

A Priority-Guided Action-Masked Proximal Policy Optimization Framework for Dynamic Scheduling of Electric Power Material Verification Tasks

Zhaolei He

Metrology CenterYunnan Power Grid Co., Ltd Kunming, China

Ao He

Metrology CenterYunnan Power Grid Co., Ltd Kunming, China

Cong Lin

Metrology CenterYunnan Power Grid Co., Ltd Kunming, China

Jing Zhao

Metrology CenterYunnan Power Grid Co., Ltd Kunming, China

Liping Gao

Metrology CenterYunnan Power Grid Co., Ltd Kunming, China

Xiang Yin

`ditty1804@yeah.net`

Kunming University of Science and Technology

Research Article

Keywords: dynamic scheduling, electric power material verification, proximal policy optimization, action masking

Posted Date: August 26th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10589894/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A Priority-Guided Action-Masked Proximal Policy Optimization Framework for Dynamic Scheduling of Electric Power Material Verification Tasks

Zhaolei He¹, Ao He¹, Cong Lin¹, Jing Zhao¹, Liping Gao¹,
Xiang Yin^{2*}

¹Metering Center, Yunnan Power Grid Co., Ltd., Kunming, 650011, China.

²Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming, 650000, China.

*Corresponding author(s). E-mail(s): ditty1804@gmail.com;

Abstract

Dynamic electric power material verification requires scheduling policies that coordinate timeliness, resource use, and interruption recovery under changing task and device states. We propose PPO-Two-Stage (PPO-TS), a priority-guided action-masked Proximal Policy Optimization framework for simulation-based verification scheduling. PPO-TS enumerates feasible task-device assignments, masks invalid actions, and retains a priority-ranked candidate subset before policy selection. The confirmatory evaluation was prospectively defined and hash-frozen before effect access. It used nine Brandimarte-based core cells, three independent training layers, a shared emergency-slack (ES) baseline, and ten frozen ScenarioTapes per cell. The global balanced weighted composite score (WCS) contrast for PPO-TS minus ES was 0.00575 (weighted-Welch 95% confidence interval, 0.00466 to 0.00683; $p = 1.20 \times 10^{-9}$). Holm-adjusted component results favored PPO-TS for average completion time in 9/9 cells and throughput in 8/9 cells, while equipment utilization was adverse in 9/9 cells. Cell-level effects were heterogeneous, including an adverse balanced-WCS result in C001. Moreover, the balanced ScenarioTape extrapolation interval included zero, so algorithm order in the broader ScenarioTape population was not statistically confirmed. These results support a conditional balanced-WCS benefit in the frozen synthetic benchmark while identifying utilization and generalization boundaries.

Keywords: dynamic scheduling, electric power material verification, proximal policy optimization, action masking

1 Introduction

Electric power material verification supports equipment reliability and safe power-grid operation. In a verification center, batches of tasks must be assigned to compatible verification devices under time, capacity, and operational constraints. Unlike a static manufacturing schedule, the verification queue evolves while scheduling is in progress. Random arrivals, urgent task insertion, variable device efficiency, and device failure can all change the feasible assignment set. A useful scheduling policy must therefore coordinate task timeliness, equipment use, and recovery from interruptions rather than optimize a single static objective.

Rule-based dispatching methods are attractive in this setting because they are transparent, fast, and easy to deploy. In particular, an emergency-slack heuristic (ES) is a strong practical baseline when urgent status and remaining slack are the dominant decision signals. However, a fixed rule usually compresses the system state into a small number of handcrafted priorities. It can be difficult to extend the same rule when queue composition, device load, failure history, and future disturbance patterns need to be fused. Deep reinforcement learning (DRL) offers a complementary route because it can learn from simulated interaction with a multi-state environment, but direct DRL policies face a large task-device action space filled with many invalid or low-value assignments.

This study addresses these issues through a priority-guided action-masked Proximal Policy Optimization (PPO) framework for dynamic electric power material verification scheduling. The scheduling process is formulated as a Markov Decision Process (MDP), and the policy is trained in a simulator with random arrivals, urgent tasks, compatible device assignments, and device failures. Before PPO action selection, infeasible task-device actions are masked and a priority-ranked candidate subset is retained. The aim is to test a structured learning framework against a strong emergency-slack heuristic, without assuming that one policy should prevail for every operational metric or preference.

The contributions are threefold. First, the method combines feasibility masking, priority-guided candidate filtering, and PPO optimization for dynamic task-device assignment. Second, the formal evaluation framework separates prospectively defined, hash-locked balanced-WCS confirmatory testing from offline weight sensitivity analysis. Third, the nine-cell experimental design tests adaptability across nominal, urgent, failure, and arrival-rate disturbance scenarios while preserving adverse and non-significant outcomes.

2 Related Work

2.1 Dynamic and Flexible Job-Shop Scheduling

Flexible job-shop scheduling offers a suitable abstraction for verification tasks because each operation can be allocated to multiple eligible machines. The Brandimarte benchmark remains a widely used source of flexible job-shop instances [30]. Subsequent work extends this setting with distributed processing, stochastic uncertainty, and multi-objective criteria. Tang et al. [1] studied distributed flexible job-shop scheduling with

integrated sequencing flexibility, while Xie et al. [2] and Li et al. [3] addressed multi-objective optimization within logistics and reinforcement learning pipelines. Dynamic uncertainty is especially important for online scheduling. Duan and Wang [4] investigated machine breakdowns and dynamic job arrivals in flexible machining systems, and Zhang et al. [5] developed robust logistics policies under uncertain travel durations. These studies motivate the dynamic verification simulation adopted in this paper.

2.2 Priority Rules and Dynamic Event Handling

Priority dispatching rules remain widely adopted due to transparency and low computational overhead. Panwalkar and Iskander [27] and Haupt [6] provided comprehensive surveys of priority-rule scheduling. Recent work develops dynamic priority mechanisms for online decision making. Su et al. [7] designed multi-feature dynamic priority for network scheduling, and Lei et al. [8] studied static-priority energy-saving online scheduling. These methods demonstrate the practical value of priority information, yet handcrafted rules lack adaptability in heavily disturbed verification environments. PPO-TS therefore uses priority scores for candidate filtering guidance rather than as the sole decision rule.

