Journal of Manufacturing Systems 86 (2026) 546–568

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Review

Reinforcement learning for autonomous production planning and control: A systematic literature review

Jesse Mayerhoff [a,b,*], Matthias Schmidt [c]

[a] Leuphana University of Lüneburg, Institute for Production Technology and Systems (IPTS), Lüneburg, Germany
[b] Volkswagen AG, Wolfsburg, Germany
[c] Leibniz University Hannover, Institute of Production Systems and Logistics (IFA), Garbsen, Germany

ARTICLE INFO

Keywords:
Reinforcement Learning
Production Planning and Control
Smart Manufacturing
Production Scheduling
Autonomous Manufacturing

ABSTRACT

The increasing complexity of modern manufacturing systems demands advanced decision-making approaches for production planning and control (PPC). Reinforcement learning (RL), as part of machine learning, has gained attention in recent years due to its ability to learn optimal policies for decision-making through trial-and-error interaction with a dynamic environment. This systematic literature review synthesizes 196 peer-reviewed publications from 2018 to 2024 on RL for PPC. Using an established RL framework, we analyze algorithm families, decision mechanisms, optimization objectives, evaluation practices, and industrial maturity. Results show a strong concentration on operational control, especially dispatching, with increasing adoption of policy-gradient methods and multi-agent formulations. Reward design remains dominated by time-based objectives such as makespan and tardiness, while cost, sustainability, and risk-oriented objectives are mainly treated as secondary terms. We identify a persistent structural gap between academic validation and industrial adoption. The majority of studies validate in synthetic simulations, only a small subset uses real industrial data, and very few connect trained policies to physical testbeds. No reviewed case study reports sustained closed-loop autonomous control in a live production system under continuous operation. We consolidate reported research gaps into an actionable agenda focused on environment fidelity, transfer governance, standardized evaluation, and safety and assurance mechanisms that enable scalable industrial deployment.

1. Introduction

Production logistics performance is critical to meeting customer demands, making PPC a key driver of business success [1]. Traditional methods such as heuristics, simulation-based control and operations research are increasingly challenged by the growing complexity, uncertainty and dynamics of modern manufacturing systems [2]. Industry 4.0 technologies provide the foundation for adaptive, data-driven decision-making that overcomes these limitations [3]. AI, particularly RL, offers innovative solutions to address the dynamic and complex challenges of PPC by improving KPIs through autonomous learning [4]. The theoretical foundations of RL date back to the 1950s, but its potential surged in the last decade with neural network integration [5]. This was exemplified by the victory of AlphaGo over the Go world champion [6] and the release of OpenAI Gym [7].

RL in PPC enables adaptive and proactive control of production processes even in dynamic and complex manufacturing environments. Understanding the state of the art in RL for PPC is crucial to facilitate its

Abbreviations: A2C, Advantage actor-critic; A3C, Asynchronous advantage actor-critic; AGV, Automated guided vehicle; AI, Artificial intelligence; AMR, Autonomous mobile robot; D3QN, Double dueling deep Q-network; DDPG, Deep deterministic policy gradient; DES, Discrete-event simulation; DNN, Deep neural network; Double DQN, Double deep Q-network; DQN, Deep Q-network; DRL, Deep reinforcement learning; Dueling DQN, Dueling deep Q-network; EDD, Earliest due date; FIFO, First in, first out; FJS, Flexible job shop; GA, Genetic Algorithms; GNN, Graph neural network; HaSupMo, Hanoverian Supply Chain Model; IoT, Internet of Things; IT, Information Technology; JSSP, Job shop scheduling problem; KPI, Key performance indicator; LinUCB, Linear upper confidence bound; MARL, Multi-agent reinforcement learning; MCTS, Monte Carlo tree search; MOO, Multi-objective optimization; OT, Operational Technology; PoC, Proof of concept; PPC, Production planning and control; PPO, Proximal policy optimization; RL, Reinforcement learning; RQ, Research question; SA, Simulated Annealing; SAC, Soft actor-critic; SARL, Single-agent reinforcement learning; SLR, Systematic literature review; SOO, Single-objective optimization; SPT, Shortest processing time; TD3, Twin delayed DDPG; TRPO, Trust region policy optimization; WIP, Work in progress.

* Corresponding author at: Leuphana University of Lüneburg, Institute for Production Technology and Systems (IPTS), Lüneburg, Germany.
E-mail address: jesse.mayerhoff@stud.leuphana.de (J. Mayerhoff).

https://doi.org/10.1016/j.jmsy.2026.03.023
Received 24 October 2025; Received in revised form 3 March 2026; Accepted 25 March 2026
Available online 2 April 2026
0278-6125/© 2026 The Authors. Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

further development. SLRs provide a structured methodology for evaluating and interpreting approaches in RL to problem-solving [8]. To assess the current landscape of RL applications in PPC, several SLRs have been conducted, providing insights into different aspects of the field. A comprehensive review on RL in PPC was conducted by Esteso et al. [9], who analyzed publications from 1994 to 2021. Their study identified the main challenges and limitations of RL applied to PPC. A more specialized perspective was provided by Panzer & Bender [10], who categorized RL publications into production domains and investigated implementation challenges. Their review offered a structured view of RL applications across various production functions but did not explore the decision mechanisms underlying these applications. Beyond PPC, Rolf et al [11]. developed a classification framework for RL applications in supply chain management, analyzing literature published between 2000 and 2021. Studies analyzed key factors such as algorithms, data availability, and industrial adoption. Similarly, C. Li et al [12]. conducted a review of DRL applications in smart manufacturing, highlighting its widespread adoption across various stages of the engineering lifecycle. Kayhan & Yildiz [13] conducted a more focused analysis of RL applications for machine scheduling problems between 1995 and 2020.

Although existing reviews provide valuable insights into RL applications in PPC and related fields, three limitations still constrain an integrated understanding of the field. First, multiple reviews are either outdated given the rapid progress in DRL tooling and architectures or intentionally narrow in scope, which prevents a comprehensive synthesis of recent work. Second, most reviews structure findings primarily by application domain and therefore do not systematically compare how decision mechanisms differ across PPC tasks and production scenarios. Third, the industrialization maturity of RL in PPC is still assessed inconsistently, leaving limited evidence on how simulation results translate to data-grounded settings, testbeds, and real control loops. These limitations motivate the present SLR, which consolidates recent innovations, analyzes decision mechanisms across the PPC hierarchy, and evaluates industrial maturity using a dedicated classification.

Accordingly, this review provides three complementary contributions. First, it delivers an up-to-date and comprehensive synthesis of RL applications in PPC by analyzing 196 peer-reviewed publications between 2018 and 2024. Second, it goes beyond domain-based categorization by systematically mapping RL approaches to the decision mechanisms they implement across the PPC hierarchy, offering a deployment-oriented perspective on how learned policies operationalize control authority under production-logistics constraints such as WIP coupling, resource limits, and routing structure. Third, it introduces a maturity classification that assesses the industrialization level of RL implementations, thereby making the persistent gap between simulator-centric validation, data-grounded studies, physical testbeds, and sustained closed-loop operation explicit. Collectively, these contributions consolidate the current state of knowledge and provide an actionable agenda for future research and industrial adoption.

To address the ambiguity often found in PPC literature, this review grounds its scope in the HaSupMo [14]. Rather than treating PPC as a monolithic block, we see it as a hierarchical system of interdependent decision levels. This perspective allows us to distinguish between Plan Production (e.g., calculate lot size, schedule throughput) and Control Production (e.g., release order, dispatching). This distinction is vital because these levels are inherently coupled, as for example order release decisions directly influence the queue lengths that dispatching agents must navigate.

We organize the remainder of this paper as follows. Section 2 introduces the RL fundamentals and the PPC scope required to interpret the review. Section 3 describes the review method, including the six-step process used for study identification and selection, the data extraction scheme, and the three research questions. Section 4 presents the in-depth results analysis along the core elements of the RL framework, covering agent design (algorithms and architectures), decision mechanisms (action spaces) mapped to the HaSupMo hierarchy, reward formulations and optimization objectives, environment characteristics including scenario types and industrial maturity, benchmarking practices, and reported future research needs. Section 5 synthesizes these findings to answer RQ1–RQ3 and discusses limitations. Section 6 concludes with the main takeaways and an actionable agenda for future research and industrial deployment.

2. Basics

This section establishes the conceptual foundation for the SLR. First, we outline the RL framework that guides the categorization of agent design, decision mechanisms, reward formulations, and environment implementations in the reviewed studies. Second, we delimit the production logistics scope by positioning PPC tasks within the HaSupMo hierarchy, providing a consistent structure for mapping RL applications across planning and control levels.

2.1. RL framework

This review analyzes RL applications using the standard framework established by Sutton & Barto [4]. RL enables agents to learn decision-making policies by interacting with their environment. The environment offers dynamic context, tasks, and conditions under which RL agents operate. At each step, the agent observes the environment's state, takes an action, and receives a numerical reward (Fig. 1). The goal of the agent is to maximize cumulative rewards over time. The sets of possible states and actions form the state and action spaces, which can be either discrete or continuous.

We categorize each study using a consistent RL taxonomy covering algorithm family, agent architecture, reward design, and environment implementation. RL algorithms are primarily distinguished by whether they are model-based or model-free [4]. Model-free algorithms are further divided into value-based, policy-based, or hybrid approaches. Value-based algorithms provide a robust foundation for discrete control tasks. A foundational value-based algorithm is Q-learning, a model-free approach that learns the value of state-action pairs [15]. DQN is an extension of Q-learning by using neural networks to approximate Q-functions [5]. Key advancements such as prioritized experience replay, fixed Q-targets, and reward clipping improve training stability. Double DQN reduces Q-value overestimation by decoupling action selection and evaluation [16]. Dueling DQN further improves learning efficiency and generalization by separating state value and action advantage estimates [17]. D3QN combines the stability of Double DQN with the efficiency of Dueling DQN. In contrast to value-based methods, policy-based algorithms directly optimize policies. While many popular algorithms in this group, such as PPO and TRPO, use an actor-critic architecture, we classify them by their policy-gradient training principle [18]. TRPO ensures stable policy improvements within a trust region [19]. PPO is a simplified variant of TRPO with clipped objectives [18]. Hybrid actor-critic methods combine the strengths of value-based and policy-based algorithms for more stable and sample-efficient learning. A3C accelerates learning through asynchronous updates [20].

Beyond the algorithmic logic, researchers must also differentiate between SARL and MARL [21] to suit the organizational structure of the

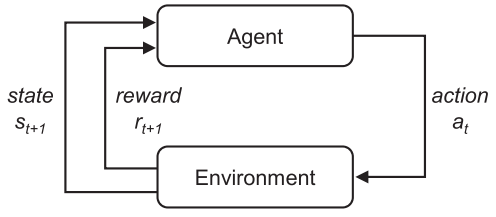

Fig. 1. The RL framework based on the agent-environment interaction loop [4].

production system. Finally, the reward function specifies the optimization objective, which can be designed as SOO or MOO approaches [22]. For further insights, please refer to additional literature such as Sutton & Barto [4].

2.2. PPC scope

To clarify the specific PPC layers and address the system level hierarchy, we adopt the definitions from HaSupMo. It serves as a fundamental descriptive model that captures the functional relationships between logistic objectives and the underlying control tasks. Within this framework, we focus on two primary areas: Plan Production and Control Production. This focus is chosen because these domains represent the core "internal" supply chain where RL agents directly interact with physical shop-floor dynamics and WIP levels. As shown in Fig. 2, these tasks form a hierarchical chain of command.

Plan Production begins with lot sizing to balance setup and inventory costs, followed by throughput scheduling to define preliminary order dates. To ensure feasibility, detailed resource planning balances load peaks by adjusting capacity supply before plan approval finalizes the orders for execution. Subsequently, Control Production manages short-term dynamics through verify availability to prevent resource bottlenecks. Once verified, order release regulates the WIP by determining the timing of order entry, while capacity control allows for short-term adjustments like processing speeds or overtime to ensure schedule reliability.

Finally, sequencing determines the order in which jobs are carried out at specific workstations, within the constraints established by upstream release and capacity decisions. In this paper, we use the term dispatching to describe local, real-time job selection and machine assignment at execution. We distinguish dispatching from scheduling by both time horizon and mechanism. Scheduling refers to the creation of a medium-term, predictive timetable, whereas dispatching addresses immediate, reactive decision-making when a resource becomes available, without projecting a future sequence.

The physical shop-floor layout and the complexity of the routing are important impact factors for RL models. This review classifies the literature into distinct scenarios. Classical Layouts include JS, where products follow individual routings, FS with its linear, unidirectional flow, and SMS focusing on isolated bottlenecks. Complex Variants feature increased flexibility through parallel machines or alternative routes (FJS, HFS) or constant job sequences (PFS), often including reentrant flows (RJS, RHFS) where jobs return to the same machine for multiple operations. Advanced and Distributed systems represent further modern manufacturing trends, including DJS across multiple sites, Matrix and Modular Production for highly individualized output, and the BAP, which is characterized by a stationary workpiece where various assembly blocks or resources are brought to the product in a specific sequence.

3. Review method

This SLR follows the guidelines of Tranfield et al [23]. and Durach et al [24]. and applies six steps in the review process: (1) defining research questions, (2) determining study characteristics, (3) retrieving literature, (4) selecting pertinent studies, (5) synthesizing findings, and (6) reporting results.

3.1. Step 1 – Define the research question

Our primary goal is to provide a comprehensive overview of RL in PPC, guiding future research by summarizing current approaches, challenges, and needs. This review is guided by the following research questions:

RQ1. : What trends have recently emerged in the development and application of RL agents in PPC?

RQ2. : What progress have RL approaches made in implementing the environment in real-world production systems?

RQ3. : What are the key research needs and gaps for deploying RL in PPC?

Addressing these questions deepens the understanding of RL in PPC and supports the long-term goal of applying autonomous agents for decision-making in real-world production.

3.2. Step 2 – Determine the required characteristics of primary studies

We used a multi-stage process to identify relevant primary studies and ensure a comprehensive and representative selection of literature. Selecting and combining suitable search terms is critical to ensuring relevant literature coverage. We developed a search string targeting RL

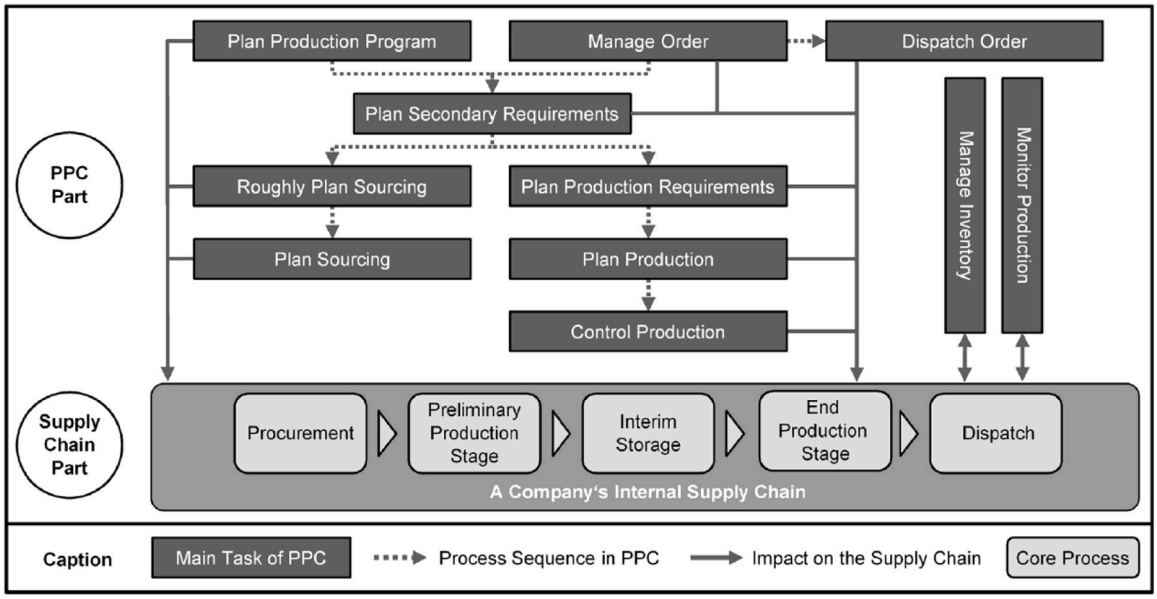

Fig. 2. Structure of the Hanoverian Supply Chain Model [14].

applications in PPC:

"Reinforcement Learning" AND(("production management" OR "production planning" OR "production control") OR ("manufacturing management" OR "manufacturing planning" OR "manufacturing control"))

We defined a set of inclusion and exclusion criteria based on the research questions to ensure relevance and avoid biased selection [8]. Only peer-reviewed English studies published from 2018 to 2024 are included. Furthermore, inclusion criteria required a concrete case study of an RL application in PPC. Papers were excluded for lacking a production context or providing insufficient implementation details.

3.3. Step 3 – Retrieve a sample of potentially relevant literature

To obtain a comprehensive overview of current research, we used the three databases Scopus, IEEE Xplore, and Web of Science. These databases were selected due to their extensive coverage of AI, manufacturing, and engineering research, ensuring a broad and relevant literature base. Searches were conducted across titles, keywords, and abstracts in all three databases, covering the full publication period from 2018 to 2024. The results are displayed in Fig. 3.

3.4. Step 4 – Select the pertinent literature

We selected the results that met the inclusion criteria for further analysis and employed a multi-stage selection process. After removing duplicates, 601 results remained for further examination. Screening titles, keywords, and abstracts excluded 352 results. We subjected the remaining 249 results to full-text analysis, which led to the exclusion of 83 papers not meeting the pre-established criteria. Consequently, we selected 166 papers for further analysis. With forward and backward searches according to Webster & Watson [25], we identified 30 additional papers, resulting in a total of 196 articles (Fig. 4).

3.5. Step 5 – Synthesize the literature

For data synthesis, we systematically extracted general publication data and RL-specific details (e.g., algorithm, agent architecture, reward structure, environment). The key findings for each paper are summarized in Table 1. Some publications contain multiple agents or multiple mechanisms. These are categorized separately in the quantitative analyses, which can result in higher category totals than the number of publications in some cases.

3.6. Step 6 – Report the results

Finally, we analyze the extracted data using quantitative methods to identify statistical trends and qualitative methods to explore research gaps and future directions. Fig. 5 displays the count of selected publications from 2018 to 2024, revealing steady annual growth in research on this topic.

The selected contributions include 61 conference papers and 135 journal articles. The journal articles span 51 different journals, with "Computers & Industrial Engineering" and the "Journal of Manufacturing Systems" being the most frequently represented, each contributing ten publications (Fig. 6). Most of the journals focus on production and manufacturing engineering, industrial computing, artificial intelligence in engineering, and automation systems.

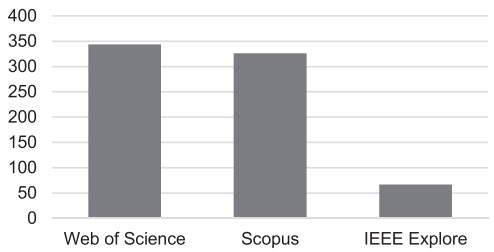

Fig. 3. Number of papers in the databases.

The conference papers were presented at 40 different conferences, with the "Winter Simulation Conference" contributing the most papers, at five contributions (Fig. 7). The conferences covered topics such as manufacturing engineering, systems engineering, computational intelligence, simulation technology, and production logistics. These events reflected a mix of production logistics and IT-focused themes.

For examining key RL areas, the presentation of the results is primarily concerned with the examination of RL algorithms (agent), decision mechanisms (action), optimization objectives (reward), and the system interaction (environment) according to the RL framework (Fig. 1). Due to inconsistent reporting in the source material, a detailed state space analysis was excluded. Insights related to state spaces are included in the discussion on the environment, but they are not the focus of a standalone analysis in this SLR. To complete the in-depth analysis of the results, we conduct a categorized consideration of proposed future research needs from the publications examined.

4. In-depth results analysis

This section synthesizes the reviewed literature along the core elements of the RL framework: agent design (algorithm and architecture), decision mechanisms (action space), optimization objectives (reward), and training and testing context (environment). The analysis is grounded in the HaSupMo hierarchy to ensure that each RL approach is interpreted within a consistent PPC task structure. To enable a rigorous quantitative synthesis, the analysis considers each agent instance in its assigned category as the primary unit, rather than the publication. A single paper may implement multiple agents (for example separate agents for job selection and resource allocation) or evaluate multiple decision mechanisms and algorithms. Consequently, counts reported in the figures and subsections refer to agents or mechanisms, and totals can exceed the number of publications (N = 196). Where studies span multiple PPC tasks, each agent is mapped to the HaSupMo task that best matches its primary decision mechanism, ensuring a consistent allocation across planning and control layers.

