

Review

Not peer-reviewed version

Industrial Scheduling in the Digital Era: Challenges, State-of-the-Art Methods, and Deep Learning Perspectives

[Alina Itu](#) *

Posted Date: 26 August 2025

doi: 10.20944/preprints202508.1120.v2

Keywords: industrial scheduling; deep reinforcement learning; digital twin; Industry 4.0; robust and adaptive optimization



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Review

Industrial Scheduling in the Digital Era: Challenges, State-of-the-Art Methods, and Deep Learning Perspectives

Alina Itu

Automation and Information Technology, Transilvania University of Brasov, Mihai Viteazu nr. 5, 5000174 Brasov, Romania; alina.itu@unitbv.ro

Abstract

Industrial scheduling remains a pivotal discipline for efficiency, resilience, and competitiveness in manufacturing and service operations. The digital transformation, driven by paradigms such as Industry 4.0, has fundamentally reshaped the scheduling landscape, introducing pervasive connectivity, real-time data flows, and cyber-physical integration. This review synthesizes advances across three enduring challenges: **(i)** scalability and computational complexity in large-scale, high-dimensional scheduling environments, **(ii)** robustness and adaptability to uncertainty and disruptions, and **(iii)** integration with digitalization through IIoT, digital twins, cloud-edge architectures, and interoperable, secure infrastructures. We highlight the surge of **AI-enhanced and deep learning-driven methods** that are redefining state-of-the-art practice. Deep reinforcement learning (DRL) now underpins policy learning for dynamic dispatching and rescheduling; graph neural networks (GNNs) and attention models enable generalization across diverse shop configurations; and digital twin-in-the-loop frameworks provide safe training and rapid adaptation under real-world volatility. At the same time, neural architectures are increasingly embedded within decomposition, metaheuristics, and multi-agent systems—forming hybrid stacks that combine the guarantees of operations research with the adaptability of learning-based approaches. The industrial impact of these developments is evident across semiconductor fabrication, flexible job shops, supply-chain-intensive production, and distributed, autonomous networks. Yet critical challenges persist in interpretability, trust, data quality, and legacy integration. To advance, future research must prioritize **explainable and certifiable neural schedulers, standardized datasets and benchmarks, seamless AI-IoT-DT integration, federated and privacy-preserving collaboration, and human-in-the-loop frameworks**. Collectively, these directions chart a path toward scheduling systems that are not only more efficient and scalable, but also transparent, secure, and resilient in the digital, interconnected era.

Keywords: industrial scheduling; deep reinforcement learning; digital twin; Industry 4.0; robust and adaptive optimization

1. Introduction

Industrial scheduling encompasses the decision-making processes that allocate limited resources—machines, workforce, tools, and materials—to jobs over time with one or more performance objectives, such as minimizing makespan, reducing cost, increasing throughput, meeting due dates, and balancing workloads or inventories. Foundational treatments formalize these models and objectives across single-machine, flow/parallel/job shops, and flexible shop environments, and review setup-related complications common in practice. (Pinedo, 2016; Allahverdi et al., 2008; Gupta & Stafford, 2006; Blazewicz et al., 2007).

The field is renowned for both its practical significance and computational difficulty: many canonical variants are NP-hard, ruling out polynomial-time exact algorithms for large instances and

motivating approximation, heuristic, and metaheuristic approaches. Classic complexity results in deterministic sequencing and scheduling, alongside broader NP-completeness theory, underpin this assessment and continue to guide algorithm design. (Graham et al., 1979; Garey & Johnson, 1979).

Against this backdrop, recent progress increasingly blends operations research with data-driven and learning-based methods—ranging from learned components inside exact/heuristic solvers to end-to-end deep reinforcement learning (DRL) policies that learn dispatching rules from experience and generalize across scales. (Bengio et al., 2021; Zhang et al., 2020).

Scheduling decisions reverberate throughout operations. Effective schedules shape lead times, inventory positions, utilization, energy consumption, and service levels across manufacturing and services; in many settings, they also form the last-mile link between planning and control. (Herrmann, 2006; Gahm et al., 2016).

The digital transformation (Industry 4.0) has fundamentally altered the information context of scheduling. Industrial IoT, cyber-physical production systems, cloud/edge computing, and digital twins generate continuous data streams and enable tighter sense–decide–act loops—but they also introduce interoperability and latency constraints that shape feasible scheduling architectures. These changes simultaneously heighten problem complexity and unlock data-driven, closed-loop scheduling integrated with shop-floor automation. (Wang et al., 2016; Monostori, 2016; Zhong et al., 2017; Ivanov & Dolgui, 2020a).

Beyond productivity and cost, scheduling now contributes to strategic goals in sustainability and human-centric operations. Energy-aware models and policies can reduce consumption and emissions, while human-in-the-loop, “Operator 4.0” concepts target ergonomics and well-being in digitally enhanced workplaces. (Fang et al., 2011; Romero et al., 2019/2020).

At the same time, volatile supply and demand conditions, equipment disruptions, and external shocks expose the limits of static plans, elevating robustness and rapid recovery as first-class requirements. Digital-twin-enabled monitoring and OR methods for ripple-effect management illustrate how predictive and reactive scheduling can be fused for resilience. (Ivanov & Dolgui, 2021; Ivanov & Dolgui, 2020a).

This review synthesizes developments across these fronts and focuses on three persistent, intertwined challenges that define industrial scheduling in the digital era:

- Scalability and computational complexity in large, high-dimensional environments;
- Robustness and adaptability to uncertainty and real-world disruptions;
- Integration with digitalization (IIoT, cloud/edge platforms, and cyber-physical systems).

The first major challenge is **the scalability and computational complexity** of scheduling large and dynamic systems. As industrial operations expand—covering dense machine networks, diversified product portfolios, and multi-echelon supply chains—the combinatorial search space grows superlinearly, making many variants NP-hard and rendering exact methods impractical beyond modest sizes (even with clever modeling and cutting planes). This reality motivates scalable heuristics, metaheuristics, and hybrid AI–OR approaches that deliver strong anytime solutions under tight latency budgets (Graham et al., 1979; Pinedo, 2016). A recent thrust is learning to optimize: machine-learned components accelerate solvers themselves (e.g., learned branching and node policies for MILP) or synthesize high-quality heuristics end-to-end. Graph-based models guide branch-and-bound more effectively than hand-crafted rules (Khalil et al., 2016; Gasse et al., 2019), while neural combinatorial optimization—pointer networks and attention models—yields powerful constructive policies that increasingly transfer to shop-floor dispatching (Vinyals et al., 2015; Kool et al., 2018; Zhang et al., 2020). These techniques do not remove worst-case hardness, but they can shift the practical frontier, offering better solution quality under fixed compute and generalization to larger instances (Bengio et al., 2021).

The second central challenge is **robustness and adaptability to uncertainty**. Industrial systems face equipment failures, stochastic processing times and arrivals, rush orders, material shortages, and upstream disruptions; static schedules—even optimal at release—degrade quickly on the shop floor. Classical robust optimization provides tunable protection against uncertainty sets (Bertsimas & Sim,

2004), while the predictive–reactive literature offers periodic and event-driven rescheduling strategies to recover performance (Vieira et al., 2003; Ouelhadj & Petrovic, 2009). On the data-driven side, deep reinforcement learning (DRL) learns reactive dispatching and repair policies that adapt online; systematic evidence shows improvements in tardiness, throughput, and resilience across manufacturing test beds (Panzer & Bender, 2022; Zhang et al., 2024). Effective stacks increasingly fuse forecasts, robust baselines, and DRL policies with feasibility guards, giving rapid recovery while containing variance in outcomes.

The third, increasingly critical challenge is **integration with digitalization and Industry 4.0**. Cyber-physical production systems and the Industrial IoT generate continuous, heterogeneous streams (status, quality, energy, and context data) that can close the loop between sensing, scheduling, and actuation—yet they also impose strict interoperability and latency constraints that shape feasible architectures (edge vs. cloud; publish/subscribe vs. polling). Digital twins serve as simulation-in-the-loop substrates for scheduling: they enable safe policy training, what-if analysis, and proactive, state-aware rescheduling once deployed (Monostori, 2016; Xu et al., 2018b; Tao & Zhang, 2017; Zhang et al., 2022). Realizing this promise requires data pipelines that manage drift and noise, hardened APIs for shop-floor integration, and algorithms that meet real-time deadlines with verifiable constraint satisfaction.

Recent AI advances further reshape the design space. Beyond GNN-augmented solvers and DRL dispatching, large language models can act as optimizers by prompting (OPRO)—iteratively proposing and refining heuristics under programmatic evaluation. When coupled with simulation-based validation and action masking, such models can rapidly tailor heuristic rules or hyper-policies to new product mixes and resource pools, complementing DRL and classical OR rather than replacing them (Yang et al., 2024; Bengio et al., 2021).

These three challenges—(i) scalability and computational complexity, (ii) robustness and adaptability, and (iii) digital integration—thus frame our review of methods, evidence of industrial impact, and opportunities for future research.

2. Scalability and Computational Complexity

2.1. The Combinatorial Nature of Industrial Scheduling

Industrial scheduling problems are intrinsically combinatorial: the number of feasible schedules typically grows factorially (or worse) with jobs, machines, precedence/eligibility constraints, and setup interactions. For classical models—job shop and flow shop—this explosion is well documented, and most realistic variants are NP-hard once we account for precedence, batching, sequence-dependent setups, machine flexibility, release/due dates, or multi-objective criteria. The consequence is a persistent gap between exact optimality and practical tractability on large or highly dynamic instances, even as computing hardware and solvers improve. Foundational surveys and texts remain the touchstones for this complexity landscape and motivate approximations, decomposition, and learning-enhanced methods that deliver strong anytime performance at scale.

2.2. Recent Methodological Developments

The profound computational complexity of industrial scheduling has motivated three complementary streams of scalable methods: (i) metaheuristics and hybrids; (ii) decomposition and parallelization; and (iii) AI-/data-driven approaches. Below, each stream is expanded with emphasis on post-2020 progress and deep learning trends.

2.2.1. Metaheuristics and Hybrid Algorithms

Metaheuristics remain a workhorse for large instances because they can explore vast, rugged search spaces quickly and flexibly. Recent work emphasizes problem-aware hybridization, adaptive control, and learning-enhanced neighborhoods.

