Innovative Computing Perspectives

https://icp.cultechpub.com/icp

Cultech Publishing

Article

Intelligent Scheduling in Reconfigurable Manufacturing Systems Using Petri Nets and Hybrid Optimization

Salah Hammedi[1,2,*]

[1]Electrical Engineering Department, National School of Engineers of Monastir, University of Monastir, Monastir, Tunisia

[2]NOCCS Laboratory, National Engineering School of Sousse, University of Sousse, Sousse, Tunisia

[*]Corresponding author: Salah Hammedi, salahhammedi@yahoo.com

Abstract

This paper proposes an intelligent scheduling methodology for Reconfigurable Manufacturing Systems (RMS) that integrates Petri Net (PN) modeling with heuristic and metaheuristic optimization techniques. The framework is validated through an industrial case study in automotive component manufacturing characterized by fluctuating demand, variable order sizes, and stochastic machine downtimes. Quantitative results show that the proposed approach reduces average production delays by 20%, increases resource utilization from 75% to 90%, and improves responsiveness to urgent orders by 25% compared with traditional scheduling methods. A comparative analysis further demonstrates that while rule-based scheduling and PN-based heuristics provide limited improvements under high variability, the integration of Genetic Algorithms (GA) and Ant Colony Optimization (ACO) ensures superior scalability, maintaining low tardiness even when the number of jobs increases significantly. By coupling a formal PN representation with adaptive decision mechanisms, the proposed methodology achieves both efficient scheduling and practical industrial applicability, addressing limitations of existing approaches that treat modeling and optimization separately.

Keywords

Petri Nets, Reconfigurable manufacturing systems, Scheduling optimization, Metaheuristics, Intelligent manufacturing

Article History

Received: 27 November 2025
Revised: 05 January 2026
Accepted: 18 March 2026
Available Online: 30 March 2026

1. Introduction

In the face of increasing industrial competition and rapidly changing market demands, manufacturing companies must continually adapt to fluctuations in customer requirements. Reconfigurable Manufacturing Systems (RMS) have emerged as a promising solution to address these challenges, as noted by Koren et al. [1]. RMS are characterized by their modularity and flexibility, enabling rapid reconfiguration of production resources to accommodate varying product types and quantities while optimizing operational efficiency.

Despite their advantages, RMS introduce significant complexity in production planning and scheduling. Unlike traditional manufacturing systems, RMS require dynamic scheduling approaches capable of addressing resource conflicts, balancing workloads, and adhering to strict deadlines. Conventional scheduling techniques, such as fixed-priority rules or linear programming methods, often fail to capture the dynamic and stochastic nature of reconfigurable systems [2]. This calls for the adoption of advanced modeling and optimization techniques tailored to the unique demands of RMS.

Petri Nets (PNs) have been widely recognized as an effective tool for modeling and analyzing complex discrete-event systems, including manufacturing environments [3]. Their ability to represent concurrency, synchronization, and resource constraints makes them particularly suitable for scheduling problems in RMS.

In Recent research has therefore moved toward hybrid approaches, combining Petri Net (PN) modeling with heuristic and metaheuristic optimization techniques, such as Shortest Processing Time (SPT), Earliest Due Date (EDD), Genetic Algorithms (GA), and Ant Colony Optimization (ACO) [4,5]. These approaches aim to exploit the strengths of PNs in system representation while leveraging intelligent optimization methods to improve scheduling performance. Nevertheless, much of the existing literature remains either theoretical, simulation-based, or evaluated on benchmark problems, with limited validation in real industrial settings. Moreover, systematic quantitative comparisons between traditional scheduling, PN-based heuristics, and PN-based metaheuristics under identical operating conditions are still scarce.

Motivated by these gaps, this study proposes and evaluates an intelligent scheduling framework that tightly integrates PN modeling with heuristic and metaheuristic optimization strategies in a real-world industrial case study [6]. The selected case involves an automotive components manufacturer operating under frequent reconfiguration, variable order sizes, machine downtime, and urgent delivery constraints conditions that realistically reflect the challenges faced by modern RMS. Through this case study, the paper goes beyond a purely conceptual contribution by providing quantitative performance comparisons across different scheduling paradigms.

The main objectives of this work are twofold:

1) To demonstrate the effectiveness of PN-based intelligent scheduling by combining formal system modeling with heuristic and metaheuristic decision mechanisms in a reconfigurable manufacturing context [7].

2) To provide a comparative and quantitative evaluation of traditional scheduling approaches, PN-based heuristics, and PN-based metaheuristics, highlighting their respective strengths and limitations in terms of delay reduction, resource utilization, scalability, and adaptability.

By addressing both modeling and optimization aspects within a unified and practically validated framework, this study contributes to bridging the gap between theoretical developments in intelligent scheduling and their industrial deployment in RMS.

The remainder of this paper is organized as follows. Section 2 reviews the related work, covering traditional scheduling methods, PN applications in manufacturing systems, and recent intelligent and hybrid optimization approaches, with the aim of identifying the research gaps addressed in this study. Section 3 presents the industrial case study and details the proposed methodology, including PN modeling, the integration of heuristic and metaheuristic scheduling strategies, and the overall architecture of the intelligent scheduling framework. Section 4 reports the experimental results, focusing on quantitative performance indicators such as production delay reduction, resource utilization, and scalability under increasing workload conditions. Section 5 provides a comprehensive discussion of the results and their practical implications, including industrial applicability, computational trade-offs, scalability considerations, and deployment perspectives in real manufacturing environments. Finally, Section 6 concludes the paper by summarizing the main findings, highlighting the limitations of the current study, and outlining future research directions, particularly in relation to IoT-enabled integration, reinforcement learning (RL), and sustainable production scheduling.

2. Related Work

The scheduling of RMS has been extensively studied in both theoretical and applied research. To contextualize the contribution of this study, this section reviews traditional approaches, classical applications of PNs, and recent intelligent and hybrid methods (2020-2025).

2.1 Traditional Approaches in RMS Scheduling

Early research on RMS scheduling relied on traditional methods such as fixed-priority rules, linear programming, and simulation-based optimization. These methods provided mathematically rigorous and structured solutions [8,9]. However, they were limited in adaptability when addressing fluctuating demand and frequent system reconfigurations [10,11]. In particular, static scheduling models often failed to capture real-time disturbances such as machine breakdowns, operator unavailability, and changes in customer requirements. While valuable as a foundation, these methods highlighted the need for dynamic and adaptive scheduling strategies.

2.2 PNs in Manufacturing Systems

PNs have long been applied to discrete-event systems, offering a powerful graphical and mathematical tool to represent concurrency, synchronization, and resource allocation. Seminal works such as Murata [12] and Silva & Recalde [13] established their role in formal verification, deadlock detection, and performance analysis.

Over the years, PNs have been used in job-shop scheduling [14], assembly systems modeling [15], and resource sharing optimization [16]. Their ability to represent reconfiguration and task concurrency makes them particularly suited for RMS. Recent works have extended PN models to Stochastic PNs [17], Colored PNs [18], and Timed PNs [19], which allow better modeling of timing variability, task differentiation, and resource constraints.

