Technische
Hochschule
Rosenheim

Faculty of Computer Science

Study Program

Reinforcement Learning for Scheduling: A Systematic Literature Review

Master Thesis

von

Max Mustermann

Date of Submission: dd.mm.yyyy
First Examiner: Prof. Dr. N. N.
Second Examiner: Prof. Dr. N. N.

Declaration of Originality

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Place, November 20, 2025

Full Name

Kurzfassung

text

Schlagworte:

Contents

List of Figures

List of Tables

Listings

1 Introduction

Provide the motivation for reinforcement learning (RL) in scheduling, outline objectives and research questions, and summarize contributions and scope. Link to the systematic review protocol for transparency.

1.1 Context and Motivation

1.2 Objectives and Research Questions

1.3 Contributions and Thesis Structure

2 Background

Summarize scheduling fundamentals and RL essentials to set common ground.

2.1 Scheduling Fundamentals

Job shop, flow shop, flexible job shop, parallel machine, hybrid flow shop; deterministic vs. stochastic vs. dynamic arrivals; constraints (due dates, setup times, resource calendars).

2.2 Reinforcement Learning Essentials

MDPs, policies, value-based vs. policy-gradient vs. model-based RL; on-policy vs. off-policy; exploration strategies; multi-objective settings.

2.3 Benchmarks and Metrics

Makespan, tardiness, flow time, energy; OR baselines (dispatching rules, MILP/CP, metaheuristics); RL evaluation norms.

Planned figure Taxonomy of scheduling problem classes.

Planned table Metrics and baseline families used in RL scheduling studies.

3 Methodology

Describe the systematic literature review protocol and data-extraction process.

3.1 Search Strategy

Databases: IEEE Xplore, ACM Digital Library, Scopus, Web of Science, Google Scholar (for snowballing). Time window: 2016–2025. Search strings (examples, adapted per indexer): `(řeinforcement learningÖR d̈eep reinforcement learningÖR DQN OR PPO OR SAC) AND (scheduling OR j̈ob shopÖR f̈low shopÖR p̈roduction schedulingÖR d̈ispatchingÖR p̈roduction planningÖR čloud schedulingÖR ëdge scheduling)̈`. Apply backward/-forward snowballing on key papers and the provided surveys. Screening follows PRISMA-style phases: deduplicate, title/abstract screen, full-text eligibility, inclusion.

3.2 Inclusion and Exclusion Criteria

- **Inclusion**: peer-reviewed conference/journal papers (2016–2025) applying RL/DRL to scheduling, dispatching, production planning/control, cloud/edge scheduling; reports quantitative results against baselines.
- **Exclusion**: non-RL approaches, purely conceptual with no evaluation, non-English, inaccessible full text, duplicates.

3.3 Quality Assessment

Baseline strength, reproducibility (code/data), statistical validity (multiple seeds, confidence intervals), clarity of environment/problem specification, constraint handling.

3.4 Data Extraction Schema

Problem type, environment, state/action/reward, algorithm, baselines, metrics, constraints, generalization tests, code availability.

Planned figure PRISMA flow diagram for study selection.

Planned table Data-extraction codebook.

4 Methodology (Draft Text)

This placeholder will be replaced by the full protocol once screening counts are known. To include: finalized search strings per database, PRISMA counts (identification/screening/eligibility/inclusion), justification for the 2016–2025 window, and the data-extraction schema aligned to the literature matrix columns. Add a PRISMA diagram and a summary table of quality assessment criteria (baseline strength, reproducibility, statistical validity, constraint handling).

5 RL Methods for Scheduling: Taxonomy

Organize the landscape of RL approaches tailored to scheduling.

5.1 Value-Based Methods

DQN/DDQN/Dueling, distributional variants; action masking for constraints.

5.2 Policy-Gradient and Actor-Critic Methods

A2C/A3C, PPO, SAC, deterministic policy gradients.

5.3 Model-Based and Simulation-Augmented RL

World models, lookahead, Dyna-style, differentiable simulators.

5.4 Meta-RL, Transfer, and Curriculum Learning

5.5 State, Action, Reward Design Patterns

Graph/state encodings, machine/job-centric actions, reward shaping for due dates/setups; constraint handling (penalties, masking, Lagrangian). *Progress note:* Initial extraction shows strong use of graph encodings (disjunctive graphs, GNN dual-attention) and action masking for feasibility in JSS/FJSS. Rewards are typically weighted makespan/tardiness, with penalties for constraint violations.

Planned figure Taxonomy diagram: methods vs. scheduling settings.

Planned table State/action/reward design patterns by problem class.