- **Genetic algorithms (GAs) and memetic hybrids.** Modern GA variants integrate local search, path relinking, or destroy-and-repair moves to accelerate convergence on very large instances and complex shop settings; hybrids tuned for industrial-scale unrelated/parallel machines and sequence-dependent setups are increasingly common. Representative examples show GA+local-search hybrids scaling to hundreds of machines/jobs while retaining solution quality (Blum & Roli, 2003; Ferreira et al., 2022).
- **Simulated annealing (SA) and tabu search (TS).** Classical SA/TS ideas—probabilistic uphill moves and adaptive memory—continue to underpin strong baselines. Contemporary implementations pair TS with constraint-aware neighborhoods or embed instance-specific neighborhoods learned from data to reduce cycling and improve the intensification/diversification balance. Conceptual surveys still frame best practices for hybrid design (Blum & Roli, 2003).
- **Large-neighborhood search (LNS) and learning-enhanced LNS.** LNS “destroy-and-repair” is particularly effective under tight timing constraints. Recent neural LNS variants use deep networks (often graph-based) to propose destroy sets or repair decisions, yielding large speed/quality gains across combinatorial problems and increasingly in scheduling (Hottung & Tierney, 2022).
- **Hyper-heuristics (rule selection/generation).** Instead of solving a schedule directly, hyper-heuristics learn which heuristic to deploy when. A recent line uses deep reinforcement learning (DRL) hyper-heuristics to select operators on-the-fly, improving generalizability across shop configurations (Panzer & Bender, 2022; Smit et al., 2024).
- **Learning-assisted parameter control & initialization.** Reviews highlight the benefit of machine-learned parameter schedules, warm-starts, and population initializers to stabilize metaheuristics on high-variance instance distributions—especially for multi-objective settings (Bengio et al., 2021).

Overall, the trend is clear: state-of-the-art metaheuristics increasingly embed learned guidance (policies, surrogates, or predictors) while retaining the robustness and portability that made them dominant in practice (Blum & Roli, 2003; Bengio et al., 2021).

2.2.2. Decomposition and Parallelization

Decomposition breaks a monolith into solvable parts; parallelization exploits modern hardware and solver frameworks. Together they are the main levers for scaling exact and hybrid methods.

- **Logic-Based Benders Decomposition (LBBD).** LBBD separates combinatorial assignment/sequence decisions (handled by CP/MIP/heuristics) from schedule-feasibility subproblems, iteratively exchanging powerful logic cuts. Recent papers demonstrate strong performance on flexible/distributed job-shops and highlight modeling patterns and cut design that make LBBD competitive on industrial testbeds (Naderi et al., 2022; Juvin et al., 2023).
- **Hierarchical/rolling-horizon schemes.** Multi-level decompositions—e.g., plan vs. schedule, coarse time windows vs. fine sequencing—remain essential when the full horizon is prohibitive. Newer work integrates domain constraints from chemical/process systems and uses decomposition to keep digital-twin/CP models responsive at runtime; learning-guided rolling horizons are emerging to adapt window sizes and priorities on the fly (Liñán & Méndez, 2024; Forbes & Kelly, 2024).
- **Dantzig-Wolfe/column generation and branch-and-price.** Modern implementations in open frameworks (e.g., SCIP/GCG) expose decomposition hooks, enabling practitioners to combine exact and heuristic components and to scale on shared/distributed memory (Bestuzheva et al., 2021).
- **Parallel solver ecosystems.** Documented advances from 2001–2020 show order-of-magnitude speedups from algorithmic and hardware progress; contemporary suites include UG, a unified framework for parallelizing branch-and-bound/price/cut across cores and clusters. These

capabilities benefit both pure MIP/CP scheduling and hybrid MH+MIP workflows (Koch et al., 2022; Bestuzheva et al., 2021).

Pragmatically, decomposition + parallelization are how many plants deploy provably strong methods within real wall-clock limits, and they combine naturally with the AI techniques below (e.g., learned cut/branching within a decomposed master) (Koch et al., 2022).

2.2.3. AI-Driven and Data-Driven Methods

AI brings policy learning, structure learning, and fast approximations—often on graph representations of shops—and is the most active area since 2020.

- **Deep reinforcement learning (DRL)** for dispatching and end-to-end scheduling.
 - **Learned dispatching rules.** GNN-based DRL learns to choose the next operation/machine given a disjunctive-graph state, outperforming hand-crafted rules and transferring to larger instances (Zhang et al., 2020).
 - **Systematic evidence** (2022–2024). Surveys map model choices (GNNs, attention/transformers), training regimes, robustness/generalization gaps, and industrial case studies—useful for selecting architectures and evaluation protocols (Panzer & Bender, 2022; Zhang et al., 2024; Smit et al., 2024).
 - **Digital-twin-in-the-loop training and deployment.** Coupling DRL with twins improves sample efficiency and safety prior to shop-floor rollout (Zhang et al., 2022).
- **Learning-augmented optimization (L4CO)** for exact solvers.
 - Cut selection via RL/imitation. DRL policies for cutting-plane selection in MILP and successors (2020–2024) reduce nodes/time across instance families; these techniques directly accelerate large MIP/CP models of scheduling (Tang et al., 2020; Huang et al., 2022; Wang et al., 2023).
 - Learned branching/diving and node selection. Neural policies guide B&B traversal and primal heuristics, improving primal-dual gaps and anytime behavior on real MIP workloads (Nair et al., 2020; Bengio et al., 2021).
- **Neural Large-Neighborhood Search (Neural-LNS).** Deep networks propose destroy/repair actions within LNS, maintaining metaheuristic scalability while injecting structural priors (Hottung & Tierney, 2022).
- **Supervised and interpretable learning of rules/policies.** Data-driven mining of dispatching rules from near-optimal schedules and interpretable learned rules (e.g., sparse/structured models) offer transparent alternatives for regulated environments—often used to warm-start DRL or guide MH neighborhoods (Ferreira et al., 2022).
- **Surrogate-assisted optimization.** ML surrogates approximate expensive objective/simulation evaluations (e.g., multi-objective, dynamic shops), enabling deeper search within fixed time budgets and stabilizing online rescheduling (Ferreira et al., 2022; Panzer & Bender, 2022).
- **Foundation-model ideas (early stage).** “LLMs as optimizers” (OPRO) and LLM-guided search/planning are being tested as meta-controllers—suggesting heuristic templates or operator sequences that a solver or metaheuristic then refines. While nascent, this strand aims at zero-/few-shot generalization across plants and products (Yang et al., 2024; Bengio et al., 2021).

In short, the most effective recent systems are hybrids: a decomposed/parallel exact core or robust metaheuristic scaffold, augmented by learning (DRL policies, neural destroy/repair, learned cuts/branching, surrogates) to navigate huge decision spaces under tight time limits (Koch et al., 2022; Bestuzheva et al., 2021).

Table 1 summarizes the above-described methods from a scalability point of view.

Table 1. Comparative Analysis of Recent Methods for Scalability in Industrial Scheduling.

Approach	Core strengths	Limitations	Typical application areas	Representative references
Genetic algorithms & memetic hybrids	Flexible; multi-objective ready; easy to hybridize with local search/repair; robust on heterogeneous constraints	Parameter tuning; stochastic variance; may plateau without strong neighborhoods	Parallel/flow/flexible job shops; sequence-dependent setups; large unrelated-machine problems	(Blum & Roli, 2003; Ferreira et al., 2022)
Simulated annealing / Tabu search	Simple and effective baselines; good intensification/diversification; easy to embed constraints	Cooling/tenure sensitivity; may require problem-specific neighborhoods	Job/flow shops; batching; setup-heavy sequencing	(Blum & Roli, 2003)
Large-neighborhood search (LNS) / Neural-LNS	Powerful destroy–repair exploration; learned destroy/repair improves speed & quality; anytime behavior	Designing repairs that preserve feasibility; training data/compute for neural variants	High-mix shops; near-real-time improvement; rolling re-optimization	(Hottung & Tierney, 2022)
Hyper-heuristics (selection / generation)	Generalizes across instance types; automates rule choice; compatible with DRL	Performance ceiling if candidate pool is weak; requires meta-level data	Mixed-model production; variable routing/loads	(Panzer & Bender, 2022; Smit & Van Vliet, 2024)
Logic-Based Benders Decomposition (LBBD)	Strong logic cuts; separates assignment/sequence from timing; integrates CP/MIP/heuristics	Modeling effort; cut engineering; potential many iterations	Flexible/distributed job shops; process/chemical scheduling	(Naderi et al., 2022; Juvin et al., 2023; Liñán & Méndez, 2024; Forbes & Kelly, 2024)
Hierarchical / rolling-horizon schemes	Scales long horizons; aligns with planning→scheduling tiers; supports simulation-in-the-loop	Coordination overhead; myopic decisions if horizons too short	Plant-level planning with shop-floor dispatch; digital-twin what-if analysis	(Liñán & Méndez, 2024; Forbes & Kelly, 2024)
Column generation / branch-and-price frameworks	Decompose by columns/routes; strong bounds; mix with heuristics	Pricing complexity; stabilization needed; parallelization non-trivial	Large machine/route generation models; transportation–production links	(Bestuzheva et al., 2021; Koch et al., 2022)
Parallel solver ecosystems	Multicore/cluster speedups; parallel B&B/price/cut (UG); mature tooling	Needs HPC resources; solver engineering expertise	Large MIP/CP scheduling; scenario-decomposed planning	(Bestuzheva et al., 2021; Koch et al., 2022)
DRL dispatching policies	Learns size-agnostic rules; reacts online; strong anytime performance	Sample efficiency; stability/robustness	Dynamic job/flexible job shops; real-time dispatch	(Zhang et al., 2020; Panzer & Bender,

Approach	Core strengths	Limitations	Typical application areas	Representative references
(GNN/attention)		ss; policy explainability		2022; Zhang et al., 2024; Smit & Van Vliet, 2024)
Learning-augmented optimization (ML for OR)	Learned branching/cuts/node selection; warm-starts; improves primal-dual gaps	Generalization across distributions; integration into certified workflows	Large MIP/CP scheduling; hybrid MH+MIP stacks	(Nair et al., 2020; Tang et al., 2020; Tian et al., 2021; Wang et al., 2022; Bengio et al., 2021)
Surrogate-/supervised rule learning	Fast evaluations; interpretable policies; good for high-volume data	Surrogate bias; retraining under drift; limited exploration	Repetitive/flow environments; KPI-specific rule mining	(Ferreira et al., 2022; Gil-Gala et al., 2023)
Digital twin-in-the-loop RL	Safe policy training; proactive, state-aware rescheduling; sim-to-real transfer	Twin fidelity/sync cost; integration complexity	Smart factories; semiconductor/assembly lines	(Zhang et al., 2022)
Foundation-model-guided heuristics (OPRO)	Rapid heuristic design/tuning; few-shot adaptability; complements DRL/OR	Very early stage; needs feasibility guards and evaluation harness	Rapid ramp-up for new product mixes/lines	(Yang et al., 2024; Bengio et al., 2021)

2.3. Industrial Impact

The adoption of scalable scheduling methods has produced tangible benefits across manufacturing and service operations. In high-mix job/flow shops, robust metaheuristic baselines—often hybridized with local improvement and problem-aware neighborhoods—continue to reduce lead times and work-in-process while maintaining schedule feasibility under complex constraints (Blum & Roli, 2003; Ferreira et al., 2022). Learning-enhanced search further expands this impact: neural large-neighborhood search and related hybrids provide fast, high-quality improvements under tight decision latencies, a pattern now being translated from routing to production scheduling settings (Hottung & Tierney, 2022; Bengio et al., 2021).