2.3 Intelligent and Hybrid Approaches (2020–2025)

Recent advances in RMS scheduling emphasize the integration of PNs with artificial intelligence and metaheuristics. Several hybrid approaches have emerged:

- PNs with GA: Time PN-GA (TPGA) approaches have been applied to flexible job-shop scheduling, efficiently handling both logical dependencies and timing constraints [20,21].
- Hybrid PSO-GA with PNs: These approaches improve convergence speed in dynamic order processing [22].
- Scatter Search with PNs: Applied to Flexible Manufacturing Systems (FMS), showing improvements in utilization and robustness against machine breakdowns [23].
- RL with PNs: Deep Q-Networks (DQN) combined with Graph Convolutional Networks (GCN) over PN structures achieved superior adaptability and computational efficiency [24].
- PetriRL framework: An explainable RL-PN hybrid enabling scalability and online adaptability without retraining, well suited to RMS [25].
- Constraint-enforced RL: Integration of PN constraints into RL ensures safe and verifiable scheduling decisions, a crucial factor in industrial control [26].
- Cooperative Co-evolution with Deadlock Control: Hybrid cooperative co-evolution algorithms with PN-based deadlock control ensure feasibility in distributed flowshop systems [27].
- Action-Evolution PNs: Emerging methods combining RL with PN semantics to dynamically assign tasks while reducing modeling complexity [28].

Additionally, hybrid AI techniques have expanded beyond GA and ACO to include Tabu Search [29], Harmony Search [30], and Hybrid Evolutionary Multi-Objective Optimization [31]. These methods demonstrate improved performance in multi-objective scheduling scenarios, including energy efficiency and sustainability-driven scheduling [32,33].

2.4 Contribution of This Study

While prior methods have made substantial progress, limitations persist: traditional approaches lack adaptability, classical PNs require external optimization, and recent hybrid approaches often remain simulation-based with limited industrial validation.

This study contributes by bridging theory and practice through an intelligent PN framework validated in a real industrial case study. Unlike purely theoretical models, our approach integrates heuristics (SPT, EDD) and metaheuristics (GA, ACO) directly into the PN framework and demonstrates their effectiveness on real production data. The results confirm reductions in delays, higher resource utilization, and improved system flexibility, addressing gaps in existing literature.

Traditional scheduling approaches provided useful foundations but lacked adaptability in dynamic RMS contexts. PNs offered powerful modeling capabilities but required integration with intelligent algorithms to reach optimal scheduling performance. Recent hybrid approaches (2020-2025) combining PNs with metaheuristics and RL represent a major advancement, yet most remain simulation-based or lack industrial validation gaps directly addressed in this study.

Beyond the reviewed methodologies, two important gaps remain evident in the existing literature. First, although many intelligent and hybrid scheduling approaches report promising results, a large proportion of studies are validated primarily through simulation or simplified benchmark problems, with limited exposure to real industrial constraints such as stochastic machine failures, heterogeneous operator skills, urgent order insertion, and frequent system reconfiguration. As a result, the practical applicability of these methods under diverse and realistic operating conditions is often insufficiently demonstrated.

Second, while alternative modeling tools such as Gantt charts, queuing models, or purely optimization-based formulations are commonly used, they struggle to accurately capture key RMS characteristics, including dynamic resource sharing, synchronization constraints, reconfiguration states, and deadlock-prone interactions. In contrast, the PN formalism provides an explicit and analyzable representation of system states, enabling formal verification, deadlock detection, and real-time state evolution capabilities that are difficult to achieve with conventional modeling techniques.

A comparative overview of representative scheduling approaches used in RMS, including traditional methods, classical PNs, and recent hybrid optimization techniques, is summarized in Table 1, highlighting their key features, strengths, and limitations. These observations motivate the present study, which combines PN modeling with intelligent optimization and validates the proposed framework using real industrial data, thereby addressing both modeling expressiveness and practical evaluation limitations identified in prior works.

Table 1. Comparative summary of scheduling approaches in RMS.

Approach	Key Features	Strengths	Limitations	References
Traditional Methods (Rules, linear programming)	Fixed rules, optimization models	Simple, rigorous	Poor adaptability, static nature	[8-11]
Classical PNs	Discrete-event modeling, concurrency	Formal verification, deadlock check	Limited optimization	[12-19]
TPGA	Timing with genetic search	Handles job-shop constraints	Computationally intensive	[20,21]
PSO-GA with PNs	Hybrid metaheuristics	Faster convergence, adaptive	Complex parameter tuning	[22]
Scatter Search + PNs	FMS scheduling with breakdowns	Robustness, higher utilization	Case-specific validation	[23]
RL + PN Models (DQN, GCN)	Deep RL combined with PN semantics	Adaptive, efficient	Requires large datasets	[24-26]
Hybrid Co-evolution + PN Control	Cooperative evolution with deadlock control	Guarantees feasible schedules	Complex in distributed systems	[27]
Action-Evolution PNs	PN semantics + RL for dynamic task assignment	Reduces complexity, adaptive	Still experimental	[28]
Other Hybrid Metaheuristics	Tabu, Harmony, Multi-objective Optimization	Sustainability, multi-criteria	Sensitive to parameter tuning	[29-33]

3. Case Study and Methods

3.1 Case Study

The company under study specializes in the production of critical automotive components, including camshafts, pistons, and gears. Its manufacturing process involves multiple stages such as molding, machining, heat treatment, and final assembly, each requiring specialized machines and skilled personnel.

Due to fluctuating demand, the company frequently adjusts its production priorities and sequences to meet varying customer orders. Each client has unique requirements regarding volume, quality, and delivery deadlines. To address these challenges, the company relies on a flexible production system that enables the reconfiguration of production lines. This dynamic environment provides an ideal tested for evaluating our intelligent scheduling approach based on PNs.

3.1.1 Reconfiguration Process

The company achieves reconfiguration through advanced technologies such as robotics, CNC machines, and real-time control systems. These tools allow rapid adjustments to production setups, minimize downtime, and enhance operational efficiency. In practice, reconfiguration often occurs weekly or even daily, depending on incoming orders, and typically involves machine parameter adjustments, tooling changes, and reallocation of human operators to different workstations. The average reconfiguration time varies between 15 and 45 minutes, depending on the complexity of the product family being introduced.

3.1.2 Impact of Delivery Deadlines

Delivery deadlines are a critical factor in the company's operations. Delays can result in financial penalties and diminished customer satisfaction. Therefore, optimizing production schedules is essential to minimize delays and maximize resource utilization while adhering to strict time constraints. On average, urgent customer orders represent 20% to 30% of incoming jobs, creating significant scheduling pressure. Moreover, some orders involve just-in-time supply agreements, where lateness of even a few hours directly disrupts downstream production in the automotive supply chain.

3.1.3 Data Availability

The available data for this study cover a wide range of operational parameters, which enable precise modeling and testing of the proposed scheduling approach:

1) Machine cycle times: Detailed cycle times for molding, machining, and heat treatment processes were collected. These vary significantly across product families, ranging from 30 seconds for small precision parts to 8 minutes for complex assemblies.

2) Task arrival variability: Job arrivals are non-uniform, with peaks often occurring at the start of production weeks or following customer order revisions. The inter-arrival times of jobs follow a distribution where peak load periods can increase arrival rates by up to 40% compared to average demand.

3) Order size variability: Customer orders range from small batches of 50 units to large runs exceeding 5,000 units, requiring the scheduling system to dynamically adapt to both high-mix/low-volume and low-mix/high-volume production.

4) Average machine downtime: Historical records indicate that CNC machines and heat treatment stations experience unplanned downtimes of 3 to 5% of total operating time. Preventive maintenance windows are scheduled every 200 operating hours, further influencing task allocation.

