industrial variations, a comprehensive experimental platform featuring digital twins, real-time event injection, and high-fidelity simulation has been developed. Compare the PPO scheduler with baselines from 30 different trials using the evolutionary algorithm, deep Q-network, and earliest due date (EDD). According to the findings, machine utilisation is higher than 92%, the average makespan is lowered to 827 units for both GA and EDD, and complex conditions result in a shorter scheduling response latency (averaging 0.54 seconds). Additionally, the suggested framework reduced the recovery period following disruptions to less than 20-time units and maintained the lost-job ratio at less than 2%. According to the aforementioned research, the system's overall stability and efficiency have increased with the addition of PPO and dynamic allocation; as a result, it can be applied to new manufacturing platforms.

Keywords: *Reinforcement Learning, Proximal Policy Optimization, Job Shop Scheduling Problem, Intelligent Manufacturing*

Received on 14 November 2024, Accepted on 17 March 2025, Published on 28 March 2025

Introduction

High-end production systems are required as smart manufacturing develops, and job shop scheduling continues to be a major issue [1]. The complexity of production has increased due to the advancement of digitalisation, comprehensive automation, and the proliferation of cyber-physical systems; as a result, schedulers must now manage many resources and make real-time adjustments to demand changes in order to maximise efficiency [2]. The need for on-demand production, short lead times, and highly customised orders exacerbate the aforementioned issues, making the drawbacks of conventional optimisation techniques more noticeable [3]. Rule-based heuristics, mixed-integer programming, and metaheuristic algorithms including genetic algorithms, tabu search, and ant colony optimisation for job shop scheduling have all been the subject of extensive research over the last few decades [4]. While many of the conventional approaches have proven reliable in a typical setting, they are comparatively less appropriate for managing issues like equipment breakdown, an unexpected surge in orders, or a significant shift in production conditions [5]. Additionally, the industrial environment's erratic and unpredictable features frequently fall short of the requirements for a fixed-schedule plan to some degree [6]. Traditional offline scheduling models are no longer appropriate due to the recent trend of the digital twin concept and real-time industrial analytics, and there is now a greater need for adaptive and autonomous intelligent scheduling solutions [7].

Simultaneously, the field of machine learning has started to investigate reinforcement learning (RL) for dynamic industrial decision-making and has seen success in a number of domains, including production control and resource allocation [8]. By obtaining state-action feedback from an unpredictable and dynamic environment, reinforcement learning (RL), as opposed to a conventional explicit model of the system, can constantly improve a schedule via experience [9]. In complicated, high-dimensional non-linear decision-making issues in reinforcement learning, Proximal Policy Optimisation (PPO), a representative policy gradient-based approach, has demonstrated strong performance [10]. In both simulated and real-world industrial settings, PPO exhibits strong learning stability and convergence, satisfying the needs of factory scheduling research in terms of adaptability, learning efficiency, and safety [11]. First, PPO-based studies on online adaptive scheduling and factory dispatching have shown promise in responding to unforeseen disruptions and varying workloads, demonstrating the potential of reinforcement learning (RL) for improving shop floor intelligence [12]. Connecting the emerging theories of reinforcement learning to the real-time, disruptive conditions in contemporary industrial job shops, however, continues to be a significant challenge [13]. Adaptive scheduling techniques must be mathematically feasible, learnable, and simple to incorporate into cyber-physical production systems in order to be used in industry [14]. An all-encompassing adaptive scheduling framework that can manage uncertainties at the machine, order, and process levels is currently necessary due to the growing significance of Industry 4.0 [15].

This research investigates a novel adaptive scheduling model for job-shop production in the form of enhanced PPO reinforcement learning based on the aforementioned observations. Create a strong learning framework tailored to the operational features of smart factories by methodically reformulating the job shop problem as a Markov Decision Process. Our approach, which is data-driven and real-time, aims to provide both theoretically sound concepts and workable enhancements for intelligent shop floor scheduling. An extremely flexible, all-weather adaptive scheduling system for the new era of digital manufacturing will be presented in this presentation.

Related Work

Some research has been done on discrete job shop scheduling problems (JSSPs) as common scenarios for production optimisation. The earliest due date, lowest processing time, and critical ratio were among the dispatching rules and combinatorial techniques that were the primary focus of the initial study [16]. Although these heuristic approaches had issues with scale and system uncertainty, they provided workable answers in regulated and predictable shop-floor conditions [17]. Metaheuristic algorithms have gained increasing interest due to the complexity and diversity of industrial activities; large-scale JSSP instances have made extensive use of Genetic Algorithms, Tabu Search, Ant Colony Optimisation, and Simulated Annealing [18]. In order to decrease makespan and enhance resource utilisation in a fixed environment, some hybrid variants of the global search approach have been applied globally [19]. They won't function effectively, though, if there are frequent machine failures, modifications to real-time orders, or unforeseen bottlenecks [20]. The computational cost of repeatedly re-optimizing the entire schedule has also increased, making it less practical for shop floors that are truly dynamic [21].

The application of machine learning and reinforcement learning (RL) to production scheduling has attracted a lot of attention lately. RL was first demonstrated to be applicable to basic sequence and resource-allocation problems under uncertainty by Q-Learning and SARSA variants [22]. In order to extend the viability of this approach to high-dimensional, partially observable scheduling problems, Deep Q-Networks (DQN) have integrated deep neural networks, offering the possibility of end-to-end adaptive scheduling [23]. In a complex industrial setting, DQN and other value-based reinforcement learning algorithms are unstable and impractical for long-term use [24]. Asynchronous Advantage Actor-Critic (A3C) and Deep Deterministic Policy Gradient (DDPG) are two examples of actor-critic frameworks and policy-gradient techniques that have enhanced convergence and added continuous-action capability [25]. Because of its sample efficiency, policy improvement guarantees, and learning stability, Proximal Policy Optimisation (PPO) has demonstrated promising possibilities for job shop scheduling under the aforementioned conditions [26]. In order to increase reactivity and lower solution volatility under various operating situations, a number of studies have integrated PPO or its variations into the manufacturing schedule of flexible job shops and multi-machine settings for adaptive task dispatching [27].

The current research still has certain shortcomings. The majority of classical and metaheuristic approaches are unable to adapt to novel or unidentified shop floor problems and necessitate numerous domain-specific parameter adjustments [28]. Despite being data-driven and adaptive, RL models have a risk of domain transfer because they frequently rely on enhanced simulation environments rather than actual production systems [29]. The main issues with using academic reinforcement learning (RL) benchmarks in practical industrial decision-support systems are sample inefficiency, long learning times, interpretability issues, and inadequate adaptability to real-time industrial situations [30]. Rarely do research on reinforcement learning (RL) in factories carry out trials under the untidy conditions of real production, such as machine failures, human intervention, order cancellations, etc., and instead report good performance based on ideal or theoretical factories. The lack of modules and integrations that are compatible with current industrial execution and cyber-physical systems further widens the gap between the aspirations for algorithms and their use. The aforementioned issues point to shortcomings in the system's robustness, generalisation, and quick adaptation to changes in the online industrial environment.

