Benchmarking Multi-Agent Reinforcement Learning for Stochastic OSAT Scheduling: A Reproducible Computational Study for Semiconductor Operations Management

Mai Ngoc Huy*

Doctor of Philosophy Program, KIMT University

***Corresponding Author**

DOI: https://doi.org/10.47772/IJRISS.2026.100800210

Received: 19 August 2026; Accepted: 24 August 2026; Published: 31 August 2026

ABSTRACT

Outsourced Semiconductor Assembly and Test (OSAT) operations require repeated scheduling decisions across interdependent stages under processing-time variability, equipment failures, product-mix changes, and demand uncertainty. This study evaluates whether alternative multi-agent reinforcement-learning (MARL) coordination structures provide measurable short-horizon scheduling benefits when compared in a common stochastic OSAT environment. Four learned policy structures—H-MARL-CTDE, H-MARL-Independent, H-MARL-Value, and Flat-MARL—were evaluated against FIFO, SPT, and EDD. The executable simulator represents Die Attach, Wire Bond, Encapsulation, Final Test, and Quality Control, advances in five-minute steps over six simulated hours, and uses active stochastic arrivals, product sampling, lognormal processing-time variation, and MTBF/MTTR-based failure and repair dynamics. The final matched design comprises 27 variability-demand-failure scenarios, ten replications, seven policies, and 1,890 run-level observations. Policy differences were assessed using matched-block Friedman tests and paired Wilcoxon signed-rank tests with Holm correction. H-MARL-Independent achieved the highest mean throughput (16.4932 units/hour) and the lowest energy per completed unit (3.8375), while H-MARL-CTDE achieved the highest OEE proxy (53.690%) and a mean flow time of 0.5848 h. SPT achieved the lowest flow time (0.5838 h). Global policy differences were significant for throughput, flow time, energy efficiency, and OEE (all $p<.001$), but not makespan ($p=.987$). CTDE and Independent significantly outperformed FIFO on the four discriminating KPIs after Holm correction, but neither consistently outperformed SPT. The results therefore support selective, KPI-specific learned-policy benefits rather than universal MARL superiority. The primary contribution is an auditable common benchmark that separates computational evidence from production claims and provides a basis for longer-horizon, multi-seed, factory-calibrated, and hybrid RL-heuristic validation in semiconductor operations management.

Keywords: OSAT manufacturing; multi-agent reinforcement learning; semiconductor scheduling; operations management; computational benchmarking

INTRODUCTION

Semiconductor back-end production combines assembly, packaging, testing, and quality activities whose queues and resource states are interdependent. Scheduling decisions therefore operate under heterogeneous equipment, stochastic processing times, equipment failures and repairs, product-mix variation, due-date pressure, and shifting demand. These characteristics make OSAT scheduling a sequential, multi-objective decision problem rather than a static sequencing exercise. Classical scheduling theory remains essential for interpretable baselines, while reinforcement learning (RL) provides a mechanism for learning state-dependent decisions from interaction with a modeled environment [13,19].

RL has already been applied to several semiconductor back-end subproblems. Park and Park [12] demonstrated deep-RL scheduling for semiconductor packaging facilities; Zhang et al. [21] formulated semiconductor final-test scheduling with Sarsa; Cao et al. [2] combined RL with cuckoo search and surrogate

modeling for a semiconductor testing facility; Lin et al. [8] used Q-learning as a high-level hyper-heuristic for final testing; Zhang et al. [22] combined Q-learning with an estimation-of-distribution algorithm; and Chiu et al. [3] integrated deep RL with simulation optimization for backend assembly. A bibliographically verified scoping review undertaken as part of the underlying doctoral research contained 84 records, but only these six (7.1%) directly addressed OSAT or semiconductor back-end RL. The direct evidence is therefore concentrated in packaging, testing, and selected backend-assembly problems rather than integrated multi-stage coordination.