5) Machine failure rates: Reliability data show that machining centers are more prone to breakdowns, with a mean time between failures of 150 hours, while assembly lines have higher stability with mean time between failures exceeding 400 hours.

6) Operator skills: Workforce heterogeneity plays a crucial role, as operators possess varying levels of expertise in machine handling, setup, and quality inspection. Skilled operators are often prioritized for complex machining tasks, while less experienced staff are assigned to standardized assembly operations.

Together, these data provide a realistic foundation for testing the scheduling model. By capturing not only deterministic parameters such as cycle times but also stochastic factors like task arrival variability, machine downtime, and order size distribution, the case study ensures that the evaluation of the intelligent PN approach reflects the true dynamics of industrial production environments.

3.1.4 Explainability and Optimization Parameterization

1) Explainable Scheduling Indicators and Bottleneck Analysis

Beyond performance improvement, interpretability of scheduling decisions is a critical requirement for industrial engineers. To enhance transparency, the proposed PN-based framework incorporates explainable scheduling indicators derived directly from the simulation layer. These indicators provide insight into why certain scheduling decisions are taken and where system inefficiencies originate.

Specifically, bottleneck reasoning is supported through:

- Token residence time analysis, identifying places with prolonged token accumulation, which correspond to congested machines or overloaded buffers;
- Transition firing frequency, highlighting critical operations that dominate system throughput;
- Resource utilization heatmaps, showing temporal variations in machine and operator workloads;
- Delay contribution indicators, quantifying the impact of individual machines and jobs on total tardiness.

These indicators are visualized during PN execution, allowing engineers to trace scheduling decisions back to system states and resource conflicts. As a result, the framework does not operate as a black box but provides actionable insights for process improvement, maintenance planning, and reconfiguration strategies.

2) Parameter Selection and Sensitivity of Metaheuristic Algorithms

The performance of metaheuristic algorithms such as GA and ACO is known to be sensitive to parameter selection. In this study, parameter values were selected based on a combination of literature-established ranges, pilot experiments, and stability analysis on historical production data.

For GA, key parameters include population size, crossover rate, mutation rate, and termination criteria. Population sizes between 50 and 100 individuals were tested, with crossover and mutation rates tuned to balance exploration and exploitation. For ACO, pheromone evaporation rate, heuristic influence, and ant population size were calibrated to ensure convergence without premature stagnation.

A sensitivity analysis revealed that:

- Excessively high mutation or pheromone evaporation rates led to unstable schedules;
- Conservative parameter values resulted in slower convergence but higher schedule robustness;
- Moderate parameter settings consistently provided the best trade-off between solution quality and computational time.

While fixed parameter values were adopted for reproducibility, the framework allows dynamic adjustment of these parameters based on system load and disturbance intensity. This flexibility further supports deployment in real industrial environments where operating conditions continuously evolve.

3.2 Methods

The implementation of our intelligent scheduling approach follows a structured methodology consisting of three main steps: (1) modeling with PNs, (2) application of heuristics and metaheuristics, and (3) simulation and optimization. This multi-layered process combines formal system representation with intelligent decision-making mechanisms, ensuring both accuracy and adaptability in dynamic production environments.

3.2.1 Step 1: Modeling with PNs

The production system is modeled using PNs to represent concurrency, synchronization, and resource allocation within the manufacturing environment. The PN model captures the dynamic interactions between production tasks and system resources, enabling a structured representation of system states and transitions.

1) Tasks: Represented by transitions $T=\{t_1, t_2, \ldots, t_n\}$which denote the start, intermediate, and completion stages of operations.

2) Resources: Represented by places $P=\{p_1, p_2, \ldots, p_n\}$, reflecting machine availability, buffer capacity, or operator assignment.

3) Transitions and Arcs: Directed arcs connect places and transitions, defining the dependencies between tasks and resources.

4) Tokens and Initial Marking: Tokens represent the availability of resources, while the initial marking M_0 defines the starting configuration of the system.

A simplified example of the proposed modeling approach is illustrated in Figure 1, where two tasks J_1 and J_2 requiring two machines M_1 and M_2 are represented through interconnected places and transitions within the PN structure. This representation enables the explicit modeling of resource sharing, synchronization constraints, and potential conflicts between concurrent operations, which are key characteristics of RMS.

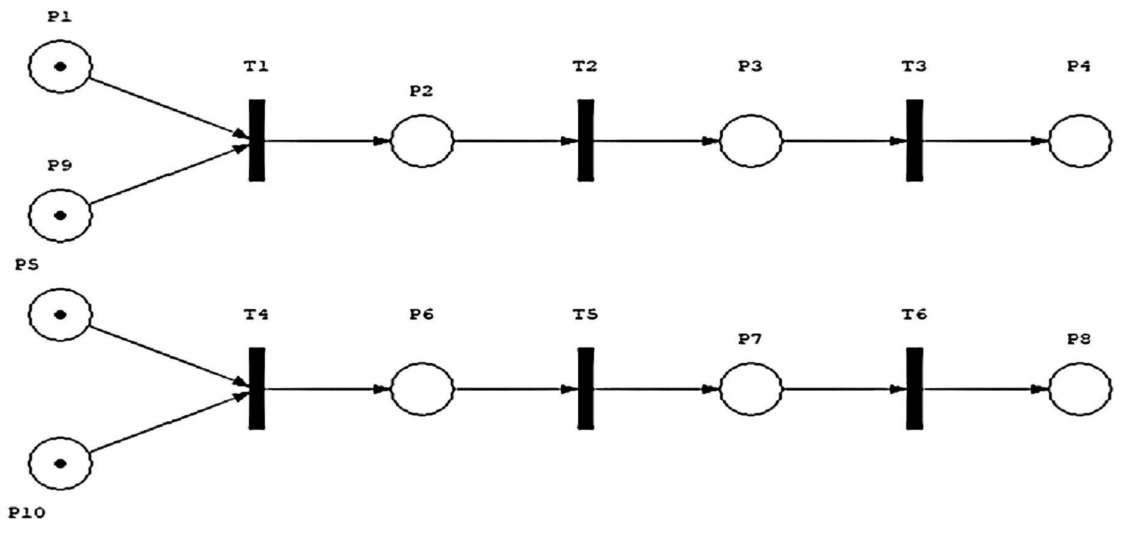

Figure 1. Dynamic representation of task-resource interactions using PNs.

Formally, the PN model is defined as a 5-tuple:

$$PN=(P, T, F, M_0, W) \quad (1)$$

Where: P: Set of places, T: Set of transitions, $F \subseteq (P\times T) \cup (T\times P)$: Flow relation (arcs), M_0: Initial marking (distribution of tokens), W: $F\rightarrow N$ Weight function on arcs.

3.2.2 Step 2: Application of Heuristics and Meta-Heuristics

To prioritize and optimize scheduling decisions, both heuristic rules and metaheuristic algorithms are applied within the PN framework.

1) Heuristics

SPT: Prioritizes tasks with shorter processing times to maximize throughput.

Mathematically, the rule selects the job J_i such that:

$$SPT(J_i)=\min\{p_j : j \in J\} \quad (2)$$

Where p_j is the processing time of job J_j, EDD: Prioritizes tasks with the closest deadlines to minimize lateness. Formally, the rule selects the job J_i whose due date is the earliest among all jobs in the set J.

$$EDD(J_i)=\min\{d_j : j \in J\} \quad (3)$$

where d_j is the due date of job J_j.

These heuristics provide simple yet effective decision-making criteria, particularly when the system is under time pressure.