4.1. Agent analysis – algorithm distribution and evolution

The choice of learning algorithm is a core design decision for RL agents in PPC. This section presents the distribution and temporal evolution of 20 algorithm types identified in the reviewed literature (Fig. 8). Most implementations are built on common Python ecosystems, including SB3, PyTorch, TensorFlow, and Keras.

Fig. 8 shows that value-based algorithms remain widely used, particularly for discrete decision problems. Classical Q-learning appears in 11 applications, whereas DQN dominates value-based implementations with 57 occurrences. Among the reviewed studies, established DQN extensions are Double DQN in 26 applications, Dueling DQN in 6 applications and D3QN in 11 applications. Furthermore, policy-based methods constitute a significant proportion of the reviewed literature. TRPO is used in 9 applications, while PPO is the most frequently reported algorithm overall, used in 60 applications. Hybrid actor-critic variants are less common. For example, A3C appears in 7 applications. Beyond these families, the literature includes a smaller set of specialized approaches such as DDPG, TD3, and LinUCB, typically applied in more specific control settings. Within the dataset, model-based RL is infrequently observed, whereas the reviewed studies predominantly employ model-free learning approaches.

Fig. 9 shows that SARL is the dominant agent architecture in the reviewed literature. Within SARL, PPO is used in 49 applications, followed by DQN in 35 applications and DQN extensions in 24

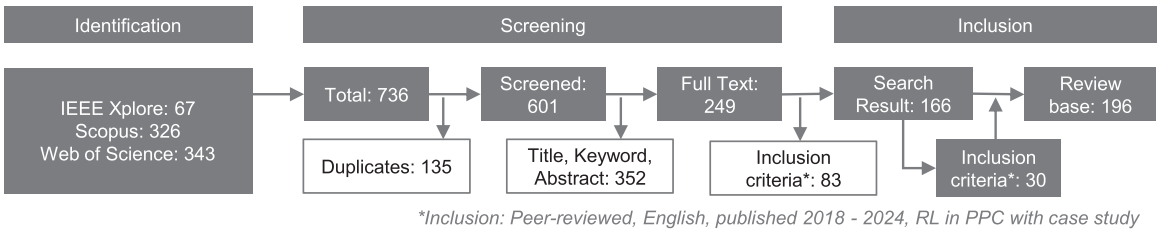

Fig. 4. Search and selection process.

applications. MARL is less frequent overall, but its algorithm mix differs. DQN is the most common choice in 23 applications, followed by DQN extensions in 14 applications and PPO in 13 applications, while A2C, A3C, Q-learning, and TRPO appear only sporadically. This distribution shows that SARL studies predominantly rely on PPO and DQN families, whereas MARL studies more often employ value-based methods.

An analysis of the temporal distribution from 2018 to 2024 shows a clear change in algorithm prevalence (Fig. 10). In 2018–2021, DQN is the most frequently used algorithm, and its refined variants become more visible from 2022 onward (e.g., Double/Dueling DQN and D3QN). In 2023 and 2024, PPO becomes the leading approach, surpassing DQN in frequency. While value-based methods remain common for discrete control settings, the increased share of policy-based methods and the broader "other" category indicate a diversification toward algorithms suited for higher-dimensional and more stochastic control problems.

An analysis of RL architectures from 2018 to 2024 shows that SARL remains the dominant modeling paradigm across the reviewed studies (Fig. 11). Over the observation period, the proportion of MARL approaches gradually increases, with the most pronounced growth in the final years. This temporal pattern shows that multi-agent formulations are being adopted more frequently alongside the established single-agent baseline, particularly in recent work. Despite this increase, MARL remains a minority compared to SARL across the full 2018–2024 period. Having characterized which learning algorithms dominate the field and how this mix evolves over time, we next examine how these agents are operationalized in PPC through their decision mechanisms.

4.2. Action space analysis – decision mechanisms within the PPC hierarchy

This section analyzes the decision mechanisms that define the RL agent's action space and maps them to PPC tasks using the HaSupMo hierarchy. To maintain a consistent synthesis, each identified agent is assigned to the HaSupMo task that best matches its primary decision mechanism.

As Fig. 12 illustrates, the reviewed literature primarily focuses on production control (86%), with production planning representing the remaining 14%. The following subsections present the task distribution and decision-making mechanisms within each domain. Across both domains, RL agents typically formulate their action spaces based on three main mechanisms: (1) job selection from a list (selecting the next job from a list of options), (2) resource allocation (assigning available resources to tasks), and (3) integrated job and resource allocation (often implemented via the selection of priority rules or heuristics rather than direct assignment). Importantly, planning and control are distinguished not by different mechanisms but by the time horizon at which these mechanisms are applied. In studies reviewed that implement multiple agents for different tasks, mechanisms are counted per agent. This may result in category totals that exceed the number of publications.

4.2.1. Plan Production

The production planning domain constitutes 14% of the identified literature. As Fig. 13 illustrates, the planning-related studies focus primarily on throughput scheduling, while lot size calculation appears less frequently and tasks such as detailed resource planning and production plan approval are rare. Across the 30 identified planning agents, SARL is predominant and only four papers employ MARL.

Following the hierarchical logic of the HaSupMo, lot size calculation with RL is used to determine batch quantities under setup and inventory trade-offs. Next in the hierarchy, throughput scheduling follows, which dominates the planning domain and therefore serves as the primary basis for analyzing planning level decision mechanisms. To understand the operational nature of throughput scheduling RL agents, we examine the specific decision mechanisms employed within this domain.

As Fig. 14 shows, most throughput scheduling agents use job selection from a list. Fourteen applications employ this mechanism. Integrated mechanisms that combine job selection and resource allocation are less common, appearing in five applications. Three applications use standalone resource allocation from a list, and all of these use MARL architectures [71,72,98]. Beyond scheduling and lot sizing, the dataset contains only limited evidence of RL in detailed resource planning (Fig. 13). Similarly, production plan approval appears only once in the reviewed literature. Therefore, RL agents at the planning level focus on mechanisms that directly construct or adjust a schedule, while higher level planning functions are sparsely covered. The following section contrasts this planning profile with the mechanisms identified for production control.

4.2.2. Control Production

Control Production accounts for 86% of the identified literature. Compared to the planning domain, the control domain shows a higher share of MARL implementations with 58 out of 178 applications, particularly within dispatching tasks (Fig. 15).

A notable observation is the absence of RL approaches addressing the HaSupMo sub-task verifying availability. In the reviewed papers, availability is either treated as an exogenous constraint or addressed by non-learning-based mechanisms outside the RL agent's decision space. The function of order release represents a smaller segment of the literature, with 13 implementations. In these studies, RL agents regulate work entering the system through action mechanisms such as binary release decisions and WIP cap management.

The largest research focus within production control is dispatching, with 103 SARL and 55 MARL applications. Dispatching mechanisms are related to scheduling but are applied with an execution-oriented time scope. Some studies integrate dispatching into broader architectures, for example by using dispatching policies to control a digital twin and feed outcomes into higher level decisions [122], or by linking planning and control within a blockchain-enabled framework [180]. In this review, such approaches are classified as dispatching when the agent's primary decision is applied online at execution time and without any planning horizon.

Fig. 16 summarizes the decision mechanisms used for dispatching. Of the 91 applications addressing job selection, 43 use direct selection from a list, 39 rely on rule-based selection, and nine use other mechanisms. The 54 applications that combine job selection and resource allocation show a similar relative distribution. For resource allocation, 14 out of 17 applications use list-based selection mechanism. Beyond dispatching, a small cluster of seven applications addresses capacity control (Fig. 15). These applications adjust available processing power using mechanisms such as workstation speed scaling, time stretching or compressing,

Table 1
Selected studies.

Ref.	Production scenario	RL algorithm	PPC task(HaSupMo mapping)	Decision mechanism(action design)	SARL / MARL		Single-/ multi-objective	
[26]	FJS	PPO	dispatching	job sel. & res. alloc.	SA			MO
[27]	FS	DQN	dispatching	job sel.	SA		SO	
[28]	FJS	Q-learning	dispatching	job sel.		MA	SO	
[29]	RHJS	DQN	dispatching, order release	job sel. & inter-arrival times	SA			MO
[30]	FS	DQN, Double DQN	capacity control	workstation ON/OFF & battery control	SA			MO
[31]	FS, JS	Q-learning	scheduling	job sel.	SA		SO	
[32]	FJS	AC	dispatching	job sel.		MA	SO	
[33]	FJS	Double DQN	dispatching	job sel. & res. alloc.	SA		SO	
[34]	FJS	D3QN	dispatching	job sel. & res. alloc.	SA			MO
[35]	FJS	PPO	dispatching	job sel.	SA		SO	
[36]	JS	A3C	scheduling	job sel.	SA			MO
[37]	JS	PPO	dispatching	job sel.	SA			MO
[38]	FS	PPO	dispatching	job sel.	SA		SO	
[39]	JS	Double DQN	dispatching	job sel.	SA		SO	
[40]	HFS	PPO	scheduling, plan resources	res. alloc. & period assignment	SA			MO
[41]	DHFS	DDPG	dispatching	job sel.		MA		MO
[42]	FJS	PPO	dispatching	job sel. & res. alloc.	SA			MO
[43]	JS	DQN	dispatching	res. alloc.		MA		MO
[44]	PFS	REINFORCE	scheduling	job sel.	SA		SO	
[45]	DFJS	Double DQN	scheduling	job sel. & res. alloc.	SA			MO
[46]	JS	DQN	capacity control	speed scaling	SA			MO
[47]	JS	PPO	scheduling	job sel.	SA			MO
[48]	Modular	Double DQN	dispatching	job sel. & res. alloc.	SA	MA	SO	
[49]	HFS	A2C	dispatching	job sel.	SA			MO
[50]	SSS	Q-learning	dispatching	job sel.	SA		SO	
[51]	JS	TRPO	dispatching	job sel., res. alloc.		MA		MO
[52]	FJS	D3QN	dispatching	job sel. & res. alloc.	SA			MO
[53]	FS	DQN, MCTS	dispatching	job sel.	SA	MA	SO	
[54]	FJS	PPO	scheduling	job sel. & res. alloc.	SA			MO
[55]	FS	A2C, PPO	capacity control	time stretching	SA			MO
[56]	FJS	DDPG	dispatching	job sel.		MA	SO	
[57]	JS	D3QN	dispatching	job sel.	SA		SO	
[58]	Modular	DQN	dispatching	job sel.	SA		SO	
[59]	JS	Double DQN	order release	job sel.	SA		SO	
[60]	JS	A3C, PPO	dispatching	res. alloc.	SA	MA	SO	
[61]	FJS	PPO	dispatching	res. alloc.		MA	SO	
[62]	FJS	PPO, other SB3-Lib	dispatching	job sel. & res. alloc.	SA	MA	SO	MO
[63]	FJS	REINFORCE	dispatching	job sel. & res. alloc.	SA		SO	
[64]	Matrix	DQN	dispatching	job sel. & res. alloc.		MA	SO	
[65]	DJS	DQN	dispatching	job sel., job sel. & res. alloc.		MA		MO
[66]	DJS	A3C	dispatching	job sel.	SA		SO	
[67]	Modular	DQN	dispatching	job sel. & res. alloc.	SA			MO
[68]	FJS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[69]	SSS	A2C	scheduling	job sel.	SA			MO
[70]	SSS	PPO	scheduling	job sel.	SA			MO
[71]	FFS	Q-learning	scheduling	job sel., res. alloc.		MA		MO
[72]	FFS	Q-learning	lot sizing, scheduling	job sel., res. alloc.		MA		MO
[73]	FJS	Double DQN	dispatching	job sel.		MA	SO	
[74]	JS	DQN	dispatching	res. alloc.		MA	SO	
[75]	PFS	DQN	dispatching	job sel.	SA		SO	
[76]	SMS	PPO	dispatching	job sel.	SA		SO	
[77]	FJS	TRPO	dispatching	job sel. & res. alloc.	SA			MO
[78]	FJS	TRPO	dispatching	job sel. & res. alloc.	SA			MO
[79]	FJS	TRPO	dispatching	job sel. & res. alloc.	SA			MO
[80]	FJS	TRPO	dispatching	job sel. & res. alloc.	SA			MO
[81]	FS	SB3-Lib	plan resources	capacity supply adjustment	SA		SO	
[82]	FJS	DQN	dispatching	res. alloc., job sel.		MA		MO
[83]	HFS	DQN	dispatching	job sel. & res. alloc.	SA		SO	
[84]	RFJS	DQN	dispatching	job sel.	SA		SO	
[85]	FFS	D3QN	dispatching	job sel. & res. alloc.	SA		SO	
[86]	SSS	A2C	scheduling	job sel.	SA			MO
[87]	DJS	DQN, Double/Dueling DQN, TRPO, PPO	dispatching	job sel. & res. alloc.	SA		SO	
[88]	FJS	Double DQN, PPO	dispatching	job sel., rescheduling decision, res. alloc.		MA	SO	
[89]	FS	D3QN	dispatching	job sel.	SA		SO	
[90]	FJS	Double DQN	dispatching, order release	binary release, job sel. & res. alloc.		MA		MO
[91]	FJS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[92]	FJS	Double DQN	dispatching	job sel. & res. alloc.	SA			MO
[93]	FS	TD	dispatching	job sel.	SA			MO
[94]	JS	OC	dispatching	job sel.	SA		SO	
[95]	JS	PPO	dispatching	job sel.	SA		SO	

(*continued on next page*)

Table 1 (*continued*)

Ref.	Production scenario	RL algorithm	PPC task(HaSupMo mapping)	Decision mechanism(action design)	SARL / MARL		Single-/ multi-objective	
[96]	JS	DQN	dispatching	job sel.	SA		SO	
[97]	DFJS	Double DQN	dispatching	job sel. & res. alloc.	SA		SO	
[98]	RHFS	DQN	scheduling	job sel., res. alloc.		MA	SO	
[99]	JS	DDPG	dispatching	job sel.		MA	SO	
[100]	FJS	Double DQN	dispatching	job sel., res. alloc.		MA	SO	
[101]	RFJS	A3C	dispatching, order release	WIP Limit & release amount & job sel.	SA			MO
[102]	JS	Double DQN	dispatching	job sel.		MA	SO	
[103]	SSS	DQN	dispatching	job sel.	SA		SO	
[104]	RHFS	PPO	capacity control, dispatching	job sel. & res. alloc., worker assignment		MA	SO	
[105]	FJS	Double DQN	dispatching	job sel. & res. alloc.	SA			MO
[106]	JS	PPO	dispatching	job sel.	SA		SO	
[107]	JS	PPO	dispatching	job sel.	SA		SO	
[108]	SSS	DQN, PPO, TRPO	capacity control	workstation ON/OFF	SA			MO
[109]	FFS	DQN, PPO, TRPO	capacity control	workstation ON/OFF	SA			MO
[110]	FJS	Double DQN	dispatching	job sel. & res. alloc.	SA			MO
[111]	FJS	Double DQN	dispatching	job sel. & res. alloc.	SA			MO
[112]	FJS	Double DQN	dispatching	job sel. & res. alloc., goal sel.		MA	SO	
[113]	RHFS	Double DQN	dispatching	job sel.	SA			MO
[114]	FJS	DQN	scheduling	job sel.	SA		SO	
[115]	FJS	Double DQN, DQN	dispatching	job sel.		MA		MO
[116]	JS	TD3	dispatching	job sel.		MA		MO
[117]	FS	DQN	order release	release amount	SA		SO	
[118]	FS	DQN	dispatching	job sel.	SA		SO	
[119]	FS	DQN	dispatching	job sel.	SA			MO
[120]	Matrix	PPO	dispatching	job sel., res. alloc.		MA		MO
[121]	Modular	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[122]	FFS	DQN	dispatching	job sel.	SA		SO	
[123]	PFS	PPO	scheduling	job sel.	SA			MO
[124]	SSS	PPO	dispatching	job sel. & res. alloc.	SA			MO
[125]	JS	PPO	scheduling	job sel.	SA		SO	
[126]	FS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[127]	FS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[128]	SSS	DQN	scheduling	job sel. & res. alloc.	SA		SO	
[129]	Modular	DQN	dispatching	job sel.		MA		MO
[130]	Modular	DQN	dispatching	job sel.		MA		MO
[131]	Matrix	DQN	dispatching	job sel.		MA		MO
[132]	RFJS	DQN	dispatching	job sel.		MA	SO	
[133]	Modular	Dueling DQN	dispatching	job sel.	SA			MO
[134]	RJS	Dueling DQN	dispatching	job sel. & res. alloc.	SA			MO
[135]	RFJS	DDPG	dispatching	job sel.	SA		SO	
[136]	FS	PPO	order release	binary release	SA			MO
[137]	FJS	DQMIX	dispatching	res. alloc.		MA	SO	
[138]	FJS	DQN	dispatching	res. alloc.		MA	SO	
[139]	FJS	PPO	dispatching	job sel.		MA		MO
[140]	FJS	PPO	dispatching	job sel.		MA		MO
[141]	FJS	PPO	dispatching	job sel. & res. alloc.		MA		MO
[142]	FJS	Dueling DQN	dispatching	job sel.		MA		MO
[143]	FJS	Dueling DQN	dispatching	job sel.		MA	SO	
[144]	FFS	DQN	dispatching	job sel.	SA		SO	
[145]	FS, JS	Q-learning	order release	WIP Limit	SA			MO
[146]	FS, JS	DQN	dispatching	job sel.	SA		SO	
[147]	JS	PPO	dispatching	job sel.		MA		MO
[148]	FJS	PPO	dispatching	job sel. & res. alloc.	SA			MO
[149]	SMS	PPO	dispatching	job sel.	SA			MO
[150]	RFJS	DQN	dispatching	job sel.		MA		MO
[151]	JS	DQN, SAC	dispatching	job sel.	SA		SO	
[152]	FJS	MCTS	dispatching	job sel.	SA		SO	
[153]	FS	A3C	order release	lead time control	SA			MO
[154]	JS	PPO	plan resources	overtime alloc.	SA			MO
[155]	FS	DQN	order release	lead time control		MA		MO
[156]	JS	DQN	order release	job sel.	SA		SO	
[157]	JS	PPO	dispatching	job sel.	SA			MO
[158]	FFS, JS	DQN	dispatching	job sel.		MA	SO	
[159]	FS	DQN	order release	WIP Limit	SA			MO
[160]	FS, JS	PPO	order release	WIP Limit	SA			MO
[161]	JS	D3QN	dispatching	job sel.	SA		SO	
[162]	FJS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[163]	RFJS	DQN	dispatching	job sel. & res. alloc.	SA			MO
[164]	FJS	PPO	scheduling	job sel. & res. alloc.	SA		SO	
[165]	FJS	DQN	dispatching	job sel.		MA		MO
[166]	Modular	D3QN	dispatching	job sel.	SA		SO	
[167]	FJS	PPO	dispatching	job sel. & res. alloc.	SA			MO
[168]	FFS	PPO	dispatching	job sel.		MA		MO
[169]	JS	DQN	dispatching	job sel.	SA			MO

(*continued on next page*)

Table 1 (*continued*)

Ref.	Production scenario	RL algorithm	PPC task(HaSupMo mapping)	Decision mechanism(action design)	SARL / MARL		Single-/ multi-objective	
[170]	JS	PPO	dispatching	job sel.	SA		SO	
[171]	SMS	PPO	lot sizing	lot sizing	SA			MO
[172]	PFS	Q-learning	scheduling	job sel.	SA		SO	
[173]	HFS	PPO	lot sizing	job sel.	SA			MO
[174]	FJS	TD3	dispatching	job sel. & res. alloc.	SA			MO
[175]	FJS	SAC	dispatching	job sel. & res. alloc.	SA		SO	
[176]	JS	PPO	dispatching	job sel.	SA		SO	
[177]	SMS	Q-learning	dispatching	job sel.	SA			MO
[178]	HFS	Double DQN	dispatching	job sel., res. alloc.		MA	SO	
[179]	FJS	Double DQN	dispatching	job sel. & res. alloc.	SA			MO
[180]	DJS	Q-learning	dispatching, scheduling	job sel.	SA			MO
[181]	JS	PPO	dispatching	job sel.		MA	SO	
[182]	FJS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[183]	RFJS	DQN	dispatching	job sel.		MA	SO	
[184]	RFJS	DQN	dispatching	job sel.		MA	SO	
[185]	SMS	PPO	lot sizing	lot sizing	SA			MO
[186]	JS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[187]	BAP	A3C	scheduling	shift/fix scheduling	SA		SO	
[188]	JS	PPO	dispatching	job sel.	SA		SO	
[189]	JS	PPO	dispatching	job sel.	SA		SO	
[190]	FJS	DQN	dispatching	job sel. & res. alloc.	SA			MO
[191]	FS	R-learning	order release	direct release decision	SA			MO
[192]	FJS	D3QN	dispatching	job sel.	SA			MO
[193]	FS	DQN	dispatching	job sel.	SA			MO
[194]	FJS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[195]	HFS	Q-learning	dispatching	job sel.	SA			MO
[196]	SMS	R-learning	dispatching	job sel.	SA			MO
[197]	PFS	Double DQN	dispatching	job sel.	SA		SO	
[198]	DPFS	A2C	scheduling	job sel.	SA		SO	
[199]	JS	Double DQN	dispatching	job sel.	SA		SO	
[200]	FJS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[201]	FJS	Double DQN	dispatching	job sel. & res. alloc.	SA			MO
[202]	FS	DQN	dispatching	capacity control		MA	SO	
[203]	JS	PPO	dispatching	job sel.	SA		SO	
[204]	JS	Dueling DQN	dispatching	job sel. & res. alloc.		MA	SO	
[205]	FJS	PPO	dispatching	res. alloc.		MA		MO
[206]	FJS	DQN	dispatching	job sel., res. alloc.		MA	SO	
[207]	FJS	PPO	dispatching	job sel. & res. alloc.	SA		SO	
[208]	FJS	SAC, D3QN	dispatching	job sel., res. alloc.		MA	SO	
[209]	JS	SAC	dispatching	job sel.	SA		SO	
[210]	HFS	DQN	dispatching	job sel. & res. alloc.		MA		MO
[211]	FJS	D3QN	dispatching	res. alloc. & other	SA			MO
[212]	FJS	D3QN	dispatching	job sel., res. alloc.		MA	SO	
[213]	JS	DQN	dispatching	job sel.		MA	SO	
[214]	DFS	Q-learning	scheduling	job sel. & res. alloc.	SA		SO	
[215]	FJS	AC	dispatching	job sel.	SA		SO	
[216]	SSS	DQN	dispatching	job sel. & res. alloc.	SA		SO	
[217]	SMS	DQN	capacity control, dispatching	job sel. & speed scaling	SA			MO
[218]	FJS	DQN	dispatching	res. alloc.	SA			MO
[219]	FJS	LinUCB	dispatching	job sel. & res. alloc.	SA		SO	
[220]	FJS	Double DQN	dispatching	job sel. & res. alloc.		MA		MO
[221]	FJS	DQN	dispatching	job sel. & res. alloc.	SA		SO	

dynamic worker assignment, and energy-oriented control via workstation ON/OFF switching.