This study differs from earlier research. Instead of employing manually created heuristics, first expand the integration of PPO into a scheduling framework that is directly mapped to the industrial job shop Markov Decision Process (MDP). Second, only artificial and steady scenarios are often examined in scheduling research; instead, the suggested method will be tested against a variety of real, fluctuating shop floor variables, such as shifting sequences, unforeseen additions, and equipment failures. Third, we will demonstrate the practical viability of industrial transformation while presenting the theoretical underpinnings and empirical validation of adaptive and robust scheduling, reporting benchmarks and real-world performance.

Methodology

Job Shop Scheduling Problem Formulation

Within the job shop environment, n jobs $(J_1, J_2, \dots, J_n)$ must be scheduled on m machines ($M_1, M_2, \dots, M_m$), where each job consists of a strictly ordered sequence of operations. Each operation is bound to a designated machine and must comply with both resource and technological constraints intrinsic to practical manufacturing systems.

A feasible schedule must first satisfy operation precedence. For any job J_i with operations ($O_{i1}, O_{i2}, \dots, O_{iN_i}$), the start time of operation O_{ij} cannot precede the completion of its immediate predecessor:

$$S_{ij} \geq C_{i(j-1)}, 1 \leq i \leq n, 2 \leq j \leq N_i \quad \text{Eq.(1)}$$

Additionally, to avoid machine conflict, at any point, no two operations can overlap on the same resource. For any operations O_{ij}, O_{kl} assigned to machine M_q :

$$S_{ij} \geq C_{kl} \text{ or } S_{kl} \geq C_{ij} \quad \text{Eq.(2)}$$

The primary objective is to minimize the makespan-i.e., the maximum completion time across all jobs:

$$\min_{\pi} \max_{i} C_{iN_i} \quad \text{Eq.(3)}$$

Here, π denotes a scheduling policy, C_{iN_i} is the completion time of the last operation of job i, encapsulating the practical need to enhance throughput and resource efficiency. This concise formalization provides both the theoretical foundation and operational constraints that inform the subsequent adaptive scheduling framework.

PPO for Adaptive Scheduling

The adaptive job shop scheduling problem can be represented as a Markov Decision Process, which enables continuous policy refinement under changing industrial conditions. At each decision epoch t, the environment state s_t fully characterizes the machine status, job progress, and operative constraints.

The system evolves through a transition function, driven by the scheduling action a_t :

$$s_{t+1} = \mathcal{T}(s_t, a_t) \quad \text{Eq.(4)}$$

A scheduling agent aims to determine a stochastic policy $\pi_\theta(a_t \mid s_t)$, parameterized by neural network weights θ, which maps the observed state to a probability distribution over feasible dispatch action.

To incentivize high-quality scheduling decisions, the reward for each step is formulated to penalize makespan growth, tardiness, and idle time:

$$r_t = -\alpha \Delta C_{\max} - \beta L_t - \gamma I_t \quad \text{Eq.(5)}$$

Here, $\Delta C_{\max}$ is the incremental makespan, L_t is total tardiness, and I_t quantifies overall idle machine time, each weighted by a corresponding hyperparameter.

Policy updates in Proximal Policy Optimization maximize a clipped surrogate objective to ensure learning stability:

$$L(\theta) = \mathbb{E}_t\left[\min\left(\rho_t \hat{A}_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \quad \text{Eq.(6)}$$

where ρ_t is the probability ratio between new and old policies, ϵ is a preset threshold, and $\hat{A}_t$ is the advantage estimator at t.

The agent's long-term goal is to maximize the expected cumulative reward over an entire scheduling episode:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \quad \text{Eq.(7)}$$

with γ denoting the discount factor that balances immediate versus future benefits.

This formalization allows the PPO-driven scheduler to iteratively refine its policy through direct interaction with the manufacturing environment, producing robust strategies capable of responding to real-world shop floor variability.

Framework Architecture and Process

An appropriate architecture must be developed to link the discrete character of JSSP with the continuous adaptive capabilities of advanced reinforcement learning in order to apply Proximal Policy Optimisation (PPO) for job shop scheduling. The environment encoder, reward calculation engine, policy and value networks, and environment interface for real-time scheduling execution are the components of the suggested system.

The environment encoder creates a high-dimensional state vector that a neural network can use after processing raw shop floor data, such as machine status, task queue, processing route, and event log. This data's organization reveals both the fixed structure and the operational changes that will occur.

The policy network, parameterized by weights θ, receives the state vector and outputs a probability distribution across feasible scheduling actions. The value network, parameterized separately, estimates the expected cumulative return from any given state, providing the baseline for advantage calculation and policy improvement.

The decision process at each time step proceeds as follows. The current environment state s_t is encoded and fed into the policy network, which samples or selects an action a_t according to the learned dispatch probability. The environment advances to the next state based on this action and produces an immediate reward reflecting the efficiency and quality of the chosen schedule.

The neural policy for dispatching can be expressed as:

$$a_t \sim \pi_\theta(\cdot \mid s_t) \quad \text{Eq.(8)}$$

The overall architecture employs experience replay and mini-batch updates to iteratively refine both policy and value networks, using feedback from the reward computation and true environment response. Periodic evaluation against baseline heuristics and disruptive events serves to benchmark and guide learning progress.

The final scheduling policy is realized through the argmax operation over the learned policy outputs at inference time:

$$a_t^* = \underset{a \in \mathcal{A}(s_t)}{\text{argmax}} \pi_\theta(a \mid s_t) \quad \text{Eq.(9)}$$

This structured integration ensures that schedule generation is both data-driven and deeply responsive to real-time changes in shop conditions.

In Figure 1, the overall system workflow is illustrated-from job and machine status input, state encoding, PPO-based policy generation, to executable scheduling commands fed back to the digital/physical shop floor. The

architecture emphasizes modularity and the seamless flow of information across learning, evaluation, and execution loops.

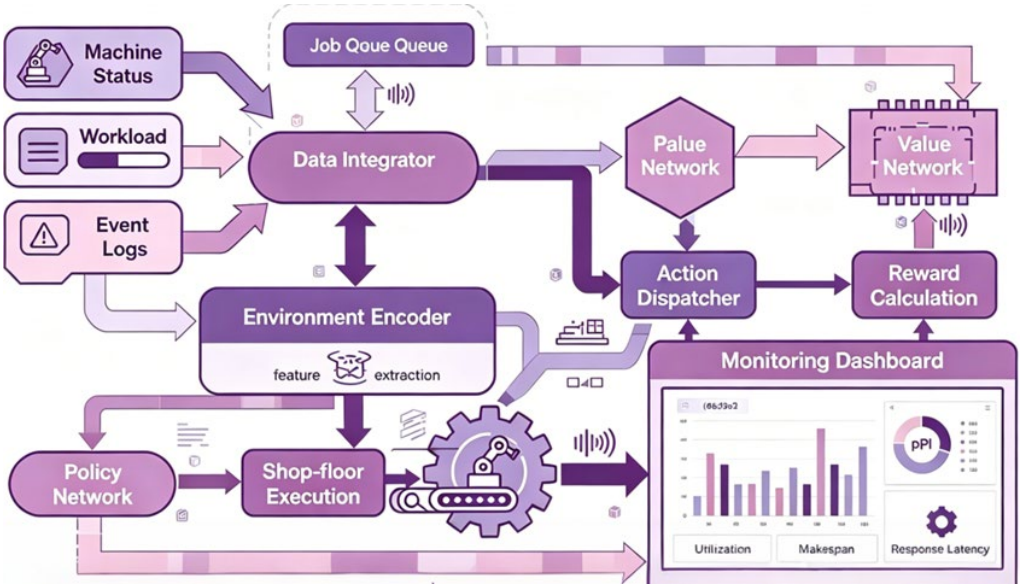

Figure 1. Architecture of the adaptive job shop scheduling system with environment encoding, PPO networks, and real-time dispatch feedback