2.3 Deep Reinforcement Learning and PPO for Scheduling

Reinforcement learning has been applied widely to scheduling and resource allocation due to its ability to learn from sequential environmental interaction. Ahamed et al. [9] discussed reinforcement learning in power system scheduling and control. Xu et al. [10] and Yuan et al. [11] deployed meta-reinforcement learning for adaptive industrial IoT and resource allocation. Within scheduling research, Li et al. [12] used DRL for dynamic satellite task scheduling, and Li et al. [13] adopted a deep Q-network variant for stochastic flexible integrated scheduling. Sun et al. [14], Wang et al. [15], Shi et al. [16], and Qiao et al. [17] further extended DRL to job-shop, production-logistics, and routing tasks. PPO has also been deployed for flexible job-shop scheduling as shown in DRL-PPO and improved PPO scheduling studies [19, 21]. These works confirm PPO as a viable scheduling learner while highlighting the importance of constraining oversized action spaces.

2.4 Action Masking and Structured Policy Learning

Structured policy learning reduces redundant exploration over invalid or low-value actions. Invalid action masking has been analyzed within policy gradient algorithms [26], and recent scheduling research combines DRL with graph neural networks, operator optimization, resource redistribution, or action masking [18, 22–24]. Wang et al. [20] also explored feature extraction for flexible job-shop scheduling with reinforcement learning. In the present work, feature gating acts as a lightweight encoder component, while the core structural innovation combines priority-guided action masking and top- k candidate filtering. This architecture directly targets feasible task-device assignment for dynamic electric power material verification.

3 Problem Formulation

3.1 Dynamic Verification Scheduling Scenario

Electric power material verification scheduling is treated as an online assignment problem. Verification tasks arrive dynamically. Each task follows a predefined sequence of operations inherited from flexible job-shop instances, and each operation can only be processed by compatible verification devices. The operation index j is tracked inside the simulator as each task's current operation pointer. The simulator injects stochastic task arrivals, urgent-task labels, sampled device efficiencies, and stochastic device failures. A scheduling decision is triggered whenever the environment contains ready tasks and available devices. The decision assigns one visible ready task to one compatible idle device, or the environment advances if no feasible assignment exists.

The task set, device set, and discrete time horizon are defined as:

$$\mathcal{J} = \{1, \dots, n\}, \quad \mathcal{M} = \{1, \dots, m\}, \quad \mathcal{T} = \{0, \dots, T_{\max}\}. \quad (1)$$

Here, $\mathcal{J}$ indexes verification tasks, $\mathcal{M}$ indexes verification devices, and $\mathcal{T}$ indexes simulation time steps. The episode termination horizon is denoted by $T_{\max}$.

For task i , the scheduling attributes are summarized as:

$$\xi_i = (a_i, p_i, d_i, \tau_i, c_i). \quad (2)$$

The vector collects arrival time a_i , nominal processing requirement p_i , due date d_i , task type τ_i , and capacity contribution c_i .

For a compatible task-device pair, the realized processing duration is defined as:

$$p_{ik}^{\text{eff}}(t) = \left\lceil \frac{p_{ik}}{\max(0.5, e_k(t))} \right\rceil. \quad (3)$$

The variable p_{ik} denotes the scaled nominal processing time on device k , and $e_k(t)$ denotes the sampled device-efficiency factor. The denominator lower bound prevents unrealistically small efficiency values from producing unstable durations.

The main notation used in the formulation and method is summarized in Table 1.

Table 1: Notation used in Sections 3 and 4.

Category	Symbol	Meaning
Sets	$\mathcal{J}, \mathcal{M}, \mathcal{T}$	Task set, verification-device set, discrete simulation-time set.
Indices	i, j, k, t	Task index, operation index, device index, time-step index.
Task attributes	$a_i, p_i, d_i, \tau_i, c_i$	Arrival time, nominal processing requirement, due date, task type, capacity contribution of task i .
Processing	$p_{ik}, p_{ik}^{\text{eff}}, e_k(t)$	Nominal compatible processing time, realized processing duration, sampled device efficiency.
Decision/state	$x_{ik}(t), s_k(t), \sigma_i(t), r_i(t), C_i$	Assignment indicator, device availability, task state, remaining minimum processing time, completion time.
Priority	$\beta_i(t), TD_i, \omega_i(t), \omega_{ik}(t), \alpha$	Urgent-task indicator, static time-demand ratio, task priority, task-device priority score, retained top- k action fraction.
RL model	$\mathcal{S}, \mathcal{A}, P, R, \gamma$	State space, action space, transition kernel, reward function, discount factor of MDP.
RL sample	s_t, a_t, r_t	State, action, scalar reward observed at time step t .
Metrics	ACT, TR, EU, IRC, WCS	Average completion time, tardiness rate, equipment utilization, interruption recovery cost, weighted composite score.

3.2 Scheduling Variables and Feasibility Conditions

The binary assignment variable is defined as:

$$x_{ik}(t) \in \{0, 1\}. \quad (4)$$

The variable equals one only when task i is assigned to device k at time t . The same decision is represented in the policy action space by a visible task slot and device index.

Each device can process at most one task at a time:

$$\sum_{i \in \mathcal{J}} x_{ik}(t) \leq 1, \quad \forall k \in \mathcal{M}, t \in \mathcal{T}. \quad (5)$$

This constraint enforces device capacity exclusiveness. It is enforced inside the simulator by only permitting assignments to devices marked idle.

Device-state feasibility is written as:

$$\sum_{i \in \mathcal{J}} x_{ik}(t) \leq s_k(t), \quad \forall k \in \mathcal{M}, t \in \mathcal{T}. \quad (6)$$

Here, $s_k(t) = 1$ indicates that device k is available, and $s_k(t) = 0$ indicates that it is busy or broken.

Task-state feasibility is defined as:

$$x_{ik}(t) = 0 \quad \text{if } \sigma_i(t) \notin \{\text{ready}\} \text{ or } k \notin \mathcal{O}_i(t). \quad (7)$$

This formula states that only ready tasks can be assigned, and only devices in the current operation's compatible option set $\mathcal{O}_i(t)$ can be selected. The simulator implements this rule by building feasible actions from the visible ready queue and each task's current operation options. Operation order is therefore enforced by the current operation index, not by a separate setup-time constraint.

3.3 Evaluation Metrics

Evaluation metrics are computed from completed tasks and device accounting variables. Average completion time (ACT) and task tardiness rate (TR) are defined as:

$$\text{ACT} = \frac{1}{N_c} \sum_{i=1}^{N_c} (C_i - a_i), \quad \text{TR} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbb{I}(C_i > d_i). \quad (8)$$

Here, N_c is the number of completed tasks, C_i is the completion time, and $\mathbb{I}(\cdot)$ is an indicator function.