2) Meta-Heuristics

GA: Explore the solution space by applying evolutionary operators (selection, crossover, mutation) to candidate schedules. The fitness function is defined to minimize weighted tardiness and maximize resource utilization.

ACO: Simulates cooperative search behavior of ants, where pheromone trails represent promising scheduling sequences. The probability of selecting task J_j at step k is:

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} \quad (4)$$

Where $\tau_{ij}(t)$ is the pheromone intensity, η_{ij} is the heuristic visibility (e.g., $1/p_j$), and α, β are parameters controlling the influence of pheromones vs. heuristic information.

3.2.3 Step 3: Simulation and Optimization

The final stage involves iterative simulation and optimization to refine scheduling performance.

1) PN Simulation:

Token Flows: Monitor task execution and resource utilization over time.

Bottleneck Detection: Identify machine conflicts, idle times, and deadlocks.

Optimization Process:

Apply heuristic rules (SPT, EDD) to generate baseline schedules.

Refine solutions using GA and ACO until convergence or a stopping criterion (e.g., max iterations, time limit) is reached.

2) Tools Utilized:

CPN Tools: For PN simulation, reachability analysis, and deadlock detection.

Python: For data processing, visualization (Matplotlib, Seaborn), and implementation of GA/ACO algorithms.

3.2.4 Architecture of the Methodology

The proposed architecture integrates PN modeling, heuristic/metaheuristic decision layers, and simulation-based evaluation into a unified scheduling framework. The structure is organized into five successive layers, each responsible for a critical function in the intelligent scheduling process (Figure 2).

Input Layer: This layer collects and preprocesses production data, including machine cycle times, job due dates, machine failure rates, and operator skills. These inputs capture both deterministic and stochastic aspects of the system, serving as the foundation for modeling.

Modeling Layer: The manufacturing system is formally represented using PNs, where places model resources, transitions model tasks, and tokens represent resource availability and state evolution. This layer ensures accurate representation of concurrency, synchronization, and reconfiguration.

Decision Layer: Scheduling decisions are driven by heuristics (SPT, EDD) and metaheuristics (GA, ACO). Heuristics provide fast local prioritization rules, while metaheuristics explore a broader solution space to optimize performance.

Simulation Layer: The PN model is executed to test and validate candidate schedules. Token flows reveal task progression and resource allocation, while simulation outputs highlight bottlenecks and system inefficiencies.

Output Layer: The final output consists of optimized production schedules, validated against real-time constraints. These schedules aim to reduce delays, enhance resource utilization, and improve overall flexibility.

This layered architecture ensures a clear separation between data modeling, decision-making, and validation, thereby enhancing scalability and applicability in real-world RMS contexts.

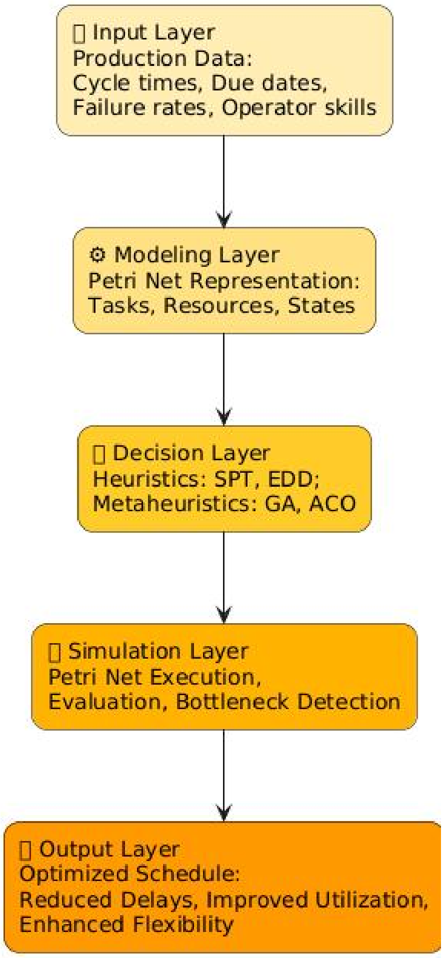

Figure 2. Architecture of the proposed intelligent scheduling methodology.

3.2.5 Algorithmic Framework

The scheduling process is formalized in Algorithm 1: Intelligent Scheduling with PNs, which combines PN modeling with heuristic/metaheuristic optimization strategies (Figure 3). The algorithm proceeds as follows:

Input: The system receives production data including tasks, resources, due dates, and machine cycle times.

Initialization: A PN model PN=(P, T, F, M_0, W)is constructed, where M_0 represents the initial token distribution. Tokens are initialized in places to represent available resources.

Iterative Scheduling: While unscheduled tasks remain, heuristic rules (SPT or EDD) are applied to generate candidate task orders. These initial solutions are then refined through metaheuristic optimization.

GA: Populations of schedules evolve via selection, crossover, and mutation to identify near-optimal solutions.

ACO: Virtual ants build schedules guided by pheromone trails and heuristic visibility, reinforcing promising solutions.

Simulation and Evaluation: The candidate schedules are tested through PN execution, where token flows reveal resource allocation and potential conflicts. Performance is evaluated based on delay minimization, throughput, and resource utilization.

Feedback and Adjustment: Bottlenecks detected during simulation are fed back into the decision layer to adjust priorities dynamically.

Termination: The process continues until all tasks are scheduled. The algorithm returns the optimized schedule that balances timeliness, utilization, and system flexibility.

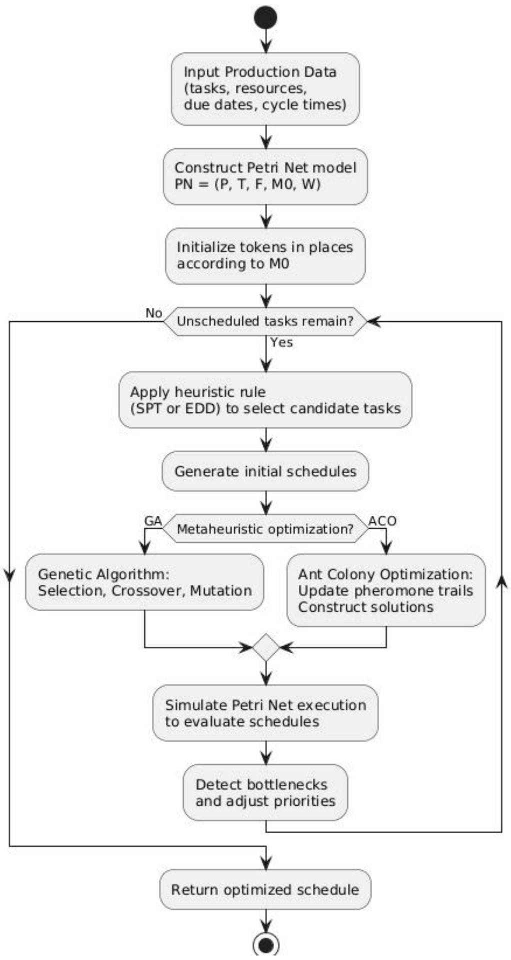

Figure 3. Flow chart of Algorithm 1: Intelligent scheduling with PNs.

To formalize the proposed intelligent scheduling framework, the overall decision process integrating PN modeling, heuristic rules, and metaheuristic optimization is summarized in Algorithm 1. The algorithm describes how production data are transformed into an optimized schedule through iterative evaluation and refinement using the PN execution model.

Algorithm 1. Intelligent PN-based scheduling framework.