The principal gap is architectural comparison. Independent learning, centralized training with decentralized execution (CTDE), value-factorization, flat MARL, and hierarchical MARL make different assumptions about information sharing, credit assignment, and decomposition [6,9,14,18]. Yet the verified OSAT literature did not identify a controlled study comparing these coordination structures under the same stochastic operating conditions. Reproducibility is a related limitation: published studies often use study-specific problem formulations, simulators, and validation protocols [10,11].

This study addresses that gap with a common executable OSAT scheduling benchmark. Four learned policy structures—H-MARL-CTDE, H-MARL-Independent, H-MARL-Value, and Flat-MARL—are compared with FIFO, shortest processing time (SPT), and earliest due date (EDD). The contribution is deliberately bounded: the study evaluates learned scheduling behavior inside a six-hour stochastic simulation and does not claim a factory-calibrated digital twin or production-ready autonomy. This distinction is consistent with real-world RL guidance emphasizing validation, distribution shift, safety, monitoring, and staged deployment before operational control [4,17].

The study asks a practical operations-management question: when all policies face the same variability, demand, failure settings, and matched random replications, do more sophisticated learned coordination structures deliver repeatable benefits over simple dispatching rules? The answer is evaluated using operational KPIs, matched nonparametric inference, scenario diagnostics, and explicit reporting of negative as well as positive results.

Related Work And Research Gap

RL and MARL for manufacturing scheduling

An MDP represents sequential decisions through states, actions, transitions, rewards, and a policy that seeks to maximize expected return [19]. Hierarchical reinforcement learning introduces temporal or organizational abstraction, allowing a higher-level policy to guide lower-level behaviors [1]. Multi-agent reinforcement learning distributes decisions among interacting agents; CTDE allows broader information during training while retaining decentralized decision structures at execution [9]. Value-decomposition approaches, including VDN and QMIX, address cooperative credit assignment by relating local agent values to team-level value [14,18].

The present benchmark uses proximal policy optimization (PPO) as a common optimization backbone for the learned policy families. PPO constrains policy updates through a clipped surrogate objective and is paired with generalized advantage estimation to control the bias-variance trade-off in advantage estimation [15,16]. A common optimizer reduces confounding between coordination structure and unrelated learning algorithms.

OSAT-specific gap

The six directly relevant OSAT/back-end studies establish that RL can contribute to packaging, final-test, testing-facility, and backend-assembly scheduling [2,3,8,12,21,22]. However, none of the six provides a common multi-stage benchmark spanning independent, CTDE-oriented, value-oriented, flat, and hierarchical structures. The gap is therefore not whether RL can be used in semiconductor scheduling; rather, it is how alternative coordination structures compare when environmental conditions, evaluation seeds, and KPI definitions are held constant.

A second gap concerns industrial translation. Reviews of deep RL in production systems and recent semiconductor deployment work emphasize that simulation performance does not itself establish external validity [10,11,17]. Factory deployment requires calibration to operational data, robust evaluation across changing conditions, monitoring, safe fallbacks, and human oversight. Accordingly, the present work treats reproducibility and falsifiability as first-order design requirements: heuristic baselines are permitted to outperform learned policies, non-significant results are retained, and the evidence is bounded to the simulator population.

METHODOLOGY

Executable OSAT environment

The final environment is an executable stochastic scheduling simulation representing five modeled OSAT stages: Die Attach, Wire Bond, Encapsulation, Final Test, and Quality Control. Time advances in five-minute steps. Jobs arrive according to a Poisson process whose rate is modified by demand and variability settings; arriving product classes are sampled from a configured product mix. Stage processing durations are sampled from lognormal distributions around stage-specific base times. Equipment failures are generated from stage MTBF-based probabilities and repair durations are sampled around configured MTTR values. Machines explicitly occupy idle, busy, or down states, and jobs progress through stage queues under the selected policy actions.