Equipment utilization (EU), interruption recovery cost (IRC), throughput (TP):

$$\text{EU} = \frac{\sum_{k \in \mathcal{M}} T_k^{\text{busy}}}{|\mathcal{M}|T}, \quad \text{IRC} = 2N_{\text{interrupt}} + N_{\text{migration}}, \quad \text{TP} = \frac{N_c}{T}. \quad (9)$$

The variable T_k^{busy} is the accumulated busy time of device k , T is the elapsed simulation time, $N_{\text{interrupt}}$ counts device-failure interruptions, and $N_{\text{migration}}$ counts interrupted tasks returned to the ready queue.

The weighted composite score (WCS) is an evaluation summary score:

$$\text{WCS} = 0.25\widetilde{\text{ACT}} + 0.25\widetilde{\text{TR}} + 0.20\widetilde{\text{EU}} + 0.20\widetilde{\text{IRC}} + 0.10\widetilde{\text{TP}}. \quad (10)$$

WCS summarizes normalized ACT, TR, EU, IRC, and TP into one scalar comparison score. It is used as a compact reporting index for the selected operational priorities and is not a Pareto-dominance volume.

The normalized components are:

$$\widetilde{\text{ACT}} = \frac{1}{1 + \text{ACT}/20}, \quad \widetilde{\text{TR}} = 1 - \text{clip}(\text{TR}, 0, 1), \quad \widetilde{\text{EU}} = \text{clip}(\text{EU}/0.65, 0, 1). \quad (11)$$

$$\widetilde{\text{IRC}} = \frac{1}{1 + \text{IRC}/10}, \quad \widetilde{\text{TP}} = \text{clip}(\text{TP}/0.35, 0, 1). \quad (12)$$

These normalization rules are used only for WCS reporting. The static evaluation metrics measure final scheduling performance, while the dynamic priority score in Section 4 is used for action filtering and policy guidance.

4 Proposed Method

This section presents the priority-guided action-masked PPO framework, denoted as PPO-TS. The framework operates within a dynamic scheduling simulator where available tasks, device states, and disturbance information evolve during each episode. PPO-TS contains four functional modules: dynamic scheduling simulation, feasible-action masking, priority-guided top- k candidate filtering, and PPO policy learning. A lightweight feature-gating layer is included as an optional state encoding component, yet the core scheduling architecture relies on feasibility control, candidate filtering, and reinforcement learning optimization.

The dynamic scheduling process is modeled as an MDP [25]. At each decision time, the simulator releases newly arrived tasks, updates device states, constructs feasible task-device actions, and passes a masked candidate set to the PPO policy. The policy output is either a task-device assignment or a no-assignment transition when no feasible assignment exists. The overall process is shown in Figure 1.

4.1 MDP Modeling and State Representation

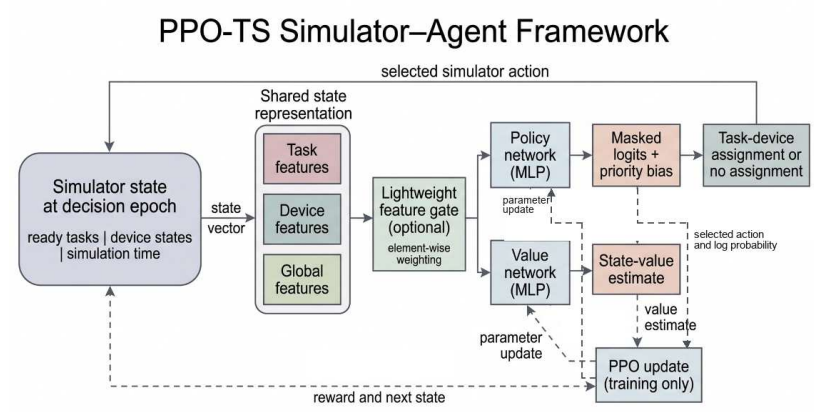
The electric power material verification scheduling process is formulated as a Markov Decision Process (MDP):

$$\mathcal{M}_{\text{RL}} = (\mathcal{S}, \mathcal{A}, P, R, \gamma). \quad (13)$$

The components denote the state space, action space, transition kernel, reward function, and discount factor [25]. The discount factor is set to $\gamma = 0.99$ for return computation.

The state observed by the policy at time t is:

$$s_t \in \mathcal{S}, \quad s_t = \left[s_t^{\text{task}}, s_t^{\text{device}}, s_t^{\text{global}} \right]. \quad (14)$$



No independent sequence encoder is implemented; feature gating is a lightweight optional encoding component.
Fig. 1: Simulator-agent framework implemented for PPO-TS. The simulator provides task, device, and global state features. The same optionally feature-gated state representation is consumed by separate policy and value multilayer perceptrons. Invalid actions are masked and priority scores bias candidate logits before the simulator receives an assignment or no-assignment action. No independent sequence encoder is used.

The state vector is constructed from the simulator state at a decision epoch and is consumed by the actor and critic networks. The task component records urgent status, urgency, normalized remaining work, normalized slack, operation progress, and capacity contribution for visible ready tasks. The device component records idle status, broken status, normalized remaining busy time, cumulative utilization, and sampled efficiency. The global component records normalized simulation time and queue-size ratios.

The discrete action selected by PPO is:

$$a_t = (q_t, k_t). \quad (15)$$

Here, q_t is a visible ready-task slot, and k_t is a verification device. A discrete action is decoded into this task-device pair after feasibility filtering, so invalid assignments are excluded before policy sampling.

Table 2 summarizes the action components.

Table 2: Action components in the assignment space.

Component	Description
Task slot	Select one task from the currently visible ready-task queue.
Device assignment	Select a compatible device that is currently idle and not broken.
No assignment	If no feasible action exists, the environment advances without assigning a task.

4.2 Feasible Action Masking and Top- k Filtering

At time t , the environment first enumerates feasible actions before masked policy selection [26]:

$$\mathcal{A}_t^{\text{feas}} = \{(q, k) : i(q) \in Q_t, k \in \mathcal{O}_{i(q)}(t), s_k(t) = 1\}. \quad (16)$$

The set is built from the visible ready-task queue, each task’s current operation options, and the current device-status vector. Q_t denotes the visible ready-task slots, $i(q)$ maps a slot to its task, $\mathcal{O}_{i(q)}(t)$ is the compatible device set for the current operation, and $s_k(t) = 1$ indicates that device k is idle and not broken.