A descriptive, algorithm level cross-mapping further complements the action space analysis. In the field of dispatching, DQN-based methods and PPO dominate the literature. The most frequently reported algorithms are 46 DQN applications, 44 PPO applications, and 22 Double DQN applications. This indicates that dispatching is currently the main arena where algorithm families could be compared under shared benchmarks. Scheduling presents a second opportunity for comparison, with PPO- and DQN-based approaches both being used repeatedly. Order release and capacity control are less prevalent and therefore offer less robust evidence for algorithm selection guidance. Since these counts reflect algorithm usage rather than controlled performance comparisons, they primarily serve to identify areas where consistent benchmark evaluations would be most impactful. Because action design determines what an agent can control in practice, we next analyze reward formulations to understand which production objectives these mechanisms are trained to optimize.

4.3. Reward analysis – optimization objectives

Reward functions operationalize production-logistics objectives as learning signals and therefore define how agent behavior is aligned with PPC targets. To systematize the heterogeneous objective terms reported in the reviewed studies, we apply the Production Logistics Target System by Wiendahl [1], which differentiates production logistic performance from production logistics costs and makes explicit that many objectives imply inherent trade-offs (Fig. 17).

Across the corpus, reward specifications span a broad spectrum of objectives, including classical PPC metrics (e.g., makespan, tardiness) as well as additional targets beyond the core logistics target system (e.g., energy-related objectives). For consistent coding, we harmonized semantically equivalent terms into unified categories (e.g., inventory, storage, warehousing consolidated as inventory) and aggregated

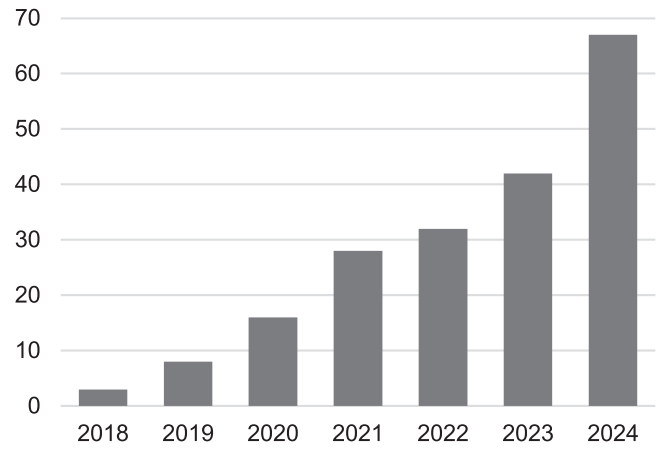

Fig. 5. Number of papers per year (2018–2024).

objectives that share the same mathematical basis (e.g., mean and standard deviation variants of the same target). We then mapped each study's main optimization objective to one of the four quadrants of Wiendahl's target system, schedule reliability and lead time for production logistics performance, as well as processing costs and working capital costs for production logistics costs. Objectives not directly represented in the Wiendahl framework were assigned to the additional "other" category (e.g., energy efficiency, safety). Fig. 18 summarizes the resulting distribution.

Production logistics performance objectives dominate reward design in the reviewed literature, with delivery-time related metrics appearing most frequently. Makespan is the most prevalent single objective and is used as a productivity proxy in 59 SOO and 33 MOO formulations. Its frequent use is consistent with its well-defined mathematical structure and its compatibility with standard scheduling and dispatching benchmarks. At the same time, the strong reliance on makespan concentrates the learning signal on aggregate completion performance and can underrepresent variability-sensitive phenomena such as flow-time dispersion and stochastic delays. Delivery reliability, in 49 case studies operationalized through tardiness minimization, forms the second major class of performance-oriented objectives.

Process-cost and capital-cost objectives are rarely used as stand-

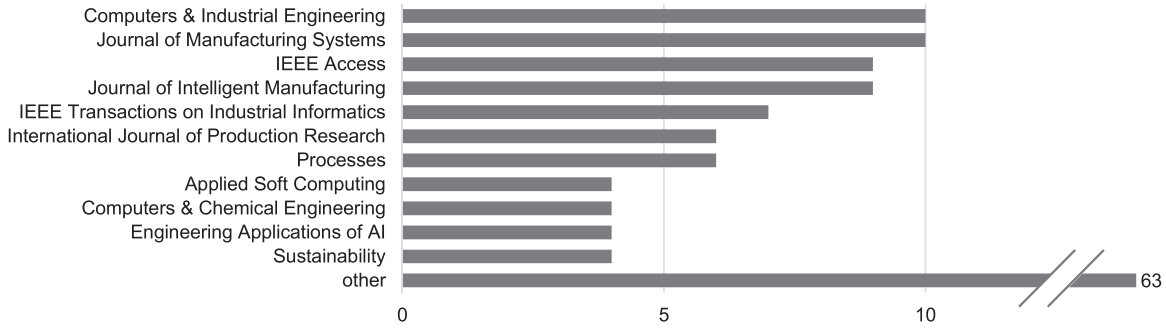

Fig. 6. Number of publications per journal.

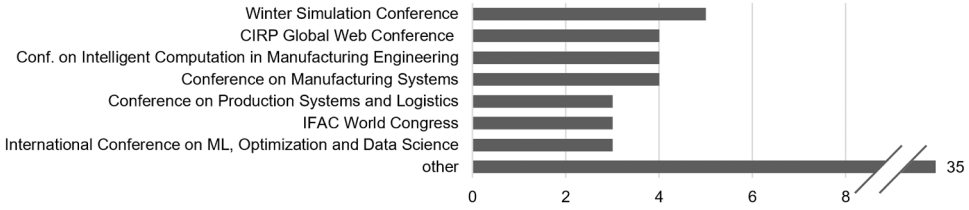

Fig. 7. Number of publications per conference.

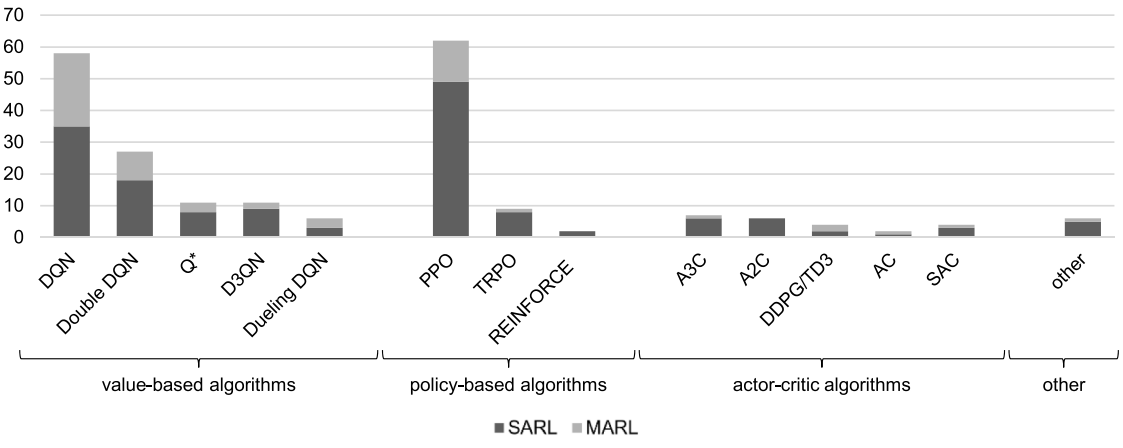

Fig. 8. Distribution of RL learning algorithms.

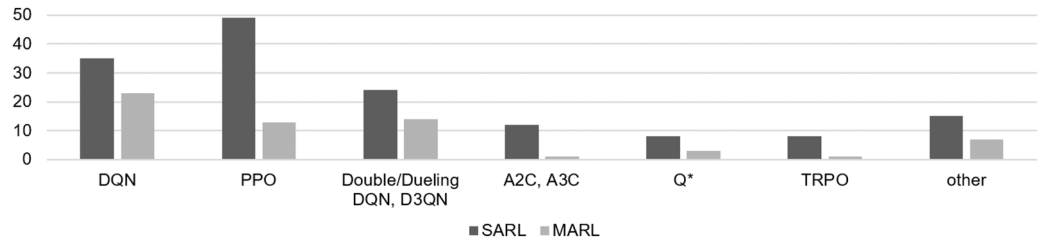

Fig. 9. SARL and MARL approaches distributed by RL algorithms.

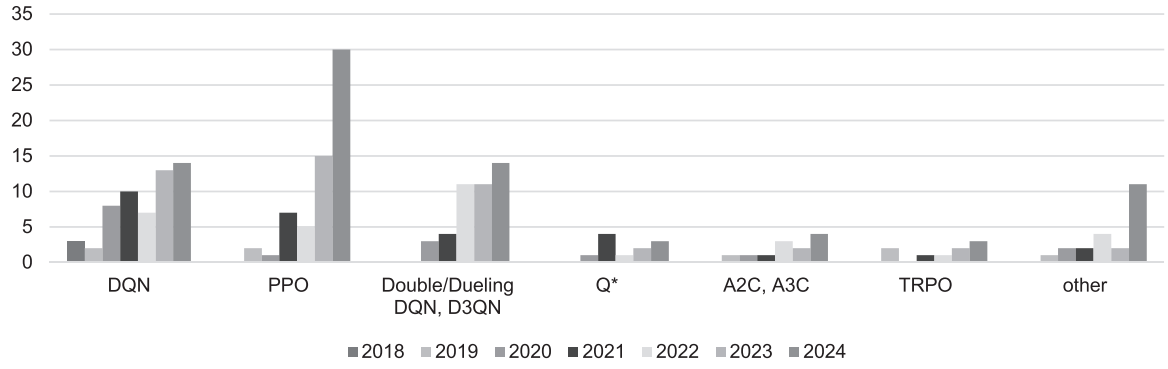

Fig. 10. Distribution of RL algorithms in PPC (2018–2024).

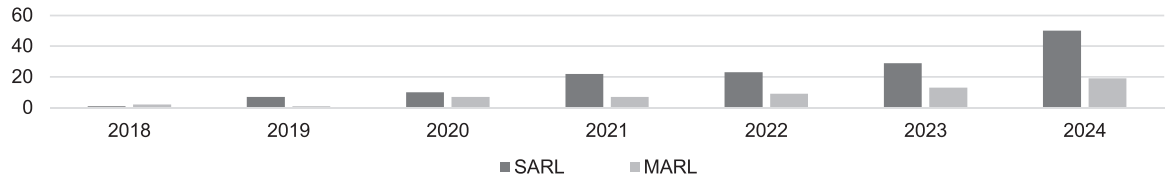

Fig. 11. Distribution of SARL and MARL approaches in PPC (2018–2024).

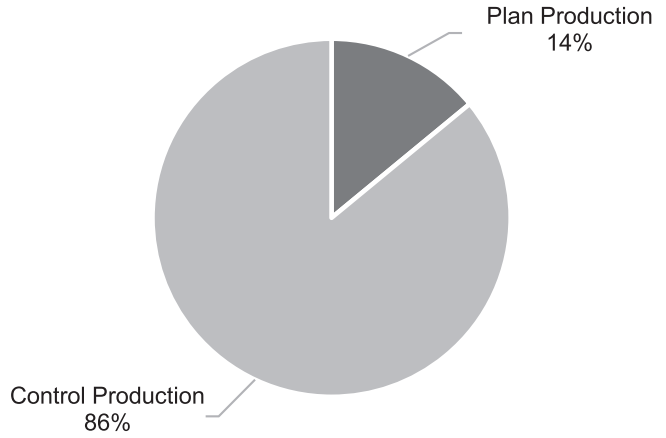

Fig. 12. Distribution of identified RL approaches between production planning and control (N = 197).

alone optimization targets. Within process costs, resource utilization is the most common metric with 19 applications, but it is predominantly embedded in multi-objective rewards rather than treated as a primary reward. Similarly, working-capital related targets such as WIP and inventory reduction are mainly introduced as additional terms in MOO formulations. This pattern mirrors the classic trade-off in production logistics between high utilization and low WIP and shows that cost-oriented objectives are typically represented as balancing terms relative to time-based performance goals, rather than as independent optimization drivers.

Beyond the target system quadrants, the smaller "other" cluster captures energy-related metrics, safety constraints, and robustness-oriented criteria. These objectives are commonly integrated via additive terms in MOO rewards, indicating an expansion of reward formulations toward constraints and externalities that extend beyond traditional throughput and due-date performance. After characterizing which targets are optimized, we analyze how conflicting objectives are coordinated in MOO settings. Fig. 19 provides a taxonomy of coordination mechanisms used to operationalize trade-offs within reward design.

A recurring pattern in MOO reward design is the need to distinguish coordination from reward shaping. Coordination defines how conflicting targets are prioritized, for example the intended trade-off between WIP and throughput. Shaping refines the learning signal to support convergence, for example by densifying feedback, without changing that preference structure. In many reviewed studies, both are combined but not reported separately, which reduces transparency and limits comparability across approaches.

Against this background, we analyze how studies coordinate conflicting targets in MOO settings. Across the 93 identified MOO approaches, the four coordination archetypes linear scalarization, hierarchical or sequential logic, Pareto-based optimization, and implicit reward engineering are observed. Linear scalarization is the dominant standard and is implemented primarily via weighted sums, covering 81% of the reviewed MOO literature. The remaining approaches appear as specialized alternatives. Hierarchical or sequential logic separates objectives structurally, Pareto-based methods approximate trade-off

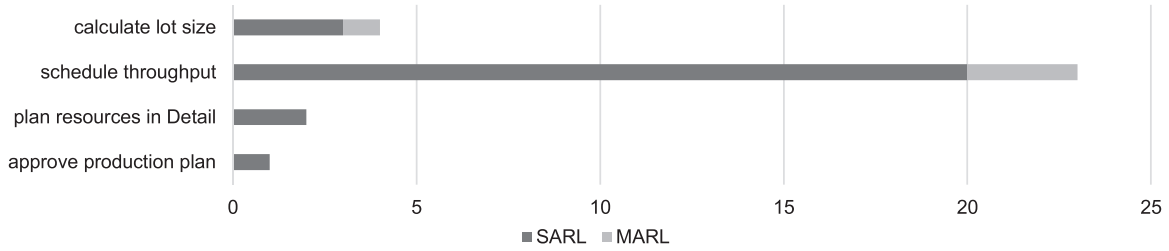

Fig. 13. Distribution of RL tasks within the Production Planning domain.

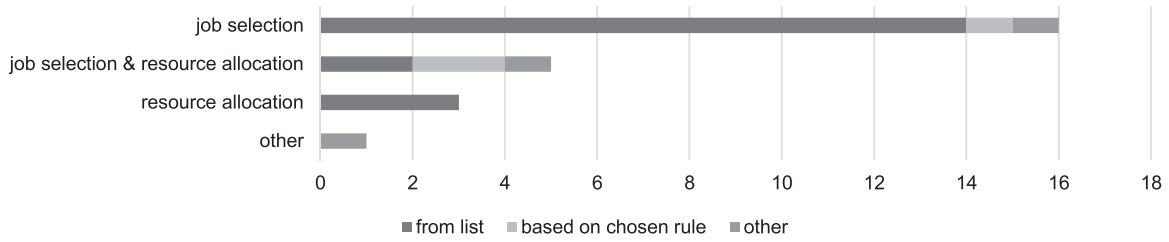

Fig. 14. Decision mechanisms within the throughput scheduling domain.

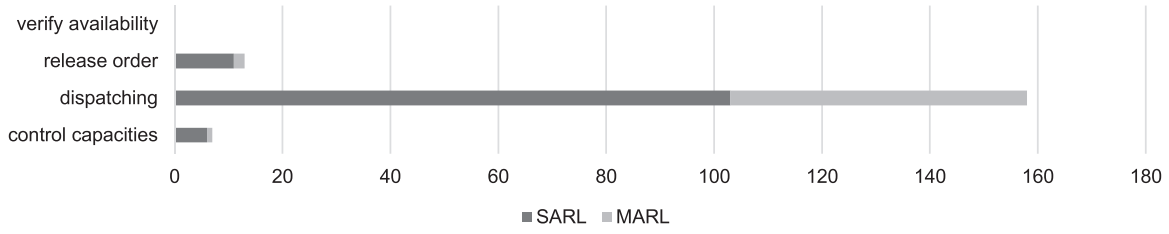

Fig. 15. Distribution of RL research across production control sub-tasks.

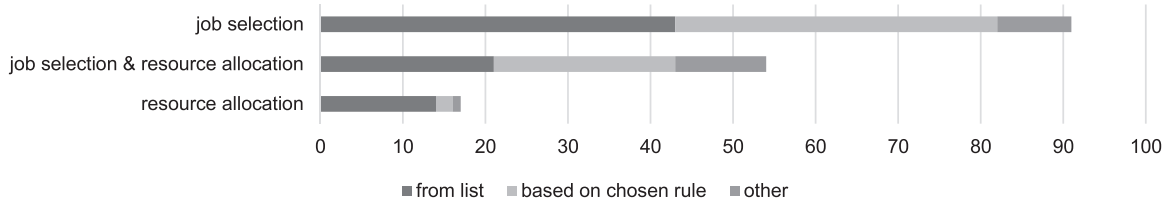

Fig. 16. Decision mechanisms within the dispatching domain.

surfaces explicitly, and implicit engineering embeds compromises through problem formulation or indirect signals rather than explicit additive aggregation. Notably, advanced reward specification paradigms based on iterative human feedback or behavioral inference, such as human-in-the-loop methods or inverse reinforcement learning, are not represented in the analyzed production logistics papers, indicating that reward design is predominantly defined by human process experts rather than feedback calibrated.

4.4. Environment analysis – maturity assessment

The training environment determines the production logic, constraints, and information structure an RL agent is exposed to during learning. This section reports the reviewed literature along physical production scenario types with their structural characteristics, and implementation maturity levels for industrial adoption.

4.4.1. Classification of production scenarios and implementation contexts

To define the operational scope of the reviewed studies, we classify case studies by production scenario type (Fig. 20). The scenario categories capture key structural constraints such as routing flexibility, machine order, resource coupling, and buffer limitations that shape the environment dynamics and the resulting control problem.

The distribution shows a strong concentration on JS and FS environments. JS configurations represent the largest cluster with 123 identified cases. These environments typically combine flexible routing or alternative resource choices with sequencing constraints and stochastic processing dynamics, and they are frequently used to study operational decision-making under high combinatorial complexity. FS configurations constitute the second-largest cluster with 50 case studies. These environments emphasize stage synchronization and throughput stability, and they commonly incorporate constraints such as bottleneck shifts, mix restrictions, and sequence-dependent setup effects.