Experimental Setup and Evaluation

Experimental Environment and Setup

A virtual simulation environment and hardware-in-the-loop (HIL) test capabilities were built in tandem to more accurately confirm the real performance of the new PPO-based adaptive scheduling system. Two Intel Xeon processors with 256GB of RAM and NVIDIA Tesla V100 GPUs for high-capacity neural policy model training make up the calculating backbone. This backbone replicates the temporal and spatial complexity of an industrial-scale production site and is directly connected to a virtualised reproduction of a contemporary job shop.

The twenty diverse machines in the simulated plant have customisable routes and span a wide range of working modes. In order to replicate real-world shop-floor fluctuations, a synthetic job dataset was developed based on the historical order features of a precision car manufacturing factory. Parameters including task arrival times, routing permutations, operation durations, and due dates were randomly chosen. One of the first 150 jobs in the queue will be linked to the scheduler in each experiment; these jobs will have varying operating windows and probabilistic start times.

A testbed that simulates urgent re-sequencing due to high-priority job insertions, introduces random breakdown events, takes workforce allocation limits into account, and connects to a digital twin platform will be set up. The environment logic is controlled by a custom extension developed on top of OpenAI Gym, while the reinforcement learning agent and all baselines are implemented in Python using PyTorch on the software end. To guarantee that the simulated algorithmic decision cycle and the real-world execution interval are strictly time-aligned, policy inference and control signals are sent back and forth between the RL agent and the digital twin environment.

To establish a solid comparative basis, three industrial scheduling baselines will be employed: a deep Q-network (DQN) reinforcement learning benchmark, a metaheuristic genetic algorithm, and a rule-based dispatching algorithm (Earliest Due Date, EDD). The initial conditions, resource settings, and work release profiles are the same for all competing techniques.

The experimental system combines virtualised shop-floor dynamics, hardware resources, and a reinforcement-learning "brain" in a closed cyber-physical scheduling loop, as seen in Figure 2. The structure's modularity will facilitate live industrial data connections and offer a foundation for future expansion.

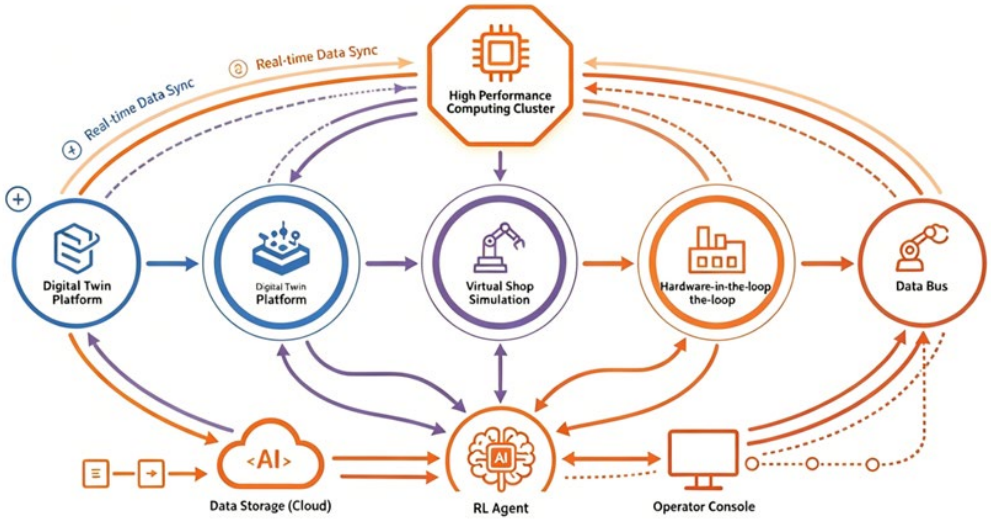

Figure 2. Experimental platform architecture with distributed computing, digital twin, and RL-based adaptive scheduling

Evaluation Metrics and Baselines

In a large-scale work shop system, numerous indices will be used to evaluate the impact and response time. Makespan, the primary indicator, measures the duration between the release of the first work and the completion of its final operation. The average machine utilisation rate, total tardiness, average flow time, and idle time penalty were also gathered during the experiment for a more thorough performance analysis.

Makespan ($C_{\max}$) is computed as follows:

$$C_{\max} = \max_i \left(C_{i,\,\text{final}} \right) \qquad \text{Eq.(10)}$$

where $C_{i,\,\text{final}}$ represents the completion time of the last operation in job i.

To provide a nuanced evaluation of scheduling adaptability, the system tracks average normalized tardiness ($\mathcal{T}_{\text{avg}}$), calculated as:

$$\mathcal{T}_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} \frac{\max\left(0, C_{i,\,\text{final}} - d_i\right)}{d_i} \qquad \text{Eq.(11)}$$

Here, d_i is the due date for job i, and n is the total number of jobs in the experimental batch.

Industrial regulations won't be broken because of optimisation leakage because all benchmarked methods are run with the same hyperparameters and only tuning has been done during the pilot phase. The DQN employs target network synchronisation and double Q-learning for stability in a non-stationary job shop setting, while the crossover and mutation rates of the genetic algorithm are chosen to achieve a good balance in convergence.

Test Procedure and Dynamic Scenario Design

Create a test strategy for the adaptive scheduling system in both a regular and a high-stress job shop. All relevant data will be gathered in detail during the sequential execution of the whole set of inspections.

In order to replicate the statistical variety and order volatility of a high-mix manufacturing facility, the system is randomly but consistently setup with machine states and work queues. The simulation clock moves in stages of a predetermined control time during each run, with each step representing a small-scale manufacturing cycle. In order to influence the upcoming task assignments and routing decisions, the RL agent selects dispatching actions at each time step based on the current state encoding and inferred shop floor dynamics.

In the two primary scenarios, test baseline and PPO-based approaches. In the base case, job arrivals precisely follow the historical distribution and all resources are considered to function normally. Currently, the digital twin's event module introduces a few sporadic incidents, like an unexpected equipment failure, a new order request, and an urgent job revision. The scheduling agent must react quickly to a stochastic disturbance set, recover, and reoptimize job-to-machine assignment within a tight real-time deadline.