PPO-TS then keeps the highest-scoring fraction of feasible actions. This filtering follows the scheduling practice of priority-rule guidance [6, 27]:

$$\mathcal{A}_t^{\text{cand}} = \text{Top}_{\lceil \alpha |\mathcal{A}_t^{\text{feas}}| \rceil} \{(a, \omega_a(t)) : a \in \mathcal{A}_t^{\text{feas}}\}. \quad (17)$$

The parameter α is the retained candidate fraction. The experiments use $\alpha = 0.35$ when top- k filtering is enabled.

4.3 Dynamic Priority Mechanism

The priority score is used for candidate filtering and, when enabled, as a logit prior. It follows the long tradition of dispatching-rule guidance in scheduling [6, 27]. The notation $\omega_i(t)$ refers to task-side priority when device-specific terms are not separated, while the task-device score is denoted by $\omega_{ik}(t)$. For a candidate task-device assignment, the score is:

$$\begin{aligned} \omega_{ik}(t) = & 0.30u_i(t) + 0.14\beta_i(t) + 0.18a_i^w(t) + 0.20q_i(t) \\ & + 0.04c_i + 0.10v_{ik}(t) + 0.04b_k(t). \end{aligned} \quad (18)$$

The terms denote urgency $u_i(t)$, urgent-task indicator $\beta_i(t)$, waiting age $a_i^w(t)$, short-job preference $q_i(t)$, capacity contribution c_i , device-speed suitability $v_{ik}(t)$, and load-balance preference $b_k(t)$.

The score components are defined as:

$$u_i(t) = \frac{1}{1 + \max(0, d_i - t - r_i(t))}. \quad (19)$$

This term assigns higher urgency to tasks with smaller nonnegative slack after remaining processing time $r_i(t)$ is considered.

$$q_i(t) = \frac{1}{1 + r_i(t)}, \quad a_i^w(t) = \text{clip}\left(\frac{t - a_i}{60}, 0, 1\right). \quad (20)$$

These terms favor shorter remaining work and longer waiting time. The clipping operation bounds waiting age between 0 and 1.

$$v_{ik}(t) = \frac{e_k(t)}{\max(1, p_{ik})}, \quad b_k(t) = 1 - \frac{T_k^{\text{busy}}}{t+1}. \quad (21)$$

These terms favor faster compatible devices and lower accumulated load. In the ablation setting without dynamic priority, the urgency, urgent-task, and waiting-age terms are set to zero. The static ratio TD_i appears only as background notation for traditional priority rules and is not used in PPO-TS.

4.4 Policy Network and PPO Update

The policy network contains shared state encoding, actor logits, and a critic value head. Feature gating is used as a lightweight state transformation rather than as a core independent contribution. When feature gating is enabled, the state vector is gated elementwise:

$$z_t = s_t \odot \sigma(W_g s_t + b_g). \quad (22)$$

The gated vector z_t is passed to the actor and critic. W_g and b_g are trainable parameters, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes elementwise multiplication.

The actor produces logits for the full discrete action space. Non-candidate actions receive a large negative mask value, which implements invalid-action masking during policy selection [26]:

$$\ell'_a = \begin{cases} \ell_a + \kappa \hat{\omega}_a(t), & a \in \mathcal{A}_t^{\text{cand}}, \\ -10^9, & a \notin \mathcal{A}_t^{\text{cand}}. \end{cases} \quad (23)$$

The masked logit vector assigns negligible probability to non-candidate actions. ℓ_a is the actor logit, $\hat{\omega}_a(t)$ is the min-max normalized priority score, and κ is the prior strength. PPO samples or greedily selects from the resulting categorical distribution.

For policy learning, returns are computed recursively:

$$G_t = r_t + \gamma G_{t+1}. \quad (24)$$

This standard reinforcement-learning return definition is used in the PPO update [28]. The return sequence is normalized before optimization, and return-minus-value is used as the advantage estimate for the clipped objective.

The PPO clipped objective is:

$$L^{\text{clip}}(\theta) = \mathbb{E}_t \left[\min \left(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right]. \quad (25)$$

$$\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}. \quad (26)$$

The clipped surrogate objective follows standard PPO [28], with clipping coefficient $\epsilon = 0.2$.

The optimized loss combines policy, value, and entropy terms:

$$\mathcal{L}(\theta) = -L^{\text{clip}}(\theta) + 0.5\mathcal{L}_V(\theta) - 0.01\mathcal{H}(\pi_\theta). \quad (27)$$

Here, $\mathcal{L}_V$ is the critic mean-squared-error loss and $\mathcal{H}$ is the categorical-policy entropy term [28].

4.5 Dynamic Response and Reward Mechanism

Dynamic events are generated inside the simulation, following dynamic job-shop settings with new job arrivals and device-state disturbances [4]. Each verification task enters the future-task stream with a stochastic arrival time. A task is marked urgent with a fixed probability, which shortens its deadline slack and adds an emergency feature to the state and priority score. At each step, newly arrived tasks are moved into the ready queue.

Device failures occur stochastically while a device is busy [4]. When a failure occurs, the current task is returned to the front of the ready queue, the device is marked broken for a sampled repair duration, and interruption and migration counters are incremented. Broken devices are masked from feasible assignments. During later steps, a broken device returns to the idle state when its repair counter expires or a repair event is sampled. The module output is the updated device state, ready queue, and interruption counters.

The dynamic response process is illustrated in Figure 2.

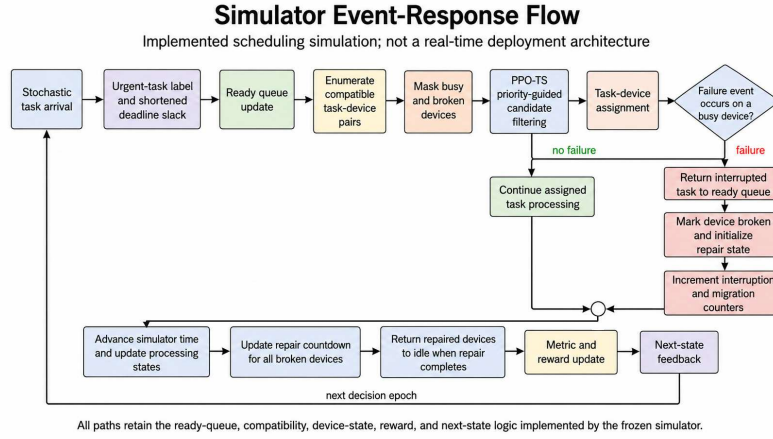


Fig. 2: Simulator event-response flow implemented for the synthetic scheduling environment. At each simulator step, arrivals and urgent labels update ready tasks; compatible assignments are enumerated, busy and broken devices are masked, and PPO-TS applies priority-guided candidate filtering before assignment. Busy devices can fail stochastically, interrupted tasks return to the ready queue, repair counters update device state, and repaired devices return to idle before metric/reward and next-state feedback. This figure depicts a simulator event-response flow, not a real online deployment system.