Beyond JS and FS, 11 case studies investigate matrix or modular layouts. Notably, all identified matrix scenario studies employ MARL architectures. In these environments, the control problem is formulated around decentralized routing and localized decision-making across modular resources. Finally, 16 studies focus on single-system configurations. In this review, a single-system refers to a solitary processing stage, such as a single machine or an isolated bottleneck station,

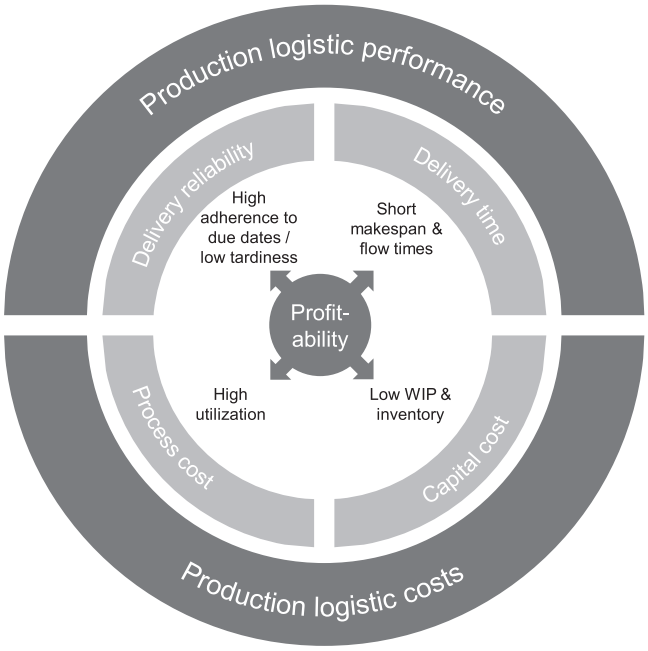

Fig. 17. Target system of production logistics [1].

operating independently of upstream or downstream routing constraints. These cases typically isolate a limited subset of environment dynamics and decision mechanisms and are used as simplified evaluation contexts alongside more complex shop-floor scenarios.

To provide a more granular view, we decompose the JS and FS scenarios into specific architectural variants (Table 2). Variants are not mutually exclusive as a study may be counted in multiple modifier categories. Within JS, 61% of studies focus on FJS configurations. These settings typically require joint decisions on sequencing and machine assignment across parallel resources. Within FS, hybrid and flexible variants represent a prominent share of the identified cases, reflecting a continued emphasis on flow-oriented systems with additional routing or stage flexibility. Distributed variants appear only infrequently in both JS and FS, and reentrant variants also remain a small subset of the reviewed corpus. Reentrant systems, in which jobs revisit the same workstation multiple times, are particularly relevant for high-precision industries such as semiconductor manufacturing. From a modeling perspective, these environments introduce repeated processing loops and longer decision dependencies compared to standard shop configurations. In the reviewed literature, however, only a limited number of studies implement reentrant variants, indicating that most evaluations are still conducted on non-reentrant shop structures.

4.4.2. Maturity levels and the sim-to-real gap

To assess readiness for industrial adoption, we classify the reviewed literature into four maturity levels based on the training and testing environment: (1) pure simulation with synthetic data, (2) simulation with real industrial data, (3) hybrid testing in simulations and physical testbeds, and (4) full industrial implementation. Fig. 21 summarizes the distribution of these levels across the reviewed studies.

Approximately 87% of studies remain at Level 1. These approaches typically use synthetic data to support initial concept validation and algorithm refinement without requiring real system connectivity. Implementations frequently combine standard RL libraries (e.g., Gymnasium) with custom DES environments. Of the 68 publications that specify their simulation environment, 21 mention SimPy [54,59,74,79,109,126], followed by Tecnomatix Plant Simulation with 13 mentions [43,46,50,81,121,133,134,198].

Level 2 is adopted by around 10% of approaches and integrates real industrial data to increase behavioral fidelity. Across the reviewed corpus, three recurring patterns are observed. First, operational datasets

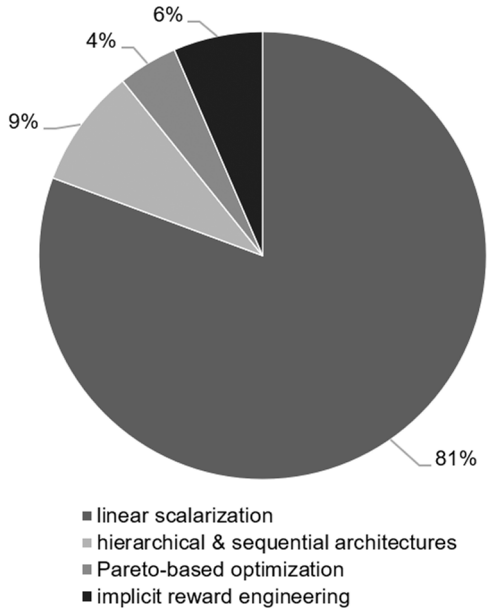

Fig. 19. Taxonomy of MOO Coordination Mechanisms in PPC (N = 93).

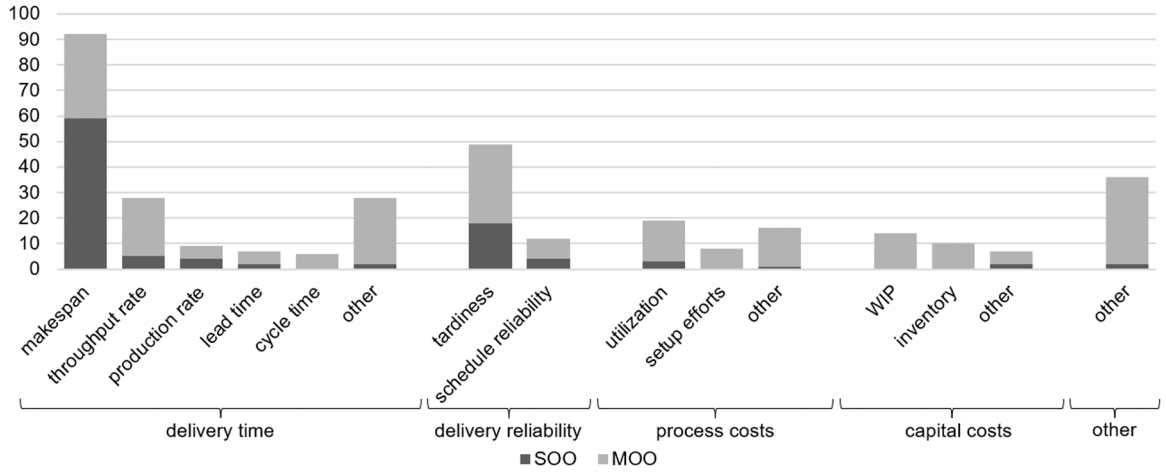

Fig. 18. Main optimization objectives (clustered based on Fig. 17).

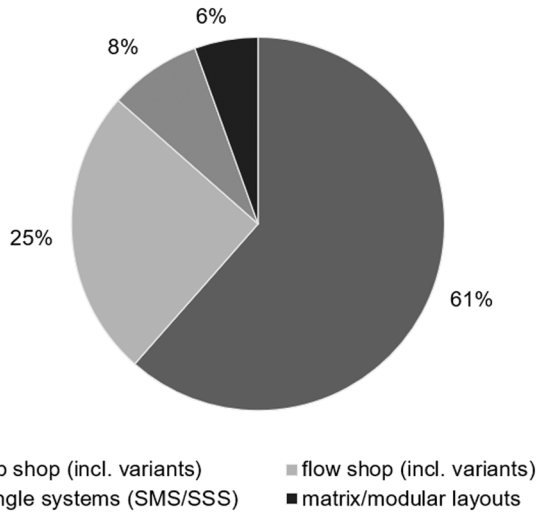

Fig. 20. Distribution of physical production scenarios across reviewed literature (N = 200).

Table 2
Distribution of variants within JS and FS scenarios.

Variant Dimension	JS (n = 123)	FS (n = 50)
base only (no modifiers)	43 (35%)	24 (48%)
flexible (F) or hybrid (H)	75 (61%)	19 (38%)
distributed (D)	6 (5%)	3 (6%)
reentrant (R)	10 (8%)	3 (6%)
permutation (P)	0 (0%)	6 (12%)

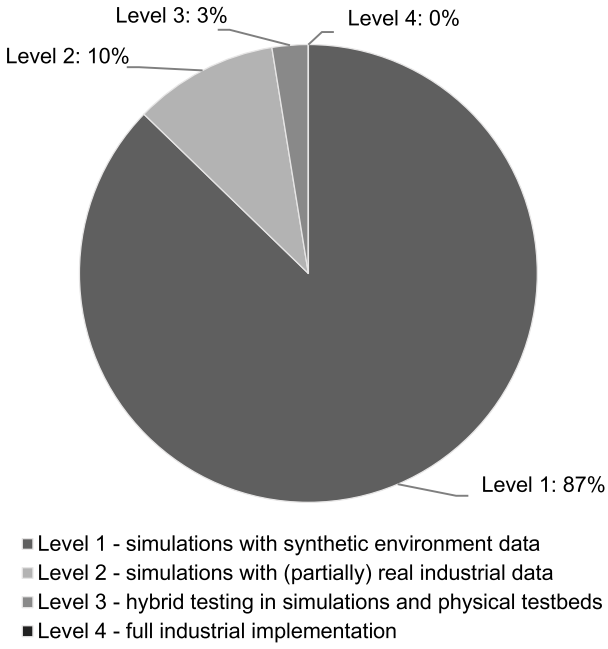

Fig. 21. Maturity level for industrial real-world application (N = 196).

are used to model shop-floor challenges in sectors such as automotive [85,89], assembly [62,81], pharma [28], and shipyards [124,187]. Second, energy-related datasets are used to train energy-aware agents, incorporating electricity pricing [47,202], photovoltaic availability [147], energy consumption [148], or stochastic energy rationing schedules [216]. Third, some studies integrate business-system data, including historical ERP data [156] or broader production datasets obtained from manufacturing partners [54].

Only about 3% of studies reach Level 3, where agents are evaluated in hybrid setups that connect simulation-trained policies to physical testbeds. These studies employ explicit integration layers that enable interaction with industrial hardware via standardized instruction pipelines. Reported enablers include industrial operating systems (for example FabOS) [129], edge computing nodes (IPCs) [205], and REST-based microservices [173] for interfacing with PLCs, AGVs, and robots. Safety and feasibility constraints are implemented through mechanisms such as action masking to avoid infeasible actions [173] or hyper-heuristics as an additional safety layer [129]. Several Level 3 studies also report transfer-oriented techniques, including two-stage transfer-learning and pre-training on benchmarks before physical deployment [141,164], as well as GNN-based representations to handle varying workstation configurations [141]. Where reported, real-time constraints require sub-second inference latency to meet the timing demands of physical hardware [164].

No identified studies demonstrate Level 4 industrial deployment in which an RL agent autonomously controls a live production system with continuous closed-loop feedback over sustained operation.

4.5. Performance analysis – comparison with benchmarks

The reviewed studies assess RL agent performance by benchmarking against established baselines, most frequently dispatching rules and heuristic policies, and less frequently optimization methods, human planners, or existing industrial practices. Reported improvements are study-specific and reflect heterogeneous instances, objectives, and baseline configurations. Values are therefore summarized as reported rather than normalized.

4.5.1. Benchmarking against dispatching rules and heuristics

Benchmarking against dispatching rules and heuristics is the most common validation approach in the reviewed literature. Baselines frequently include FIFO, SPT, LPT, and EDD. Reported improvements of RL-based approaches cover multiple objective types. For flow and time related objectives, studies report makespan reductions of 7% compared to SPT and LPT [27], throughput increases of 13.9–21% compared to FIFO and other practical dispatching rules [84,127], and lead time reductions of up to 63% [84]. Under disruption conditions such as machine failures, one study reports a slightly higher mean throughput rate compared to a nearest job first heuristic [221]. For delivery reliability, tardiness reductions of 4–12% relative to EDD and SPT are reported [75]. For cost-oriented objectives, one study reports profit increases of up to 7.3% through reduced penalty costs [131], and another reports total cost reductions of 20% compared to the best static or random planning heuristics via overtime scheduling decisions [154]. A recurrent mechanism in this benchmark class is the use of RL-based hyper--heuristics, where the agent selects among dispatching rules conditional on the system state. In this context, adaptive rule selection outperforms composite rules in 83.7% of test instances [111] and achieves lower makespans than any single-rule strategy in the reported experiments [96].

4.5.2. Benchmarking against metaheuristics and optimization solvers

A smaller subset benchmarks RL against metaheuristics (e.g., GA, SA) and exact solvers (e.g., CPLEX, Gurobi). Across these comparisons, studies report substantially lower runtime for trained RL policies, e.g. 4.59 s in [137] and “few” seconds in [57]. In comparable settings, GA or intelligent search is reported to require more than 1000 seconds or up to 51 min. One MARL system produced a production plan in 84 s, while a comparable SA method required three hours [147].

Several studies report that lower runtime does not necessarily coincide with lower objective performance. Improvements up to 46.3% over GA and Particle Swarm Optimization are reported in high-complexity scenarios [213]. Feasibility outcomes are also reported explicitly. In

one study, a PPO agent achieved a 100% feasibility rate across trials, whereas GA failed to find feasible schedules in 38% of cases [123]. Other comparisons emphasize a quality-time trade-off, such as 98% of SA solution quality at 2% of the computation time [55], or similar solution quality with considerably lower computational cost [138].

Against exact solvers, studies report strong performance for tools such as Gurobi and CP-SAT on small instances, with runtimes increasing as instance sizes grow [52,148]. In contrast, RL agents are reported to maintain low-latency execution while reaching 93–100% of CPLEX performance as instance size increases [70]. Finally, hybrid configurations integrate DRL into GAs to guide or improve search performance [54].

4.5.3. Benchmarking against human experts and industrial practice

Several studies benchmark RL agents against human planners, historical operational decisions, or manual planning baselines. Across these comparisons, reported improvements cover core logistics KPIs. Examples include a 24.2% throughput increase over conventional scheduling protocols [85], a 20% increase in production line utilization [27], and a 19.6% improvement in workload balancing compared to manual planning [187]. In addition, one study reports 5% total cost savings when comparing RL-based decisions to historical decisions from experienced planners [173].

Beyond outcome KPIs, some contributions quantify development effort. One study reports that a Double DQN agent reduced total development time from 40.4 h to 3 h by learning autonomously [113], indicating that RL can reduce reliance on manually crafted decision logic. Complementary evidence shows that sufficiently trained agents can match the performance of human students when applying complex heuristics to scheduling problems [31]. Collectively, these benchmarks document that RL policies can be competitive with, and in several settings exceed, human-derived or historically used decision baselines.

4.6. Research gaps analysis – proposed future research needs

Despite clear progress in applying RL to PPC, the literature highlights significant research gaps that require future attention. This section analyzes these needs by clustering them according to the components of the RL framework (Fig. 1). We supplement this structure with two overarching categories, “evaluation and transparency” and “other” (Fig. 22).

4.6.1. Future research – environment

Future research requirements for the RL environment focus on increasing fidelity to bridge the persistent sim-to-real gap, primarily through expanded systems complexity and seamless model integration (Fig. 23).

First, the literature emphasizes dynamic systems and uncertainty modeling, including resource volatility such as stochastic machine breakdowns or deterioration [33,46,60,65,66,82,84,111,112,128,148,169,178,206]. Parallel needs target demand and order fluctuations, including dynamic job arrivals, rush orders, and cancellations [33,52,66,82,105,116,148,154,166,169,180,206], as well as fluctuating demand levels [72,136,153]. Additional calls address higher operational variability through uncertain processing times [33,46,52,65,66,111,112,165], increased product variety [86,138,168], energy-related volatility such as price fluctuations [47,72,202], and other external irregularities [45,178,204].

Second, the literature calls for expanded structural complexity and broader scenario coverage, including more flexible production scenarios [47,57,221], semiconductor manufacturing [86], open-shop tasks [138], and multi-line variety production [186]. Scaling these models from single lines to full factory and supply chain representations remains a significant frontier [58,108,133,140]. This expanded operational scope requires integrating specialized logistical factors such as AGVs and inter-work cell coordination [220], tool management, and constrained buffer capacities [186], as well as adjacent processes like site transport [75] and final assembly [53].

Third, future work statements repeatedly emphasize validation and integration pathways that test generalizability, robustness, and scalability in real-world physical production systems [27,113] and real operational environments [55,126,127,130,138,160,186,190]. Upscaling applications to handle industrially relevant complexities, such as systems with more than 20 stations, is identified as a vital prerequisite for industrial acceptance [121]. To support this transition, the literature frequently proposes the integration of RL agents into digital twin frameworks. These frameworks serve as key enablers by providing essential connections to real-time data from manufacturing operations systems, decision support tools, and fleet management software [27,113,136,145].

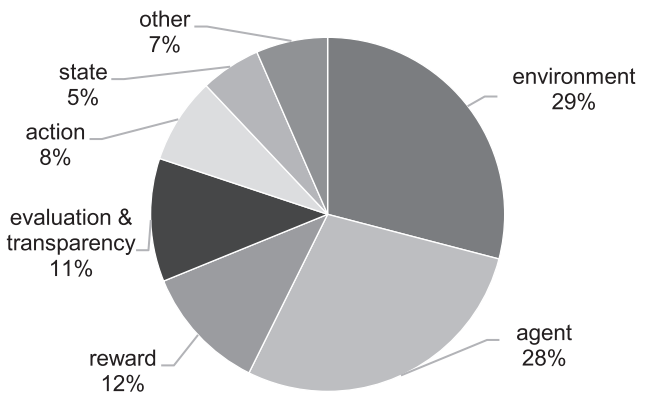

Fig. 22. Research directions for further RL development in PPC (N = 380).

4.6.2. Future research – agent

The agent-centric research needs reported in the reviewed literature primarily target advancing algorithmic foundations, learning efficiency, and model architectures (Fig. 24).

A frequently stated direction is the extension beyond traditional value-based baselines [47,57,105,111,112,168,178]. Multiple studies explicitly call for broader evaluation of policy-based methods, including PPO [30,33,34,92,111,112,151] and TRPO [30,57,111,192]. In addition, authors propose increased attention to algorithms supporting continuous action spaces [58,82] and to hybrid paradigms that combine RL with complementary optimization methods, such as game-theoretic mechanisms, metaheuristics, or hyperheuristics [30,92]. Several papers further emphasize the need for systematic benchmarking of less frequently used algorithms, including A3C [33,57,92,111,155,192], DDPG [34,92], and TD3 [192].

A second cluster of research needs addresses the learning process, with calls to improve training speed, robustness, and generalization for large-scale instances [44,73,135,151]. Proposed means include parallel and distributed training [54,121,218,219] and the use of transfer-oriented learning frameworks, such as meta-learning [135,181], transfer-learning [58,73], and representation learning via autoencoders for varying state spaces [58].

A third cluster focuses on model and agent architectures to accommodate heterogeneous inputs and changing system structures. Suggested directions include advanced deep learning designs [152,197], such as GNNs [162,170], recurrent networks [128,132], and convolutional layers [149,165], to support variable-sized inputs and evolving layouts [28,66,95,179]. Emerging proposals also include collaborative computation concepts (e.g., swarm intelligence) [197] and automated architecture search [149].

Finally, multiple studies highlight the expansion toward multi-agent formulations to address cooperative or competitive dynamics [69,74,101,181]. Related needs include hierarchical structures [53,155], specialized agent roles for sequencing and multi level planning [77,82,

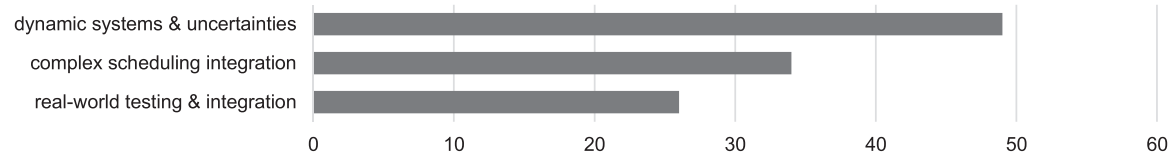

Fig. 23. Distribution of research need statements in the area of environment.

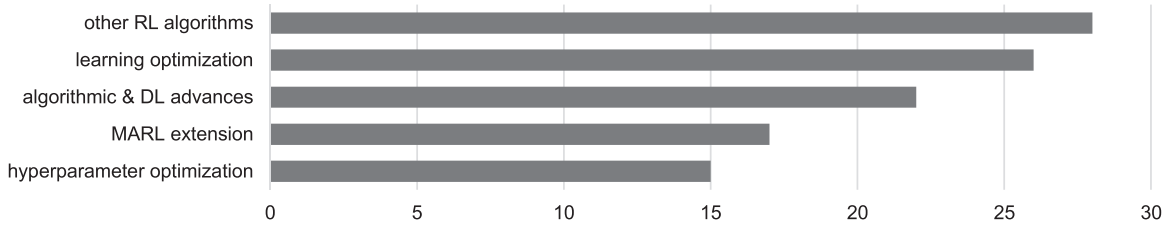

Fig. 24. Distribution of research needs in the area of agents.