For **large, highly constrained plants** (e.g., semiconductor wafer fabs and flexible job shops), decomposition and parallel solver ecosystems have been crucial. Logic-based Benders and related decompositions separate assignment/sequence choices from timing feasibility, enabling strong cuts and subproblem specialization; reported results on flexible/distributed job shops show competitive anytime performance with reliable convergence behavior (Juvin et al., 2023). In parallel, advances in MILP/CP solver engineering and HPC frameworks—particularly unified parallelization of branch-and-bound/price/cut—have delivered order-of-magnitude speedups over the last two decades, narrowing the gap between optimality guarantees and industrial wall-clock deadlines (Bestuzheva et al., 2021; Koch et al., 2022). In wafer-fab settings, such tooling integrates naturally with established production-planning and dispatching practices (Mönch et al., 2013).

AI-driven approaches increasingly complement these stacks. **Deep reinforcement learning (DRL)** policies trained on disjunctive-graph representations learn size-agnostic dispatching rules that generalize to larger instances and volatile shop conditions; systematic reviews report consistent gains in tardiness, throughput, and resilience across testbeds (Zhang et al., 2020; Panzer & Bender, 2022;

Zhang et al., 2024). **Digital-twin-in-the-loop scheduling** strengthens the path to deployment: twins enable safe policy training and allow proactive, state-aware rescheduling once online (Zhang et al., 2022), aligning with the broader shift toward cyber-physical production systems and IIoT platforms (Monostori, 2016; Xu et al., 2018a).

Despite these advances, important **gaps** remain for industrial adoption. Stakeholders frequently request interpretable, auditable decision logic—especially in regulated domains—driving interest in **interpretable rule learning** and hybrid DRL+rule designs (Ferreira et al., 2022). Multi-objective trade-offs (e.g., service, energy, emissions) are increasingly prominent, calling for methods that deliver scalable, **explainable** Pareto policies and that remain stable under distribution shift (Panzer & Bender, 2022; Zhang et al., 2024). Finally, realizing end-to-end impact requires reliable **data/compute infrastructure**—edge/cloud orchestration, streaming quality control, and human-in-the-loop decision support (Romero et al., 2020; Xu et al., 2018a). Bridging these elements—decomposition and parallel solvers, learning-augmented heuristics, digital twins, and human-centered interfaces—will continue to move scalable scheduling from research prototypes to robust, real-time industrial decision systems.

3. Robustness and Adaptability to Uncertainty

3.1. The Prevalence of Uncertainty in Industrial Scheduling

Industrial environments are rife with uncertainties—machine breakdowns, unpredictable processing times, urgent rush orders, supply-chain disruptions, and human factors, to name a few (Herrmann, 2006; Ivanov & Dolgui, 2020b). Traditional static schedules, even if optimal under assumed conditions, often falter when such disturbances arise, leading to inefficiencies, missed deadlines, and costly rework (Pinedo, 2016). As systems scale and markets become more volatile, robust and adaptive scheduling becomes imperative. The digital transformation of operations increases both the visibility of stochastic dynamics and the opportunities to respond. High-frequency streams from shop-floor sensors and MES/ERP logs enable online detection of anomalies, delay predictions, and the learning of proactive control policies. Recent advances—deep reinforcement learning (DRL), graph neural networks (GNNs), neural surrogates, and digital-twin-in-the-loop training—are pushing beyond fixed “robust plans,” enabling continuous, data-driven adaptation under uncertainty while preserving computational tractability (Zhang et al., 2022; Zhang et al., 2022; Zhang et al., 2024).

3.2. Recent Methodological Developments

Efforts to increase robustness and adaptability fall into three main streams: robust optimization; stochastic/probabilistic modeling; and real-time, predictive, and reactive scheduling. Below we summarize key ideas, with an emphasis on post-2020 developments and AI-enabled techniques.

3.2.1. Robust Optimization

Robust optimization explicitly models uncertainty and seeks schedules that perform well across a range of realizations (Kouvelis & Yu, 1997; Bertsimas & Sim, 2004). In modern deployments, robust models are often hybridized with learning components for forecasting, dynamic parameterization of uncertainty sets, or warm-starting.

- **Min-max and min-max regret formulations.** These guard against worst-case or worst-regret scenarios—useful where delivery penalties or rework costs are high (Aissi et al., 2009). While conservative, recent practice tunes uncertainty budgets to balance robustness and performance, often informed by empirical variance estimates extracted from shop data (Bertsimas & Sim, 2004).
- **Adjustable robust optimization (ARO).** Defers part of the decision (e.g., dispatching, batching) until information is revealed, improving adaptability versus static designs (Ben-Tal et al., 2004).

Rolling-horizon ARO for job shops with uncertain processing times demonstrates strong performance under continuous disturbances (Cohen et al., 2023).

- **Interval/set-based uncertainty.** Interval activity durations and release dates yield tractable robust counterparts and are attractive in regulated or contract-driven environments; hybrid robust approaches for projects exemplify this trend (Bruni et al., 2017).
- **Learning-in-the-loop robust models.** Robust parameters (e.g., uncertainty budgets, scenario weights) can be calibrated from historical trace data or forecasts and periodically retuned; neural surrogates speed robust evaluation when embedded inside metaheuristics or rolling-horizon loops (Zhang et al., 2022).

3.2.2. Stochastic and Probabilistic Modeling

Stochastic models represent uncertainty via probability distributions or stochastic processes and optimize expectations, risk measures, or violation probabilities (Pinedo, 2016; Birge & Louveaux, 2011).

- **Chance-constrained scheduling.** Constraints (e.g., due-date adherence) are enforced with high probability, enabling explicit trade-offs between service levels and efficiency (Birge & Louveaux, 2011). In data-rich plants, estimated distributions are kept up to date from streaming data and predictive models.
- **Markov decision processes (MDP).** MDP formulations capture sequential uncertainty and state transitions. For job-shop settings with stochastic processing times, MDPs provide a principled foundation and also underpin modern DRL policies (Zhang et al., 2017; Puterman, 2005).
- **Simulation-based evaluation and design.** Monte Carlo/discrete-event simulation (DES) remains essential when analytic tractability is limited. It supports proactive design of robust schedules, stress-tests rollout policies, and serves as a safe training ground for learning-based controllers (Vieira et al., 2003; Mönch et al., 2013).

3.2.3. Real-Time, Predictive, and Reactive Scheduling

These approaches adapt schedules dynamically in response to real-time information, disturbances, or new job arrivals. Recent work blends classical repair/rolling-horizon controls with DRL, GNNs, and digital twins.

- **Rescheduling and repair algorithms.** Minimal-perturbation repairs stabilize operations after disruptions, reducing shop-floor turbulence. Frameworks and taxonomies remain highly relevant (Vieira et al., 2003; Ouelhadj & Petrovic, 2009), and are increasingly combined with learned predictors of disruption impact to prioritize repairs.
- **Rolling-horizon and event-driven updates.** Periodic or event-triggered reoptimization integrates naturally with MES/ERP. State-of-practice implementations use hierarchical decompositions and fast heuristics/MIP models, often parallelized, to refresh plans at high cadence (Vieira et al., 2003; Weng et al., 2022).
- **Predictive analytics and machine learning.** Supervised models forecast delays, failures, and congestion; DRL agents learn dispatching policies that generalize across shop states. Reviews synthesize model choices (GNNs, attention/transformers), training regimes, and robustness/generalization gaps (Serrano-Ruiz et al., 2021; Zhang et al., 2024).
- **Digital-twin-in-the-loop decision-making.** Twins provide high-fidelity simulators for safe testing and sample-efficient training/deployment of real-time policies (Zhang et al., 2020; Zhang et al., 2022).
- **Multi-agent and self-organizing control.** Decentralized agent-based frameworks enhance resilience by localizing decisions while coordinating globally through negotiation/market or contract-net mechanisms—well aligned with cyber-physical production systems (Leitão et al., 2016a; Seitz et al., 2021).

- **End-to-end AI stacks at scale.** In practice, the strongest systems are hybrids: fast decomposed MIP/CP or robust metaheuristics at the core, augmented by DRL policies, learned repair operators, neural surrogates, and digital twins to navigate vast decision spaces under tight time limits (Lee & Lee, 2022; Zhang et al., 2024).

Table 2 summarizes the above-described methods from the point of view of robustness and adaptability in industrial scheduling.

Table 2. Comparative Analysis of Methods for Robustness and Adaptability in Industrial Scheduling.