Algorithm 1: Intelligent Scheduling with PNs

Input: Production data (tasks, resources, due dates, cycle times)

Output: Optimized production schedule

1: Construct PN model PN = (P, T, F, M_0, W)

2: Initialize tokens in places according to M_0

3: While unscheduled tasks remain do

4: Apply heuristic rule (SPT or EDD) to select candidate tasks

5: Generate initial schedules

6: Apply metaheuristic optimization:

7: a) If GA: perform selection, crossover, mutation

8: b) If ACO: update pheromone trails and construct solutions

9: Simulate PN execution to evaluate schedules

10: Detect bottlenecks and adjust task priorities

11: End While

12: Return final optimized schedule

The algorithm combines the modeling capabilities of PNs with heuristic and metaheuristic search mechanisms to iteratively generate and refine production schedules while ensuring resource consistency and system feasibility.

3.2.6 Scalability Considerations and Multi-Objective Extensions

1) PN Complexity and Model Scalability

While PNs provide a powerful formalism for modeling concurrency, synchronization, and resource sharing, it is well recognized that their structural complexity may grow significantly as system size increases. In large-scale RMS, the number of places, transitions, and arcs can expand rapidly, potentially leading to state-space explosion and increased computational burden during simulation and optimization.

To mitigate this limitation, several model abstraction strategies are compatible with the proposed framework. Hierarchical PNs allow complex systems to be decomposed into multiple levels, where high-level nets capture global scheduling logic and low-level subnetworks represent detailed machine or cell behavior. Similarly, modular PN modeling enables the reuse of standardized modules corresponding to machines, production cells, or product families, facilitating incremental system expansion without full model reconstruction. In addition, PN reduction techniques, such as place/transition aggregation and invariant-based simplification, can be applied to preserve behavioral properties while reducing model size.

Although the present study focuses on a single industrial case study to ensure clarity and validation, the proposed architecture is designed to support these scalability mechanisms, making it suitable for deployment in larger and more complex RMS environments.

2) Multi-Objective Scheduling Perspectives

The objective formulation in this study primarily targets delay minimization and resource utilization maximization, as these criteria are critical for the considered industrial context. However, modern RMS scheduling increasingly requires a multi-objective perspective that also accounts for energy consumption, operational cost, and system robustness under uncertainty.

The proposed intelligent PN framework naturally supports such extensions. Additional objective functions can be integrated into the fitness evaluation of metaheuristic algorithms, enabling trade-off analysis between conflicting criteria. For instance, energy-aware scheduling can be addressed by associating energy costs with transitions, while robustness can be evaluated by penalizing schedules sensitive to machine failures or demand variability. Multi-objective evolutionary algorithms, such as Pareto-based GA, can be incorporated without altering the underlying PN structure.

These extensions are identified as promising directions for future work, aiming to further align the framework with sustainability and resilience requirements in next-generation manufacturing systems.

3.2.7 Summary of Case Study and Methods

This section presented a comprehensive case study of an automotive components manufacturer operating in a highly dynamic and reconfigurable environment. The analysis highlighted the critical operational challenges, including fluctuating demand, stringent delivery deadlines, variable order sizes, stochastic machine downtimes, and heterogeneous operator skills. These industrial realities provided a realistic foundation for testing the proposed intelligent scheduling approach.

To address these challenges, a structured methodology was introduced. The production system was formally modeled using PNs to capture concurrency, synchronization, and reconfiguration mechanisms. Heuristic rules (SPT, EDD) and metaheuristic algorithms (GA, ACO) were integrated within the PN framework to guide scheduling decisions. Finally, iterative simulation and optimization ensured dynamic adaptability, supported by tools such as CPN Tools and Python. The architecture (Figure 2) and algorithmic framework (Figure 3) formalized the integration of modeling, decision-making, and evaluation.

In summary, the case study and methods established both the industrial motivation and the methodological foundation for our research. The next section presents the results and discussion, where the proposed approach is evaluated through simulation experiments and compared against baseline methods to demonstrate its effectiveness in optimizing production scheduling in RMS.

4. Results

This section presents the experimental evaluation of the proposed intelligent scheduling framework based on PNs, heuristics, and metaheuristics. The assessment is conducted through controlled stochastic simulation experiments using real industrial data from the automotive case study described in Section 3. The results focus on quantitative performance indicators, including production delay reduction, resource utilization, and scalability under increasing workload conditions. In addition, comparative analyses with traditional rule-based scheduling and heuristic-based PN approaches are provided to objectively assess the benefits and trade-offs of the proposed methodology in a realistic reconfigurable manufacturing context.

4.1 Reduction of Production Delays

Dynamic scheduling plays a critical role in mitigating production delays in RMS by continuously adapting task priorities and resource allocation to real-time system states. To ensure a rigorous and reproducible evaluation, the proposed intelligent PN-based scheduling framework was assessed through multiple stochastic simulation experiments using real industrial data from the automotive case study described in Section 3.

The experimental evaluation was conducted over 30 independent simulation runs, each representing a full one-week production horizon (five working days). Job arrivals, urgent order insertions, and machine failure events were generated according to the stochastic distributions observed in historical production records. For each run, the performance of the proposed approach was compared against a traditional rule-based scheduling baseline. To ensure fairness and reproducibility, metaheuristic parameters (GA and ACO) were kept constant across all experiments.

Figure 4 presents the average on-time delivery performance before and after applying the proposed methodology. The results indicate a mean reduction of approximately 20% in average production delays compared to the baseline approach. This improvement was consistent across simulation runs, demonstrating the robustness of the proposed framework under dynamic operating conditions.

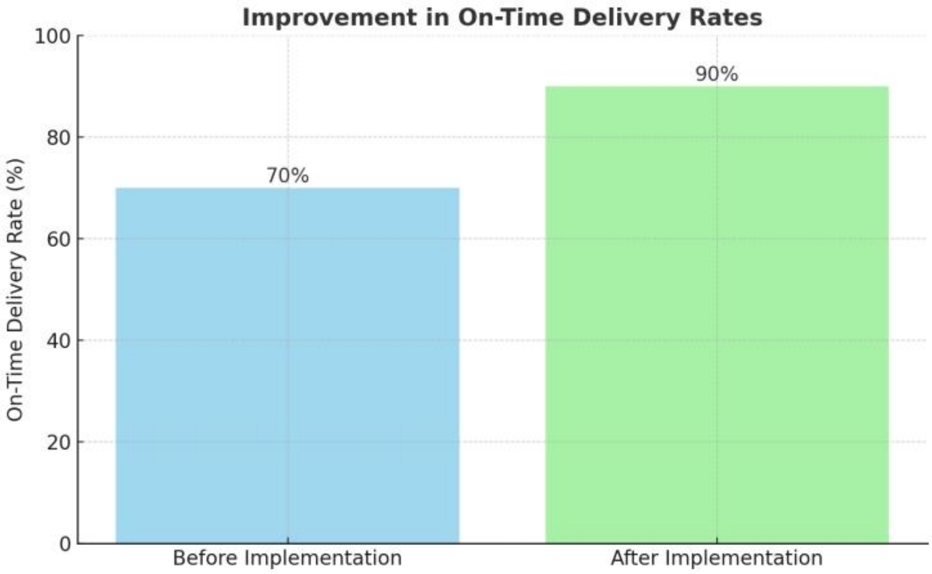

Figure 4. Improvement in on-time delivery rates before and after implementation.