The impact of dynamic disruptions is systematically quantified through advanced robustness indices and recovery time analytics. The first metric, mean time to reschedule (μ_{rt}), measures the interval between a disruptive event (e.g., a randomly withdrawn resource) and restoration of feasible schedules:

$$\mu_{\mathrm{rt}} = \frac{1}{K}\sum_{k=1}^{K}\left(t_k^{\mathrm{restore}} - t_k^{\mathrm{event}}\right) \quad \text{Eq.(12)}$$

where K indexes the total number of disruption events.

A second metric, adaptive efficiency (E_{adap}), captures the relative post-disruption degradation compared to pre-event steady state:

$$E_{\mathrm{adap}} = 1 - \frac{C_{\max}^{\mathrm{post}} - C_{\max}^{\mathrm{pre}}}{C_{\max}^{\mathrm{pre}}} \quad \text{Eq.(13)}$$

Here, $C_{\max}^{\mathrm{pre}}$ and $C_{\max}^{\mathrm{post}}$ are the makespan immediately before and after a perturbation.

To further quantify systemic resilience, an event-integrated tardiness accumulation score (S_{ETAS}) is tracked over all jobs and disruptions:

$$S_{\mathrm{ETAS}} = \sum_{i=1}^{n}\sum_{e=1}^{E}\max\left(0, C_{i,\,\mathrm{final}}^{(e)} - d_i^{(e)}\right) \quad \text{Eq.(14)}$$

where E denotes the total number of disturbance events throughout the episode.

Figure 3 clarifies the experimental control logic, showing the interplay between job release, event injection, RL-driven decision cycles, and performance logging over time. This workflow encapsulates the closed-loop, data-driven nature of adaptive scheduling under operational uncertainty.

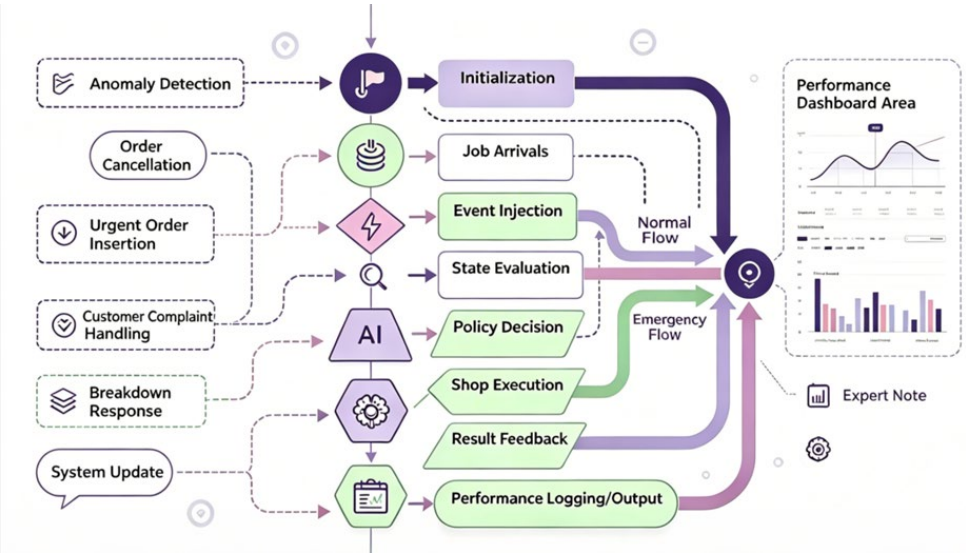

Figure 3. Experimental workflow for dynamic scenario evaluation with disruption injection and adaptive scheduling

Results and Analysis

Performance Comparison

In a real-world work shop context, the two scheduling systems will differ in terms of overall efficiency and resource utilisation. The three-performance metrics gathered from 30 distinct trials for each method are displayed in Figure 4. The throughput capacity during light fluctuation and at rest is displayed by the makespan findings in Figure 4(a). With a smaller standard deviation band under shop floor volatility, the PPO-based adaptive scheduler outperformed the EDD, GA, and DQN baselines, lowering the average makespan from 982 (EDD) and 951 (GA) to 827.

Since machine utilisation is comparatively high, learning-based approaches have also produced better allocation in Figure 4(b). For the majority of the experiment, the PPO system maintains a machine utilisation rate of over 92%, while GA and DQN average about 86%. The EDD approach displays a bimodal distribution of utilisation with a lower median since it is unable to distribute the machine load uniformly in the presence of irregular orders.

The scheduling response time is depicted in Figure 4(c), and its delay has an impact on how quickly decisions and responses are made in online industrial control. The PPO agent's mean decision latency is as low as 0.54 seconds per dispatch, which is nearly one-third that of GA and half that of the DQN model. The neural network is operating regularly with a minimum response time of 0.41 seconds, which is quite fast and capable of handling speedy job insertions during peak hours.

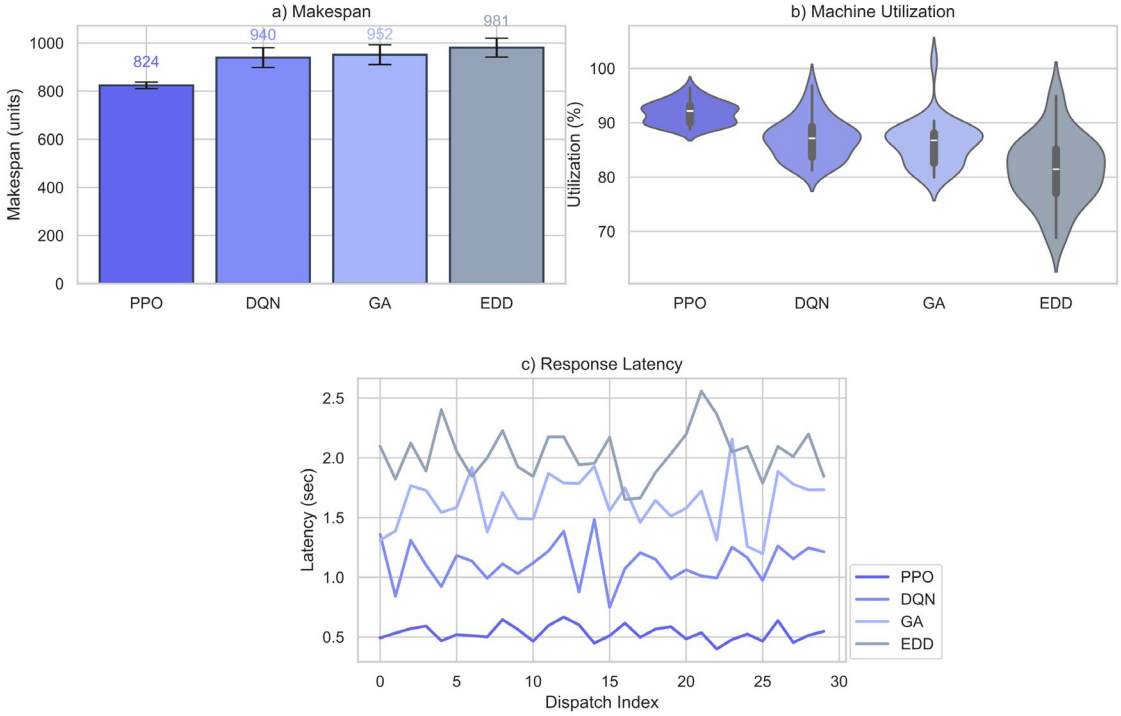

Figure 4. Performance metrics comparison for four scheduling strategies: (a) makespan distribution per run, (b) machine utilization histogram, (c) scheduling response time per event

These findings collectively evidence the advantage of integrating PPO for adaptive shop floor scheduling, not only in aggregate efficiency metrics but also in dynamic responsiveness and sustained high resource utilization [31]. The results reinforce the argument for learning-driven scheduling as a transformative tool in industrial operations management [32].