Approach	Core strengths	Limitations	Typical application areas	Representative references
Min-max & Min-max Regret Robust Optimization	Strong guarantees; interpretable; protects against penalties	Conservative; scalability issues with large scenario sets	Semiconductor fabs, aerospace, contract manufacturing	(Aissi et al., 2009; Bertsimas & Sim, 2004)
Adjustable Robust Optimization (ARO)	Balances robustness and flexibility; realistic for dynamic shops	More complex; heavier computation	Job shops with uncertain processing times	(Ben-Tal et al., 2004; Cohen et al., 2023)
Interval/Set-Based Models	Tractable; practical for bounded uncertainties	Can yield conservative schedules	Project-driven and regulated industries	(Bruni et al., 2017)
Learning-in-the-loop Robust Models	Adaptive; efficient evaluation; improves robustness	Requires quality data; explainability issues	Flexible manufacturing, online scheduling	(Zhang et al., 2022)
Chance-Constrained Scheduling	Balances service levels vs efficiency; intuitive	Relies on accurate distribution estimation	Service industries, logistics, large projects	(Birge & Louveaux, 2011)
Markov Decision Processes (MDP)	Principled sequential control; foundation for DRL	Curse of dimensionality for large systems	Stochastic job shops, batch processes	(Zhang et al., 2017; Puterman, 2005)
Simulation-Based Evaluation (DES/Monte Carlo)	Flexible; captures complex interactions; supports stress-testing	Computationally expensive	Semiconductor, project-based, high-uncertainty industries	(Vieira et al., 2003; Mönch et al., 2013)
Rescheduling & Repair Algorithms	Stable shop floor behavior; minimal disruption	Myopic if frequent disruptions occur	MES/MRP systems, dynamic job shops	(Vieira et al., 2003; Ouelhadj & Petrovic, 2009)
Rolling-Horizon/Event-Driven Updates	Continuous adaptation; ERP/MES integration	Risk of nervousness with frequent updates	High-mix, volatile production	(Vieira et al., 2003; Weng et al., 2022)
Predictive Analytics & ML	Data-driven; real-time adaptability;	Data hungry; legacy integration challenges	Smart factories, flexible electronics	(Serrano-Ruiz et al., 2021; Zhang et al., 2024)

Approach	Core strengths	Limitations	Typical application areas	Representative references
	generalizable policies			
Digital-Twin-in-the-Loop Scheduling	Safe training/testing; improves sample efficiency	Twin fidelity/synchronization cost	Intelligent manufacturing, reconfigurable factories	(Zhang et al., 2020; Zhang et al., 2022)
Multi-Agent & Self-Organizing Systems	Resilient; scalable; fault-tolerant	Coordination and global optimality issues	Cyber-physical production, distributed factories	(Leitão et al., 2016a; Seitz et al., 2021)
End-to-End AI Stacks at Scale	Hybrid performance; scalable and adaptive under real-time constraints	Engineering complexity; integration & MLOps challenges	Large-scale Industry 4.0, smart factories	(Lee & Lee, 2022; Zhang et al., 2024)

3.3. Industrial Impact

Research advances in robust and adaptive scheduling are rapidly transferring to industrial practice, enabled by Industry 4.0 and the ubiquitous digitization of factory environments. In capital-intensive domains such as semiconductor fabrication, aerospace, and high-value custom production, robust and stochastic scheduling is increasingly applied to mitigate the high costs of rescheduling and downtime (Giret et al., 2015; Mönch et al., 2013). Robust optimization models and stochastic formulations provide effective safeguards against disruptions, particularly where contractual service levels and reliability are critical.

In highly dynamic industries—food processing, agile automotive, and flexible electronics—reactive and predictive scheduling algorithms are being integrated into manufacturing execution systems. These enable reductions in downtime, service-level improvements, and higher equipment utilization through predictive analytics and real-time reoptimization (Mönch et al., 2013). Data-driven methods such as DRL-based dispatching and neural surrogate models have been especially effective in managing uncertainty while respecting tight time constraints, with successful demonstrations in flexible job shops and assembly lines (Zhang et al., 2024).

Decentralized and multi-agent scheduling approaches are also gaining traction. When combined with digital twins, these methods enhance resilience by localizing decisions and enabling distributed coordination across smart factories. Recent studies demonstrate robust multi-agent control architectures that remain stable under frequent disturbances and scale effectively in cyber-physical production systems (Seitz et al., 2021; Leitão et al., 2016b). Hybrid architectures—where metaheuristics, robust optimization, and multi-agent systems are augmented by real-time predictive analytics—are increasingly deployed in pilot Industry 4.0 testbeds, particularly for smart logistics and reconfigurable assembly (Giret et al., 2015; Zhang et al., 2022).

Despite these gains, several open challenges remain. Balancing robustness and performance efficiency is non-trivial, as overly conservative schedules may reduce throughput. Methods for uncertainty quantification and explainability of AI-driven approaches are not yet standardized, raising adoption barriers. Data privacy and cybersecurity risks emerge as predictive and decentralized systems rely heavily on shared sensor and cloud data. Finally, interoperability across platforms and legacy systems limits the seamless deployment of self-organizing, multi-agent scheduling frameworks. Addressing these issues—alongside creating benchmarks and human-in-the-loop control paradigms—will be central to advancing industrial adoption.

4. Integration with Digitalization and Industry 4.0

4.1. Industrial Scheduling in the Age of Digital Transformation

The advent of Industry 4.0 has fundamentally altered the landscape of industrial scheduling. Modern enterprises are increasingly interconnected, harnessing the Industrial Internet of Things (IIoT), big data analytics, digital twins, and cyber-physical systems (CPSs) to create adaptive and autonomous shop floors (Mourtzis et al., 2020; Wang et al., 2016). In these data-rich and sensor-driven environments, scheduling is no longer a static, offline optimization task but a dynamic, real-time decision process seamlessly embedded within production execution systems (Kusiak, 2019; Xu et al., 2018b).

This digital transformation introduces new requirements and opportunities. Scheduling algorithms must now:

- rapidly process high-frequency streaming data from sensors and MES/ERP logs,
- interact with intelligent machines and human operators in collaborative CPSs, and
- adapt autonomously to both predicted and unforeseen disruptions.

Crucially, these systems must be interoperable with digital infrastructures, including ERP, MES, cloud, and edge computing platforms, while guaranteeing security and scalability in complex industrial environments (Rauch et al., 2020).

Recent advances in AI and deep learning are reshaping this integration. Deep reinforcement learning (DRL) and graph neural networks (GNNs) are being deployed for real-time dispatching and predictive rescheduling, exploiting the graph-structured nature of job-shop networks (Zhang et al., 2024). Transformer-based architectures further enhance forecasting accuracy by capturing temporal dependencies in machine states and job arrivals (Song et al., 2023). Meanwhile, digital twin-driven scheduling frameworks enable closed-loop learning, where algorithms are trained and validated against high-fidelity virtual replicas before being deployed on the shop floor (Zhang et al., 2021).

Another important development is the emergence of cloud-edge collaborative scheduling: heavy optimization tasks are solved in the cloud, while real-time adjustments are delegated to lightweight edge agents co-located with machines (Lu et al., 2020). This architecture improves responsiveness while ensuring that AI-powered schedulers remain scalable across global production networks.

Altogether, industrial scheduling in the digital era is moving toward autonomous, learning-enabled ecosystems that blend optimization, machine learning, and distributed digital infrastructures. This convergence represents both the core opportunity and central challenge of scheduling in Industry 4.0.

4.2. Recent Methodological Developments

4.2.1. Data-Driven Scheduling and Real-Time Data Integration

The exponential increase in accessible, high-quality process data within modern industrial environments has enabled new classes of scheduling algorithms that leverage real-time information for greater agility and responsiveness.

- **Sensor-Enabled, Closed-Loop Scheduling.** Modern shop floors, equipped with IIoT sensors and CPSs, continuously generate streams of data on machine status, job progress, and environmental conditions. Scheduling algorithms can now operate in closed-loop mode, where feedback from the shop floor directly drives updates to production plans (Wang et al., 2016; Rauch et al., 2020). These approaches improve agility but also raise challenges in data quality assurance, latency management, and interoperability with legacy systems. Emerging solutions apply streaming analytics and lightweight deep models at the edge to process sensor inputs in milliseconds.

- **Digital Twin-Based Scheduling.** Digital twins (DTs)—virtual replicas of physical systems—are increasingly central to scheduling in Industry 4.0. DTs mirror the current shop state and can simulate disruptions, evaluate dispatching rules, and test repair strategies before they are deployed on the shop floor. This enables dynamic rescheduling, what-if analysis, and proactive maintenance scheduling (Uhlemann et al., 2017; Kritzing et al., 2018). Recent work links DTs with reinforcement learning agents, providing safe training environments where policies are stress-tested virtually before live deployment (Zhang et al., 2021; Leng et al., 2020).
- **Cloud and Edge Computing for Distributed Scheduling.** Cloud-based scheduling platforms offer scalable cooperative optimization, supporting multi-plant and supply-chain-level scheduling tasks with heavy computation offloaded to distributed clusters (Mourtzis & Vlachou, 2018). In contrast, edge computing brings intelligence closer to the shop floor, enabling low-latency rescheduling in response to real-time events (Lu et al., 2020). Hybrid cloud–edge architectures are gaining traction, where global optimization runs in the cloud while local edge agents handle immediate decisions, balancing responsiveness and scalability.

Together, these data-driven paradigms are shifting industrial scheduling from static planning to adaptive, self-correcting ecosystems capable of handling volatility at scale.

4.2.2. Autonomous, Intelligent, and Decentralized Scheduling

The integration of advanced artificial intelligence and distributed control frameworks is transforming scheduling decisions, fostering systems capable of high autonomy, self-adaptation, and decentralized negotiation.

- **Agent-Based and Multi-Agent Scheduling Systems:** Autonomous software agents (machines, cells, workpieces) negotiate job allocations and routing independently, supporting decentralized, modular scheduling architectures aligned with flexible manufacturing systems (Leitão et al., 2016a; Giret et al., 2017). Recent advances leverage digital twins (Siatras et al., 2024) and multi-agent reinforcement learning (Xu et al., 2025) to enhance negotiation, coalition formation, and adaptive learning for global performance.
- **Self-Optimizing and Adaptive Control Algorithms:** Self-optimizing scheduling algorithms continuously adapt parameter values, decision rules, or objectives in light of new data or predicted disturbances (Kusiak, 2017). Deep reinforcement learning methods such as multi-agent dueling DRL (Qin et al., 2023), graph-based MARL (Zhang et al., 2023), and hierarchical MARL (Wang et al., 2025) are enabling scalable and resilient scheduling in dynamic environments.
- **Emerging Architectures:** Knowledge-graph-enhanced MARL (Qin & Lu, 2024), attention-based coordination (Zheng et al., 2025), and decentralized training strategies (Malucelli et al., 2025) represent next-generation paradigms, further strengthening adaptability and autonomy in Industry 4.0 scheduling.