121], and scaling architectures to production-line or multi-facility settings [173]. In parallel, several papers identify hyperparameter selection as a recurring methodological challenge and propose moving from ad-hoc tuning toward systematic refinement via sensitivity analysis [165] and automated tools like AutoKeras [82], aiming for more predictable configurations across scenarios [43,49,129,152,215].

4.6.3. *Future research – reward*

Research need statements related to reward design focus on broadening optimization objectives and refining the structural design of the reward function to enhance agent performance (Fig. 25).

The literature calls for broader objective definitions that extend beyond makespan [33] and tardiness [73,92]. This includes reward components for plan robustness under operational disruptions [170]. A second recurring direction is the integration of cost and resource-related objectives, such as machine utilization [33,73,175], production costs [111,112,178], and labor or inventory expenses [147]. Sustainability is increasingly emphasized, with studies proposing environmental objectives such as energy consumption [33,111,112,139,178], carbon emissions [90], and waste or water reduction [139] as explicit reward components.

A second cluster of research needs targets the reward function structure. Several studies propose tailored reward functions informed by historical production data [29,96,99,103,117,119,125,135,179,218,219]. Managing conflicting objectives is frequently highlighted through calls for improved weight calibration [49,130] and for MOO structures [137,199]. Additional statements emphasize the need for more dynamic reward mechanisms, including context-sensitive formulations that adapt to changing production conditions [165] and ranking-based approaches to prioritize specific operational outcomes [151].

4.6.4. *Future research – evaluation & transparency*

Research need statements in this category primarily call for stronger, more reproducible evaluation designs and improved transparency of agent behavior (Fig. 26).

A recurring theme is more comprehensive evaluation to quantify longer-term effects, including economic and sustainability-related outcomes [30,31,39,96,125,145,155,162,191,207]. In addition, studies frequently call for assessing generalizability across different production systems [48,79,172,183,184] and adaptability to unseen or changing shop configurations [45,47,49,55,59,126,153,154,156]. Several contributions further suggest scenario-specific assessment formats, such as techno-economic evaluations, to validate feasibility and value in particular industrial contexts [130].

A second cluster concerns systematic benchmarking to enable more comparable performance claims and to clarify where RL provides advantages. Beyond continued comparisons against classical heuristics and metaheuristics [49,122], authors propose benchmarking against advanced neighborhood-based scheduling methods [55] and against specific order-release mechanisms to test robustness under alternative control logics [153–155,192]. Some papers also advocate the development of new benchmark heuristics tailored to RL evaluation needs [145,190].

Finally, multiple studies highlight the need for explainability mechanisms that make learned strategies and individual actions interpretable in operational terms [59,123,171]. Proposed directions include methods for interpreting higher level policy behavior and action level explanations that support human review in high-stakes settings [32,79].

4.6.5. *Future research – action space*

Research need statements in the action space category emphasize the expansion of decision tasks and the technical enhancement of the action space (Fig. 27).

First, the literature calls for extending RL beyond traditional scheduling toward a broader set of PPC decision mechanisms, including order release, employee allocation, capacity control, and AGV route planning [43,77,114,163,180,185,218]. These statements frequently position the scope expansion toward higher level functions such as inventory and supply chain management. To enable such extensions, authors propose hybrid control designs [120] and collaborative decision architectures in which RL interacts with other optimization components [136]. Specific coordination problems, for example transport resource planning are highlighted [195]. Several papers also explicitly request action formulations that remain applicable across varying production scenarios without requiring full retraining [132].

Second, action space enhancement statements focus on refining decision granularity and feasibility. Proposed directions include increasing the number of available actions [33,34,122] and incorporating

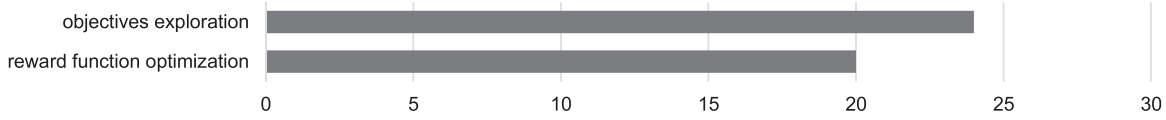

Fig. 25. Distribution of research needs in the area of reward.

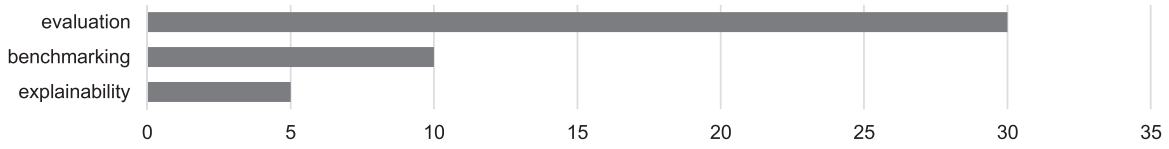

Fig. 26. Distribution of research needs in the area of evaluation & transparency.

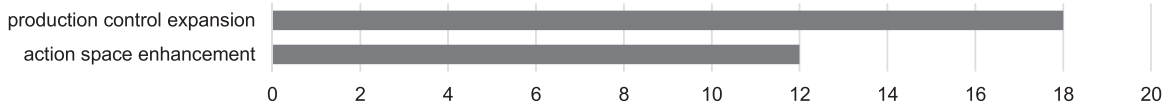

Fig. 27. Distribution of research needs in the area of action.

industry-oriented dispatching rule sets [117,118,205], alongside methodological needs for coping with resulting high-dimensional action spaces [171]. Continuous action spaces are discussed as a pathway to more fine-grained control [55,202]. In addition, multiple studies highlight the need for scalable state and action representations that are less dependent on system-specific constraints [128], including locally specialized action spaces aligned with local objectives [129]. Further proposals include structured action designs such as tuple-based decisions for lot sizing and scalable mechanisms for machine-specific tasks [173,213].

4.6.6. Future research – state space

The literature calls for expanding state representations and improving their practical suitability for industrial settings. Proposed feature extensions include operational metrics such as buffer utilization [29], workstation utilization and queue lengths [145], material transportation time [179], and product condition information [190]. In addition, several studies emphasize integrating maintenance-related information and indicators of system reliability to better reflect real operating conditions [69]. Beyond shop-floor signals, authors propose incorporating economic and planning-relevant inputs, including transportation costs [65], price fluctuations [69], and demand forecasts [171], to support higher level decision-making. Finally, a recurring direction is the integration of sustainability-related state variables, especially energy consumption and the availability of green energy sources such as wind or photovoltaic power [108,192].

5. Discussion

In the following section we discuss the results of this SLR by addressing the research questions, interpreting findings in relation to existing studies, and identifying study limitations. Overall, the evidence indicates that while RL algorithms are increasingly aligning with industrial PPC requirements, the bottleneck for real-world deployment has shifted to environment maturity, necessitating a future research focus on end-to-end system engineering, safety assurance, and standardized sim-to-real governance.

5.1. RQ 1 – What trends have recently emerged in the development and application of RL agents in PPC?

To answer RQ1, we structure the discussion along with the agent, action, and reward layers of the RL framework. The evidence suggests consolidation toward deployment readiness. This includes action abstractions that align with established PPC loops, and reward designs that shift the bottleneck from algorithms to explicit preference governance under industrial trade-offs.

5.1.1. Agent level trends

At the agent level, algorithm selection is increasingly driven by robustness and reproducibility under stochastic shop-floor dynamics, along with practical control interfaces and standard toolchains. In parallel, research is shifting from centralized control toward decentralized MARL architectures, reframing the core challenge from learning a single policy to orchestrating coordinated decisions across coupled models. As a consequence, coordination mechanisms, reward coupling, credit assignment, and state information sharing become dominant scaling constraints and shape how PPC problems must be formulated at the decision-mechanism level. Finally, the limited use of model-based approaches is a signal of feasibility rather than a lack of interest. Realizing this potential requires tractable abstractions, calibrated environment models or learned surrogates, as well as safety-aware planning that remains reliable under evolving routes, products, and equipment. This shift redefines contribution value from algorithmic improvements to maintainability and operational robustness under evolving factory configurations.

5.1.2. Action level trends

At the action level, the evidence shows consolidation around decision mechanisms that align with established PPC tasks and can be evaluated with clear KPIs. Rather than introducing novel action formulations, studies combine proven PPC logic with standard RL and increasingly position RL agents as meta-controllers that select among validated rules or policies. This hierarchical action design reduces action space complexity, stabilizes training, and improves interpretability under operational constraints, which makes it attractive for industrial adoption. However, the literature reveals a capability gap in action mechanisms requiring system-wide feasibility reasoning and cross-horizon coordination. These mechanisms include availability verification, multi-dimensional capacity control, and planning approval workflows. In these settings, limited state visibility, coupled objectives, and oversight requirements increase the difficulty for observation definition, safe learning, and evaluation. This explains the comparatively smaller footprint of WIP regulation and continuous control actions (e.g., batch size, energy, speed adjustment), which depend on context-specific targets and typically demand more sample efficient training procedures.

5.1.3. Reward level trends

At the reward level, the bottleneck has shifted from algorithm choice to preference specification and multi-objective trade-offs. Most reward designs deliberately optimize a narrow, KPI-centric portion of production value because trainability and interpretability dominate design decisions. This stabilizes learning, but it systematically underrepresents financial and cross-functional objectives and leaves trade-offs implicit and vulnerable to context shifts (e.g., mix changes, disruptions, and energy constraints). Going forward, reward engineering should be approached as a control-stack design problem by separating a strategic layer that specifies preferences or constraints from a shaping layer that improves learning without distorting intent. This enables context-adaptive preference settings and constraint-aware formulations that align agent behavior with operational risk tolerance and business priorities. The practical implication is that reward reporting should adopt

standardized templates (objectives, weights/constraints, shaping terms, and sensitivity analyses) so policies that appear strong based on isolated KPIs remain reliable under deployment-time trade-offs and distribution shift.

5.2. RQ 2 – What progress have RL approaches made in implementing the environment in real-world production systems?

To answer RQ2, we examine progress in realizing and integrating RL environments for industrial PPC through environment maturity, scenario selection patterns, and sim-to-real transfer governance. The evidence suggests that the environment is the primary maturity gate for RL in PPC. Overall, this is driven by realism gaps, infrastructure constraints, and missing validation protocols, which explains why most work remains limited to the simulator despite strong algorithmic results.

5.2.1. Environment maturity as the primary bottleneck

Throughout the literature, environments are primarily designed as research instruments for algorithm development rather than as engineered interfaces for industrial control. This is a rational posture in the early phases because virtual environments enable rapid iteration, controlled scenario design, and risk-free experimentation. However, this approach creates a maturity ceiling. When noise, delays, missing data, partial observability, and execution constraints are abstracted away, policy performance becomes tightly coupled to simulator assumptions, and strong results may reflect simulator fit rather than operational robustness.

Progress toward realism is not a simple upgrade from synthetic to real data. It requires an end-to-end capability stack that encompasses consistent observation and event semantics, enforceable data quality governance, and repeatable calibration pipelines that keep the environment aligned with changing production logic. This dependency chain explains why real industrial data connections and physical testbeds remain rare. The limiting factor is as much organizational readiness and cross-boundary data access as it is technical integration effort. A further constraint is the fidelity complexity trade-off in environment design. Large, highly coupled real-world PPC processes are difficult to model at sufficient fidelity for training without making the environment itself complex, slow to simulate, or costly to calibrate. This undermines scalability and limits the agent's ability to learn strategies that transfer reliably to reality.

5.2.2. Scenario Bias and the Deployment Readiness Gap

Accordingly, studies favor replayable, easily parameterized environment designs, which accelerates algorithm iteration but does not necessarily translate into progress toward industrialization. These scenario selection patterns reinforce this maturity signal. Research environments are often chosen where RL can demonstrate visible gains over simple baselines in the presence of combinatorial complexity and stochastic disruption. While this approach supports a clear value narrative, it leaves high-volume, high-precision, low-error-margin domains underrepresented, despite their strategic importance. In such settings, exploration risk is unacceptable, and the environment must provide not only behavioral fidelity but also credible constraint satisfaction and safety assurances before any real execution is considered. The current underrepresentation is therefore best interpreted as a deployment readiness gap, not a relevance gap.

5.2.3. Sim-to-Real Transfer and Infrastructure Requirements

A central opportunity is to further develop systematic transfer governance. The limited set of near-real demonstrations shows that sim-to-real is usually achieved through tailored engineering with case specific interfaces, safety layers, and validation choices. What is missing is a reference process with comparable criteria that defines what evidence is required to move from simulation to testbed and from testbed to live operations. Without shared protocols, generalization remains difficult to benchmark and scale across plants.

Environment maturity is closely related to infrastructure maturity. Industrial embedding requires a reliable real-time loop, stable interface standards, explicit latency budgets, cybersecurity constraints, and monitoring of failure modes. The operational challenge is less about computing actions and more about ensuring timely observations, feasible execution, and safe degradation under missing or inconsistent inputs. This shifts environment engineering from a research artifact toward a product architecture requirement. Once trained, RL is repeatedly positioned as an adaptive decision engine capable of delivering high-quality actions under disturbances with low inference latency. This capability becomes decisive when classical optimization cannot be rerun frequently enough at scale. The industrial value proposition is therefore centered on near real-time, disturbance-aware control, not on marginal offline optimality.

Taken together, these findings indicate that transfer readiness is primarily an environment engineering and governance challenge, consistent with prior observations by Panzer & Bender [10], Rolf et al. [11], and with challenges encountered in other RL application fields [222]. This motivates the cross-cutting gaps and research agenda synthesized in RQ3.

5.3. RQ 3 – What are the key research needs and gaps for deploying RL in PPC?

To answer RQ3, we synthesize the reported barriers to industrial adoption from the literature and translate them into an actionable research agenda. The research gaps analysis in Section 4.6 catalogues future research needs (environment fidelity, agent algorithms/architectures, expanded state/action spaces, reward design, and evaluation transparency). Building on this, we first consolidate the recurring root causes behind the sim-to-real transfer barrier. Then, we derive system engineering priorities that can transition RL in PPC from prototypes to deployment-grade control.

5.3.1. Research Gaps

The transfer barrier is not caused by the absence of a single component. Rather, it results from a series of closely interrelated socio-technical causes spanning data accessibility and semantic instability, IT/OT integration complexity, safety assurance and accountability under exploration, and human-machine collaboration. An additional evaluation bottleneck compounds these issues by limiting the field's ability to measure, compare, and communicate progress systematically.

A root cause for the development of training environments is data accessibility and semantic instability, which directly constrains the derivation and maintenance of training environments. PPC environments have heterogeneous state and event semantics across production control systems (e.g., ERP, MES, SCADA), and equipment controllers. The available logs reflect operational reality rather than curated research datasets. This makes it difficult to define transferable observation representations, validate rewards against business outcomes, and reproduce experiments across sites. In practice, access is further constrained by ownership boundaries, cybersecurity requirements, and the effort required to extract, clean, and align operational histories. The complexity of integrating IT and OT levels complicates data provision and model integration. Heterogeneous interfaces, site-specific architectures, and strict latency and reliability requirements make environment integration and policy execution hard to transfer and resource intensive to maintain. Safety assurance and accountability under exploration is a further gap, which is critical in high-stakes, low-error-margin environments. Trial and error learning can be economically unacceptable and may violate hard operational constraints. This shifts the research problem from learning performance to assurance. Industrial stakeholders need a safety case with evidence of constraint compliance and predictable behavior under disturbances, sensor noise, and rare events.

Beyond these structural causes, an emerging requirement is human- and operator-aware control. In addition to processing and machine data, other information is relevant for a full understanding of the system. This includes operator skills, fatigue, shift schedules, and guidelines for manual override as part of the RL state, scope of action, or reward structure. This gap is becoming increasingly relevant in contexts where human-centricity, resilience, and workforce constraints determine feasibility and acceptance. Deployment-grade PPC therefore requires shared autonomy, governance, and responsibility at the human-machine interface.

Finally, an evaluation bottleneck limits cumulative progress. Comparability remains weak because environments, metrics, disturbances, baselines, and reporting practices vary widely, making it difficult to distinguish methodological advances from favorable scenario design. Without shared benchmarks, reporting templates, and robustness protocols, generalization claims remain narrative and deployment decisions lack a consistent evidence basis.

5.3.2. Research Agenda

To close the identified gaps, the field should prioritize system engineering for industrial readiness, without neglecting algorithmic and methodological advances that improve robustness, safety, and learning efficiency. The following agenda translates the analysis into specific research goals and follows the deployment pipeline from environment generation and integration, to validation and governance, to human adoption and scalable application.

(1) **Automated, Scalable Learning Environments for Complex Production Systems:** To effectively train RL agents in these systems, we need methods and procedures that provide learning environments with reasonable effort and sufficiently accurate reflection of relevant real-world dynamics. This requires consistent fidelity–complexity governance. The environment must be realistic enough to enable transferable strategies while remaining modular, maintainable, and computationally efficient to ensure scalable training and iteration. A key factor is the automated derivation and maintenance of RL learning environments from real production data, including semantic layers (data and event semantics), model derivation, calibration, drift detection, and recalibration.

(2) **Reference Architectures and Sim-to-Real Guides for Transferable RL Systems:** Reference architectures should standardize the interfaces between the learning environment, decision model, and execution level. This includes clearly defining real-time and latency requirements, cybersecurity and monitoring interfaces, and technical operating agreements, such as data formats, update frequencies, and fallback mechanisms. These standardizations make RL system designs transferable across sites and reduce the effort required for case- or plant-specific IT/OT integration solutions. Additionally, reference guides should operationalize the transfer from simulation to reality by combining standardized stress tests (e.g., layout changes, demand fluctuations, disturbances, sensor noise, and constraint violations) with repeatable procedures for calibrating and recalibrating the environment.

(3) **Operationalized Deployment and Benchmarking Infrastructure:** Stage gates should specify the minimum requirements for transitioning from simulation to testbed, from testbed to shadow mode, and from shadow mode to live operation. These criteria can serve as standardized templates for assessment and reporting, as well as reference benchmark suites that enable the repeatable comparison of PPC tasks, scenarios, and sites. This development provides industry with a clear framework for deciding whether to implement RL projects, assists with testing and change management (versioning, rollback, and monitoring), and helps convert generalization claims into deployment decisions.

(4) **Employee integration and trust as design goals (human-in-the-loop):** Future research should emphasize the integration of employees in the introduction of RL systems and building trust as independent design goals. This includes systematically transferring implicit experiential knowledge into model development. It also includes integrating workforce availability, qualification, and deployment restrictions into appropriate state representations. At the same time, RL systems should enable shared autonomy by providing explicit override, approval, and escalation mechanisms, rather than relying exclusively on autonomous decision-making. Evaluations of these approaches should include not only logistical KPIs but also work-related outcomes such as workload, stability of task distribution, intervention/override rates, and trust in automated recommendations. This closes the gap between humans and machines, positioning RL as a collaborative decision-making tool rather than a fully autonomous controller. Transparent and traceable decisions are required so that human decision-makers can review, validate, and take responsibility for the RL control strategy.

(5) **Expanding the Range of PPC Tasks and Scaling Hierarchical MARL Systems:** Another key research focus is gathering experience across the entire range of PPC tasks and evaluating RL for both established and new control mechanisms. This includes tasks and mechanisms that are currently only partially feasible due to technical, or organizational restrictions but could become feasible in the future thanks to improved data availability, system integration, and RL assistance concepts. At the same time, we should investigate hierarchical approaches and large-scale MARL more closely to robustly orchestrate coordinated decisions across coupled resources and time horizons. However, this scaling requires safety assurance and operational monitoring to remain primary design goals. For MARL systems, verifiable compliance with constraints and predictable behavior under uncertainty are critical to success and trust. Therefore, exploring new tasks and mechanisms must consistently involve assurance-oriented validation, constraint handling, and continuous monitoring.

In sum, the central research need is a transition from isolated policy performance toward end-to-end operational capability. The most valuable future contributions will define and validate repeatable transfer guides and governance criteria that make RL systems portable, auditable, and safe under real production variability. This is the pathway from isolated demonstrations to scalable deployment programs across plants and sectors.

5.4. Limitations

Although this SLR was conducted using systematic methods, it is subject to certain limitations. First, limitations in our review process include the potential for missed papers due to the specific search terms and databases used. Another factor is subjectivity in study selection, as inter-rater reliability was not formally assessed. Including both journal articles and conference papers may introduce bias regarding the rigor of the findings. Since positive results are more likely to be published, publication bias is another potential limitation.

Second, significant limitations arise from the heterogeneity of the primary literature itself. We found that varying levels of detail in methodological descriptions, coupled with diverse evaluation metrics and test conditions, complicated direct comparisons across different approaches. This lack of standardization makes it difficult to draw overarching conclusions about the added value of specific RL models. Furthermore, our qualitative analysis revealed thematic overlaps where single studies influence multiple research areas, which could potentially distort the absolute numbers presented in our findings. Future reviews could address these limitations through broader database coverage and by incorporating additional sources, such as expert interviews, to enrich

the analysis.