The reward mechanism embeds assignment quality and dynamic event feedback into reinforcement learning. A small step penalty is applied at each decision. A feasible assignment receives a score-based reward, urgent tasks receive an additional bonus when assigned, and a ready queue that cannot be assigned receives a penalty. After the environment advances, throughput and equipment utilization add positive reward terms, while tardiness and interruption recovery cost add penalties. The weighted scalar form follows the general composite-indicator practice of normalizing

and weighting heterogeneous indicators [29]:

$$R_t = -0.01 + 0.30\omega_a + 0.08\mathbb{I}_{\text{urgent}} - 0.04\mathbb{I}_{\text{unassigned}} + 0.10\text{TP} - 0.08\text{TR} + 0.03\text{EU} - 0.01\text{IRC}. \quad (28)$$

Here, ω_a is the selected action score, $\mathbb{I}_{\text{urgent}}$ indicates that the assigned task is urgent, and $\mathbb{I}_{\text{unassigned}}$ indicates that ready tasks remain but no assignment was made. The metric terms are computed from the simulator state after the time step advances.

4.6 Training and Evaluation Procedures

4.6.1 Algorithm 1. Dynamic scheduling environment and feasible action generation

Algorithm 1 specifies the environment-side input-output flow. Its inputs are a problem instance, scenario seed, current time, ready and future task queues, device states, and configuration parameters. Its outputs are the next state, reward, done flag, feasible or candidate actions, and updated metric counters.

Algorithm 1 Dynamic Scheduling Environment

- 1: Initialize the task stream, device states, repair counters, utilization counters, and event counters.
 - 2: At each time step, release all tasks whose arrival time is not greater than the current simulation time.
 - 3: Build the visible ready-task queue and read the current operation options for each visible task.
 - 4: For each visible task-device pair, keep the action only if the device is compatible, idle, and not broken.
 - 5: Score each feasible action using $\omega_{ik}(t)$ and keep the top $\lceil \alpha |\mathcal{A}_t^{\text{feas}}| \rceil$ actions when top- k filtering is enabled.
 - 6: Apply the selected action, advance processing by one time step, update completed tasks, and process stochastic breakdown or repair events.
 - 7: Return the next state, scalar reward, done flag, and updated metric counters.
-

4.6.2 Algorithm 2. PPO-TS training and evaluation procedure

Algorithm 2 specifies the policy-side training and evaluation flow. Its training inputs are environment states, feasible candidate actions, rewards, and stored log probabilities. Its outputs are trained network parameters and evaluation records used by the frozen confirmatory protocol.

Algorithm 2 PPO-TS Training and Evaluation

- 1: For each problem instance and random seed, initialize the dynamic scheduling environment and PPO-TS agent.
 - 2: During training, generate feasible candidate actions using Algorithm 1 and compute masked policy logits.
 - 3: Sample an action from the masked categorical policy, execute the environment step, and store state, action, log probability, reward, and done flag.
 - 4: After an episode or a sufficient buffer length, compute discounted returns and update the actor-critic network with the PPO clipped objective.
 - 5: Repeat the training loop for the configured number of episodes.
 - 6: During evaluation, run the trained policy deterministically on the frozen ScenarioTapes and record ACT, TR, EU, IRC, TP, and WCS.
 - 7: Preserve training-layer identities, the shared ES baseline, Mk10 checkpoint blocks, and cell labels for the frozen confirmatory aggregation.
-

5 Experimental Setup

5.1 Frozen Benchmark Design

The evaluation used Brandimarte flexible job-shop instances as synthetic verification tasks and devices [30]. The simulator added frozen task arrivals, urgent-task states, device failures, and operation-device compatibility. The confirmatory comparison was restricted to PPO-TS and the emergency-slack heuristic (ES). ES prioritizes urgent tasks and smaller remaining slack and therefore provides a strong rule-based reference for the study’s scheduling setting.

Nine core cells were evaluated in the fixed order shown in Table 3. C001 and C002 formed independent Mk01 and Mk03 training layers. C005–C011 shared the Mk10 training layer and remained a seven-cell checkpoint block in all resampling. The design therefore contained three independent training layers rather than one cross-instance seed panel.

Each PPO-TS layer used ten independent TrainingSeeds. Each cell used ten frozen ScenarioTapes. ES was a single shared baseline within each cell and was not copied into pseudo-seeds. The final input comprised 990 eligible evaluation records: 900 PPO-TS records and 90 ES records. All nine cells received equal weight in the global balanced-WCS estimand. Same-named TrainingSeed indices were never paired across training layers.

Table 3: Frozen nine-cell confirmatory evaluation design.

Cell	Instance	Condition	Frozen disturbance setting
C001	Mk01	Nominal	No additional disturbance parameter.
C002	Mk03	Nominal	No additional disturbance parameter.
C005	Mk10	Nominal	Reference condition for Mk10 disturbance contrasts.
C006	Mk10	Urgency, low	Emergency probability 0.05.
C007	Mk10	Urgency, high	Emergency probability 0.30.
C008	Mk10	Failure, low	Breakdown probability 0.01.
C009	Mk10	Failure, high	Breakdown probability 0.05.
C010	Mk10	Arrival, low	Arrival-gap range 3–6.
C011	Mk10	Arrival, high	Arrival-gap range 1–2.

5.2 Confirmatory Inference and Sensitivity Boundary

The primary estimand was the nine-cell equal-weight PPO-TS-minus-ES contrast in balanced WCS. Positive values favor PPO-TS after the frozen benefit-oriented normalization. The main interval and two-sided raw p -value used a weighted Welch analysis with Welch–Satterthwaite degrees of freedom [31, 32]. A conditional TrainingSeed bootstrap resampled the three training layers independently for 10,000 replicates, while preserving each Mk10 seven-cell block. An exact hierarchical sign-flip audit enumerated all 2^{30} training-seed sign configurations.

Three pre-specified and hash-frozen Holm families controlled cell and component follow-up claims [33]. They contained nine cell-WCS contrasts, six Mk10 disturbance contrasts, and 45 component contrasts. A global balanced-WCS gate had to pass before any lower-level result could support a confirmatory claim. The component directions were frozen before analysis: lower ACT, TR, and IRC were favorable, while higher EU and TP were favorable.