4.2.3. Interoperability, Standardization, and Security

The effectiveness of digitalized scheduling also hinges on robust interface design, standardized interoperable frameworks, and secure handling of the growing volume and variety of critical scheduling data exchanged across industrial networks.

- **Interoperable Architectures.** Modern scheduling stacks integrate with heterogeneous ERP/MES/SCM ecosystems via standardized information models and open APIs. OPC UA-centric service models and Asset Administration Shell (AAS)-based dataspace connectors enable plug-and-operate exposure of machine capabilities and scheduling services across sites and partners—supporting decentralized optimization and rapid reconfiguration (Beregi et al., 2021; Neubauer et al., 2023).
- **Semantically Enriched, AI-Ready Data Layers.** Knowledge-graph and model-driven integration (e.g., KG-backed twins, auto-generated data collection architectures) provide a

common vocabulary across planning, dispatching, and control. This boosts data quality and feature consistency for deep learning and RL schedulers, shortens data engineering cycles, and improves cross-system explainability (Wan et al., 2024; Trunzer et al., 2021).

- **Security and Data Provenance.** As scheduling moves onto IIoT/cloud fabrics, compliance-by-design with ICS/IIoT security baselines (e.g., IEC 62443 mappings, NIST ICS guidance) is essential. End-to-end provenance and tamper-evident audit trails—sometimes blockchain-anchored and paired with ML for predictive auditing—help ensure integrity, confidentiality, and traceability of schedule decisions and event logs across organizational boundaries (Cindrić et al., 2025; NIST, 2023; Hu et al., 2020; Umer et al., 2024).
- **Data Sovereignty & Federated Collaboration (added).** Dataspace-oriented integration (AAS + policy-enforced connectors) supports inter-company scheduling use cases (capacity sharing, subcontracting) while retaining usage-control over shared datasets and learned models—key for privacy-preserving, multi-party optimization (Neubauer et al., 2023).
- **Operational Hardening for AI-Driven Scheduling (added).** As DL/RL components enter the loop, interface standards and security controls must extend to model artifacts and pipelines (versioned data/model registries, signed inference services, and policy-aware event buses), ensuring reproducibility and trustworthy deployment in time-critical rescheduling scenarios (Beregi et al., 2021; NIST, 2023).

Table 3 summarizes the above-described methods addressing the integration with digitalization and industry 4.0.

Table 3. Comparative Analysis of Methods for Integration with Digitalization and Industry 4.0.

Approach	Core strengths	Limitations	Typical application areas	Representative references
Sensor-Enabled, Closed-Loop Scheduling	Real-time responsiveness; immediate adaptation to shop-floor events; integration of IIoT/CPS data streams	Data quality and latency challenges; integration with legacy systems; requires robust edge analytics	High-variability shop floors; condition-based rescheduling; flow-shop monitoring	Wang et al. (2016); Rauch et al. (2020)
Digital Twin-Based Scheduling	Virtual experimentation; safe training/testbed for RL agents; proactive rescheduling and predictive maintenance	High development and synchronization costs; computationally intensive	Job-shop/flexible shop scheduling; disruption management; predictive control	Uhlemann et al. (2017); Kritzinger et al. (2018); Zhang et al. (2021); Leng et al. (2020)
Cloud and Edge Computing for Distributed Scheduling	Scalable optimization (cloud); low-latency local response (edge); hybrid setups balance global and local	Security and data-transfer overhead; partitioning optimization tasks is complex	Multi-plant coordination; distributed supply chains; real-time edge rescheduling	Mourtzis & Vlachou (2018); Lu et al. (2020)
Agent-Based and Multi-Agent Scheduling Systems	Decentralization, modularity, and negotiation capabilities; well-	Coordination overhead; global optimality hard to guarantee	Flexible job-shop systems; distributed resource allocation	Leitão et al. (2016a); Giret et al. (2017); Siatras et al. (2024); Xu et al. (2025)

Approach	Core strengths	Limitations	Typical application areas	Representative references
	suited to flexible manufacturing			
Self-Optimizing and Adaptive Control Algorithms	Continuous adaptation to data and disturbances; reinforcement learning and heuristic evolution enable resilience	Sample inefficiency in RL; difficulty in explainability; requires large/high-quality datasets	Dynamic job-shop scheduling; mass personalization; adaptive planning	Kusiak (2017); Qin et al. (2023); Zhang et al. (2023); Wang et al. (2025)
Emerging Architectures (KG-MARL, attention-based, decentralized training)	Enhanced context-awareness; improved coordination; scalable decentralized learning	Complexity of design; limited industrial deployments; integration with legacy IT/OT	Smart manufacturing scheduling; dynamic flow/assembly shops	Qin & Lu (2024); Zheng et al. (2025); Malucelli et al. (2025)
Interoperable Architectures (OPC UA, AAS, open APIs)	Seamless integration across ERP/MES/SCM; supports plug-and-operate scheduling services	Requires ecosystem-wide standard adoption; potential vendor lock-in	Multi-system integration; cross-site scheduling; Manufacturing-X initiatives	Beregi et al. (2021); Neubauer et al. (2023)
Semantically Enriched, AI-Ready Data Layers	Standard vocabulary for heterogeneous data; improves explainability and feature quality for DL/RL	Knowledge graph development overhead; ontology alignment challenges	Digital twins; predictive scheduling; cross-enterprise scheduling	Wan et al. (2024); Trunzer et al. (2021)
Security and Data Provenance	Ensures integrity, confidentiality, and traceability of scheduling data; supports compliance (IEC 62443, NIST)	Added overhead in performance; blockchain solutions not yet fully scalable	Regulated supply chains; critical infrastructures; cloud manufacturing	Cindrić et al. (2025); NIST (2023); Hu et al. (2020); Umer et al. (2024)
Data Sovereignty & Federated Collaboration	Policy-enforced data sharing across organizations; supports privacy-preserving optimization	Governance complexity; interoperability still evolving	Inter-company scheduling; collaborative supply chains; subcontracting	Neubauer et al. (2023)
Operational Hardening for AI-Driven Scheduling	Secure and reproducible ML pipelines; signed model artifacts; trustworthy rescheduling	Requires ML lifecycle governance; raises infrastructure complexity	AI-driven job-shop scheduling; cloud-edge rescheduling services	Beregi et al. (2021); NIST (2023)

4.3. Industrial Impact

Digitalization is reshaping the production floor from plan–execute to sense–decide–adapt loops. Real-time sensor streams fused into digital twins (DTs) are shortening the time from deviation to decision: shops detect anomalies earlier, evaluate counterfactuals virtually, and deploy schedule repairs with less risk. Demonstrations in discrete manufacturing show DT-driven anomaly detection and rolling-window rescheduling that cut response latency and improve throughput; complementary work uses DTs to train RL agents safely for dispatching and policy control before go-live—key for automotive, electronics, and high-value custom manufacturing where disruptions and mix variability are high (Uhlemann et al., 2017; Xia et al., 2021; Li et al., 2023).

At network scale, cloud–edge scheduling stacks are proving decisive. Cloud back-ends coordinate heavy optimization across plants and suppliers, while edge controllers execute fast, local rescheduling under machine/AGV events; recent DT-enabled flexible job-shop deployments report real-time responsiveness with compute pushed to the edge and global plans synchronized from the cloud. This division of labor is now common in supply-chain-intensive sectors and multi-factory groups (Ma et al., 2020; Gao et al., 2024; Mourtzis, 2022).

Decentralized and agent-based control is also moving from concept to impact. Industrial case work shows MAS+DT architectures that localize negotiation and routing while preserving global KPIs; in parallel, expert studies across production/supply networks identify concrete MAS use cases that lift resilience—e.g., autonomous replanning, distributed bottleneck mitigation, and exception handling—supporting modular, small-batch, and reconfigurable lines (Siatras et al., 2024; Nitsche et al., 2023).

Finally, the diffusion of interoperability and security baselines is a practical accelerator. Asset Administration Shell (AAS) models and service interfaces are easing plug-and-operate integration of scheduling services with ERP/MES/CPS, while updated OT/ICS security guidance formalizes segmentation, provenance, and hardening requirements for IIoT-connected scheduling—critical for regulated sectors and cross-border collaboration (Abdel-Aty et al., 2022; NIST, 2023). Remaining blockers—data/model standardization, legacy coupling, and assurance of real-time decision quality at scale—are active research and deployment fronts (Parente et al., 2020).

5. Conclusions and Research Directions

Industrial scheduling remains a cornerstone of modern operations yet continues to face three intertwined hurdles: **(i)** scaling to large, high-dimensional instances, **(ii)** staying robust and adaptive under uncertainty and disruptions, and **(iii)** integrating deeply with digitalization—IIoT, digital twins, cloud/edge, and secure interoperable ecosystems. Across these fronts, the past five years have seen notable progress: faster metaheuristics and decomposition, more mature rescheduling for real-time events, and increasingly “software-defined” factories where data and models flow among ERP/MES/SCM, device layers, and analytics stacks (Parente et al., 2020; Mourtzis, 2022). Still, industrial impact hinges on standardization, trust, and rigorous engineering of ML/OR pipelines end-to-end (Abdel-Aty et al., 2022; NIST, 2023).

A decisive shift is the rise of deep learning and deep reinforcement learning (DRL) as practical tooling for complex, dynamic scheduling. Recent work demonstrates: (1) policy learning that reacts in milliseconds to shop-floor events, (2) training “in the twin” to de-risk deployment, and (3) stronger generalization via graph-structured and attention-based models. Case studies now span semiconductor packaging, flexible job shops, and distributed production networks—where DRL outperforms rules/metaheuristics or offers comparable quality at much lower decision latency (Xia et al., 2021; Liu et al., 2022; Park & Park, 2023; Wang et al., 2021).

Deep models are beginning to demonstrate clear value in several areas of industrial scheduling, most notably in the following domains:

- **Policy learning for real-time decisions.** DRL agents trained in simulation or digital twins learn dispatching, routing, and batching policies that scale to many machines and diverse job mixes, offering competitive makespan/tardiness with tight reaction times. Centralized or multi-agent

variants increasingly handle disturbances and changing shop states (Kovács et al., 2022; Zhang et al., 2022; Wang et al., 2021).