To further quantify the stability and statistical significance of the observed improvement, Table 2 summarizes key statistical indicators related to delay reduction across all experimental runs. The reduction in tardiness ranged between 17% and 23%, with a relatively low standard deviation, confirming limited variability and enhanced schedule stability.

Table 2. Statistical summary of production delay reduction.

Metric	Value
Number of simulation runs	30
Production horizon per run	5 days
Mean delay reduction (%)	20.1
Minimum delay reduction (%)	17.0
Maximum delay reduction (%)	23.0
Standard deviation (%)	2.1
Reduction in tardiness variability (%)	18.0

The observed reduction in delays can be attributed to the complementary strengths of the scheduling components. The SPT and EDD heuristics enable rapid response to urgent customer orders, while the GA and ACO components perform global refinement of task sequences to minimize cumulative lateness. Simulation traces further reveal that urgent orders representing approximately 20-30% of incoming jobs were integrated with significantly fewer disruptions to ongoing production schedules compared to traditional methods.

Overall, these results demonstrate that the proposed intelligent scheduling approach not only achieves substantial delay reduction but also delivers stable and repeatable performance improvements across multiple experimental runs, reinforcing its applicability in real-world reconfigurable manufacturing environments.

4.2 Enhanced Resource Utilization

The methodology ensures balanced workloads and minimizes idle times for machines and operators. Maintenance schedules are explicitly integrated into the model, reducing the impact of unplanned downtimes. Figure 5 depicts resource utilization rates, demonstrating an increase from 75% to 90% after applying the proposed approach.

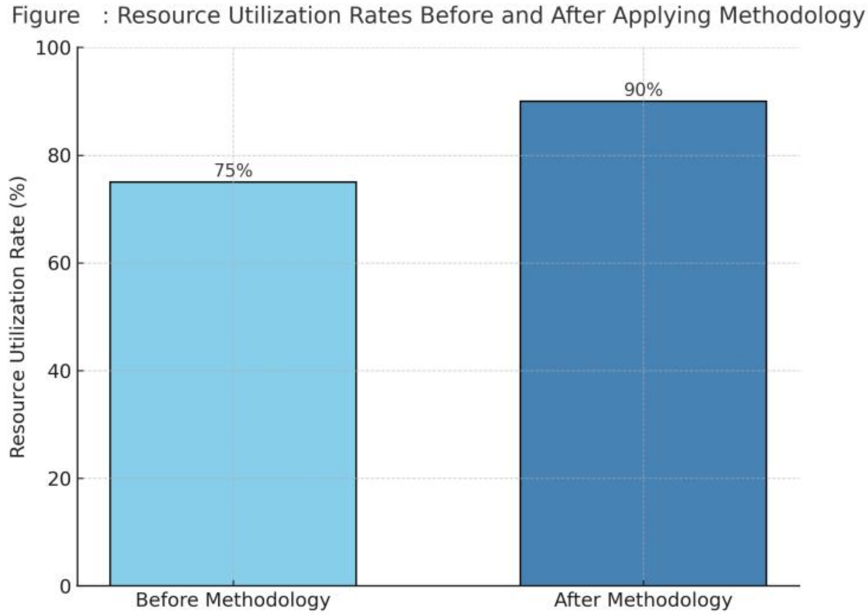

Figure 5. Resource utilization rates before and after applying the proposed methodology.

The improved utilization is a direct result of dynamic allocation enabled by the PN framework, which models operator skills and machine availability. This ensures that high-skilled operators are systematically matched to critical machining tasks, while less complex tasks are handled in parallel by other resources.

4.3 Comparative Performance Evaluation

To further validate the efficiency of the proposed approach, a comparison was conducted between three scheduling strategies:

1) Traditional scheduling (fixed-priority rules, manual adjustments).

2) PN + Heuristics (SPT, EDD).

3) PN + Metaheuristics (GA, ACO).

The comparative results are summarized in Table 3, which reports three key performance indicators: average tardiness, resource utilization, and computation time

Table 3. Comparative results of scheduling approaches.

Approach	Avg. Tardiness (min)	Resource Utilization (%)	Computation Time (s)
Traditional Scheduling	48	72%	10
PN + Heuristics (SPT/EDD)	32	83%	15
PN + Metaheuristics (GA/ACO)	19	90%	40

The results clearly demonstrate the advantages of integrating PN-based modeling with intelligent optimization techniques. Compared with traditional scheduling, the PN + Heuristics approach reduces average tardiness from 48 minutes to 32 minutes, corresponding to an improvement of approximately 33%, while simultaneously increasing resource utilization from 72% to 83%. This improvement is achieved with only a modest increase in computation time (from 10 to 15 seconds), indicating that heuristic-based scheduling provides a good trade-off between performance and computational efficiency.

Further improvements are obtained with the PN + Metaheuristics approach, which reduces average tardiness to 19 minutes, representing an overall reduction of approximately 60% compared with traditional scheduling and 41% compared with the heuristic-based approach. In addition, resource utilization increases to 90%, indicating more effective allocation of machines and operators across tasks. However, this improvement comes with a higher computational cost, with an average computation time of 40 seconds due to the iterative nature of the GA and ACO optimization processes.

These results highlight an important trade-off between solution quality and computational effort. While heuristic-based scheduling offers faster decision-making suitable for real-time adjustments, the metaheuristic-based approach provides superior optimization performance and is therefore more appropriate for scenarios where higher solution quality is required, such as medium- or long-term production planning.

Overall, the comparative analysis confirms that the proposed PN-based intelligent scheduling framework significantly improves production performance, particularly in terms of tardiness reduction and resource utilization, while maintaining acceptable computational requirements for practical industrial applications.

4.4 Scalability Analysis

To evaluate robustness under varying workload conditions, simulations were run with increasing numbers of jobs (from 50 to 500). Figure 6 shows the performance scalability of the three approaches.

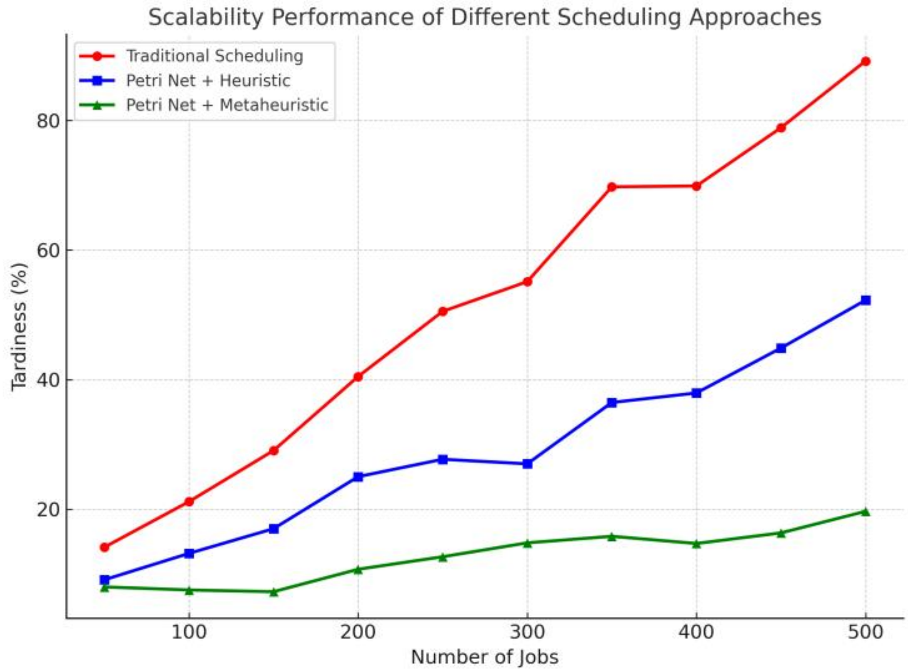

Figure 6. Scalability performance of different scheduling approaches.