6. Conclusion

This SLR of 196 publications from 2018 to 2024 shows that RL in PPC is progressing from early proof-of-concept studies toward deployment-oriented agent designs, compatible action representations, and more explicit handling of industrial trade-offs. The trend is reflected in the shift from dominant DQN-based approaches toward stability-oriented on-policy approaches (e.g., PPO), alongside growing interest in MARL for decentralized production settings. At the same time, industrial maturity remains limited. Most contributions are validated in simulator-centric setups, and only a small share moves toward physical testing or sustained operational embedding. Taken together, these findings suggest that the bottleneck has moved from demonstrating that RL can optimize isolated PPC KPIs to engineering transferable and governable control stacks. Closing the sim-to-real gap will require higher-fidelity environments, standardized and transparent evaluation protocols, and safety and assurance mechanisms that enable trustworthy integration into real-time PPC loops.

In line with the discussion, environment maturity is a primary gate for further progress. First, future work should deliver automated, scalable learning environments derived from and maintained by production data with explicit fidelity complexity governance. Second, it should establish reference architectures and sim-to-real transfer guides that standardize the interfaces between the environment, policy, and execution. Third, it should develop benchmarking and evaluation infrastructure with standardized reporting and robustness protocols. Fourth, human-in-the-loop deployment and trust-by-design should be treated as core requirements through shared autonomy, auditability, and clear operational governance. Finally, it should expand the range of PPC tasks while scaling hierarchical MARL under robust validation, verifiable constraint handling, and continuous monitoring.

For practitioners, these findings indicate that RL is not yet a general-purpose solution for broad PPC rollout. Its near-term business case is strongest in narrowly scoped, disturbance-driven control loops where rapid, repeated re-optimization is required and classical optimization cannot be run reliably within the available decision latency. We therefore recommend based on patterns observed a staged adoption pathway aligned with transfer governance. Start with decision support in simulation, progress to shadow mode with offline/online monitoring, then supervised operation with human override, and only then consider limited autonomous control for well-bounded tasks. Across all stages, success depends on a production-grade environment interface, evidence-driven robustness validation under realistic variability, and clearly defined stage gates and operational acceptance criteria co-developed with shop-floor and PPC domain experts.

CRediT authorship contribution statement

Jesse Mayerhoff: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Conceptualization. **Matthias Schmidt:** Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. J. Mayerhoff is employed by Volkswagen AG. The company had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Data availability

No new data were created or analyzed during this study. Data sharing is not applicable to this article.

References

[1] Wiendahl H.-P. Betriebsorganisation für Ingenieure. 7., aktualisierte Aufl. München: Hanser; 2010.
[2] Zhang W, Bao X, Hao X, Gen M. Metaheuristics for multi-objective scheduling problems in industry 4.0 and 5.0: a state-of-the-arts survey. Front Ind Eng 2025;3:1540022. https://doi.org/10.3389/fieng.2025.1540022.
[3] Ghobakhloo M. Industry 4.0, digitization, and opportunities for sustainability. J Clean Prod 2020;252:119869. https://doi.org/10.1016/j.jclepro.2019.119869.
[4] Sutton RS, Barto A. Reinforcement learning: an introduction. Second edition. Cambridge, Massachusetts London, England: The MIT Press; 2018.
[5] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Play Atari Deep Reinf Learn 2013.
[6] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016;529:484–9. https://doi.org/10.1038/nature16961.
[7] Brockman G., Cheung V., Pettersson L., Schneider J., Schulman J., Tang J., et al. OpenAI Gym [Internet]. arXiv; 2016 [cited 2024 Apr 1]. ⟨http://arxiv.org/abs/1606.01540⟩. Accessed 1 Apr 2024.
[8] Kitchenham B. Proced Perform Syst Rev 2004.
[9] Esteso A, Peidro D, Mula J, Díaz-Madroñero M. Reinforcement learning applied to production planning and control. Int J Prod Res 2023;61:5772–89. https://doi.org/10.1080/00207543.2022.2104180.
[10] Panzer M, Bender B. Deep reinforcement learning in production systems: a systematic literature review. Int J Prod Res 2022;60:4316–41. https://doi.org/10.1080/00207543.2021.1973138.
[11] Rolf B, Jackson I, Müller M, Lang S, Reggelin T, Ivanov D. A review on reinforcement learning algorithms and applications in supply chain management. Int J Prod Res 2023;61:7151–79. https://doi.org/10.1080/00207543.2022.2140221.
[12] Li C, Zheng P, Yin Y, Wang B, Wang L. Deep reinforcement learning in smart manufacturing: A review and prospects. CIRP J Manuf Sci Technol 2023;40:75–101. https://doi.org/10.1016/j.cirpj.2022.11.003.
[13] Kayhan BM, Yildiz G. Reinforcement learning applications to machine scheduling problems: a comprehensive literature review. J Intell Manuf 2023;34:905–29. https://doi.org/10.1007/s10845-021-01847-3.
[14] Schmidt M, Schäfers P. The Hanoverian Supply Chain Model: modelling the impact of production planning and control on a supply chain's logistic objectives. Prod Eng 2017;11:487–93. https://doi.org/10.1007/s11740-017-0740-9.
[15] Watkins CJCH. Learning from Delayed Rewards [phd thesis]. Oxford: King's College; 1989.
[16] van Hasselt H., Guez A., Silver D. Deep Reinforcement Learning with Double Q-learning [Internet]. arXiv; 2015 [cited 2024 Apr 3]. ⟨http://arxiv.org/abs/1509.06461⟩. Accessed 3 Apr 2024.
[17] Wang Z., Schaul T., Hessel M., van Hasselt H., Lanctot M., de Freitas N. Dueling Network Architectures for Deep Reinforcement Learning [Internet]. arXiv; 2016 [cited 2024 Apr 3]. ⟨http://arxiv.org/abs/1511.06581⟩. Accessed 3 Apr 2024.
[18] Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O. Proximal Policy Optimization Algorithms [Internet]. arXiv; 2017 [cited 2024 Apr 2]. ⟨http://arxiv.org/abs/1707.06347⟩. Accessed 2 Apr 2024.
[19] Schulman J., Levine S., Moritz P., Jordan M.I., Abbeel P. Trust Region Policy Optimization [Internet]. arXiv; 2017 [cited 2024 Apr 2]. ⟨http://arxiv.org/abs/1502.05477⟩. Accessed 2 Apr 2024.
[20] Mnih V., Badia A.P., Mirza M., Graves A., Lillicrap T.P., Harley T., et al. Asynchronous Methods for Deep Reinforcement Learning [Internet]. arXiv; 2016 [cited 2024 Apr 2]. ⟨http://arxiv.org/abs/1602.01783⟩. Accessed 2 Apr 2024.
[21] Tan M. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. International Conference on Machine Learning (ICML); 1993.
[22] Roijers DM, Vamplew P, Whiteson S, Dazeley R. A Survey of Multi-Objective Sequential Decision-Making. J Artif Intell Res 2013;48:67–113. https://doi.org/10.1613/jair.3987.
[23] Tranfield D, Denyer D, Smart P. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. Br J Manag 2003;14:207–22. https://doi.org/10.1111/1467-8551.00375.
[24] Durach CF, Kembro J, Wieland A. A New Paradigm for Systematic Literature Reviews in Supply Chain Management. J Supply Chain Manag 2017;53:67–85. https://doi.org/10.1111/jscm.12145.
[25] Webster J, Watson RT. Guest Ed Anal Prep Future Writ a Lit Rev 2002.
[26] Ahn J, Yun S, Kwon J-W, Kim W-T. Literacy deep reinforcement learning-based federated digital twin scheduling for the software-defined factory. Electronics 2024;13:4452. https://doi.org/10.3390/electronics13224452.
[27] Alexopoulos K, Nikolakis N, Bakopoulos E, Siatras V, Mavrothalassitis P. Machine learning agents augmented by digital twinning for smart production scheduling. IFAC-Pap 2023;56:2963–8. https://doi.org/10.1016/j.ifacol.2023.10.1420.
[28] Ali Said NE-D, Samaha Y, Azab E, Shihata LA, Mashaly M. An online reinforcement learning approach for solving the dynamic flexible job-shop scheduling problem for multiple products and constraints. 2021 Int Conf Comput Sci Comput Intell CSCI [Internet]. Las Vegas, NV, USA. IEEE; 2021. p. 134–9. https://doi.org/10.1109/CSCI54926.2021.00095. cited 2024 Aug 1.
[29] Altenmüller T, Stüker T, Waschneck B, Kuhnle A, Lanza G. Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. Prod Eng 2020;14:319–28. https://doi.org/10.1007/s11740-020-00967-8.

[30] Ballouch M, Souissi O, El-Fenni MR. Enhancing Control in Manufacturing and Microgrid Systems: Deep Reinforcement Learning with Double Q-Learning. 2023 14th Int Conf Intell Syst Theor Appl SITA [Internet]. Casablanca, Morocco: IEEE; 2023. p. 1–7. https://doi.org/10.1109/SITA60746.2023.10373737. cited 2024 Aug 1.
[31] Burggräf P, Wagner J, Koke B, Bamberg M. Performance assessment methodology for AI-supported decision-making in production management. Procedia CIRP 2020;93:891–6. https://doi.org/10.1016/j.procir.2020.03.047.
[32] Burggräf P, Wagner J, Saßmannshausen T, Ohrndorf D, Subramani K. Multi-agent-based deep reinforcement learning for dynamic flexible job shop scheduling. Procedia CIRP 2022;112:57–62. https://doi.org/10.1016/j.procir.2022.09.024.
[33] Chang J, Yu D, Hu Y, He W, Yu H. Deep reinforcement learning for dynamic flexible job shop scheduling with random job Arrival. Processes 2022;10:760. https://doi.org/10.3390/pr10040760.
[34] Chang J, Yu D, Zhou Z, He W, Zhang L. Hierarchical reinforcement learning for multi-objective real-time flexible scheduling in a smart shop floor. Machines 2022;10:1195. https://doi.org/10.3390/machines10121195.
[35] Chang Y-H, Liu C-H, You SD. Scheduling for the flexible job-shop problem with a dynamic number of machines using deep reinforcement learning. Information 2024;15:82. https://doi.org/10.3390/info15020082.
[36] Chen J, Yu K, Rodrigues JJPC, Guizani M, Sato T. FAG-scheduler: Privacy-Preserving Federated Reinforcement Learning with GRU for Production Scheduling on Automotive Manufacturing. GLOBECOM 2023 - 2023 IEEE Glob Commun Conf [Internet]. Kuala Lumpur, Malaysia: IEEE; 2023. p. 5147–52. https://doi.org/10.1109/GLOBECOM54140.2023.10437362. cited 2024 Aug 1.
[37] Chen W, Zhang Z, Tang D, Liu C, Gui Y, Nie Q, et al. Probing an LSTM-PPO-Based reinforcement learning algorithm to solve dynamic job shop scheduling problem. Comput Ind Eng 2024;197:110633. https://doi.org/10.1016/j.cie.2024.110633.
[38] Chen J, Zhang H, Ma W, Xu G. Real-time scheduling for two-stage assembly flowshop with dynamic job arrivals by deep reinforcement learning. Adv Eng Inf 2024;62:102632. https://doi.org/10.1016/j.aei.2024.102632.
[39] Dasbach T, Olbort J, Wenk F, Ander R. Sequencing Through a Global Decision Instance Based on a Neural Network. In: Canciglieri Junior O, Noël F, Rivest L, Bouras A, editors. Prod Lifecycle Manag Green Blue Technol Support Smart Sustain Organ [Internet]. Cham: Springer International Publishing; 2022. p. 334–44. https://doi.org/10.1007/978-3-030-94335-6_24 [cited 2024 Aug 1].
[40] de Jong B, Schelthoff K, Bianco RL, van Jaarsveld W. Long-Term Rapid Scenario Planning in the Semiconductor Industry Using Deep Reinforcement Learning. 2024 Winter Simul Conf WSC [Internet]. Orlando, FL, USA: IEEE; 2024. p. 1818–29. https://doi.org/10.1109/WSC63780.2024.10838925. cited 2025 Oct 2.
[41] Di Y, Deng L, Zhang L. A collaborative-learning multi-agent reinforcement learning method for distributed hybrid flow shop scheduling problem. Swarm Evol Comput 2024;91:101764. https://doi.org/10.1016/j.swevo.2024.101764.
[42] Ding L, Guan Z, Rauf M, Yue L. Multi-policy deep reinforcement learning for multi-objective multiplicity flexible job shop scheduling. Swarm Evol Comput 2024;87:101550. https://doi.org/10.1016/j.swevo.2024.101550.
[43] Dittrich M-A, Fohlmeister S. Cooperative multi-agent system for production control using reinforcement learning. CIRP Ann 2020;69:389–92. https://doi.org/10.1016/j.cirp.2020.04.005.
[44] Dong Z, Ren T, Weng J, Qi F, Wang X. Minimizing the Late Work of the Flow Shop Scheduling Problem with a Deep Reinforcement Learning Based Approach. Appl Sci 2022;12:2366. https://doi.org/10.3390/app12052366.
[45] Du Y, Li J. A deep reinforcement learning based algorithm for a distributed precast concrete production scheduling. Int J Prod Econ 2024;268:109102. https://doi.org/10.1016/j.ijpe.2023.109102.
[46] Eriksson K, Ramasamy S, Zhang X, Wang Z, Danielsson F. Conceptual framework of scheduling applying discrete event simulation as an environment for deep reinforcement learning. Procedia CIRP 2022;107:955–60. https://doi.org/10.1016/j.procir.2022.05.091.
[47] Felder M, Steiner D, Busch P, Trat M, Sun C, Bender J, et al. Energy-Flexible Job-Shop Scheduling Using Deep Reinforcement Learning. Hannover: publish-Ing; 2023. https://doi.org/10.15488/13454.
[48] Gankin D, Mayer S, Zinn J, Vogel-Heuser B, Endisch C. Modular Production Control with Multi-Agent Deep Q-Learning. 2021 26th IEEE Int Conf Emerg Technol Fact Autom ETFA [Internet]. Vasteras, Sweden: IEEE; 2021. p. 1–8. https://doi.org/10.1109/ETFA45728.2021.9613177. cited 2024 Aug 1.
[49] Gerpott FT, Lang S, Reggelin T, Zadek H, Chaopaisarn P, Ramingwong S. Integration of the A2C Algorithm for Production Scheduling in a Two-Stage Hybrid Flow Shop Environment. Procedia Comput Sci 2022;200:585–94. https://doi.org/10.1016/j.procs.2022.01.256.
[50] Ghaleb M, Namoura HA, Taghipour S. Reinforcement Learning-based Real-time Scheduling Under Random Machine Breakdowns and Other Disturbances: A Case Study. 2021 Annu Reliab Maintainab Symp RAMS [Internet]. Orlando, FL, USA: IEEE; 2021. p. 1–8. https://doi.org/10.1109/RAMS48097.2021.9605791. cited 2024 Aug 1.
[51] Golpayegani F, Ghanadbashi S, Zarchini A. Advancing Sustainable Manufacturing: Reinforcement Learning with Adaptive Reward Machine Using an Ontology-Based Approach. Sustainability 2024;16:5873. https://doi.org/10.3390/su16145873.
[52] Gong H, Xu W, Sun W, Xu K. Multi-Objective Flexible Flow Shop Production Scheduling Problem Based on the Double Deep Q-Network Algorithm. Processes 2023;11:3321. https://doi.org/10.3390/pr11123321.
[53] Gros TP, Gros J, Wolf V. Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning. 2020 Winter Simul Conf WSC [Internet]. Orlando, FL, USA: IEEE; 2020. p. 3032–44. https://doi.org/10.1109/WSC48552.2020.9383884. cited 2024 Aug 1.
[54] Grumbach F, Badr NEA, Reusch P, Trojahn S. A Memetic Algorithm With Reinforcement Learning for Sociotechnical Production Scheduling. IEEE Access 2023;11:68760–75. https://doi.org/10.1109/ACCESS.2023.3292548.
[55] Grumbach F, Müller A, Reusch P, Trojahn S. Robust-stable scheduling in dynamic flow shops based on deep reinforcement learning. J Intell Manuf 2024;35:667–86. https://doi.org/10.1007/s10845-022-02069-x.
[56] Gui Y, Zhang Z, Tang D, Zhu H, Zhang Y. Collaborative dynamic scheduling in a self-organizing manufacturing system using multi-agent reinforcement learning. Adv Eng Inf 2024;62:102646. https://doi.org/10.1016/j.aei.2024.102646.
[57] Han B-A, Yang J-J. Research on Adaptive Job Shop Scheduling Problems Based on Dueling Double DQN. IEEE Access 2020;8:186474–95. https://doi.org/10.1109/ACCESS.2020.3029868.
[58] Harb J, Riedmann S, Wegenkittl S. Strategies for Developing a Supervisory Controller with Deep Reinforcement Learning in a Production Context. 2022 IEEE Conf Control Technol Appl CCTA [Internet]. Trieste, Italy. IEEE; 2022. p. 869–74. https://doi.org/10.1109/CCTA49430.2022.9966086. cited 2024 Aug 1.
[59] He Z, Thürer M, Zhou W. The use of reinforcement learning for material flow control: An assessment by simulation. Int J Prod Econ 2024;274:109312. https://doi.org/10.1016/j.ijpe.2024.109312.
[60] Heik D, Bahrpeyma F, Reichelt D. Application of Multi-agent Reinforcement Learning to the Dynamic Scheduling Problem in Manufacturing Systems. In: Nicosia G, Ojha V, La Malfa E, La Malfa G, Pardalos PM, Umeton R, editors. Mach Learn Optim Data Sci [Internet. Cham: Springer Nature Switzerland; 2024. p. 237–54. https://doi.org/10.1007/978-3-031-53966-4_18. cited 2024 Aug 1].
[61] Heik D, Bahrpeyma F, Metzler J, Reichelt D. EncodedRL: Solving the Dynamic Scheduling Problem using Multi-Agent Reinforcement Learning Based on an Encoded State Representation. 2024 IEEE 22nd Int Conf Ind Inform INDIN [Internet]. Beijing, China: IEEE; 2024. p. 1–8. https://doi.org/10.1109/INDIN58382.2024.10774356. cited 2025 Oct 2.
[62] Heik D, Bahrpeyma F, Reichelt D. Study on the application of single-agent and multi-agent reinforcement learning to dynamic scheduling in manufacturing environments with growing complexity: Case study on the synthesis of an industrial IoT Test Bed. J Manuf Syst 2024;77:525–57. https://doi.org/10.1016/j.jmsy.2024.09.019.
[63] Ho K-H, Cheng J-Y, Wu J-H, Chiang F, Chen Y-C, Wu Y-Y, et al. Residual Scheduling: A New Reinforcement Learning Approach to Solving Job Shop Scheduling Problem. IEEE Access 2024;12:14703–18. https://doi.org/10.1109/ACCESS.2024.3357969.
[64] Hofmann C, Krahe C, Stricker N, Lanza G. Autonomous production control for matrix production based on deep Q-learning. Procedia CIRP 2020;88:25–30. https://doi.org/10.1016/j.procir.2020.05.005.
[65] Huang J-P, Gao L, Li X-Y, Zhang C-J. A cooperative hierarchical deep reinforcement learning based multi-agent method for distributed job shop scheduling problem with random job arrivals. Comput Ind Eng 2023;185:109650. https://doi.org/10.1016/j.cie.2023.109650.
[66] Huang J-P, Gao L, Li X-Y, Zhang C-J. A novel priority dispatch rule generation method based on graph neural network and reinforcement learning for distributed job-shop scheduling. J Manuf Syst 2023;69:119–34. https://doi.org/10.1016/j.jmsy.2023.06.007.
[67] Huang J, Huang S, Moghaddam SK, Lu Y, Wang G, Yan Y, et al. Deep Reinforcement Learning-Based Dynamic Reconfiguration Planning for Digital Twin-Driven Smart Manufacturing Systems With Reconfigurable Machine Tools. IEEE Trans Ind Inf 2024;20:13135–46. https://doi.org/10.1109/TII.2024.3431095.
[68] Huang D, Zhao H, Zhang L, Chen K. Learning to Dispatch for Flexible Job Shop Scheduling Based on Deep Reinforcement Learning via Graph Gated Channel Transformation. IEEE Access 2024;12:50935–48. https://doi.org/10.1109/ACCESS.2024.3384923.
[69] Hubbs CD, Li C, Sahinidis NV, Grossmann IE, Wassick JM. A deep reinforcement learning approach for chemical production scheduling. Comput Chem Eng 2020;141:106982. https://doi.org/10.1016/j.compchemeng.2020.106982.
[70] Hwangbo S, Liu JJ, Ryu J-H, Lee HJ, Na J. Production rescheduling via explorative reinforcement learning while considering nervousness. Comput Chem Eng 2024;186:108700. https://doi.org/10.1016/j.compchemeng.2024.108700.
[71] Jabeur MH, Mahjoub S, Toublanc C, Cariou V. A reinforcement learning approach for a lot sizing and production scheduling problem with energy consideration. IFAC-Pap 2023;56:11141–7. https://doi.org/10.1016/j.ifacol.2023.10.832.
[72] Jabeur MH, Mahjoub S, Toublanc C, Cariou V. Optimizing integrated lot sizing and production scheduling in flexible flow line systems with energy scheme: A two level approach based on reinforcement learning. Comput Ind Eng 2024;190:110095. https://doi.org/10.1016/j.cie.2024.110095.
[73] Johnson D, Chen G, Lu Y. Multi-Agent Reinforcement Learning for Real-Time Dynamic Production Scheduling in a Robot Assembly Cell. IEEE Robot Autom Lett 2022;7:7684–91. https://doi.org/10.1109/LRA.2022.3184795.
[74] Kardos C, Laflamme C, Gallina V, Sihn W. Dynamic scheduling in a job-shop production system with reinforcement learning. Procedia CIRP 2021;97:104–9. https://doi.org/10.1016/j.procir.2020.05.210.
[75] Kim T, Kim Y-W, Lee D, Kim M. Reinforcement learning approach to scheduling of precast concrete production. J Clean Prod 2022;336:130419. https://doi.org/10.1016/j.jclepro.2022.130419.
[76] Krenczyk D. Digital Twins of Production Systems Based on Discrete Simulation and Machine Learning Algorithms. In: García Bringas P, Pérez García H, Martínez de Pisón FJ, Martínez Álvarez F, Troncoso Lora A, Herrero Á, et al., editors. 18th Int Conf Soft Comput Models Ind Environ Appl SOCO 2023 [Internet]. Cham:

Springer Nature Switzerland; 2023. p. 57–66. https://doi.org/10.1007/978-3-031-42536-3_6. cited 2024 Aug 1.
[77] Kuhnle A, Röhrig N, Lanza G. Autonomous order dispatching in the semiconductor industry using reinforcement learning. Procedia CIRP 2019;79:391–6. https://doi.org/10.1016/j.procir.2019.02.101.
[78] Kuhnle A, Schäfer L, Stricker N, Lanza G. Design, Implementation and Evaluation of Reinforcement Learning for an Adaptive Order Dispatching in Job Shop Manufacturing Systems. Procedia CIRP 2019;81:234–9. https://doi.org/10.1016/j.procir.2019.03.041.
[79] Kuhnle A, Kaiser J-P, Theiß F, Stricker N, Lanza G. Designing an adaptive production control system using reinforcement learning. J Intell Manuf 2021;32:855–76. https://doi.org/10.1007/s10845-020-01612-y.
[80] Kuhnle A, May MC, Schäfer L, Lanza G. Explainable reinforcement learning in production control of job shop manufacturing system. Int J Prod Res 2022;60:5812–34. https://doi.org/10.1080/00207543.2021.1972179.
[81] Kulmer F, Wolf M, Ramsauer C. Planning Assistant for Medium-term Capacity Management using Deep Reinforcement Learning. IFAC-Pap 2024;58:31–6. https://doi.org/10.1016/j.ifacol.2024.09.082.
[82] Lang S, Behrendt F, Lanzerath N, Reggelin T, Muller M. Integration of Deep Reinforcement Learning and Discrete-Event Simulation for Real-Time Scheduling of a Flexible Job Shop Production. 2020 Winter Simul Conf WSC [Internet]. Orlando, FL, USA: IEEE; 2020. p. 3057–68. https://doi.org/10.1109/WSC48552.2020.9383997. cited 2024 Aug 1.
[83] Lee C, Lee S. A Practical Deep Reinforcement Learning Approach to Semiconductor Equipment Scheduling. 2021 22nd IEEE Int Conf Ind Technol ICIT [Internet]. Valencia, Spain. IEEE; 2021. p. 979–85. https://doi.org/10.1109/ICIT46573.2021.9453533. cited 2024 Aug 1.
[84] Lee YH, Lee S. Deep reinforcement learning based scheduling within production plan in semiconductor fabrication. Expert Syst Appl 2022;191:116222. https://doi.org/10.1016/j.eswa.2021.116222.
[85] Lee S, Kim J, Wi G, Won Y, Eun Y, Park K-J. Deep Reinforcement Learning-Driven Scheduling in Multijob Serial Lines: A Case Study in Automotive Parts Assembly. IEEE Trans Ind Inf 2024;20:2932–43. https://doi.org/10.1109/TII.2023.3292538.
[86] Lee C-Y, Huang Y-T, Chen P-J. Robust-optimization-guiding deep reinforcement learning for chemical material production scheduling. Comput Chem Eng 2024;187:108745. https://doi.org/10.1016/j.compchemeng.2024.108745.
[87] Lei Y, Deng Q, Liao M, Gao S. Deep reinforcement learning for dynamic distributed job shop scheduling problem with transfers. Expert Syst Appl 2024;251:123970. https://doi.org/10.1016/j.eswa.2024.123970.
[88] Lei K, Guo P, Wang Y, Zhang J, Meng X, Qian L. Large-Scale Dynamic Scheduling for Flexible Job-Shop With Random Arrivals of New Jobs by Hierarchical Reinforcement Learning. IEEE Trans Ind Inf 2024;20:1007–18. https://doi.org/10.1109/TII.2023.3272661.
[89] Leng J, Jin C, Vogl A, Liu H. Deep reinforcement learning for a color-batching resequencing problem. J Manuf Syst 2020;56:175–87. https://doi.org/10.1016/j.jmsy.2020.06.001.
[90] Leng J, Guo J, Zhang H, Xu K, Qiao Y, Zheng P, et al. Dual deep reinforcement learning agents-based integrated order acceptance and scheduling of mass individualized prototyping. J Clean Prod 2023;427:139249. https://doi.org/10.1016/j.jclepro.2023.139249.
[91] Leng J, Ruan G, Xu C, Zhou X, Xu K, Qiao Y, et al. Deep Reinforcement Learning of Graph Convolutional Neural Network for Resilient Production Control of Mass Individualized Prototyping Toward Industry 5.0. IEEE Trans Syst Man Cyber Syst 2024;54:7092–105. https://doi.org/10.1109/TSMC.2024.3446671.
[92] Li Y, Gu W, Yuan M, Tang Y. Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep Q network. Robot Comput-Integr Manuf 2022;74:102283. https://doi.org/10.1016/j.rcim.2021.102283.
[93] Liang P, Xiao P, Li Z, Luo M, Zhang C. A novel deep reinforcement learning-based algorithm for multi-objective energy-efficient flow-shop scheduling. IET Collab Intell Manuf 2024;6:e12121. https://doi.org/10.1049/cim2.12121.
[94] Liao Z, Li Q, Dai Y, Zhang Z. Learning to Schedule Job-Shop Problems via Hierarchical Reinforcement Learning. 2022 IEEE Int Conf Syst Man Cybern SMC [Internet]. Prague, Czech Republic. IEEE; 2022. p. 3222–7. https://doi.org/10.1109/SMC53654.2022.9945585. cited 2024 Aug 1.
[95] Liao Z, Chen J, Zhang Z. Solving Job-Shop Scheduling Problem via Deep Reinforcement Learning with Attention Model. In: Fujita H, Wang Y, Xiao Y, Moonis A, editors. Adv Trends Artif Intell Theory Appl [Internet. Cham: Springer Nature Switzerland; 2023. p. 201–12. https://doi.org/10.1007/978-3-031-36822-6_18. cited 2024 Aug 1].
[96] Lin C-C, Deng D-J, Chih Y-L, Chiu H-T. Smart manufacturing scheduling with edge computing Using multiclass deep Q network. IEEE Trans Ind Inf 2019;15:4276–84. https://doi.org/10.1109/TII.2019.2908210.
[97] Lin C-C, Peng Y-C, Chen Z-YA, Fan Y-H, Chin H-H. Distributed flexible job shop scheduling through deploying fog and edge computing in smart factories using dual deep Q networks. Mob Netw Appl 2024;29:886–904. https://doi.org/10.1007/s11036-024-02302-2.
[98] Lin C-C, Peng Y-C, Chang Y-S, Chang C-H. Reentrant hybrid flow shop scheduling with stockers in automated material handling systems using deep reinforcement learning. Comput Ind Eng 2024;189:109995. https://doi.org/10.1016/j.cie.2024.109995.
[99] Liu C-L, Chang C-C, Tseng C-J. Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. IEEE Access 2020;8:71752–62. https://doi.org/10.1109/ACCESS.2020.2987820.
[100] Liu R, Piplani R, Toro C. Deep reinforcement learning for dynamic scheduling of a flexible job shop. Int J Prod Res 2022;60:4049–69. https://doi.org/10.1080/00207543.2022.2058432.
[101] Liu J, Qiao F, Zou M, Zinn J, Ma Y, Vogel-Heuser B. Dynamic scheduling for semiconductor manufacturing systems with uncertainties using convolutional neural networks and reinforcement learning. Complex Intell Syst 2022;8:4641–62. https://doi.org/10.1007/s40747-022-00844-0.
[102] Liu R, Piplani R, Toro C. A deep multi-agent reinforcement learning approach to solve dynamic job shop scheduling problem. Comput Oper Res 2023;159:106294. https://doi.org/10.1016/j.cor.2023.106294.
[103] Liu C-L, Tseng C-J, Huang T-H, Wang J-W. Dynamic Parallel Machine Scheduling With Deep Q-Network. IEEE Trans Syst Man Cyber Syst 2023;53:6792–804. https://doi.org/10.1109/TSMC.2023.3289322.
[104] Liu Y, Fan J, Zhao L, Shen W, Zhang C. Integration of deep reinforcement learning and multi-agent system for dynamic scheduling of re-entrant hybrid flow shop considering worker fatigue and skill levels. Robot Comput-Integr Manuf 2023;84:102605. https://doi.org/10.1016/j.rcim.2023.102605.
[105] Liu L, Liang Z, Hong Y, Liu K, Wang H, Shang P. Multi-Constraint Flexible Job Scheduling Algorithm Based on DDQN: A Deep Reinforcement Learning Algorithm for Solving Practical Job Scheduling Problems. Proc 4th Int Conf Artif Intell Comput Eng [Internet]. Dalian China: ACM; 2023. p. 248–56. https://doi.org/10.1145/3652628.3652670 [cited 2024 Aug 1].
[106] Liu C-L, Tseng C-J, Weng P-H. Dynamic Job-Shop Scheduling via Graph Attention Networks and Deep Reinforcement Learning. IEEE Trans Ind Inf 2024;20:8662–72. https://doi.org/10.1109/TII.2024.3371489.
[107] Liu Z, Mao H, Sa G, Liu H, Tan J. Dynamic job-shop scheduling using graph reinforcement learning with auxiliary strategy. J Manuf Syst 2024;73:1–18. https://doi.org/10.1016/j.jmsy.2024.01.002.
[108] Loffredo A, May MC, Schäfer L, Matta A, Lanza G. Reinforcement learning for energy-efficient control of parallel and identical machines. CIRP J Manuf Sci Technol 2023;44:91–103. https://doi.org/10.1016/j.cirpj.2023.05.007.
[109] Loffredo A, May MC, Matta A, Lanza G. Reinforcement learning for sustainability enhancement of production lines. cited 2024 Aug 1]; J Intell Manuf [Internet] 2023. https://doi.org/10.1007/s10845-023-02258-2.
[110] Lu S, Wang Y, Kong M, Wang W, Tan W, Song Y. A Double Deep Q-Network framework for a flexible job shop scheduling problem with dynamic job arrivals and urgent job insertions. Eng Appl Artif Intell 2024;133:108487. https://doi.org/10.1016/j.engappai.2024.108487.
[111] Luo S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. Appl Soft Comput 2020;91:106208. https://doi.org/10.1016/j.asoc.2020.106208.
[112] Luo S, Zhang L, Fan Y. Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. Comput Ind Eng 2021;159:107489. https://doi.org/10.1016/j.cie.2021.107489.
[113] Ma Y, Cai J, Li S, Liu J, Xing J, Qiao F. Double deep Q-network-based self-adaptive scheduling approach for smart shop floor. Neural Comput Appl 2023;35:22281–96. https://doi.org/10.1007/s00521-023-08877-3.
[114] Magalhães R, Martins M, Vieira S, Santos F, Sousa J. Encoder-Decoder Neural Network Architecture for solving Job Shop Scheduling Problems using Reinforcement Learning. 2021 IEEE Symp Ser Comput Intell SSCI [Internet]. Orlando, FL, USA: IEEE; 2021. p. 01–8. https://doi.org/10.1109/SSCI50451.2021.9659849. cited 2024 Aug 1.
[115] Malathy V, Al-Jawahry HM, G K M, Suganya G. P R. A Reinforcement Learning Method in Cooperative Multi-Agent System for Production Control System. 2024 Int Conf Data Sci Netw Secur ICDSNS [Internet]. Tiptur, India: IEEE; 2024. p. 1–4. https://doi.org/10.1109/ICDSNS62112.2024.10691212. cited 2025 Oct 2.
[116] Malus A, Kozjek D, Vrabič R. Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. CIRP Ann 2020;69:397–400. https://doi.org/10.1016/j.cirp.2020.04.001.
[117] Marchesano MG, Guizzi G, Santillo LC, Vespoli S. A Deep Reinforcement Learning approach for the throughput control of a Flow-Shop production system. IFAC-Pap 2021;54:61–6. https://doi.org/10.1016/j.ifacol.2021.08.006.
[118] Marchesano MG, Guizzi G, Santillo LC, Vespoli S. Dynamic Scheduling in a Flow Shop Using Deep Reinforcement Learning. In: Dolgui A, Bernard A, Lemoine D, von Cieminski G, Romero D, editors. Adv Prod Manag Syst Artif Intell Sustain Resilient Prod Syst [Internet. Cham: Springer International Publishing; 2021. p. 152–60. https://doi.org/10.1007/978-3-030-85874-2_16. cited 2024 Aug 1].
[119] Marchesano MG, Guizzi G, Popolo V, Converso G. Dynamic scheduling of a due date constrained flow shop with Deep Reinforcement Learning. IFAC-Pap 2022;55:2932–7. https://doi.org/10.1016/j.ifacol.2022.10.177.
[120] May MC, Kiefer L, Kuhnle A, Stricker N, Lanza G. Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. Procedia CIRP 2021;96:3–8. https://doi.org/10.1016/j.procir.2021.01.043.
[121] Mayer S, Classen T, Endisch C. Modular production control using deep reinforcement learning: proximal policy optimization. J Intell Manuf 2021;32:2335–51. https://doi.org/10.1007/s10845-021-01778-z.
[122] Mueller-Zhang Z, Oliveira Antonino P, Kuhn T. Integrated Planning and Scheduling for Customized Production using Digital Twins and Reinforcement Learning. IFAC-Pap 2021;54:408–13. https://doi.org/10.1016/j.ifacol.2021.08.046.
[123] Müller A, Grumbach F, Kattenstroth F. Reinforcement Learning for Two-Stage Permutation Flow Shop Scheduling—A Real-World Application in Household Appliance Production. IEEE Access 2024;12:11388–99. https://doi.org/10.1109/ACCESS.2024.3355269.