- **Generalization and transfer.** Graph and attention models encode precedence, resource compatibilities, and machine–job relations, enabling transfer across families of instances and faster adaptation to new products or line configurations (Cappart et al., 2021; Peng et al., 2021; Wang et al., 2024).
- **Perception-to-schedule loops.** CNN/RNN/LSTM pipelines for predictive maintenance and anomaly detection feed early warnings to schedulers, enabling proactive repair policies and fewer bottlenecks by aligning maintenance windows with production plans (Bampoula et al., 2021).

At the same time, significant obstacles remain that currently limit the broader industrial adoption of neural schedulers:

- **Interpretability and assurance.** Black-box policies face scrutiny in regulated and safety-critical operations. Tooling for XAI/XRL, post-hoc rationales, counterfactuals, and certifiable robustness remains underused in scheduling, yet is increasingly feasible (Milani et al., 2024).
- **Data quality and benchmarks.** Many plants lack curated, labeled datasets for learning and objective comparison. Open, standardized benchmarks (including realistic simulators and DT-backed logs) are essential to measure progress and reproducibility (Parente et al., 2020; Gao et al., 2024).
- **Legacy integration and lifecycle MLOps.** Industrial IT/OT landscapes demand hardened interfaces (model registries, signed inference, versioned features), standardized semantics, and zero-downtime rollout/rollback for policies—especially when rescheduling is time-critical (Abdel-Aty et al., 2022; NIST, 2023).
- **Robustness and safety.** Policies must remain stable under distribution shift, sensor noise, or partial outages. Methods from robust and safe RL—risk-sensitive training, certified bounds, disturbance/adversary models—should be brought into the scheduling loop with plant-level validation (Moos et al., 2022).
- **Human-in-the-loop.** Operators and planners bring tacit knowledge and risk judgments. Practical systems will blend human guidance with learned policies—e.g., learning from interventions, preference feedback, or human-authored constraints—to ensure actionable, trusted decisions (Mourtzis, 2022).

Looking forward, several research priorities emerge that will be central to realizing the next wave of digital, AI-driven scheduling systems:

- **Interpretable and certifiable neural scheduling** (XRL, policy simplification, safety monitors) with plant-ready evidence artifacts (Milani et al., 2024).
- **Open datasets, simulators, and DT-based benchmarks** for dynamic shop floors (events, breakdowns, product changeovers), enabling apples-to-apples evaluation and reproducibility (Parente et al., 2020; Kovács et al., 2022).
- **Seamless integration** of AI with **IoT platforms, digital twins, and edge/cloud**, using interoperable data models/ontologies and policy-aware event buses (Abdel-Aty et al., 2022; Gao et al., 2024).
- **Federated and privacy-preserving learning** for cross-site/cross-enterprise scheduling, with model provenance and usage controls (NIST, 2023).
- **Design for robustness:** training against disturbances, runtime monitors, and rollback strategies to keep service levels under shocks (Moos et al., 2022).
- **Human-in-the-loop frameworks** that combine optimization/learning with operator intent, safety culture, and multi-objective business constraints (Mourtzis, 2022).

With these directions, industrial scheduling can fully exploit digitalization: policies that learn continuously, explain their choices, and operate safely at scale—across connected factories and supply networks.

Author Contributions: Conceptualization, A.I.; methodology, A.I.; formal analysis, A.I.; investigation, A.I.; resources, A.I.; writing—original draft preparation, A.I.; writing—review and editing, A.I.; supervision, A.I.; project administration, A.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI - UEFISCDI, project number ERANET-CHISTERA-IV-REMINDER, within PNCDI IV.

Abbreviations

The following abbreviations are used in this manuscript:

AAS	Asset Administration Shell
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
ARO	Adjustable Robust Optimization
CNN	Convolutional Neural Network
CP	Constraint Programming
CPS	Cyber-Physical System
DES	Discrete-Event Simulation
DRL	Deep Reinforcement Learning
DT	Digital Twin
ERP	Enterprise Resource Planning
GA	Genetic Algorithm
GCG	Generic Column Generation
GNN	Graph Neural Network
HPC	High-Performance Computing
ICS	Industrial Control Systems
IEC	International Electrotechnical Commission
IIoT	Industrial Internet of Things
IoT	Internet of Things
KPI	Key Performance Indicator
KG	Knowledge Graph
LBBD	Logic-Based Benders Decomposition
LLM	Large Language Model(s)
LNS	Large-Neighborhood Search
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
MES	Manufacturing Execution System
MILP	Mixed-Integer Linear Programming
MIP	Mixed-Integer Programming
ML	Machine Learning
MLOps	Machine-Learning Operations
NIST	National Institute of Standards and Technology
NP-hard	Nondeterministic Polynomial-time hard
OPC UA	Open Platform Communications Unified Architecture
OPRO	Optimizers by Prompting
OR	Operations Research
PdM	Predictive Maintenance
RNN	Recurrent Neural Network
RL	Reinforcement Learning
SA	Simulated Annealing
SCM	Supply Chain Management
SCIP	Solving Constraint Integer Programs (optimization framework)

TS	Tabu Search
UG	Unified parallelization framework for branch-and-bound/price/cut
XAI	Explainable Artificial Intelligence
XRL	Explainable Reinforcement Learning

References

1. Abdel-Aty, T. A., Negri, E., & Galparoli, S. (2022). Asset Administration Shell in Manufacturing: Applications and Relationship with Digital Twin. *IFAC-PapersOnLine*, 55(10), 2533–2538. <https://doi.org/10.1016/j.ifacol.2022.10.090>
2. Aissi, H., Bazgan, C., & Vanderpooten, D. (2009). Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2), 427–438. <https://doi.org/10.1016/j.ejor.2008.09.012>
3. Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3), 985–1032. <https://doi.org/10.1016/j.ejor.2006.06.060>
4. Bampoula, X., Siaterlis, G., Nikolakis, N., & Alexopoulos, K. (2021). A Deep Learning Model for Predictive Maintenance in Cyber-Physical Production Systems Using LSTM Autoencoders. *Sensors*, 21(3), 972. <https://doi.org/10.3390/s21030972>
5. Bengio, Y., Lodi, A., & Prouvost, A. (2021). Machine Learning for Combinatorial Optimization: a Methodological Tour d’Horizon. *European Journal of Operational Research*, 290(2), 405–421. <https://doi.org/10.1016/j.ejor.2020.07.063>
6. Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2), 351–376. <https://doi.org/10.1007/s10107-003-0454-y>
7. Beregi, R., Németh, D., Turek, P., Monostori, L., & Váncza, J. (2021). Manufacturing Execution System Integration through the Standardization of a Common Service Model for Cyber-Physical Production Systems. *Applied Sciences*, 11(16), 7581. <https://doi.org/10.3390/app11167581>
8. Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52(1), 35–53. <https://doi.org/10.1287/opre.1030.0065>
9. Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., et al. (2021). The SCIP Optimization Suite 8.0. arXiv. <https://doi.org/10.48550/arXiv.2112.08872>
10. Bestuzheva, K., Vigerske, S., Achterberg, T., et al. (2021). The SCIP Optimization Suite 8.0. arXiv. <https://doi.org/10.48550/arXiv.2112.08872>
11. Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming* (2nd ed.). Springer. <https://doi.org/10.1007/978-1-4614-0237-4>
12. Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (Eds.). (2007). *Handbook on Scheduling: From Theory to Applications*. Springer. <https://link.springer.com/book/10.1007/978-3-540-32220-7>
13. Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268–308. <https://doi.org/10.1145/937503.937505>
14. Bruni, M. E., Di Puglia Pugliese, L., Beraldi, P., & Guerriero, F. (2017). An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations. *Omega*, 71, 66–84. <https://doi.org/10.1016/j.omega.2016.09.009>
15. Cappart, Q., Chételat, D., Khalil, E., Lodi, A., Morris, C., & Veličković, P. (2021). Combinatorial optimization and reasoning with graph neural networks (Survey). *IJCAI 2021*. <https://doi.org/10.24963/ijcai.2021/595>
16. Cindrić, I., Jurčević, M., & Hadjina, T. (2025). Mapping of Industrial IoT to IEC 62443 Standards. *Sensors*, 25(3), 728. <https://doi.org/10.3390/s25030728>
17. Cohen, I., Postek, K., & Shtern, S. (2023). An adaptive robust optimization model for parallel machine scheduling. *European Journal of Operational Research*, 306(1), 83–104. <https://doi.org/10.1016/j.ejor.2022.07.018>