ScenarioTape extrapolation was evaluated separately with 10,000 bootstrap replicates. It resampled frozen ScenarioTapes while preserving PPO-TS–ES tape pairing and Mk10 structure. This interval did not replace the conditional confirmatory interval. Four alternative WCS schemes were also evaluated offline in a fixed frozen protocol order. These results are sensitivity analyses and cannot create new confirmatory claims or modify the balanced-WCS conclusion.

6 Results

6.1 Main Balanced-WCS Result

The global balanced-WCS estimate was 0.00575 for PPO-TS minus ES. The weighted-Welch 95% confidence interval was 0.00466 to 0.00683, with $p = 1.20 \times 10^{-9}$. The conditional TrainingSeed bootstrap interval was 0.00480 to 0.00674. The exact hierarchical sign-flip p -value was 0.00295. These three frozen analyses support the prospectively specified conditional balanced-WCS claim.

Figure 3 separates the confirmatory and extrapolation uncertainty targets. The balanced ScenarioTape extrapolation interval was -0.00035 to 0.01187 . Because this

interval includes zero, the algorithm order in the broader ScenarioTape population was not statistically confirmed.

6.2 Cell-Specific Balanced WCS

Seven of nine cell-WCS contrasts passed the Holm family. C002, C005, C006, C007, C008, and C010 had supported favorable directions. C001 had a supported adverse direction, with an estimate of -0.01196 , a 95% confidence interval of -0.01478 to -0.00914 , and Holm-adjusted $p = 4.55 \times 10^{-5}$.

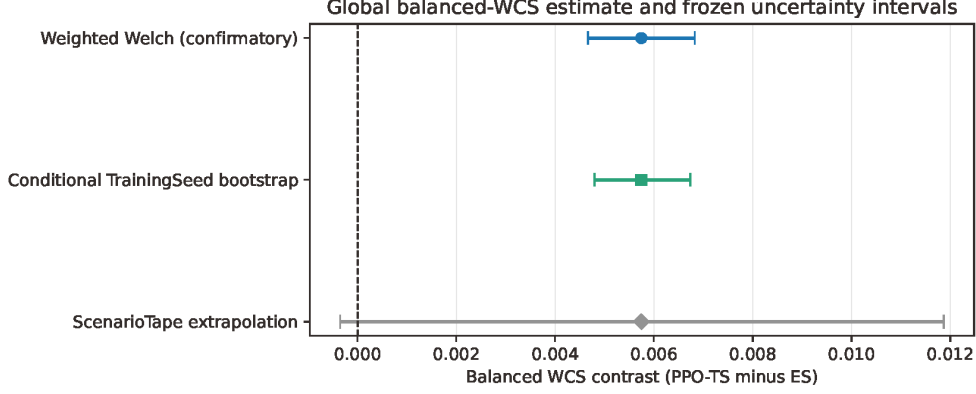


Fig. 3: Global balanced-WCS inference intervals. The weighted-Welch interval serves as primary confirmatory result; TrainingSeed bootstrap is conditional on fixed tapes, and extrapolation interval cannot confirm overall algorithm superiority.

C009 and C011 did not pass Holm adjustment. Their estimates were 0.00371 and 0.00381, respectively, and both had Holm-adjusted $p = 0.0524$. Their positive nominal intervals therefore do not support confirmatory cell-level claims. Figure 4 presents every cell in fixed protocol order and retains the adverse and unsupported results.

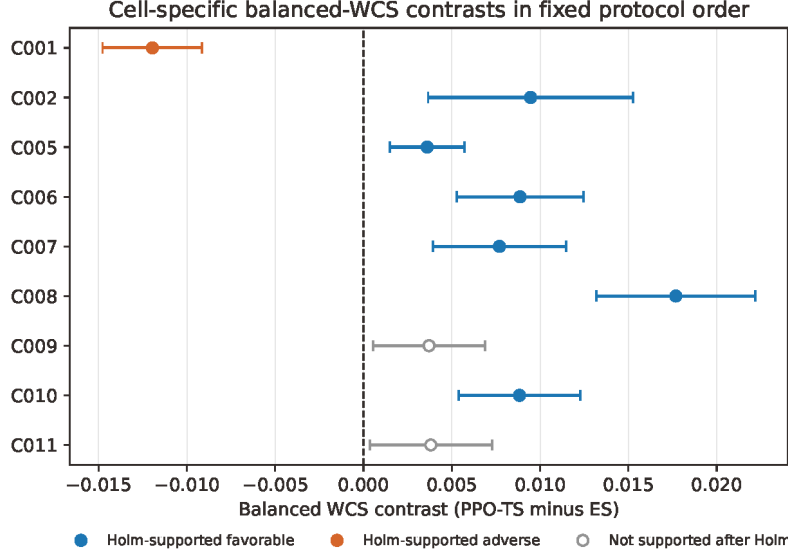


Fig. 4: Cell-specific balanced-WCS contrasts in fixed protocol order. Filled markers denote Holm-supported effects, and open markers denote effects not supported after Holm adjustment. Positive values favor PPO-TS; negative values favor ES.

6.3 Component Trade-Offs

Forty-one of 45 component contrasts passed the component-family Holm procedure. ACT favored PPO-TS in 9/9 cells, and TP favored PPO-TS in 8/9 cells. The TP contrast at C001 was not supported. In contrast, EU was adverse for PPO-TS in 9/9 cells, establishing a consistent utilization trade-off within the frozen evaluation.

TR and IRC were heterogeneous. TR was supported as favorable in C002, C007, C009, and C011, but adverse in C005 and C006. TR/C001 and TR/C008 were unsupported. IRC was supported as favorable in C002, C005, C006, C007, C008, C009, and C011, but adverse in C001. IRC/C010 was unsupported. Thus, the four unsupported component contrasts were TR/C001, TR/C008, IRC/C010, and TP/C001. Figure 5 summarizes direction and support without combining metrics into an algorithm ranking.