The manager component selects one of three high-level goals—FLOW, DUE_DATE, or THROUGHPUT—while worker actions select among FIFO, SPT, and EDD dispatch primitives. The reward combines completed jobs with penalties for work-in-process, overdue work, and incremental energy, with weighting conditioned by the manager goal. Learned policies therefore choose actual scheduling actions from observed state rather than receiving fixed policy-specific performance multipliers.

Policy structures

Four learned structures and three deterministic baselines were evaluated. The labels describe compact research implementations and should not be interpreted as canonical industrial reproductions of every referenced MARL algorithm.

Policy	Structure	Role in benchmark
H-MARL-CTDE	Manager actor, shared worker actor, centralized critic	Tests a compact CTDE-oriented manager-worker structure.
H-MARL-Independent	Separate worker actors/critics plus manager actor/critic	Tests decentralized learning without an equivalent centralized coordination critic.
H-MARL-Value	Compact value-decomposition-inspired structure	Tests value-oriented cooperation; not a canonical VDN/QMIX reproduction.
Flat-MARL	Global observation mapped to manager and stage actions without explicit hierarchy	Controls for whether learning itself, rather than hierarchy, explains performance.
FIFO	First-in-first-out dispatch	Arrival-order heuristic baseline.
SPT	Shortest-processing-time dispatch	Strong flow-time-oriented heuristic baseline.
EDD	Earliest-due-date dispatch	Due-date-oriented heuristic baseline; limited by the short horizon.

Table 1. Policy structures evaluated in the common OSAT environment.

Training and evaluation boundary

Each learned architecture was trained for 1,000 six-hour episodes. The final cross-policy evaluation used stored learned checkpoints under one corrected terminal boundary. CTDE, Flat-MARL, and Value were evaluated from archived checkpoints; Independent was retrained after correcting PPO probability-ratio bookkeeping and diagnostic logging. Complete episode-level training history is preserved only for Independent, so convergence claims are restricted to that architecture. This asymmetric evidence is retained explicitly rather than inferred from checkpoint existence.

The six-hour terminal boundary is intentionally strict. Residual work-in-process is not drained after the horizon and no extra arrival is injected at the terminal boundary. This avoids contaminating policy-specific KPIs with post-horizon dispatch under another rule. The compact horizon contains approximately 72 decision epochs and exposes repeated arrivals, queue formation, processing variability, failures, repairs, energy accumulation, and dispatch decisions, but it is not a substitute for multi-day or production-week behavior.

Experimental design

The legacy condition grid was collapsed to 27 unique physical scenarios formed by three variability levels, three demand labels, and three failure levels. Every policy was evaluated in every physical scenario using ten matched replication identifiers. The final design therefore contains 7 policies × 27 scenarios × 10 replications = 1,890 run-level observations, or 270 observations per policy. Matching means that policy comparisons within a block share the same nominal variability, demand, failure condition, and replication index.

Factor	Levels / design
Policies	FIFO; SPT; EDD; Flat-MARL; H-MARL-Independent; H-MARL-CTDE; H-MARL-Value
Variability	Low; Medium; High
Demand	Steady; Seasonal; Surge
Failure	Low; Medium; High
Unique physical scenarios	3 × 3 × 3 = 27
Replications	10 matched replications per policy-scenario cell
Canonical observations	7 × 27 × 10 = 1,890

Table 2. Matched factorial evaluation design.

Performance measures and statistical analysis

The primary KPIs are throughput (higher is better), makespan (lower), mean flow time (lower), energy per completed job (lower), and an OEE proxy (higher). Tardiness is retained in the output schema but is not a discriminating endpoint: the configured 168-hour lead time is far beyond the six-hour experiment, so all 1,890 tardiness observations are zero. The OEE value is a simulation proxy used consistently for cross-policy comparison; it is not presented as a certified factory OEE report.

Descriptive statistics include means, standard deviations, coefficients of variation, and 95% confidence intervals. Because the seven policies are observed on the same scenario-replication blocks, the Friedman rank test is the primary omnibus comparison [5]. Where global policy differences are present, paired Wilcoxon signed-rank tests [20] compare learned policies with baselines and Holm's procedure controls family-wise error across planned comparison families [7]. Factorial ordinary-least-squares models provide complementary

diagnostics for policy, variability, demand, failure, and selected policy interactions; these parametric models support interpretation but do not replace the matched nonparametric tests.