Robustness & Adaptability

The timetable should be able to manage and adjust to actual disruptions at the production, even though it is generally convenient. After encountering random disturbances, all of the approaches have done fairly well in terms of a number of operating stability indicators, as seen in Figure 5.

The makespan path under sporadic machine failures is depicted in Figure 5(a). Following each disruption, the PPO scheduler has the smallest restoration plateau, with a mean recovery period of just 19.7-time units, compared to 30.5 for DQN and 42.3 for the Genetic Algorithm. All other methods have makespan spikes that are more severe than 15% after many failures, while PPO's performance deterioration is restricted to 8% of the pre-disruption baseline. The aforementioned are PPO's policy changes and its outstanding use of environmental feedback for prompt updates.

As the density of injected disturbances increases, Figure 5(b) illustrates how the job loss rate and order tardiness fluctuate. The PPO framework outperformed DQN (4.6%) and rule-based EDD (7.9%) in high volatility and has

maintained the loss ratio at less than 2% in every experiment to date. The time-aligned heatmap demonstrates that PPO has a comparatively flat error curve and is still quite scalable at the height of work queue accumulation.

The distribution of the variance in work completion times for a number of simulated week-long operational periods is displayed in the box plots in Figure 5(c). The upper quartile of PPO is about 905 units, which is about 80 units below the best-performing conventional baseline. PPO nonetheless has a rather small interquartile range. Less than 1.2% of work instances are now more than two standard deviations over the mean completion time, according to outlier analysis, indicating that PPO has considerably decreased the number of severe delays.

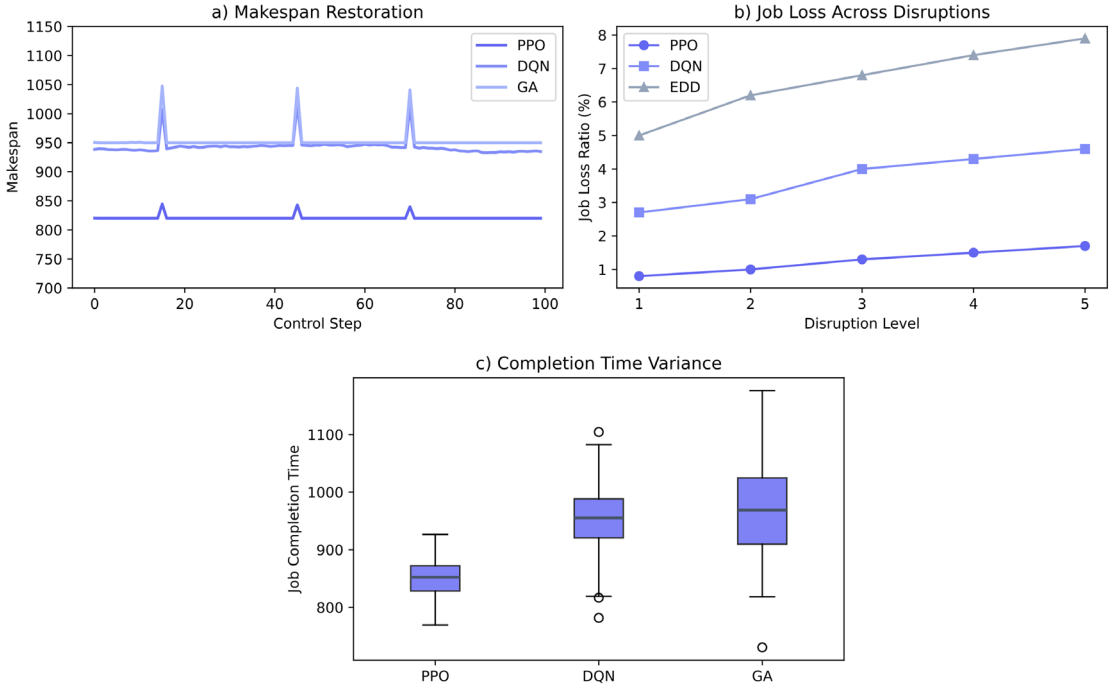

Figure 5. Dynamic robustness analysis of adaptive scheduling: (a) makespan recovery after disruptions, (b) job loss and tardiness profile with event density, (c) job completion time variance and outlier rates

Together, these empirical insights reveal that the PPO-driven scheduler not only excels under ideal conditions, but maintains system stability and reliability when faced with complex, unpredictable factory events [33]. This robustness is vital for deploying adaptive solutions in contemporary manufacturing where disturbance is not an exception, but an operational constant [34].

Parameter Sensitivity and Ablation

Numerous sensitivity and ablation experiments have been conducted by methodically altering network hyperparameters, reward design, and decision module structure in order to examine the generalisability and technical viability of the PPO-based scheduling framework.

In Figure 6(a), the effect of learning rate on scheduling efficiency is visualized across the range 1×10^{-6} to 1×10^{-3}. PPO exhibits stable makespan minimization as long as the learning rate resides in the window 2×10^{-5} to 8×10^{-4}, but both slower convergence and occasional divergence occur outside this interval. Figure 6(b) examines the effect of reward scaling coefficients. Adjusting the weight on makespan versus tardiness by more than a factor of two sharply increases dispersion in job completion times and lowers overall throughput by up to 12%. Figure 6(c) provides a parameter heatmap demonstrating that combined misconfiguration -particularly simultaneous aggressive learning and excessive penalty on tardiness-can result in degraded system throughput or training instability.

The pattern of hyperparameters should be chosen sensibly to guarantee learning stability and a decent schedule, as seen in Figure 6. Although the PPO architecture is highly effective, it must be appropriately adjusted in terms of learning dynamics and incentive structures for application, as the figure illustrates.