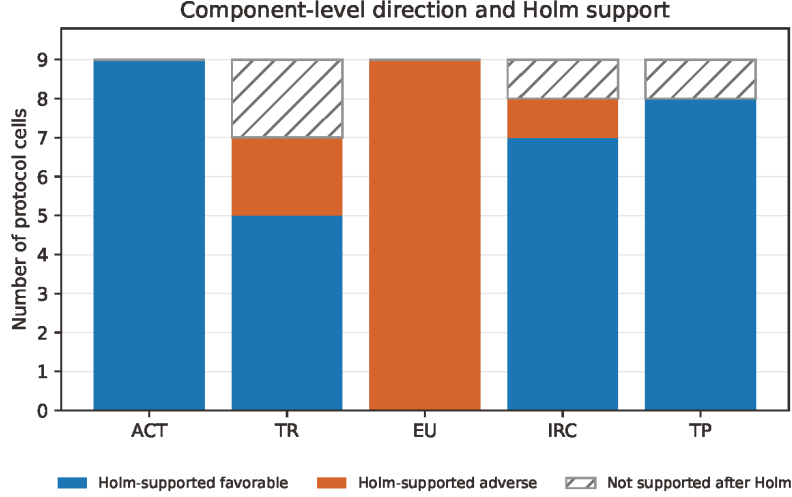


Fig. 5: Component-level direction and Holm support in the nine protocol cells. Each bar accounts for all nine cells. The fixed metric order is ACT, TR, EU, IRC, and TP.

6.4 Mk10 Disturbance Contrast

One of six Mk10 disturbance contrasts passed its Holm family. The PPO-TS-minus-ES WCS advantage at C008 exceeded that at the nominal C005 condition by 0.01409. Its 95% confidence interval was 0.00811 to 0.02006, with Holm-adjusted $p = 0.00285$. This is a relative policy-contrast difference between two cells; it does not establish an absolute performance change caused by the disturbance. The other five Mk10 disturbance contrasts did not pass their Holm family.

6.5 Offline Weight Sensitivity

Offline WCS sensitivity; non-confirmatory. The four schemes were evaluated in their frozen order and were not selected from observed effects. completion_priority was classified as ROBUST_POSITIVE, tardiness_priority as ROBUST_POSITIVE, utilization_priority as PREFERENCE_SENSITIVE, and recovery_priority as ROBUST_POSITIVE. The utilization-priority weighted-Welch interval included zero, showing that the balanced result was not stable under every frozen preference profile. Supplementary Table S1 reports all four estimates and intervals in frozen protocol order. These analyses do not create new confirmatory claims or revise the balanced-WCS result.

7 Discussion

The confirmatory analysis supports a small positive global balanced-WCS contrast for PPO-TS under the frozen conditional design. However, the cell-level results show that this average does not describe every operating condition. C001 was a confirmed adverse

WCS result, while C009 and C011 remained unsupported after Holm adjustment. These findings prevent a universal scenario-level interpretation.

The component results clarify the balanced score. PPO-TS showed consistent ACT benefits and broadly supported throughput benefits, but EU was adverse in every cell. TR and IRC also contained supported adverse directions in specific cells. Consequently, statistical support for the composite score cannot be translated into comprehensive metric improvement.

The magnitude of the global WCS contrast should be interpreted separately from its statistical evidence. No practical-importance threshold was prospectively specified. Therefore, the small estimate and its narrow interval establish direction under the frozen metric, but they do not establish engineering, operational, or economic importance.

Two uncertainty boundaries further constrain interpretation. First, the balanced ScenarioTape extrapolation interval included zero. The current evidence is therefore insufficient to confirm algorithm order in a broader ScenarioTape population. Second, the utilization-priority offline sensitivity interval also included zero. This PREFERENCE.SENSITIVE result shows that the WCS conclusion depends on the frozen operational preference profile, although it does not alter the balanced confirmatory result.

Finally, the study uses synthetic Brandimarte-based scheduling scenarios rather than field deployment data. The simulator captures structured arrivals, urgency, failures, and compatibility, but not every communication delay, observation error, maintenance practice, or multi-center constraint. Operational validation and a prospectively defined practical-importance threshold are needed before deployment claims can be assessed.

8 Conclusions

This study presented PPO-TS, which combines feasible-action masking, priority-guided candidate filtering, and PPO selection for dynamic verification scheduling. Its formal evaluation separated one balanced-WCS confirmatory target from cell, component, disturbance, extrapolation, and weight-sensitivity analyses.

Under the frozen conditional design, the global balanced-WCS contrast favored PPO-TS over ES. The component evidence supported completion-time benefits in all nine cells and throughput benefits in eight, while equipment utilization was adverse in all nine. C001 also showed an adverse balanced-WCS result, and the broader ScenarioTape algorithm order was not statistically confirmed.

The evidence therefore supports a conditional balanced trade-off in the synthetic benchmark, not universal dominance or comprehensive metric improvement. The utilization trade-off, preference sensitivity, and extrapolation uncertainty define the principal boundaries for future operational validation.

Supplementary Information. Supplementary Table S1 reports offline WCS sensitivity schemes. Machine-readable source data for figures and tables are provided.

Acknowledgements. Not applicable.

Declarations

Funding

This work is supported by the Science and Technology project of China Southern Power Grid Co., Ltd. (Grant No. YNKJXM20240384).

Conflict of Interest

The authors declare no conflicts of interest.

Ethics Approval and Consent to Participate

Not applicable.

Consent for Publication

Not applicable.

Data Availability

Synthetic data based on Brandimarte instances are included in the supplementary package. Further inquiries to the corresponding author.

Author Contribution

Conceptualization, Z.H.; methodology, Z.H.; software, C.H.; validation, C.H.; formal analysis, J.Z.; investigation, A.H.; data curation, A.H.; writing—original draft, Z.H.; writing—review and editing, J.Z. and X.Y.; resources, L.G.; supervision, X.Y.; project administration, L.G. and X.Y.

AI Statement

Generative AI assisted language editing and typesetting draft review. All scientific content, design, and analysis were independently verified by the authors.

References

- [1] Tang, J., Gong, G., Peng, N., et al. An effective memetic algorithm for distributed flexible job shop scheduling problem considering integrated sequencing flexibility. *Expert Systems with Applications*, 2024, 242: 122734. doi:10.1016/j.eswa.2023.122734.
- [2] Xie, Y., Chu, K. F., Lam, A. Y. S., et al. A multi-objective evolutionary algorithm with constraint-compliant initialization for energy transport and urban logistics in electric vehicle routing. *Applied Soft Computing*, 2025, 183: 113624. doi:10.1016/j.asoc.2025.113624.