RESULTS

Data integrity and overall performance

The canonical evaluation contains exactly 1,890 rows reconstructed from four learned-policy evaluation files (270 rows each) and one heuristic evaluation file (810 rows). The final audit identified no exact duplicate rows. Each of the seven policies has 270 observations and complete coverage of the 27 physical scenarios and ten matched replications.

Policy	Throughput	Makespan (h)	Flow time (h)	Energy/unit	OEE proxy (%)
EDD	16.250 ± 0.959	5.868 ± 0.064	0.695 ± 0.161	3.886 ± 0.185	52.799 ± 3.841
FIFO	16.290 ± 0.966	5.868 ± 0.063	0.698 ± 0.167	3.879 ± 0.186	52.844 ± 3.928
Flat-MARL	16.301 ± 1.120	5.868 ± 0.065	0.656 ± 0.122	3.877 ± 0.240	53.182 ± 4.504
H-MARL-CTDE	16.482 ± 1.076	5.868 ± 0.063	0.585 ± 0.074	3.840 ± 0.202	53.690 ± 4.124
H-MARL-Independent	16.493 ± 1.006	5.869 ± 0.063	0.592 ± 0.079	3.838 ± 0.184	53.572 ± 3.723
H-MARL-Value	16.243 ± 1.084	5.867 ± 0.066	0.705 ± 0.171	3.889 ± 0.215	52.811 ± 4.128
SPT	16.434 ± 1.068	5.868 ± 0.064	0.584 ± 0.069	3.848 ± 0.208	53.685 ± 4.189

Table 3. Overall policy performance (mean ± standard deviation; n = 270 per policy). Tardiness is omitted because all values are zero.

The descriptive ranking is multi-objective rather than dominated by one policy. H-MARL-Independent has the highest mean throughput (16.4932 units/hour) and lowest mean energy per completed unit (3.8375). H-MARL-CTDE has the highest OEE proxy (53.690%) and a mean flow time of 0.5848 h, while SPT has the lowest mean flow time at 0.5838 h. H-MARL-Value has the numerically lowest makespan, but makespan means differ by only about 0.002 h across policies and the global test is non-significant.

Relative to FIFO, H-MARL-Independent improves throughput by 1.25%, flow time by 15.16%, energy per completed unit by 1.08%, and OEE by 1.38%; its makespan change is effectively zero. H-MARL-CTDE improves throughput by 1.18%, flow time by 16.22%, energy per unit by 1.02%, and OEE by 1.60%, again without a meaningful makespan change. SPT is also strong, improving flow time by 16.36% and OEE by 1.59% relative to FIFO. These percentages are computational prototype effects and are not interpreted as measured factory savings.

Matched inferential results

KPI	Friedman $\chi^2(6)$	p-value	Conclusion
Throughput	44.345	<.001	Significant
Makespan	0.954	.987	Not significant
Flow time	267.456	<.001	Significant

Energy efficiency	57.919	<.001	Significant
OEE	36.849	<.001	Significant

Table 4. Global matched-block policy tests.

Friedman tests reject equality among the seven policies for throughput, flow time, energy efficiency, and OEE, but not makespan. Complementary factorial models identify small policy effects for throughput, energy efficiency, and OEE, a larger policy effect for flow time, and no meaningful policy effect for makespan. This pattern emphasizes the distinction between statistical repeatability and practical magnitude.

Comparison vs FIFO	**Throughput**	**Flow time**	**Energy/unit**	**OEE**
H-MARL-Independent	p=.0080	p<.001	p=.0027	p=.0148
H-MARL-CTDE	p=.0335	p<.001	p=.0077	p=.0148

Table 5. Holm-corrected paired Wilcoxon results for the two leading learned policies versus FIFO. Makespan was non-significant for both.