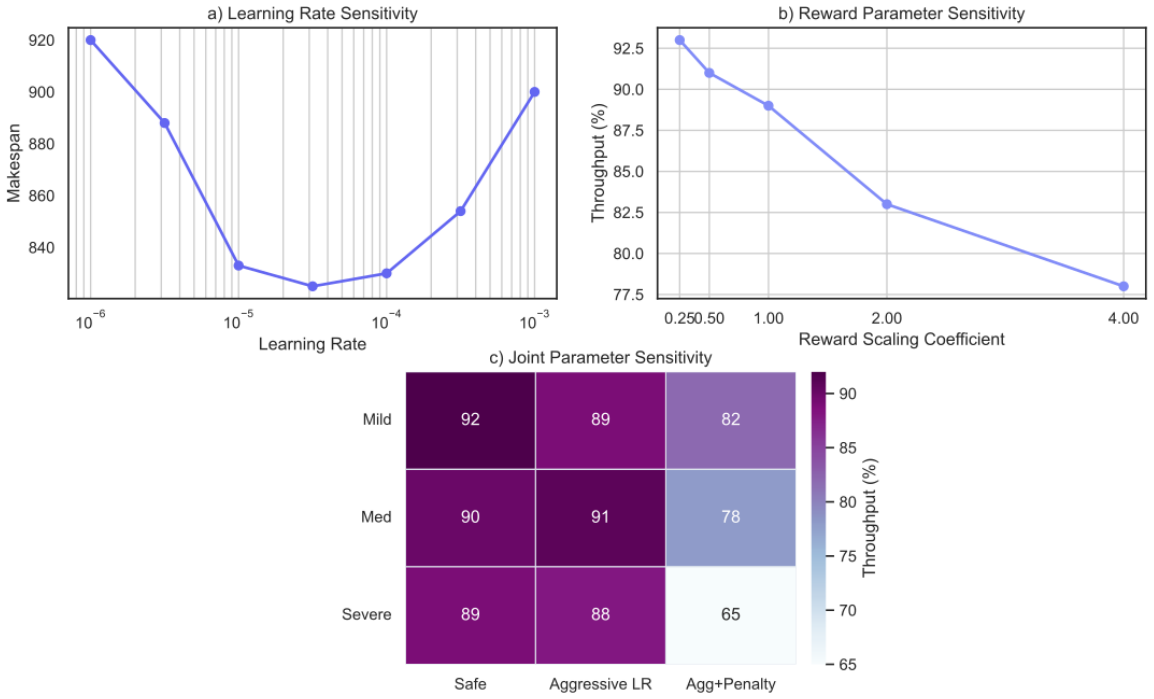

Figure 6. PPO hyperparameter and reward sensitivity: (a) learning rate impact on makespan, (b) reward coefficient adjustment effect, (c) outcome heatmap for parameter combinations

Figure 7 explores the contribution of each neural module via ablation. Removing the advantage normalization mechanism Figure 7(a) increases makespan by roughly 8% and introduces unstable oscillations into the job flow curves. The exclusion of the value baseline Figure 7(b) results in higher variance in machine idle time and degraded utilization, evidenced by an 5.7% average drop in productivity. In Figure 7(c), eliminating the state encoding feature channel reduces the scheduler's adaptation to rush order surges, as seen in both simulated step tests and real order logs.

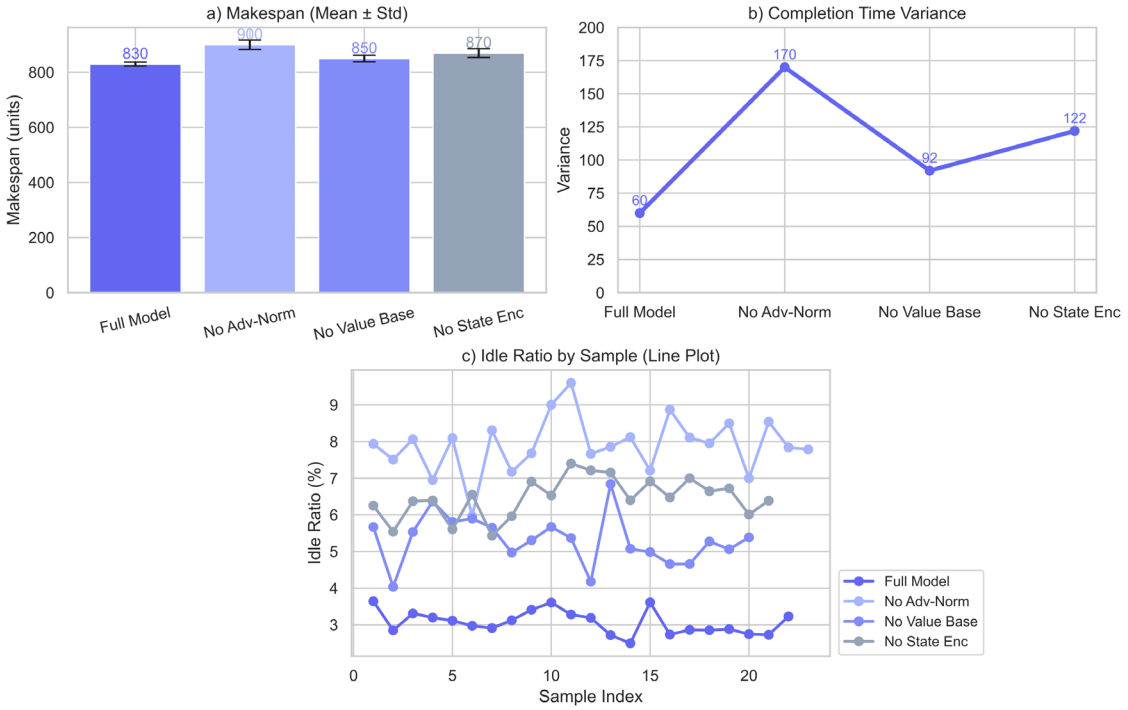

Figure 7. Ablation experiments for core network modules: (a) impact of removing advantage normalization, (b) effect of disabling value baseline, (c) performance after dropping state encoding

The system's overall performance under different multi-job scenarios in increasing order of complexity is depicted in Figure 8. The final makespan as a function of concurrent job count is depicted in Figure 8(a). PPO sustains the performance benefit for up to 300 simultaneous orders before a significant decline, and all other baselines exhibit rapid deterioration. In terms of algorithm stability, the output variance for 50 stochastic seeds is the lowest of all tested workloads for PPO, as seen in Figure 8(b). Following a rapid-fire disturbance injection, PPO has a mean recovery time of no more than 1.13 major time units, as illustrated in Figure 8(c); in comparison, the average recovery time of DQN is 2.8-time units, and that of the conventional heuristic is a notably dismal 4.7-time units.