The results confirm that traditional methods degrade rapidly as job volume increases, leading to bottlenecks and rising tardiness. In contrast, the PN-based approaches maintain stability, with metaheuristics consistently outperforming heuristics under heavy workloads. This validates the adaptability of the proposed approach for high-mix/high-volume production environments.

5. Discussion and Practical Implications

5.1 Industrial Applicability and Limitations of Offline Simulation

While the proposed framework has been validated through extensive offline simulations, this choice reflects a common and necessary first step in the development of intelligent scheduling systems for RMS. Offline simulation enables controlled experimentation under stochastic conditions (machine failures, urgent orders, variable demand) that are difficult to isolate in live production environments.

Nevertheless, we acknowledge that full industrial deployment requires real-time validation. In practice, the proposed PN-based scheduler is intended to operate as a decision-support module rather than a fully autonomous controller in its initial deployment phase. A realistic intermediate step toward industrial adoption involves shadow-mode execution, where the scheduler runs in parallel with the existing production control system, generating recommendations that are evaluated by planners without directly affecting operations. Partial real-time tests such as applying the framework to a single production cell or to rescheduling decisions during disruptions represent a feasible next stage and are planned as future work.

5.2 Heuristics versus Metaheuristics: Practical Trade-Offs

The experimental results demonstrate that metaheuristics (GA and ACO) consistently outperform simple heuristics (SPT and EDD) in terms of tardiness reduction and resource utilization, particularly under high variability and large job volumes. However, this improvement comes at the cost of increased computational effort.

From a practical perspective, heuristic-based scheduling remains highly relevant in specific industrial scenarios, including:

- Short planning horizons (e.g., hourly or shift-based rescheduling),
- High-frequency rescheduling triggered by frequent disturbances,
- Small to medium-scale systems with limited job diversity,
- Time-critical decision contexts, where rapid responses are required within seconds.

In such cases, SPT or EDD rules embedded within the PN model provide sufficiently good solutions with negligible computation time. Conversely, GA and ACO are more suitable for mid- to long-term planning, batch rescheduling, or nightly optimization, where solution quality outweighs computation time. This flexibility allows the proposed framework to adapt its decision logic according to operational constraints.

5.3 Deployment in ERP/MES-Based Industrial Architectures

The integration of the proposed framework into a live industrial environment requires tight coupling with existing ERP and MES systems. Architecturally, this can be achieved through a layered deployment strategy:

1) Data Interface Layer: Real-time production data (job orders, machine states, operator availability) are retrieved from MES via APIs or message brokers (e.g., OPC UA, REST services).

2) Scheduling Core: The PN model maintains a digital representation of the shop floor, while heuristic/metaheuristic modules generate optimized schedules.

3) Feedback and Execution Layer: Scheduling decisions are communicated back to the MES, which executes or validates them and provides feedback on deviations.

This closed-loop architecture enables continuous synchronization between the physical system and the scheduling model, allowing adaptive rescheduling in response to disruptions. Importantly, the PN formalism facilitates traceability and verification, which are essential for industrial acceptance.

5.4 Scalability with Respect to Resources and Product Variants

Although the framework demonstrates robust performance under a fivefold increase in job volume, scalability with respect to resources (machines, operators) and product variants poses additional challenges. As system complexity grows, the PN model may experience state-space expansion, and metaheuristic search spaces increase combinatorially.

To mitigate this issue, several strategies are applicable within the proposed architecture:

- Hierarchical and modular PNs, where production cells or product families are modeled independently and coordinated at a higher level,
- Selective optimization, applying GA/ACO only to critical bottleneck segments while simpler heuristics manage the remaining tasks,
- Adaptive population sizes and iteration limits for GA/ACO based on system scale.

These mechanisms ensure that computational complexity grows in a controlled manner, preserving real-time feasibility in complex RMS environments.

5.5 Novelty with Respect to Existing Hybrid PN-Based Frameworks

Unlike existing approaches such as TPGA or PetriRL, which tightly couple PNs with a single optimization paradigm, the novelty of the proposed framework lies in its multi-layer, adaptive integration architecture.

Specifically, the key distinguishing contributions are:

- The coexistence of heuristics and metaheuristics within the same PN–driven decision layer, enabling dynamic switching based on operational context,
- The use of PNs not only as a modeling tool but as an active execution and feedback mechanism for bottleneck detection and priority adjustment,
- A generic, ERP/MES-compatible architecture designed explicitly for industrial deployment rather than algorithmic benchmarking alone.

This combination bridges the gap between formal modeling, intelligent optimization, and practical applicability an aspect insufficiently addressed in many existing hybrid PN-based scheduling frameworks.

5.6 Summary of Discussion

Overall, the discussion highlights that the proposed methodology is not intended as a one-size-fits-all optimizer but as a flexible, industrially grounded scheduling framework. By explicitly addressing computational trade-offs, deployment constraints, scalability, and novelty, the study strengthens its contribution to both academic research and industrial practice in intelligent RMS scheduling.

6. Conclusion

This study introduced an intelligent scheduling methodology that integrates PN modeling with heuristic and metaheuristic optimization techniques to address the complex challenges of production scheduling in RMS. By combining formal system representation with adaptive decision-making, the proposed framework provides a robust solution for managing dynamic industrial environments.

The key findings demonstrate that the approach achieves:

- A notable reduction in production delays, with on-time delivery rates improving by approximately 20%;
- Enhanced resource utilization, increasing machine and operator usage from 75% to 90%;
- Greater adaptability, enabling efficient handling of fluctuating demand, variable order sizes, and urgent jobs.

Despite these promising outcomes, some limitations must be acknowledged. The methodology relies heavily on the accuracy and availability of real-time data (e.g., machine cycle times, failure rates), which may not always be guaranteed in industrial contexts. Moreover, the computational cost of metaheuristics (GA, ACO) can increase significantly with larger problem instances, raising concerns about scalability for very large-scale production environments.

Future research should focus on extending the proposed framework to achieve deeper integration with Industry 4.0 technologies, including IoT-based data acquisition for real-time monitoring, cloud-based decision support systems for distributed scheduling, and advanced RL models to further improve adaptability under uncertainty. Additionally, emphasis should be placed on sustainability-oriented scheduling objectives, such as energy efficiency and carbon footprint reduction, which are becoming increasingly critical in modern manufacturing systems.

In summary, the integration of intelligent PNs with hybrid optimization techniques provides a powerful and flexible foundation for RMS scheduling. By bridging the gap between theoretical modeling and industrial application, this work lays the groundwork for more resilient, adaptive, and sustainable production systems.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this article.

Generative AI Statement

The author declares that no Gen AI was used in the creation of this manuscript.