Both H-MARL-Independent and H-MARL-CTDE significantly outperform FIFO on throughput, flow time, energy efficiency, and OEE after Holm correction. However, neither learned structure significantly outperforms SPT on throughput, energy efficiency, or OEE, and SPT retains the lowest mean flow time. The evidence therefore supports baseline-specific learned-policy advantages rather than universal superiority.

Scenario effects and stability qualification

Variability significantly affects throughput (p<.001), flow time (p=.0011), energy efficiency (p<.001), and OEE (p<.001). Demand significantly affects makespan (p=.033), flow time (p=.0069), and OEE (p=.0012), but these demand effects are qualified because the six-hour horizon observes only the initial portion of the configured long-horizon demand trajectories. Failure significantly affects makespan (p<.001) but is not significant for the other four discriminating KPIs in the final factorial model.

No policy×variability or policy×failure interaction is statistically significant across the five discriminating KPIs. This supports descriptive stability within the tested factor grid but does not establish industrial robustness. Average rank across the five discriminating KPIs places H-MARL-CTDE and SPT in a tie for first (mean rank 2.5), followed by H-MARL-Independent (3.0), Flat-MARL (3.9), EDD (5.2), FIFO (5.3), and H-MARL-Value (5.6).

DISCUSSION

Selective learned-policy benefit rather than universal dominance

The main result is not that RL defeats heuristics; it is that coordination architecture affects selected KPIs under a controlled common environment. CTDE and Independent show repeatable advantages over FIFO, but SPT remains highly competitive. This outcome is scientifically useful because a valid benchmark must permit simple policies to win when their structural bias matches the metric. In a short horizon, SPT immediately exploits processing-time information and therefore has a direct mechanism for reducing flow time [13].

The effect sizes also matter. Throughput, energy-per-unit, and OEE improvements relative to FIFO are around one percent, whereas flow-time improvements are approximately fifteen to sixteen percent. A one-percent throughput shift can be operationally meaningful in high-volume manufacturing, but its business value depends on bottleneck location, demand, yield, utilization, and downstream capacity. The present experiment therefore does not translate computational percentage differences into monetary or sustainability claims without site-specific calibration.

H-MARL-Value is an informative negative result. It does not consistently improve the reported KPIs relative to FIFO and has the weakest average multi-KPI rank. This should not be read as evidence against value factorization generally. The implementation is value-decomposition-inspired rather than a canonical VDN or QMIX implementation, the training budget is finite, and the horizon is short [14,18]. The result instead identifies a concrete architecture that warrants ablation and stronger canonical comparison.

Implications for hybrid semiconductor scheduling

Strong heuristic performance points toward a hybrid control architecture rather than an all-neural replacement strategy. The manager-worker formulation already treats FIFO, SPT, and EDD as low-level dispatch primitives. A future learned manager could select among an expanded rule library—including setup-aware, qualification-aware, maintenance-aware, and due-date-sensitive rules—based on system state. Similar hyper-heuristic logic has already been demonstrated in semiconductor final testing [8,22].

Hybridization can also improve deployment safety. Real factories expose a learned scheduler to state distributions, exceptions, and governance requirements that are not fully represented in simulation. Rule-based fallback behavior can provide bounded decisions when confidence is low, monitoring detects drift, constraints are violated, or an operator overrides the recommendation. Recent semiconductor RL deployment literature explicitly emphasizes validation, monitoring, human involvement, and the combination of classical and learned scheduling mechanisms [17].

Operations-management implications and Vietnam context

The computational benchmark is relevant to semiconductor operations management because it formalizes how alternative decision structures trade off throughput, flow, resource efficiency, and equipment-productivity proxies under uncertainty. It also demonstrates why advanced scheduling should be evaluated against strong operations-research baselines rather than weak controls. The practical contribution is therefore a reproducible decision-evaluation workflow, not a claim that one learned policy should immediately replace established dispatch rules.