18. Fang, K., Uhan, N., Zhao, F., & Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, 30(4), 234–240. <https://doi.org/10.1016/j.jmsy.2011.08.004>
19. Ferreira, C., Figueira, G., & Amorim, P. (2022). Effective and interpretable dispatching rules for dynamic job shops via guided empirical learning. *Omega*, 111, 102643. <https://doi.org/10.1016/j.omega.2022.102643>
20. Forbes, M. A., Harris, M. G., Jansen, H. M., van der Schoot, F. A., & Taimre, T. (2024). Combining optimisation and simulation using logic-based Benders decomposition. *European Journal of Operational Research*, 312(3), 840–854. <https://doi.org/10.1016/j.ejor.2023.07.032>
21. Gahm, C., Denz, F., Dirr, M., & Tuma, A. (2016). Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research*, 248(3), 744–757. <https://doi.org/10.1016/j.ejor.2015.07.017>
22. Gao, Q., Gu, F., Li, L., & Guo, J. (2024). A framework of cloud–edge collaborated digital twin for flexible job shop scheduling with conflict-free routing. *Robotics and Computer-Integrated Manufacturing*, 86, 102672. <https://doi.org/10.1016/j.rcim.2023.102672>
23. Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman. (open copy) <https://perso.limos.fr/~palafour/PAPERS/PDF/Garey-Johnson79.pdf>
24. Gasse, M., Chételat, D., Ferroni, N., Charlin, L., & Lodi, A. (2019). Exact Combinatorial Optimization with Graph Convolutional Neural Networks. *NeurIPS 2019*. <https://arxiv.org/abs/1906.01629>
25. Giret, A., Trentesaux, D., & Prabhu, V. (2015). Sustainability in manufacturing operations scheduling: A state of the art review. *Journal of Manufacturing Systems*, 37, 126–140. <https://doi.org/10.1016/j.jmsy.2015.08.002>
26. Giret, A., Trentesaux, D., Salido, M. A., Garcia, E., & Adam, E. (2017). A holonic multi-agent methodology to design sustainable intelligent manufacturing control systems. *Journal of Cleaner Production*, 167, 1370–1386. <https://doi.org/10.1016/j.jclepro.2017.03.079>
27. Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, 5, 287–326. [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
28. Gupta, J. N. D., & Stafford, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3), 699–711. <https://doi.org/10.1016/j.ejor.2005.02.001>
29. Herrmann, J. W. (Ed.). (2006). *Handbook of Production Scheduling*. Springer. <https://doi.org/10.1007/0-387-33117-4>
30. Hottung, A., & Tierney, K. (2022). Neural large neighborhood search for routing problems. *Artificial Intelligence*, 313, 103786. <https://doi.org/10.1016/j.artint.2022.103786>
31. Hu, R., Yan, Z., Ding, W., & Yang, L. T. (2020). A survey on data provenance in IoT. *World Wide Web*, 23(2), 1441–1463. <https://doi.org/10.1007/s11280-019-00746-1>
32. Huang, Z., Wang, K., Liu, F., Zhen, H.-L., Zhang, W., Yuan, M., Hao, J., Yu, Y., & Wang, J. (2022). Learning to select cuts for efficient mixed-integer programming. *Pattern Recognition*, 124, 108353. <https://doi.org/10.1016/j.patcog.2021.108353>
33. Ivanov, D., & Dolgui, A. (2020a). A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. *Production Planning & Control*, 32(9), 775–788. <https://doi.org/10.1080/09537287.2020.1768450>
34. Ivanov, D., & Dolgui, A. (2020b). Viability of intertwined supply networks: Extending the supply chain resilience angles toward survivability. *International Journal of Production Research*, 58(10), 2904–2915. <https://doi.org/10.1080/00207543.2020.1750727>
35. Ivanov, D., & Dolgui, A. (2021). OR-methods for coping with the ripple effect in supply chains during COVID-19: Managerial insights and research implications. *International Journal of Production Economics*, 232, 107921. <https://doi.org/10.1016/j.ijpe.2020.107921>
36. Juvin, C., Houssin, L., & Lopez, P. (2023). Logic-based Benders decomposition for the preemptive flexible job-shop scheduling problem. *Computers & Operations Research*, 154, 106156. <https://doi.org/10.1016/j.cor.2023.106156>

37. Khalil, E. B., Le Bodic, P., Song, L., Nemhauser, G., & Dilkina, B. (2016). Learning to Branch in Mixed Integer Programming. AAAI 2016. <https://ojs.aaai.org/index.php/AAAI/article/view/10080>
38. Koch, T., Berthold, T., Pedersen, J., & Vanaret, C. (2022). Progress in mathematical programming solvers from 2001 to 2020. *EURO Journal on Computational Optimization*, 10, 100031. <https://doi.org/10.1016/j.ejco.2022.100031>
39. Kool, W., van Hoof, H., & Welling, M. (2018). Attention, Learn to Solve Routing Problems! arXiv:1803.08475. <https://arxiv.org/abs/1803.08475>
40. Kouvelis, P., & Yu, G. (1997). *Robust Discrete Optimization and Its Applications*. Springer. <https://doi.org/10.1007/978-1-4757-2620-6>
41. Kovács, B., Tassel, P., Gebser, M., & Seidel, G. (2022). A Customizable Reinforcement Learning Environment for Semiconductor Fab Simulation. 2022 Winter Simulation Conference, 2663–2674. <https://doi.org/10.1109/WSC57314.2022.10015524>
42. Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11), 1016–1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>
43. Kusiak, A. (2017). Smart manufacturing must embrace big data. *Nature*, 544(7648), 23–25. <https://doi.org/10.1038/544023a>
44. Kusiak, A. (2019). Fundamentals of smart manufacturing: A multi-thread perspective. *Annual Reviews in Control*, 47, 214–220. <https://doi.org/10.1016/j.arcontrol.2019.02.001>
45. Lee, Y. H., & Lee, S. (2022). Deep reinforcement learning based scheduling within production plan in semiconductor fabrication. *Expert Systems with Applications*, 191, 116222. <https://doi.org/10.1016/j.eswa.2021.116222>
46. Leitão, P., Colombo, A. W., & Karnouskos, S. (2016a). Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry*, 81, 11–25. <https://doi.org/10.1016/j.compind.2015.08.004>
47. Leitão, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., & Colombo, A. W. (2016b). Smart agents in industrial cyber-physical systems. *Proceedings of the IEEE*, 104(5), 1086–1101. <https://doi.org/10.1109/JPROC.2016.2521931>
48. Li, Y., Tao, Z., Wang, L., Du, B., Guo, J., & Pang, S. (2023). Digital twin-based job shop anomaly detection and dynamic scheduling. *Robotics and Computer-Integrated Manufacturing*, 79, 102443. <https://doi.org/10.1016/j.rcim.2022.102443>
49. Liñán, D. A., & Ricardez-Sandoval, L. A. (2024). Multicut logic-based Benders decomposition for discrete-time scheduling and dynamic optimization of network batch plants. *AIChE Journal*, 70(9), e18491. <https://doi.org/10.1002/aic.18491>
50. Liu, R., Piplani, R., & Toro, C. (2022). Deep reinforcement learning for dynamic scheduling of a flexible job shop. *International Journal of Production Research*, 60(13), 4049–4069. <https://doi.org/10.1080/00207543.2022.2058432>
51. Lu, Y., Liu, C., Wang, K. I.-K., Huang, H., & Xu, X. (2020). Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61, 101837. <https://doi.org/10.1016/j.rcim.2019.101837>
52. Ma, J., Zhou, H., Liu, C., E, M., Jiang, Z., Wang, Q., et al. (2020). Study on edge-cloud collaborative production scheduling based on enterprises with multi-factory. *IEEE Access*, 8, 30069–30080. <https://doi.org/10.1109/ACCESS.2020.2972914>
53. Malucelli, N., Domini, D., Aguzzi, G., & Viroli, M. (2025). Neighbor-Based Decentralized Training Strategies for Multi-Agent Reinforcement Learning. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25)*. <https://doi.org/10.1145/3672608.3707923>
54. Milani, S., Faraji, S., Wu, J., McCann, T., Ghassemi, M., & Santu, S. (2024). Explainable Reinforcement Learning: A Survey and Comparative Review. *ACM Computing Surveys*, 56(7), 168. <https://doi.org/10.1145/3616864>

55. Mönch, L., Fowler, J. W., & Mason, S. J. (2013). *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. Springer. <https://doi.org/10.1007/978-1-4614-4472-5>
56. Monostori, L. (2016). Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2), 621–641. <https://doi.org/10.1016/j.cirp.2016.06.005>
57. Moos, J., Hansel, K., Abdulsamad, H., Stark, S., Clever, D., & Peters, J. (2022). Robust Reinforcement Learning: A Review of Foundations and Recent Advances. *Machine Learning and Knowledge Extraction*, 4(1), 276–315. <https://doi.org/10.3390/make4010013>
58. Mourtzis, D. (2022). Advances in Adaptive Scheduling in Industry 4.0. *Frontiers in Manufacturing Technology*, 2, 937889. <https://doi.org/10.3389/fmtec.2022.937889>
59. Mourtzis, D., & Vlachou, E. (2018). A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *Journal of Manufacturing Systems*, 47, 179–198. <https://doi.org/10.1016/j.jmsy.2018.05.008>
60. Mourtzis, D., Vlachou, E., & Milas, N. (2016). Industrial Big Data as a Result of IoT Adoption in Manufacturing. *Procedia CIRP*, 55, 290–295. <https://doi.org/10.1016/j.procir.2016.07.038>
61. Naderi, B., & Roshanaei, V. (2022). Critical-path-search logic-based Benders decomposition approaches for flexible job shop scheduling. *INFORMS Journal on Optimization*, 4(1), 1–28. <https://doi.org/10.1287/ijoo.2021.0056>
62. Nair, V., Bartunov, S., Gimeno, F., et al. (2020). Solving mixed integer programs using neural networks. arXiv. <https://doi.org/10.48550/arXiv.2012.13349>
63. National Institute of Standards and Technology (NIST). (2023). *Guide to Operational Technology (OT) Security*. <https://doi.org/10.6028/NIST.SP.800-82r3>
64. Neubauer, M., Steinle, L., Reiff, C., Ajdinović, S., Klingel, L., Lechler, A., & Verl, A. (2023). Architecture for Manufacturing-X: Bringing Asset Administration Shell, Eclipse Dataspace Connector and OPC UA together. *Manufacturing Letters*, 37, 1–6. <https://doi.org/10.1016/j.mfglet.2023.05.002>
65. Nitsche, B., Brands, J., Treiblmaier, H., & Gebhardt, J. (2023). The impact of multiagent systems on autonomous production and supply chain networks: use cases, barriers and contributions to logistics network resilience. *Supply Chain Management: An International Journal*, 28(5), 894–908. <https://doi.org/10.1108/SCM-07-2022-0282>
66. Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4), 417–431. <https://doi.org/10.1007/s10951-008-0090-8>
67. Panzer, M., & Bender, B. (2022). Deep Reinforcement Learning in Production Systems: A Systematic Literature Review. *International Journal of Production Research*, 60(13), 4316–4341. <https://doi.org/10.1080/00207543.2021.1973138>
68. Parente, M., Figueira, G., Amorim, P., & Marques, A. (2020). Production scheduling in the context of Industry 4.0: Review and trends. *International Journal of Production Research*, 58(17), 5401–5431. <https://doi.org/10.1080/00207543.2020.1718794>
69. Park, I.-B., & Park, J. (2023). Scalable Scheduling of Semiconductor Packaging Facilities Using Deep Reinforcement Learning. *IEEE Transactions on Cybernetics*, 53(6), 3518–3531. <https://doi.org/10.1109/TCYB.2021.3128075>
70. Peng, Y., Choi, B., & Xu, J. (2021). Graph Learning for Combinatorial Optimization: A Survey of State-of-the-Art. *Data Science and Engineering*, 6, 119–141. <https://doi.org/10.1007/s41019-021-00155-3>
71. Pinedo, M. L. (2016). *Scheduling: Theory, Algorithms, and Systems* (5th ed.). Springer. <https://doi.org/10.1007/978-3-319-26580-3>
72. Puterman, M. L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (2nd ed.). Wiley. <https://doi.org/10.1002/9780470316887>
73. Qin, Z., & Lu, Y. (2024). Knowledge graph-enhanced multi-agent reinforcement learning for adaptive scheduling in smart manufacturing. *Journal of Intelligent Manufacturing*
74. Qin, Z., Johnson, D., & Lu, Y. (2023). Dynamic production scheduling towards self-organizing mass personalization: A multi-agent dueling deep reinforcement learning approach. *Journal of Manufacturing Systems*, 68, 242–257. <https://doi.org/10.1016/j.jmsy.2023.03.003>