[124] Nam S, Cho Y, Woo JH. Simulation-based deep reinforcement learning for multi-objective identical parallel machine scheduling problem. Int J Nav Arch Ocean Eng 2024;16:100629. https://doi.org/10.1016/j.ijnaoe.2024.100629.
[125] Nasuta A, Kemmerling M, Lütticke D, Schmitt RH. Reward Shaping for Job Shop Scheduling. In: Nicosia G, Ojha V, La Malfa E, La Malfa G, Pardalos PM, Umeton R, editors. Mach Learn Optim Data Sci [Internet. Cham: Springer Nature Switzerland; 2024. p. 197–211. https://doi.org/10.1007/978-3-031-53969-5_16. cited 2024 Aug 1].
[126] Overbeck L, Hugues A, May MC, Kuhnle A, Lanza G. Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. Procedia CIRP 2021;103:170–5. https://doi.org/10.1016/j.procir.2021.10.027.
[127] Overbeck L, Glaser V, May MC, Lanza G. Generalization of Reinforcement Learning Agents for Production Control. In: Galizia FG, Bortolini M, editors. Prod Process Prod Evol Age Disrupt [Internet. Cham: Springer International Publishing; 2023. p. 338–46. https://doi.org/10.1007/978-3-031-34821-1_37. cited 2024 Aug 1].
[128] Paeng B, Park I-B, Park J. Deep Reinforcement Learning for Minimizing Tardiness in Parallel Machine Scheduling With Sequence Dependent Family Setups. IEEE Access 2021;9:101390–401. https://doi.org/10.1109/ACCESS.2021.3097254.
[129] Panzer M, Bender B, Gronau N. A deep reinforcement learning based hyper-heuristic for modular production control. Int J Prod Res 2023:1–22. https://doi.org/10.1080/00207543.2023.2233641.
[130] Panzer M, Gronau N. Designing an adaptive and deep learning based control framework for modular production systems. cited 2024 Aug 1]; J Intell Manuf [Internet] 2023. https://doi.org/10.1007/s10845-023-02249-3.
[131] Panzer M, Gronau N. Enhancing economic efficiency in modular production systems through deep reinforcement learning. Procedia CIRP 2024;121:55–60. https://doi.org/10.1016/j.procir.2023.09.229.
[132] Park I-B, Huh J, Kim J, Park J. A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities. IEEE Trans Autom Sci Eng 2020:1–12. https://doi.org/10.1109/TASE.2019.2956762.
[133] Park KT, Son YH, Ko SW, Noh SD. Digital Twin and Reinforcement Learning-Based Resilient Production Control for Micro Smart Factory. Appl Sci 2021;11:2977. https://doi.org/10.3390/app11072977.
[134] Park KT, Jeon S-W, Noh SD. Digital twin application with horizontal coordination for reinforcement-learning-based production control in a re-entrant job shop. Int J Prod Res 2022;60:2151–67. https://doi.org/10.1080/00207543.2021.1884309.
[135] Park I-B, Park J. Scalable Scheduling of Semiconductor Packaging Facilities Using Deep Reinforcement Learning. IEEE Trans Cyber 2023;53:3518–31. https://doi.org/10.1109/TCYB.2021.3128075.
[136] Paschko F, Knorn S, Krini A, Kemke M. Material flow control in Remanufacturing Systems with random failures and variable processing times. J Remanufacturing 2023;13:161–85. https://doi.org/10.1007/s13243-023-00126-z.
[137] Peng S, Xiong G, Yang J, Shen Z, Tamir TS, Tao Z, et al. Multi-Agent Reinforcement Learning for Extended Flexible Job Shop Scheduling. Machines 2024;12:8. https://doi.org/10.3390/machines12010008.
[138] Pol S, Baer S, Turner D, Samsonov V, Meisen T. Global Reward Design for Cooperative Agents to Achieve Flexible Production Control under Real-time Constraints: Proc 23rd Int Conf Enterp Inf Syst [Internet]. Online Streaming, — Select a Country —: SCITEPRESS - Science and Technology Publications; 2021. p. 515–26. https://doi.org/10.5220/0010455805150526. cited 2024 Aug 1].
[139] Popper J, Motsch W, David A, Petzsche T, Ruskowski M. Utilizing Multi-Agent Deep Reinforcement Learning For Flexible Job Shop Scheduling Under Sustainable Viewpoints. 2021 Int Conf Electr Comput Commun Mechatron Eng ICECCME [Internet]. Mauritius, Mauritius: IEEE; 2021. p. 1–6. https://doi.org/10.1109/ICECCME52200.2021.9590925. cited 2024 Aug 1.
[140] Popper J, Ruskowski M. Using Multi-Agent Deep Reinforcement Learning For Flexible Job Shop Scheduling Problems. Procedia CIRP 2022;112:63–7. https://doi.org/10.1016/j.procir.2022.09.039.
[141] Pu Y, Li F, Rahimifard S. Multi-Agent Reinforcement Learning for Job Shop Scheduling in Dynamic Environments. Sustainability 2024;16:3234. https://doi.org/10.3390/su16083234.
[142] Qin Z, Johnson D, Lu Y. Dynamic production scheduling towards self-organizing mass personalization: A multi-agent dueling deep reinforcement learning approach. J Manuf Syst 2023;68:242–57. https://doi.org/10.1016/j.jmsy.2023.03.003.
[143] Qin Z, Lu Y. Knowledge graph-enhanced multi-agent reinforcement learning for adaptive scheduling in smart manufacturing. cited 2025 Oct 2]; J Intell Manuf [Internet] 2024. https://doi.org/10.1007/s10845-024-02494-0.
[144] Qiu J, Liu J, Li Z, Lai X. A multi-level action coupling reinforcement learning approach for online two-stage flexible assembly flow shop scheduling. J Manuf Syst 2024;76:351–70. https://doi.org/10.1016/j.jmsy.2024.08.006.
[145] Ragazzini L, Negri E, Macchi M. A Digital Twin-based Predictive Strategy for Workload Control. IFAC-Pap 2021;54:743–8. https://doi.org/10.1016/j.ifacol.2021.08.183.
[146] Rangel-Martinez D, Ricardez-Sandoval LA. A recurrent reinforcement learning strategy for optimal scheduling of partially observable job-shop and flow-shop batch chemical plants under uncertainty. Comput Chem Eng 2024;188:108748. https://doi.org/10.1016/j.compchemeng.2024.108748.
[147] Roesch M, Linder C, Bruckdorfer C, Hohmann A, Reinhart G. Industrial Load Management using Multi-Agent Reinforcement Learning for Rescheduling. 2019 s Int Conf Artif Intell Ind AI4I [Internet]. Laguna Hills, CA, USA. IEEE; 2019. p. 99–102. https://doi.org/10.1109/AI4I46381.2019.00033. cited 2024 Aug 1.
[148] Rui Z, Zhang X, Liu M, Ling L, Wang X, Liu C, et al. Graph reinforcement learning for flexible job shop scheduling under industrial demand response: A production and energy nexus perspective. Comput Ind Eng 2024;193:110325. https://doi.org/10.1016/j.cie.2024.110325.
[149] Rummukainen H, Nurminen JK. Practical Reinforcement Learning -Experiences in Lot Scheduling Application. IFAC-Pap 2019;52:1415–20. https://doi.org/10.1016/j.ifacol.2019.11.397.
[150] Sakr AH, Aboelhassan A, Yacout S, Bassetto S. Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems. J Intell Manuf 2023;34:1311–24. https://doi.org/10.1007/s10845-021-01851-7.
[151] Samsonov V, Kemmerling M, Paegert M, Lütticke D, Sauermann F, Gützlaff A, et al. Manufacturing Control in Job Shop Environments with Reinforcement Learning: Proc 13th Int Conf Agents Artif Intell [Internet]. Online Streaming, — Select a Country: SCITEPRESS - Science and Technology Publications; 2021. p. 589–97. https://doi.org/10.5220/0010202405890597. cited 2024 Aug 1].
[152] Saqlain M, Ali S, Lee JY. A Monte-Carlo tree search algorithm for the flexible job-shop scheduling in manufacturing systems. Flex Serv Manuf J 2023;35:548–71. https://doi.org/10.1007/s10696-021-09437-4.
[153] Schneckenreither M, Haeussler S. Reinforcement Learning Methods for Operations Research Applications: The Order Release Problem. In: Nicosia G, Pardalos P, Giuffrida G, Umeton R, Sciacca V, editors. Mach Learn Optim Data Sci [Internet. Cham: Springer International Publishing; 2019. p. 545–59. https://doi.org/10.1007/978-3-030-13709-0_46. cited 2024 Aug 1].
[154] Schneckenreither M, Windmueller S, Haeussler S. Smart Short Term Capacity Planning: A Reinforcement Learning Approach. In: Dolgui A, Bernard A, Lemoine D, von Cieminski G, Romero D, editors. Adv Prod Manag Syst Artif Intell Sustain Resilient Prod Syst [Internet. Cham: Springer International Publishing; 2021. p. 258–66. https://doi.org/10.1007/978-3-030-85874-2_27. cited 2024 Aug 1].
[155] Schneckenreither M, Haeussler S, Peiró J. Average reward adjusted deep reinforcement learning for order release planning in manufacturing. Knowl-Based Syst 2022;247:108765. https://doi.org/10.1016/j.knosys.2022.108765.
[156] Schuh G, Schmitz S, Maetschke J, Janke T, Eisbein H. Application of a Reinforcement Learning-based Automated Order Release in Production. Hannover: publish-Ing.; 2023. https://doi.org/10.15488/13500.
[157] Serrano-Ruiz JC, Mula J, Poler R. Job shop smart manufacturing scheduling by deep reinforcement learning. J Ind Inf Integr 2024;38:100582. https://doi.org/10.1016/j.jii.2024.100582.
[158] Siatras V, Bakopoulos E, Mavrothalassitis P, Nikolakis N, Alexopoulos K. Production Scheduling Based on a Multi-Agent System and Digital Twin: A Bicycle Industry Case. Information 2024;15:337. https://doi.org/10.3390/info15060337.
[159] Silva T, Azevedo A. Production flow control through the use of reinforcement learning. Procedia Manuf 2019;38:194–202. https://doi.org/10.1016/j.promfg.2020.01.026.
[160] Silva T, Azevedo A. Self-adapting WIP parameter setting using deep reinforcement learning. Comput Oper Res 2022;144:105854. https://doi.org/10.1016/j.cor.2022.105854.
[161] Song L, Li Y, Xu J. Dynamic Job-Shop Scheduling Based on Transformer and Deep Reinforcement Learning. Processes 2023;11:3434. https://doi.org/10.3390/pr11123434.
[162] Song W, Chen X, Li Q, Cao Z. Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. IEEE Trans Ind Inf 2023;19:1600–10. https://doi.org/10.1109/TII.2022.3189725.
[163] Stricker N, Kuhnle A, Sturm R, Friess S. Reinforcement learning for adaptive order dispatching in the semiconductor industry. CIRP Ann 2018;67:511–4. https://doi.org/10.1016/j.cirp.2018.04.041.
[164] Sun L, Shi W, Xuan C, Zhang Y. Research on Scheduling Algorithm of Knitting Production Workshop Based on Deep Reinforcement Learning. Machines 2024;12:579. https://doi.org/10.3390/machines12080579.
[165] Taghipour S, Namoura HA, Sharifi M, Ghaleb M. Real-time production scheduling using a deep reinforcement learning-based multi-agent approach. INFOR Inf Syst Oper Res 2024;62:186–210. https://doi.org/10.1080/03155986.2023.2287996.
[166] Tang J, Salonitis K. A Deep Reinforcement Learning Based Scheduling Policy for Reconfigurable Manufacturing Systems. Procedia CIRP 2021;103:1–7. https://doi.org/10.1016/j.procir.2021.09.089.
[167] Tang Y, Shen L, Han S. Low-Carbon Flexible Job Shop Scheduling Problem Based on Deep Reinforcement Learning. Sustainability 2024;16:4544. https://doi.org/10.3390/su16114544.
[168] Theumer P, Edenhofner F, Zimmermann R, Zipfel A. Explain Deep Reinf Learn Prod Control 2022.
[169] Turgut Y, Bozdag CE. Deep Q-Network Model for Dynamic Job Shop Scheduling Problem Based on Discrete Event Simulation. 2020 Winter Simul Conf WSC [Internet]. Orlando, FL, USA: IEEE; 2020. p. 1551–9. https://doi.org/10.1109/WSC48552.2020.9383986. cited 2024 Aug 1.
[170] Van Ekeris T, Meyes R, Meisen T. Discovering Heuristics And Metaheuristics For Job Shop Scheduling From Scratch Via Deep Reinforcement Learning. Hannover: publish-Ing; 2021. https://doi.org/10.15488/11231.
[171] van Hezewijk L, Dellaert N, Van Woensel T, Gademann N. Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem. Int J Prod Res 2023;61:1955–78. https://doi.org/10.1080/00207543.2022.2056540.
[172] Vijayan S, Parameshwaran Pillai T. Application of a Machine Learning Algorithm in a Multi Stage Production System. Trans FAMENA 2022;46:91–102. https://doi.org/10.21278/TOF.461033121.
[173] Voss T, Bode C, Heger J. Dynamic Lot Size Optimization with Reinforcement Learning. In: Freitag M, Kinra A, Kotzab H, Megow N, editors. Dyn Logist [Internet. Cham: Springer International Publishing; 2022. p. 376–85. https://doi.org/10.1007/978-3-031-05359-7_30. cited 2024 Aug 1].

[174] Wan L, Cui X, Zhao H, Fu L, Li C. A novel method for solving dynamic flexible job-shop scheduling problem via DIFFormer and deep reinforcement learning. Comput Ind Eng 2024;198:110688. https://doi.org/10.1016/j.cie.2024.110688.
[175] Wan L, Fu L, Li C, Li K. Flexible job shop scheduling via deep reinforcement learning with meta-path-based heterogeneous graph neural network. Knowl-Based Syst 2024;296:111940. https://doi.org/10.1016/j.knosys.2024.111940.
[176] Wang L, Hu X, Wang Y, Xu S, Ma S, Yang K, et al. Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. Comput Netw 2021;190:107969. https://doi.org/10.1016/j.comnet.2021.107969.
[177] Wang H, Yan Q, Zhang S. Integrated scheduling and flexible maintenance in deteriorating multi-state single machine system using a reinforcement learning approach. Adv Eng Inf 2021;49:101339. https://doi.org/10.1016/j.aei.2021.101339.
[178] Wang M, Zhang J, Zhang P, Cui L, Zhang G. Independent double DQN-based multi-agent reinforcement learning approach for online two-stage hybrid flow shop scheduling with batch machines. J Manuf Syst 2022;65:694–708. https://doi.org/10.1016/j.jmsy.2022.11.001.
[179] Wang H, Cheng J, Liu C, Zhang Y, Hu S, Chen L. Multi-objective reinforcement learning framework for dynamic flexible job shop scheduling problem with uncertain events. Appl Soft Comput 2022;131:109717. https://doi.org/10.1016/j.asoc.2022.109717.
[180] Wang H, Yan Q, Wang J. Blockchain-secured multi-factory production with collaborative maintenance using Q-learning-based optimisation approach. Int J Prod Res 2023;61:3685–702. https://doi.org/10.1080/00207543.2021.2002968.
[181] Wang Z, Liao W. Smart scheduling of dynamic job shop based on discrete event simulation and deep reinforcement learning. cited 2023 Nov 30]; J Intell Manuf [Internet] 2023. https://doi.org/10.1007/s10845-023-02161-w.
[182] Wang Y, Wang R, Sun J, Wang G. Unified DRL for Enhanced Flexible Job-Shop Scheduling with Transportation Constraints. 2024 China Autom Congr CAC [Internet]. Qingdao, China: IEEE; 2024. p. 877–82. https://doi.org/10.1109/CAC63892.2024.10864691. cited 2025 Oct 2.
[183] Waschneck B, Reichstaller A, Belzner L, Altenmüller T, Bauernhansl T, Knapp A, et al. Deep reinforcement learning for semiconductor production scheduling. 2018 29th Annu SEMI Adv Semicond Manuf Conf ASMC [Internet]. Saratoga Springs, NY, USA: IEEE; 2018. p. 301–6. https://doi.org/10.1109/ASMC.2018.8373191. cited 2024 Aug 1.
[184] Waschneck B, Reichstaller A, Belzner L, Altenmüller T, Bauernhansl T, Knapp A, et al. Optimization of global production scheduling with deep reinforcement learning. Procedia CIRP 2018;72:1264–9. https://doi.org/10.1016/j.procir.2018.03.212.
[185] Wesendrup K, Hellingrath B. ML2-enabled Condition-based Demand, Production, Inventory, and Maintenance Planning. IFAC-Pap 2023;56:6600–5. https://doi.org/10.1016/j.ifacol.2023.10.358.
[186] Wong VWH, Kim SH, Park J, Park J, Law KH. Generating Dispatching Rules for the Interrupting Swap-Allowed Blocking Job Shop Problem Using Graph Neural Network and Reinforcement Learning. J Manuf Sci Eng 2024;146:011009. https://doi.org/10.1115/1.4063652.
[187] Woo JH, Kim B, Ju S, Cho YI. Automation of load balancing for Gantt planning using reinforcement learning. Eng Appl Artif Intell 2021;101:104226. https://doi.org/10.1016/j.engappai.2021.104226.
[188] Wu X, Yan X. A spatial pyramid pooling-based deep reinforcement learning model for dynamic job-shop scheduling problem. Comput Oper Res 2023;160:106401. https://doi.org/10.1016/j.cor.2023.106401.
[189] Wu X, Yan X, Guan D, Wei M. A deep reinforcement learning model for dynamic job-shop scheduling problem with uncertain processing time. Eng Appl Artif Intell 2024;131:107790. https://doi.org/10.1016/j.engappai.2023.107790.
[190] Wurster M, Michel M, May MC, Kuhnle A, Stricker N, Lanza G. Modelling and condition-based control of a flexible and hybrid disassembly system with manual and autonomous workstations using reinforcement learning. J Intell Manuf 2022;33:575–91. https://doi.org/10.1007/s10845-021-01863-3.
[191] Xanthopoulos AS, Chnitidis G, Koulouriotis DE. Reinforcement learning-based adaptive production control of pull manufacturing systems. J Ind Prod Eng 2019;36:313–23. https://doi.org/10.1080/21681015.2019.1647301.
[192] Xia M, Liu H, Li M, Wang L. A dynamic scheduling method with Conv-Dueling and generalized representation based on reinforcement learning. Int J Ind Eng Comput 2023;14:805–20. https://doi.org/10.5267/j.ijiec.2023.6.003.
[193] Xia B, Li Y, Gu J, Peng Y. Research on Sustainable Scheduling of Material-Handling Systems in Mixed-Model Assembly Workshops Based on Deep Reinforcement Learning. Sustainability 2024;16:10025. https://doi.org/10.3390/su162210025.
[194] Xu S, Li Y, Li Q. A Deep Reinforcement Learning Method Based on a Transformer Model for the Flexible Job Shop Scheduling Problem. Electronics 2024;13:3696. https://doi.org/10.3390/electronics13183696.
[195] Xu K, Ye C, Gong H, Sun W. Reinforcement Learning-Based Multi-Objective of Two-Stage Blocking Hybrid Flow Shop Scheduling Problem. Processes 2024;12:51. https://doi.org/10.3390/pr12010051.
[196] Yang H, Li W, Wang B. Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning. Reliab Eng Syst Saf 2021;214:107713. https://doi.org/10.1016/j.ress.2021.107713.
[197] Yang Y, Qian B, Hu R, Zhang D. Deep Reinforcement Learning Algorithm for Permutation Flow Shop Scheduling Problem. In: Huang D-S, Jo K-H, Jing J, Premaratne P, Bevilacqua V, Hussain A, editors. Intell Comput Methodol [Internet. Cham: Springer International Publishing; 2022. p. 473–83. https://doi.org/10.1007/978-3-031-13832-4_39. cited 2024 Aug 1].
[198] Yang SL, Wang JY, Xin LM, Xu ZG. Verification of intelligent scheduling based on deep reinforcement learning for distributed workshops via discrete event simulation. Adv Prod Eng Manag 2022;17:401–12. https://doi.org/10.14743/apem2022.4.444.
[199] Yang Z, Bi L, Jiao X. Combining Reinforcement Learning Algorithms with Graph Neural Networks to Solve Dynamic Job Shop Scheduling Problems. Processes 2023;11:1571. https://doi.org/10.3390/pr11051571.
[200] Yuan E, Wang L, Cheng S, Song S, Fan W, Li Y. Solving flexible job shop scheduling problems via deep reinforcement learning. Expert Syst Appl 2024;245:123019. https://doi.org/10.1016/j.eswa.2023.123019.
[201] Yue L, Peng K, Ding L, Mumtaz J, Lin L, Zou T. Two-stage double deep Q-network algorithm considering external non-dominant set for multi-objective dynamic flexible job shop scheduling problems. Swarm Evol Comput 2024;90:101660. https://doi.org/10.1016/j.swevo.2024.101660.
[202] Yun L, Wang D, Li L. Explainable multi-agent deep reinforcement learning for real-time demand response towards sustainable manufacturing. Appl Energy 2023;347:121324. https://doi.org/10.1016/j.apenergy.2023.121324.
[203] Zhang C, Song W, Cao Z, Zhang J, Tan PS, Xu C. Learn Dispatch Job Shop Sched via Deep Reinf Learn 2020.
[204] Zhang L, Yang C, Yan Y, Hu Y. Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning. IEEE Trans Ind Inf 2022;18:8999–9007. https://doi.org/10.1109/TII.2022.3178410.
[205] Zhang Y, Zhu H, Tang D, Zhou T, Gui Y. Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems. Robot Comput-Integr Manuf 2022;78:102412. https://doi.org/10.1016/j.rcim.2022.102412.
[206] Zhang J-D, He Z, Chan W-H, Chow C-Y. DeepMAG: Deep reinforcement learning with multi-agent graphs for flexible job shop scheduling. Knowl-Based Syst 2023;259:110083. https://doi.org/10.1016/j.knosys.2022.110083.
[207] Zhang M, Wang L, Qiu F, Liu X. Dynamic scheduling for flexible job shop with insufficient transportation resources via graph neural network and deep reinforcement learning. Comput Ind Eng 2023;186:109718. https://doi.org/10.1016/j.cie.2023.109718.
[208] Zhang W, Zhao F, Li Y, Du C, Feng X, Mei X. A novel collaborative agent reinforcement learning framework based on an attention mechanism and disjunctive graph embedding for flexible job shop scheduling problem. J Manuf Syst 2024;74:329–45. https://doi.org/10.1016/j.jmsy.2024.03.012.
[209] Zhang W, Zhao F, Yang C, Du C, Feng X, Zhang Y, et al. A novel Soft Actor–Critic framework with disjunctive graph embedding and autoencoder mechanism for Job Shop Scheduling Problems. J Manuf Syst 2024;76:614–26. https://doi.org/10.1016/j.jmsy.2024.08.015.
[210] Zhang J, Guo B, Ding X, Hu D, Tang J, Du K, et al. An adaptive multi-objective multi-task scheduling method by hierarchical deep reinforcement learning. Appl Soft Comput 2024;154:111342. https://doi.org/10.1016/j.asoc.2024.111342.
[211] Zhang L, Yang C, Yan Y, Cai Z, Hu Y. Automated guided vehicle dispatching and routing integration via digital twin with deep reinforcement learning. J Manuf Syst 2024;72:492–503. https://doi.org/10.1016/j.jmsy.2023.12.008.
[212] Zhang L, Yan Y, Hu Y. Dynamic flexible scheduling with transportation constraints by multi-agent reinforcement learning. Eng Appl Artif Intell 2024;134:108699. https://doi.org/10.1016/j.engappai.2024.108699.
[213] Zhao Y, Wang Y, Tan Y, Zhang J, Yu H. Dynamic Jobshop Scheduling Algorithm Based on Deep Q Network. IEEE Access 2021;9:122995–3011. https://doi.org/10.1109/ACCESS.2021.3110242.
[214] Zhao F, Liu Y, Xu T, Jonrinaldi. A reinforcement learning hyper-heuristic algorithm for the distributed flowshops scheduling problem under consideration of emergency order insertion. Appl Soft Comput 2024;167:112461. https://doi.org/10.1016/j.asoc.2024.112461.
[215] Zhao C, Deng N. An actor-critic framework based on deep reinforcement learning for addressing flexible job shop scheduling problems. Math Biosci Eng 2024;21:1445–71. https://doi.org/10.3934/mbe.2024062.
[216] Zhao Y, Ma S, Mo X, Xu X. Data-driven optimization for energy-constrained dietary supplement scheduling: A bounded cut MP-DQN approach. Comput Ind Eng 2024;188:109894. https://doi.org/10.1016/j.cie.2024.109894.
[217] Zheng X, Chen Z. An improved deep Q-learning algorithm for a trade-off between energy consumption and productivity in batch scheduling. Comput Ind Eng 2024;188:109925. https://doi.org/10.1016/j.cie.2024.109925.
[218] Zhou T, Tang D, Zhu H, Wang L. Reinforcement learning with composite rewards for production scheduling in a smart factory. IEEE Access 2020;9:752–66. https://doi.org/10.1109/ACCESS.2020.3046784.
[219] Zhu H., Zhang Y., Liu C., Shi W. An Adaptive Reinforcement Learning-Based Scheduling Approach with Combination Rules for Mixed-Line Job Shop Production. Abiyev R., editor. Math Probl Eng. 2022;2022:1–14. https://doi.org/10.1155/2022/1672166.
[220] Zhu X, Xu J, Ge J, Wang Y, Xie Z. Multi-task multi-agent reinforcement learning for real-time scheduling of a dual-resource flexible job shop with robots. Processes 2023;11:267. https://doi.org/10.3390/pr11010267.
[221] Zisgen H, Miltenberger R, Hochhaus M, Stöhr N. Dynamic scheduling of gantry robots using simulation and reinforcement learning. 2023 Winter Simul Conf WSC [Internet]. San Antonio, TX, USA: IEEE; 2023. p. 3026–34. https://doi.org/10.1109/WSC60868.2023.10407159. cited 2024 Aug 1.
[222] Nozarijouybari Z, Fathy HK. Machine learning for battery systems applications: Progress, challenges, and opportunities. J Power Sources 2024;601:234272. https://doi.org/10.1016/j.jpowsour.2024.234272.