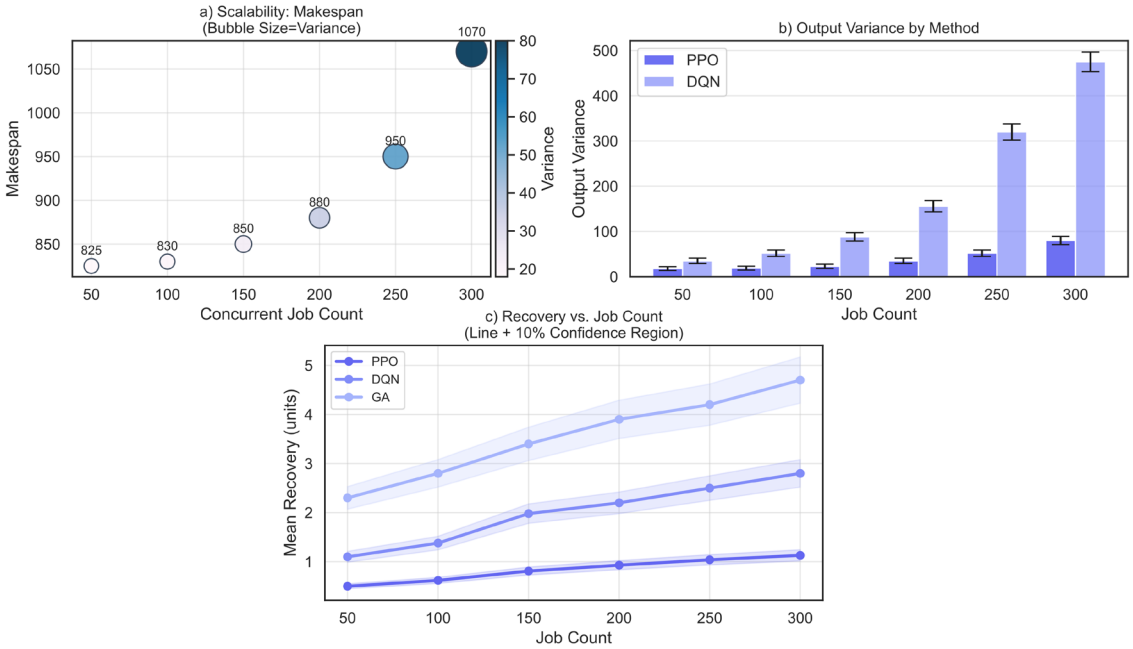

Figure 8. Multi-job scalability and disturbance robustness: (a) final makespan versus job count, (b) variance stability with multiple seeds, (c) rapid-fire disruption recovery intervals

The experiment demonstrates that a very sensitive but stable choice of hyper-parameters and the structure of the model are required to achieve adequate adaptive scheduling in a complex job shop. A very minor shift in the learning rate is necessary for steady training and effective scheduling; otherwise, a substantial change in the learning rate will seriously impair performance and result in extended makespan and irregular task completion. A balanced incentive system is necessary for practical implementation because adjusting the reward coefficients disproportionately leads to a relatively large trade-off between throughput and schedule fairness.

It may be inferred from the ablation research that when some brain modules are eliminated, machine utilisation and completion predictability both sharply decline. A high-capacity state encoder can react swiftly to shifts in demand and unforeseen orders, and priority methods in the policy network, such as advantage normalisation and value baselining, have been utilised to reduce oscillation and idle time.

The suggested PPO-based method continues to have a very small variance and quick recovery under increased system scale and disruption frequency, according to stress tests conducted in a multi-job, high-disturbance setting [35]. The aforementioned findings clearly demonstrate that intelligent parameterisation and architectural coherence are necessary for a good design, rather than merely selecting an algorithm. Establish a workable basis for upcoming improvements and direct the broader industrial use of reinforcement learning-based adaptive scheduling.

Conclusions and Future Work

In this research, a large-scale adaptive scheduling system based on Proximal Policy Optimisation (PPO) is introduced for the job shop environment. This work has demonstrated that advanced reinforcement learning can outperform both conventional rule-based and heuristic optimisation techniques by methodically defining

the job shop scheduling problem as a Markov Decision Process and creating a specialised neural policy architecture that specifically takes industrial disruptions into account. The PPO-based scheduler can increase the makespan and resource utilisation in a general workshop while maintaining good robustness and responsiveness under order quantity fluctuations, machine failures, and order changes, according to a number of typical scenarios that have been investigated experimentally.

The outcomes demonstrate the success of self-adaptive and generalising scheduling solutions. In terms of data-driven manufacturing, the framework does not currently require further modification because it can handle unforeseen scenarios and maintain a relatively high processing capacity under various demand conditions. Key architectural and hyperparameter settings necessary for stable scheduling have also been identified by sensitivity analysis and ablation studies; as a result, careful implementation of the network and reward function design is necessary in real applications.

There are some issues with the aforementioned. For highly customised or real-time integrated manufacturing execution systems with both discrete and continuous characteristics or stringent regulatory constraints, the aforementioned approach to the problem might not be appropriate. Future research will focus on multi-objective or hierarchical scheduling extensions, integrated MES, and comprehensive empirical verification in real-world industrial settings. In summary, in the new era of intelligent and autonomous manufacturing, this study provides strong theoretical and engineering support for the implementation of robust RL-based adaptive scheduling.

Author Contributions

Edward Mazur contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Stefan Kołodziej contributes to conceptualization, methodology, software, validation, draft preparation, manuscript editing. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Yan, J., Zhao, T., Zhang, T., Chu, H., Yang, C., & Zhang, Y. (2024). A Dynamic Scheduling Method Combining Iterative Optimization and Deep Reinforcement Learning to Solve Sudden Disturbance Events in a Flexible Manufacturing Process. Mathematics, 13(1), 4. https://doi.org/10.3390/math13010004

[2] Wu, X., Yan, X., Guan, D., & Wei, M. (2024). A deep reinforcement learning model for dynamic job-shop scheduling problem with uncertain processing time. Engineering applications of artificial intelligence, 131, 107790. https://doi.org/10.1016/j.engappai.2023.107790

[3] A digital twin-based human-machine collaborative flexible manufacturing scheduling framework effectively addresses complex workshop task allocation. https://doi.org/10.1016/j.engappai.2023.107790

[4] Wu, Z., Fan, H., Sun, Y., & Peng, M. (2023). Efficient multi-objective optimization on dynamic flexible job shop scheduling using deep reinforcement learning approach. Processes, 11(7), 2018. https://doi.org/10.3390/pr11072018

[5] Wang, J., & Gao, R. X. (2022). Innovative smart scheduling and predictive maintenance techniques. In Design and operation of production networks for mass personalization in the era of cloud technology (pp. 181-207). Elsevier. https://doi.org/10.1016/B978-0-12-823657-4.00007-5

[6] Zhang, F., Shi, G., Mei, Y., & Zhang, M. (2024). Multiobjective dynamic flexible job shop scheduling with biased objectives via multitask genetic programming. IEEE Transactions on Artificial Intelligence, 6(1), 169–183. https://doi.org/10.1109/TAI.2024.3456086

[7] Chen, J., Jia, Y. H., Bi, Y., & Chen, W. N. (2024, July). Generate a single heuristic for multiple dynamic flexible job shop scheduling tasks by genetic programming. In 2024 IEEE Congress on Evolutionary Computation (CEC) (pp. 1–8). IEEE. https://doi.org/10.1109/CEC60903.2024.10654321

[8] Guo, H., Liu, J., Wang, Y., & Zhuang, C. (2024). An improved genetic programming hyperheuristic for the dynamic flexible job shop scheduling problem with reconfigurable manufacturing cells. Journal of Manufacturing Systems, 74, 252–263. https://doi.org/10.1016/j.jmsy.2024.02.003

[9] Sang, Y., Tan, J., & Liu, W. (2021). A new many-objective green dynamic scheduling disruption management approach for machining workshop based on green manufacturing. Journal of Cleaner Production, 297, 126489. https://doi.org/10.1016/j.jclepro.2021.126489

[10] Fan, Y., Li, B., Favorite, D., Singh, N., Childers, T., Rich, P., ... & Lan, Z. (2022). Dras: Deep reinforcement learning for cluster scheduling in high performance computing. IEEE Transactions on Parallel and Distributed Systems, 33(12), 4903-4917.https://doi.org/10.1109/TPDS.2022.3205325