- [3] Li, K., Zhang, T., Wang, R. Deep reinforcement learning for multiobjective optimization. *IEEE Transactions on Cybernetics*, 2021, 51(6): 3103–3114. doi:10.1109/TCYB.2020.2977661.
- [4] Duan, J., Wang, J. Robust scheduling for flexible machining job shop subject to machine breakdowns and new job arrivals considering system reusability and task recurrence. *Expert Systems with Applications*, 2022, 203: 117489. doi:10.1016/j.eswa.2022.117489.
- [5] Zhang, H., Pan, C., Gen, M. A robust bi-objective optimization model and heuristic solution for truck-drone collaboration in humanitarian logistics under truck travel time uncertainty. *Swarm and Evolutionary Computation*, 2026, 101: 102309. doi:10.1016/j.swevo.2026.102309.
- [6] Haupt, R. A survey of priority rule-based scheduling. *OR Spektrum*, 1989, 11(1): 3–16. doi:10.1007/BF01721162.
- [7] Su, X., Li, Y., Zong, N., et al. A real-time network scheduling algorithm based on multi-feature dynamic priority. *Journal on Communications*, 2020, 41(5): 159–167. (in Chinese)
- [8] Lei, T., Hu, X., Zhou, X. A new static-priority online energy-saving scheduling algorithm. *Journal of University of Science and Technology of China*, 2006(2): 219–224. (in Chinese)
- [9] Ahamed, T. P. I., Jasmin, E. A., Al-Ammar, E. Reinforcement learning in power system scheduling and control: A unified perspective. In: *2011 IEEE Symposium on Computers & Informatics*, 2011: 650–655. doi:10.1109/ISCI.2011.5958993.
- [10] Xu, H., Wu, J., Pan, Q., et al. Digital twin and meta RL empowered fast-adaptation of joint user scheduling and task offloading for mobile industrial IoT. *IEEE Journal on Selected Areas in Communications*, 2023, 41(10): 3254–3266. doi:10.1109/JSAC.2023.3310081.
- [11] Yuan, Y., Zheng, G., Wong, K. K., et al. Meta-reinforcement learning based resource allocation for dynamic V2X communications. *IEEE Transactions on Vehicular Technology*, 2021, 70(9): 8964–8977. doi:10.1109/TVT.2021.3098854.
- [12] Li, K., Xiong, S., Dai, P., et al. A timeliness optimization method for real-time dynamic satellite task scheduling based on deep reinforcement learning. *Scientia Sinica Informationis*, 2024, 54(10): 2443–2469. (in Chinese)
- [13] Li, X., Ren, X., Zhou, Y., et al. Flexible integrated scheduling with stochastic processing times based on an improved D3QN algorithm. *Journal of System Simulation*, 2025, 37(2): 474–486. (in Chinese)

- [14] Sun, A., Lei, Q., Song, Y. Solving joint scheduling of job-shop machines and AGVs based on deep reinforcement learning. *Control and Decision*, 2024, 39(1): 253–262. (in Chinese)
- [15] Wang, L., Wang, C., Li, X., et al. Multi-agent deep reinforcement learning for dynamic scheduling of distributed heterogeneous job shops. *Computer Integrated Manufacturing Systems*, 2025, 31(12): 4608–4620. (in Chinese)
- [16] Shi, J., Qiao, F., Ma, Y., et al. Production-logistics collaborative scheduling in dynamic flexible job shops via multi-objective deep reinforcement learning. In: *2024 IEEE International Conference on Systems, Man, and Cybernetics*, 2024: 5123–5128. doi:10.1109/SMC54092.2024.10831278.
- [17] Qiao, Y., Miao, J., Huang, X. Carbon-aware logistics routing via decomposition-guided deep reinforcement learning and parameter transfer. *Sustainable Computing: Informatics and Systems*, 2026, 50: 101314. doi:10.1016/j.suscom.2026.101314.
- [18] Chandrasiri, S., Meedeniya, D. Multi-objective cloud workflow scheduling with two-stage deep reinforcement learning and graph neural networks. *IEEE Access*, 2026, 14: 34802–34829. doi:10.1109/ACCESS.2026.3669772.
- [19] Dong, H., Lu, T. Solving flexible job shop scheduling based on DRL-PPO. *Application Research of Computers*, 2025, 42(9): 2722–2729. (in Chinese)
- [20] Wang, H., Li, C., Li, T. A reinforcement learning method with dual-layer attention networks for flexible job-shop scheduling. *Computer Science*, 2026, 53(1): 231–240. (in Chinese)
- [21] Wang, Y., Fu, W., Zhang, J., et al. Flexible job-shop scheduling based on an improved proximal policy optimization algorithm. *Control and Decision*, 2025, 40(6): 1883–1891. (in Chinese)
- [22] Guo, Z., Wang, J., Gong, H., et al. Streaming task scheduling integrating deep reinforcement learning and operator optimization. *Journal of Zhengzhou University (Natural Science Edition)*, 2025, 57(1): 15–22. (in Chinese)
- [23] Chen, J., Wang, Y., Wu, Z., et al. Cloud-edge collaborative task migration and resource redistribution optimization based on deep reinforcement learning. *Computer Science*, 2024, 51(S2): 713–722. (in Chinese)
- [24] Zhou, M., Wan, K., Fei, J., et al. Research on deep reinforcement learning based adaptive job-shop scheduling. In: *2025 International Conference on Cyber-Physical Social Intelligence*, 2025: 1–6. doi:10.1109/CPSI66656.2025.11343891.
- [25] Sutton, R. S., Barto, A. G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.

- [26] Huang, S., Ontañón, S. A closer look at invalid action masking in policy gradient algorithms. *The International FLAIRS Conference Proceedings*, 2022, 35. doi:10.32473/flairs.v35i.130584.
- [27] Panwalkar, S. S., Iskander, W. A survey of scheduling rules. *Operations Research*, 1977, 25(1): 45–61. doi:10.1287/opre.25.1.45.
- [28] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. Proximal policy optimization algorithms. arXiv, 2017, arXiv:1707.06347. doi:10.48550/arXiv.1707.06347.
- [29] OECD, European Union, Joint Research Centre–European Commission. *Handbook on Constructing Composite Indicators: Methodology and User Guide*; OECD Publishing: Paris, France, 2008. doi:10.1787/9789264043466-en.
- [30] Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, 1993, 41(3): 157–183. doi:10.1007/BF02023073.
- [31] Welch, B. L. The generalization of “Student’s” problem when several different population variances are involved. *Biometrika*, 1947, 34(1–2): 28–35. doi:10.1093/biomet/34.1-2.28.
- [32] Satterthwaite, F. E. An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 1946, 2(6): 110–114. doi:10.2307/3002019.
- [33] Holm, S. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 1979, 6(2): 65–70. doi:10.2307/4615733.