Vietnam provides an important application context in the underlying doctoral research because of the country's expanding semiconductor assembly, packaging, and test activities. However, no proprietary Vietnamese factory logs are used in the reported experiment, and the simulator is not calibrated to a specific Vietnamese site. The article therefore treats Vietnam as a future validation context, not as evidence that the observed KPI differences will transfer unchanged to local production. Factory-level inference requires governed operating data, longer horizons, and staged validation.

Limitations And Future Research

The study has six principal limitations. First, the six-hour horizon is a computational proof-of-concept boundary and does not represent full multi-shift or weekly operation. Second, the configured 168-hour lead time makes tardiness non-discriminating within six hours. Third, the long-horizon Surge event begins outside the observed window, and the Seasonal trajectory is only partially observed. Fourth, operating parameters are synthetic/configuration-derived rather than calibrated from proprietary factory histories. Fifth, complete episode-level training history is preserved only for Independent, while CTDE, Flat-MARL, and Value are represented by checkpoints and evaluation outputs. Sixth, CTDE and value-oriented policies are compact research implementations rather than canonical industrial stacks.

Future research should extend training and evaluation progressively to 24 h, 72 h, 168 h, and multi-week horizons; align due-date settings with the evaluation period; add product-dependent routes, setup/changeover matrices, batching, rework/retest loops, maintenance calendars, and WIP constraints; and calibrate stochastic distributions using anonymized factory data where governance permits. All architectures should be retrained across multiple independent seeds with complete reward, loss, entropy, and checkpoint histories. Canonical CTDE and VDN/QMIX variants should be placed behind the same environment interface and evaluated with ablation and reward-sensitivity studies.

Industrial translation should proceed from offline replay to shadow-mode recommendations, human-in-the-loop review, limited pilot deployment, and only then controlled closed-loop operation. This staged pathway is consistent with broader real-world RL concerns regarding distribution shift, safety, monitoring, and operational accountability [4,17].

CONCLUSION

This study provides a matched, reproducible comparison of four learned scheduling structures and three classical heuristics in a common stochastic OSAT environment. Across 1,890 run-level observations, policy choice significantly affects throughput, flow time, energy efficiency, and the OEE proxy, but not makespan. H-MARL-Independent leads mean throughput and energy efficiency; H-MARL-CTDE leads OEE and nearly matches the best flow time; and SPT achieves the lowest mean flow time. Holm-corrected paired tests show that CTDE and Independent improve the four discriminating KPIs relative to FIFO, while neither consistently surpasses SPT.

The result is a qualified but useful finding: learned coordination can add value, yet architectural complexity does not guarantee universal superiority. The strongest contribution is an auditable benchmark that allows positive, negative, and non-significant outcomes to emerge under matched stochastic conditions. For semiconductor operations management, this supports a future path based on hybrid RL-heuristic control, longer and multi-seed evaluation, factory calibration, and staged deployment rather than immediate replacement of established dispatching practice.

Data Availability

The executable code, configuration files, trained checkpoints, per-policy evaluation outputs, canonical 1,890-row dataset, statistical report generator, figures, and integrity manifests are archived in the public project repository: https://github.com/plmaihuy-cloud/OSAT_RL. The repository should be cited together with the version-of-record used for the submitted manuscript.

Ethical Considerations

The reported study is a computational simulation and literature-based analysis. It includes no human participants, no animal subjects, no completed Delphi participant dataset, and no proprietary factory production logs. Accordingly, no human- or animal-subject intervention is reported in this manuscript.

Conflict Of Interest

The author declares no conflict of interest.