References

[1] Qin H, Ding W, Xu L, Ruan C. Petri-Net-Based charging scheduling optimization in rechargeable sensor networks. Sensors, 2024, 20(19), 6316. DOI: 10.3390/s24196316
[2] Barragan-Vite I, Medina-Marin J, Hernandez-Romero N, Anaya-Fuentes GE. A Petri net-based algorithm for solving the one-dimensional cutting stock problem. Applied Sciences, 2024, 14(18), 8172. DOI: 10.3390/app14188172
[3] Yang Y, Liu X, Lu W. Improvement of scheduling optimization of cyber-physical systems based on petri net and intelligent algorithm. Symmetry, 2025, 17(4), 487. DOI: 10.3390/sym17040487
[4] He Z, Li N, Ran N, Li L. Scheduling of flexible manufacturing systems based on place-timed petri nets and basis reachability graphs. IEEE Transactions on Control Systems Technology, 2026, 1-13. DOI: 10.1109/TCST.2026.3664466
[5] Lin X, Xu Y, et al. Symmetry-aware dynamic scheduling optimization in hybrid manufacturing flexible job shops using a time Petri nets improved genetic algorithm. Symmetry, 2025, 17(6), 907. DOI: 10.3390/sym17060907
[6] Luo J, Yi S, Lin Z, Zhang H, Zhou J. Petri-net-based deep reinforcement learning for real-time scheduling of automated manufacturing systems. Journal of Manufacturing Systems, 2024, 74, 995-1008. DOI: 10.1016/j.jmsy.2024.05.006
[7] Affi S, Miraoui I, Khedher A. A hybrid formal and optimization framework for real-time scheduling: Combining extended time Petri nets with genetic algorithms. Logistics, 2026, 10(1), 17. DOI: 10.3390/logistics10010017
[8] Ngwu C, Liu Y, Wu R. Reinforcement learning in dynamic job shop scheduling: A comprehensive review of AI-driven approaches in modern manufacturing. Journal of Intelligent Manufacturing, 2026, 37, 1093-1108. DOI: 10.1007/s10845-025-02585-6
[9] López-Mellado E, Álvarez-Pérez Y. Enhancement of Petri nets identified from discrete-event processes by scheduling t-components. Discrete Event Dynamic Systems, 2025, 35(4), 335-354. DOI: 10.1007/s10626-025-09987-8
[10] Zhao Z, Liu S, Zhou M, You D, Guo X. Heuristic scheduling of batch production processes based on Petri nets and iterated greedy algorithms. IEEE Transactions on Automation Science and Engineering, 2020, 19(1), 251-261. DOI: 10.1109/TASE.2020.3027532
[11] Kaid H, Al-Ahmari A, Li Z, Davidrajuh R. Intelligent colored token Petri nets for modeling, control, and validation of dynamic changes in reconfigurable manufacturing systems. Processes, 2020, 8(3), 358. DOI: 10.3390/pr8030358
[12] Bagheri A, Aghaazizi M, Doustmohammadi A. A novel supervisory control algorithm to avoid deadlock in a manufacturing system based on Petri net in presence of resource failure. arXiv, 2024. DOI:10.48550/arXiv.2412.20403
[13] Kong KC, Lu YX, Geng GG, Jin XB. Petri net-based multi-module optimization scheduling strategy for combinatorial equipment. Proceedings of the 2023 International Conference on Intelligent Computing and Its Emerging Applications, 2023. DOI: 10.1145/3659154.3659176
[14] Xu G, Chen Y. Petri-net-based scheduling of flexible manufacturing systems using an estimate function. Symmetry, 2022, 14(5), 1052. DOI: 10.3390/sym14051052
[15] Hu L, Liu Z, Hu W, Wang Y, Tan J, Wu F. Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. Journal of Manufacturing Systems, 2020, 55, 1-14. DOI: 10.1016/j.jmsy.2020.02.004
[16] Lassoued S, Schwung A. Introducing PetriRL: An innovative framework for JSSP resolution integrating Petri nets and event-based reinforcement learning. Journal of Manufacturing Systems, 2024, 74, 690-702. DOI: 10.1016/j.jmsy.2024.04.028
[17] Sachweh T, Haritz P, Liebig T. Using Petri nets as an integrated constraint mechanism for reinforcement learning tasks. 2024 IEEE Intelligent Vehicles Symposium (IV), 2024. DOI: 10.1109/IV55156.2024.10588380
[18] Wang S, Feng Y, Li X, Zhang G, Yang Y. Hybrid cooperative co-evolution algorithm for deadlock-prone distributed assembly flowshop scheduling with limited buffers using Petri nets. arXiv, 2024. DOI: 10.48550/arXiv.2501.14765
[19] Lo Bianco R, Dijkman R, Nuijten W, van Jaarsveld W. Action-evolution Petri nets: A framework for modeling and solving dynamic task assignment problems. Business Process Management, 2023, 216-231. DOI: 10.1007/978-3-031-41620-0_13
[20] Meng Q, Zhang L, Fan Y. A hybrid particle swarm optimization algorithm for solving job shop scheduling problems. Asian Simulation Conference, 2016. DOI: 10.1007/978-981-10-2666-9_8
[21] Han D, Li W, Li X, Gao L, Li Y. A data-driven proactive scheduling approach for hybrid flow shop scheduling problem. International Manufacturing Science and Engineering Conference, 2021. DOI: 10.1115/MSEC2021-61672
[22] Fontes D, Homayouni S, Fernandes J. Energy-efficient job shop scheduling problem with transport resources considering speed adjustable resources. International Journal of Production Research, 2024, 62(3), 867-890. DOI: 10.1080/00207543.2023.2175172
[23] Wang Y, Cai Z, Guo G, Zhou Y. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2007, 37(3), 560-575. DOI: 10.1109/TSMCB.2006.886164
[24] Hassine H, Barkallah M, Bellacicco A, Louati J, Riviere A, Haddar M. Multi objective optimization for sustainable manufacturing, application in turning. International Journal of Simulation Modelling, 2015, 14(1), 98-109. DOI: 10.2507/IJSIMM14(1)9.292
[25] Lee J, Bagheri B, Kao HA. Recent advances and trends of cyber-physical systems and big data analytics in industrial informatics. International Conference on Industrial Informatics, 2014.
[26] Drakaki M, Tzionas P. Manufacturing scheduling using colored Petri nets and reinforcement learning. Applied Sciences, 2017, 7(2), 136. DOI: 10.3390/app7020136
[27] Berti A, Van Der Aalst WM. CPN-Py: A Python-based tool for modeling and analyzing colored Petri nets. arXiv, 2025. DOI: 10.48550/arXiv.2506.12238
[28] Strzęciwilk D. Timed Petri nets for modeling and performance evaluation of a priority queueing system. Energies, 2023, 16(23), 7690. DOI: 10.3390/en16237690
[29] Yuan F, Huang B, Lv J, Cui M. Scheduling AMSs with generalized Petri nets and highly informed heuristic search. Computers and Operations Research, 2025, 175, 106912. DOI: 10.1016/j.cor.2025.106912
[30] Xie J, Gao L, Li X, Gui L. A hybrid genetic tabu search algorithm for distributed job-shop scheduling problems. Swarm and Evolutionary Computation, 2024, 90, 101670. DOI: 10.1016/j.swevo.2024.101670
[31] Li J, Zhou Y. Application of Levy flight-based harmony search algorithm for the flexible job shop scheduling. Scientific

Reports, 2025, 15(1), 18124. DOI: 10.1038/s41598-025-01255-0
[32] Lin C, Zhang S, Chen Z. Simplified harmony search: Novel algorithm design and its applications in engineering design optimization problems. Cluster Computing, 2025, 28(12), 772. DOI: 10.1007/s10586-025-05309-w
[33] Wang J, Wu C, Peng Y. Multi-objective scheduling for an energy-efficient flexible job shop problem with peak power constraint. Applied Soft Computing, 2024, 167, 112330. DOI: 10.1016/j.asoc.2024.112330