75. Rauch, E., Linder, C., & Dallasega, P. (2020). Anthropocentric perspective of production before and within Industry 4.0. *Computers & Industrial Engineering*, 139, 105644. <https://doi.org/10.1016/j.cie.2019.01.018>
76. Romero, D., Stahre, J., Taisch, M., et al. (2020). The Operator 4.0: Towards socially sustainable factories of the future. *Computers & Industrial Engineering*, 139, 106128. <https://doi.org/10.1016/j.cie.2019.106128>
77. Seitz, M., Gehlhof, F., Cruz Salazar, L. A., Fay, A., & Vogel-Heuser, B. (2021). Automation platform independent multi-agent system for robust networks of production resources in Industry 4.0. *Journal of Intelligent Manufacturing*, 32(7), 2023–2041. <https://doi.org/10.1007/s10845-021-01759-2>
78. Serrano-Ruiz, J. C., Mula, J., & Poler, R. (2021). Smart manufacturing scheduling: A literature review. *Journal of Manufacturing Systems*, 61, 265–287. <https://doi.org/10.1016/j.jmsy.2021.09.011>
79. Siatras, V., Bakopoulos, E., Mavrothalassitis, P., Nikolakis, N., & Alexopoulos, K. (2024). Production Scheduling Based on a Multi-Agent System and Digital Twin: A Bicycle Industry Case. *Information*, 15(6), 337. <https://doi.org/10.3390/info15060337>
80. Smit, I. G., Zhou, J., Reijnen, R., Wu, Y., Chen, J., Zhang, C., Bukhsh, Z., Zhang, Y., & Nuijten, W. (2024/2025). Graph neural networks for job shop scheduling problems: A survey. *Computers & Operations Research*, 176, 106914. <https://doi.org/10.1016/j.cor.2024.106914>
81. Song, L., Li, Y., & Xu, J. (2023). Dynamic Job-Shop Scheduling Based on Transformer and Deep Reinforcement Learning. *Processes*, 11(12), 3434. <https://doi.org/10.3390/pr11123434>
82. Tang, Y., Agrawal, S., & Faenza, Y. (2020). Reinforcement Learning for Integer Programming: Learning to Cut. *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, PMLR 119: 9367–9376. DOI: <https://doi.org/10.48550/arXiv.1906.04859>
83. Tao, F., & Zhang, M. (2017). Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. *IEEE Access*, 5, 20418–20427. <https://doi.org/10.1109/ACCESS.2017.2756069>
84. Trunzer, E., Vogel-Heuser, B., Chen, J.-K., & Kohnle, M. (2021). Model-Driven Approach for Realization of Data Collection Architectures for Cyber-Physical Systems of Systems to Lower Manual Implementation Efforts. *Sensors*, 21(3), 745. <https://doi.org/10.3390/s21030745>
85. Uhlemann, T. H.-J., Schock, C., Lehmann, C., Freiberger, S., & Steinhilper, R. (2017). The Digital Twin: Demonstrating the potential of real-time data acquisition in production systems. *Procedia Manufacturing*, 9, 113–120. <https://doi.org/10.1016/j.promfg.2017.04.043>
86. Umer, M. A., Umer, M., Pandey, M., & Abdulla, S. (2024). Leveraging Artificial Intelligence and Provenance Blockchain Framework to Mitigate Risks in Cloud Manufacturing in Industry 4.0. *Electronics*, 13(3), 660. <https://doi.org/10.3390/electronics13030660>
87. Vieira, G. E., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1), 39–62. <https://doi.org/10.1023/A:1022235519958>
88. Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer Networks. *NeurIPS 2015*. <https://arxiv.org/abs/1506.03134>
89. Wan, Y., Liu, Y., Chen, Z., Chen, C., Li, X., Hu, F., et al. (2024). Making knowledge graphs work for smart manufacturing: Research topics, applications and prospects. *Journal of Manufacturing Systems*, 76, 1–22. <https://doi.org/10.1016/j.jmsy.2024.07.009>
90. Wang, L., Pan, Z., & Wang, J. (2021). A review of reinforcement learning-based intelligent optimization for manufacturing scheduling. *Complex System Modeling and Simulation*, 1(4), 257–270. <https://doi.org/10.23919/CSMS.2021.0027>
91. Wang, R., Wang, G., Sun, J., Deng, F., & Chen, J. (2024). Flexible Job Shop Scheduling via Dual Attention Network-Based Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3), 3091–3102. <https://doi.org/10.1109/TNNLS.2023.3306421>
92. Wang, S., Wan, J., Li, D., & Zhang, C. (2016). Implementing smart factory of Industrie 4.0: An outlook. *International Journal of Distributed Sensor Networks*, 12(1), 3159805. <https://doi.org/10.1155/2016/3159805>
93. Wang, W., Zhang, Y., Wang, Y., Pan, G., & Feng, Y. (2025). Hierarchical multi-agent deep reinforcement learning for dynamic flexible job-shop scheduling with transportation. *International Journal of Production Research* (advance online publication), 1–28. <https://doi.org/10.1080/00207543.2025.2511239>

94. Wang, Z., Li, X., Wang, J., Kuang, Y., Yuan, M., Zeng, J., Zhang, Y., & Wu, F. (2023). Learning cut selection for mixed-integer linear programming via hierarchical sequence model. arXiv. <https://doi.org/10.48550/arXiv.2302.00244>
95. Weng, W., Chen, J., Zheng, M., & Fujimura, S. (2022). Realtime scheduling heuristics for just-in-time production in large-scale flexible job shops. *Journal of Manufacturing Systems*, 63, 64–77. <https://doi.org/10.1016/j.jmsy.2022.01.006>
96. Xia, K., Sacco, C., Kirkpatrick, M., Saidy, C., Nguyen, L., Kircaliali, A., & Harik, R. (2021). A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *Journal of Manufacturing Systems*, 58, 210–230. <https://doi.org/10.1016/j.jmsy.2020.06.012>
97. Xiong, H., Shi, S., Ren, D., & Hu, J. (2022). A survey of job shop scheduling problem: The types and models. *Computers & Operations Research*, 142, 105731. <https://doi.org/10.1016/j.cor.2022.105731>
98. Xu, H., Yu, W., Griffith, D., & Golmie, N. (2018). A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective. *IEEE Access*, 6, 78238–78259. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9074819/>
99. Xu, L. D., Xu, E. L., & Li, L. (2018). Industry 4.0: State of the art and future trends. *International Journal of Production Research*, 56(8), 2941–2962. <https://doi.org/10.1080/00207543.2018.1444806>
100. Xu, Y., Wang, Z., & Li, X. (2025). Multi-agent reinforcement learning for flexible job shop scheduling: A review. *Frontiers in Industrial Engineering*, 2, 1611512. <https://doi.org/10.3389/fieng.2025.1611512>
101. Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., & Chen, X. (2024). Large Language Models as Optimizers (OPRO). ICLR 2024. <https://openreview.net/forum?id=Bb4VGOWELI>
102. Zhang, C., Juraschek, M., & Herrmann, C. (2024). Deep reinforcement learning-based dynamic scheduling for resilient and sustainable manufacturing: A systematic review. *Journal of Manufacturing Systems*, 77, 962–989. <https://doi.org/10.1016/j.jmsy.2024.10.026>
103. Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P. S., & Xu, C. (2020). Learning to dispatch for job-shop scheduling via deep reinforcement learning. arXiv. <https://doi.org/10.48550/arXiv.2010.12367>
104. Zhang, C., Wang, X., & Li, J. (2023). DeepMAG: Multi-agent graph reinforcement learning for dynamic job shop scheduling. *Knowledge-Based Systems*, 259, 110083. <https://doi.org/10.1016/j.knsys.2022.110083>
105. Zhang, F., Bai, J., Yang, D., & Wang, Q. (2022). Digital twin data-driven proactive job-shop scheduling strategy towards asymmetric manufacturing execution decision. *Scientific Reports*, 12, 1546. <https://doi.org/10.1038/s41598-022-05304-w>
106. Zhang, L., Yan, Y., Hu, Y., & Ren, W. (2022). Reinforcement learning and digital twin-based real-time scheduling method in intelligent manufacturing systems. *IFAC-PapersOnLine*, 55(10), 359–364. <https://doi.org/10.1016/j.ifacol.2022.09.413>
107. Zhang, M., Tao, F., & Nee, A. Y. C. (2020). Digital twin-enhanced dynamic job-shop scheduling. *Journal of Manufacturing Systems*, 58, 146–156. <https://doi.org/10.1016/j.jmsy.2020.04.008>
108. Zhang, T., Xie, S., & Rose, O. (2017). Real-time job shop scheduling based on simulation and Markov decision processes. *Proceedings of the 2017 Winter Simulation Conference (WSC)*, 3357–3368. <https://doi.org/10.1109/WSC.2017.8248100>
109. Zhang, Y., Zhu, H., Tang, D., Zhou, T., & Gui, Y. (2022). Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 78, 102412. <https://doi.org/10.1016/j.rcim.2022.102412>
110. Zheng, J., Zhao, Y., Li, Y., Li, J., Wang, L., & Yuan, D. (2025). Dynamic flexible flow shop scheduling via cross-attention networks and multi-agent reinforcement learning. *Journal of Manufacturing Systems*, 80, 395–411. <https://doi.org/10.1016/j.jmsy.2025.03.005>
111. Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of Industry 4.0: A review. *Engineering*, 3(5), 616–630. <https://doi.org/10.1016/j.eng.2017.05.015>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.