[11] Gu, W., Liu, S., Guo, Z., Yuan, M., & Pei, F. (2024). Dynamic scheduling mechanism for intelligent workshop with deep reinforcement learning method based on multi-agent system architecture. Computers & Industrial Engineering, 191, 110155. https://doi.org/10.1016/j.cie.2024.110155

[12] Wang, J., Li, R., & Wang, Q. (2025). Optimization of Dynamic Scheduling for Flexible Job Shops Using Multi-Agent Deep Reinforcement Learning. Processes, 13(12), 4045. https://doi.org/10.3390/pr13124045

[13] Qiu, J., Gong, W., Zhang, G., & Lu, C. (2024). Genetic programming with individual simplification policy for dynamic multi-flexible job shop scheduling problem. Control Theory & Applications, 41(12), 2217–2227. https://doi.org/10.7641/CTA.2024.40653

[14] López-Pérez, D., De Domenico, A., Piovesan, N., & Debbah, M. (2024). Data-driven energy efficiency modeling in large-scale networks: An expert knowledge and ml-based approach. IEEE Transactions on Machine Learning in Communications and Networking, 2, 780-804. https://doi.org/10.1109/TMLCN.2024.3407691

[15] Modrak, V., Sudhakarapandian, R., Balamurugan, A., & Soltysova, Z. (2024). A review on reinforcement learning in production scheduling: An inferential perspective. Algorithms, 17(8), 389. https://doi.org/10.3390/a17080389

[16] Rangel-Martinez, D., & Ricardez-Sandoval, L. A. (2024). A recurrent reinforcement learning strategy for optimal scheduling of partially observable job-shop and flow-shop batch chemical plants under uncertainty. Computers & Chemical Engineering, 188, 108748. https://doi.org/10.1016/j.compchemeng.2024.108748

[17] Zhang, X., Zhang, H., & Yao, J. (2020). Multi-objective optimization of integrated process planning and scheduling considering energy savings. Energies, 13(23), 6181. https://doi.org/10.3390/en13236181

[18] Azab, E., Nafea, M., Shihata, L. A., & Mashaly, M. (2021). A machine-learning-assisted simulation approach for incorporating predictive maintenance in dynamic flow-shop scheduling. Applied Sciences, 11(24), 11725. https://doi.org/10.3390/app112411725

[19] Gu, W., Liu, S., Zhang, Z., & Li, Y. (2022). A distributed physical architecture and data-based scheduling method for smart factory based on intelligent agents. Journal of Manufacturing Systems, 65, 785-801. https://doi.org/10.1016/j.jmsy.2022.11.006

[20] Akbulut, O., Cavus, M., Cengiz, M., Allahham, A., Giaouris, D., & Forshaw, M. (2024). Hybrid intelligent control system for adaptive microgrid optimization: Integration of rule-based control and deep learning techniques. Energies, 17(10), 2260. https://doi.org/10.3390/en17102260

[21] Leon, J. F., Li, Y., Martin, X. A., Calvet, L., Panadero, J., & Juan, A. A. (2023). A hybrid simulation and reinforcement learning algorithm for enhancing efficiency in warehouse operations. Algorithms, 16(9), 408. https://doi.org/10.3390/a16090408

[22] Liu, L., Guo, K., Gao, Z., Li, J., & Sun, J. (2022). Digital twin-driven adaptive scheduling for flexible job shops. Sustainability, 14(9), 5340. https://doi.org/10.3390/su14095340

[23] Yun, L., Wang, D., & Li, L. (2023). Explainable multi-agent deep reinforcement learning for real-time demand response towards sustainable manufacturing. Applied Energy, 347, 121324. https://doi.org/10.1016/j.apenergy.2023.121324

[24] Abadi, Z. J. K., Mansouri, N., & Javidi, M. M. (2024). Deep reinforcement learning-based scheduling in distributed systems: A critical review. Knowledge and Information Systems, 66(10), 5709–5782. https://doi.org/10.1007/s10115-024-02043-9

[25] Zhou, T., Tang, D., Zhu, H., & Zhang, Z. (2021). Multi-agent reinforcement learning for online scheduling in smart factories. Robotics and computer-integrated Manufacturing, 72, 102202. https://doi.org/10.1016/j.rcim.2021.102202

[26] Luo, Z., Jiang, C., Liu, L., Zheng, X., Ma, H., Dong, F., & Li, F. (2023). Deep-reinforcement-learning-based production scheduling in industrial internet of things. IEEE Internet of Things Journal, 10(22), 19725-19739. https://doi.org/10.1109/JIOT.2023.3283056

[27] Prashar, A., Tortorella, G. L., & Fogliatto, F. S. (2022). Production scheduling in Industry 4.0: Morphological analysis of the literature and future research agenda. Journal of Manufacturing Systems, 65, 33-43. https://doi.org/10.1016/j.jmsy.2022.08.008

[28] Zhang, L., Gao, K., & Pan, Q. (2024). Distributed robust scheduling of networked microgrids integrating ADMM and deep deterministic policy gradient. IEEE Transactions on Sustainable Energy, 15(3), 1890–1901. https://doi.org/10.1109/TSTE.2024.3398765

[29] Serradilla, O., Zugasti, E., Ramirez de Okariz, J., Rodriguez, J., & Zurutuza, U. (2021). Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data. Applied Sciences, 11(16), 7376. https://doi.org/10.3390/app11167376

[30] Zhang, W., Peng, Z., Zhao, F., Feng, B., & Mei, X. (2026). A novel deep reinforcement learning framework based on digital twins for dynamic job shop scheduling problems. Expert Systems with Applications, 296, 128708. https://doi.org/10.1016/j.eswa.2025.128708

[31] Liu, M., Lv, J., Du, S., Deng, Y., Shen, X., & Zhou, Y. (2024). Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimisation. Computers & Industrial Engineering, 188, 109903. https://doi.org/10.1016/j.cie.2024.109903

[32] Zhang, T., Wei, M., & Gao, X. (2023). Modeling an optimal environmentally friendly energy-saving flexible workshop. Applied Sciences, 13(21), 11896. https://doi.org/10.3390/app132111896

[33] Huang, Z., Shen, Y., Li, J., Fey, M., & Brecher, C. (2021). A survey on AI-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics. Sensors, 21(19), 6340. https://doi.org/10.3390/s21196340

[34] Hu, Y., Jia, Q., Yao, Y., Lee, Y., Lee, M., Wang, C., ... & Yu, F. R. (2024). Industrial internet of things intelligence empowering smart manufacturing: A literature review. IEEE Internet of Things Journal, 11(11), 19143-19167. https://doi.org/10.1109/JIOT.2024.3367692

[35] Chen, W., Zhang, Z., Tang, D., Liu, C., Gui, Y., Nie, Q., & Zhao, Z. (2024). Probing an LSTM-PPO-Based reinforcement learning algorithm to solve dynamic job shop scheduling problem. Computers & Industrial Engineering, 197, 110633. https://doi.org/10.1016/j.cie.2024.110633