Table 5.1 State, action, reward patterns observed in RL for scheduling

Problem class	State design	Action design	Reward design
Job shop (JSS) static/dynamic	Disjunctive graph embeddings (GNN size-agnostic)	Dispatch next eligible operation	$-$makespan / $-$tardiness with step penalties
Flexible JSS (routing + sequencing)	Dual attention over operations and machines	Joint machine routing and operation sequencing	Weighted makespan + tardiness; shaping for idle time
Dynamic arrivals	Queue/machine status, arrival indicators	Dispatch/route arriving jobs	Weighted tardiness/-completion; penalties on lateness
Energy-aware JSS	Machine load + energy profiles	Dispatch with energy-aware tie breaks	Combined makespan + energy cost; penalties for overconsumption
Cloud/edge scheduling	Resource utilization, SLA/backlog	Task-to-VM/offload assignment	$-$slowdown, latency, SLA penalties, energy terms
Transport/AGV	Network/vehicle positions, queue lengths	Vehicle dispatch/route choice	Throughput, delay penalties, collision avoidance penalties

6 Comparative Performance Analysis

Synthesize empirical results across studies, focusing on baselines, metrics, and robustness.

6.1 Performance vs. Classical Baselines

Dispatching rules, MILP/CP, metaheuristics; domain-wise comparison.

6.2 Generalization and Robustness

Out-of-distribution instances, dynamic arrivals, noise/perturbations.

6.3 Sample Efficiency and Ablations

Replay strategies, curriculum, reward shaping. *Progress note:* Recent GNN/PPO and dual-attention actor-critic schedulers outperform classic PDRs on JSS/FJSS benchmarks and generalize to larger unseen instances; exact OR tools still stronger on some cases.

Planned tables Performance comparison per domain; robustness/generalization results; sample-efficiency summaries.

Planned figure Heatmap of methods vs. benchmarks and win/loss vs. baselines.

6.4 Constraint Handling and Feasibility

Penalty shaping, masking, Lagrangian/shields. Table 6.2 summarizes common techniques.

Table 6.1 Baselines and metrics across domains

Domain	Typical baselines	Metrics
JSS/FJSS	PDRs (EDD, SPT, LPT), NEH, tabu/-GA/SA; MILP/CP on small instances	Makespan, tardiness, total weighted tardiness (TWT)
Dynamic shop/fab	Dispatching rules + simulation heuristics; myopic OR heuristics	Throughput, cycle time, tardiness, service level
Cloud/edge	SJF, Tetris, round-robin, heuristics, OR-Tools	Makespan, slowdown, latency, SLA adherence, energy
Transport/AGV	Nearest-vehicle/greedy dispatch, rule-based logistics heuristics	Throughput, travel time, tardiness
Energy-aware	Energy-aware heuristics, metaheuristics	Energy consumption, makespan, tardiness

Table 6.2 Constraint-handling techniques in RL scheduling

Technique	Examples	Notes
Action masking	Feasible machines/operations only; block violating routes	Stabilizes training, keeps feasibility; used in JSS/FJSS and constrained routing
Penalty shaping	Add cost for lateness, setups, energy overuse, SLA violation	Simple to implement; may struggle with hard constraints if penalties mis-tuned
Shields/filters	Safety layer vetoes unsafe actions (collisions, overruns)	Effective for safety-critical transport/production; requires rule base
Lagrangian/dual	Penalty multipliers updated during training	Better balance feasibility vs performance; needs tuning
Curricula	Start relaxed, tighten constraints over training	Improves learning stability under heavy constraints

7 Application Domains

Short vignettes for key sectors and their specific constraints.

7.1 Semiconductor and Flexible Manufacturing

7.2 Logistics and Transportation

7.3 Cloud and Edge Computing

7.4 Energy-Aware and Sustainable Scheduling

Planned tables Domain-specific datasets/benchmarks and metrics; constraint profiles per domain.

Planned figure Timeline of notable RL-in-scheduling papers per domain.

8 Cross-Cutting Challenges

Discuss systemic issues in applying RL to scheduling.

8.1 Stability and Variance

Seed sensitivity, policy brittleness.

8.2 Constraint Handling

Hard vs. soft constraints, feasibility preservation, masking vs. penalties.

8.3 Simulation-to-Real Gap

Domain randomization, robust policies, transfer.

8.4 Interpretability and Safety

Action rationale, override strategies, safe RL.

8.5 Reproducibility

Open code/data, hyperparameter documentation, evaluation protocols.

Planned tables Constraint-handling techniques; reproducibility checklist.

Planned figure Sim-to-real mitigation strategies.

9 Open Gaps and Future Directions

Identify promising research avenues grounded in observed gaps.

9.1 Hybrid RL and Operations Research

Learning-augmented heuristics, RL-guided search, primal-dual methods.

9.2 Offline, Safe, and Risk-Sensitive RL

9.3 Transfer, Meta-Learning, and Continual Adaptation

9.4 Benchmarking and Standardization

Need for standardized environments, seeds, reporting.

Planned figure Roadmap of future research directions and milestones.

10 Conclusion

Synthesize insights, answer research questions, and highlight practical implications for deploying RL in scheduling.

A Appendix

Reserved for supplementary material (extended tables, hyperparameters, additional plots, reproducibility checklists). Remove if no appendix is required.

Bibliography