REFERENCES

1. Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. Discrete Event Dynamic Systems, 13, 41–77. https://doi.org/10.1023/A:1022140919877
2. Cao, Z., Lin, C., Zhou, M., & Huang, R. (2019). Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling. IEEE Transactions on Automation Science and Engineering, 16(2), 825–837. https://doi.org/10.1109/TASE.2018.2862380
3. Chiu, C.-C., Lai, C.-M., Liao, Y.-S., & Yeh, W.-C. (2026). Integration of deep reinforcement learning with simulation optimization applied to semiconductor backend assembly scheduling problem. Swarm and Evolutionary Computation, 100, 102252. https://doi.org/10.1016/j.swevo.2025.102252
4. Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., & Hester, T. (2021). Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. Machine Learning, 110, 2419–2468. https://doi.org/10.1007/s10994-021-05961-4
5. Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association, 32(200), 675–701. https://doi.org/10.1080/01621459.1937.10503522

6. Ghavamzadeh, M., Mahadevan, S., & Makar, R. (2006). Hierarchical multi-agent reinforcement learning. Autonomous Agents and Multi-Agent Systems, 13(2), 197–229. https://doi.org/10.1007/s10458-006-7035-4
7. Holm, S. (1979). A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6(2), 65–70.
8. Lin, J., Li, Y.-Y., & Song, H.-B. (2022). Semiconductor final testing scheduling using Q-learning based hyper-heuristic. Expert Systems with Applications, 187, 115978. https://doi.org/10.1016/j.eswa.2021.115978
9. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in Neural Information Processing Systems, 30, 6379–6390.
10. Mayerhoff, J., & Schmidt, M. (2026). Reinforcement learning for autonomous production planning and control: A systematic literature review. Journal of Manufacturing Systems, 86, 546–568. https://doi.org/10.1016/j.jmsy.2026.03.023
11. Panzer, M., & Bender, B. (2022). Deep reinforcement learning in production systems: A systematic literature review. International Journal of Production Research, 60(13), 4316–4341. https://doi.org/10.1080/00207543.2021.1973138
12. Park, I. B., & Park, J. (2023). Scalable scheduling of semiconductor packaging facilities using deep reinforcement learning. IEEE Transactions on Cybernetics, 53(6), 3518–3531. https://doi.org/10.1109/TCYB.2021.3128075
13. Pinedo, M. L. (2022). Scheduling: Theory, algorithms, and systems (6th ed.). Springer. https://doi.org/10.1007/978-3-031-05921-6
14. Rashid, T., Samvelyan, M., Schroeder de Witt, C., Farquhar, G., Foerster, J., & Whiteson, S. (2018). QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. Proceedings of the 35th International Conference on Machine Learning, 80, 4295–4304. https://proceedings.mlr.press/v80/rashid18a.html
15. Schulman, J., Moritz, P., Levine, S., Jordan, M. I., & Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. International Conference on Learning Representations. https://arxiv.org/abs/1506.02438
16. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv. https://doi.org/10.48550/arXiv.1707.06347
17. Stöckermann, P., Feudel, S., Immordino, A., Hayen, N., Altenmüller, T., Gebser, M., Schekotihin, K., Seidel, G., Wegmann, M., & Higgins, F. (2025). Reinforcement learning based dispatching solutions in semiconductor manufacturing: A literature review on validation and deployment. Production & Manufacturing Research, 13(1), 2582472. https://doi.org/10.1080/21693277.2025.2582472
18. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., & Graepel, T. (2018). Value-Decomposition Networks for cooperative multi-agent learning based on team reward. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2085–2087.
19. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.
20. Wilcoxon, F. (1945). Individual comparisons by ranking methods. Biometrics Bulletin, 1(6), 80–83. https://doi.org/10.2307/3001968
21. Zhang, Z., Zheng, L., Hou, F., & Li, N. (2011). Semiconductor final test scheduling with Sarsa(λ, k) algorithm. European Journal of Operational Research, 215(2), 446–458. https://doi.org/10.1016/j.ejor.2011.05.052
22. Zhang, L., Lin, Y., Xu, C., & Liu, M. (2024). A new EDA algorithm combined with Q-learning for semiconductor final testing scheduling problem. Computers & Industrial Engineering, 193, 110259. https://doi.org/10.1016/j.cie.2024.110259