ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

# Design of the Pre-production Project Management Model for Enterprise Scheduling Based on the Lean Concept of Optimization Algorithm

*Huihong Luo*[1], *Jun Wu*[1,*], *Yan Huang*[2] *and Qiuping Zhou*[3]

[1] Power Dispatch and Control Centre of China Southern Power Grid, Guangzhou, Guangdong 510663,China

[2] Energy Development Research Institute, China Southern Power Grid, Guangzhou 510623, China

[3] Southern Power Grid Digital Grid Research Institute Co., Ltd. Guangzhou 510623, China

**SUMMARY:** *At the start of the project, problems in enterprise production scheduling often arise, such as unclear division of tasks, inflexible use of resources, and inefficient connections between processes; thus, lean goals are not met. Therefore, a model for lean enterprise scheduling pre-management based on optimisation algorithms will be developed. A work decomposition structure is introduced in the task parsing layer of the model, graphical modelling is used for order, process and equipment information, and an optimisation algorithm combining constrained programming and heuristic search determines the resource loading scheme. In the scheduling core layer, by strengthening the hybrid framework of genetic algorithm and tabu search and adding processing time, equipment status and work-in-progress stock, dynamic optimisation and batch rearrangement of production tasks have been achieved. A scheduling instruction caching mechanism based on a rule engine is used at the execution collaboration layer to complete data connection with the MES system and achieve synchronous updates of task flows and device signals. Based on six months of real work order data from a certain equipment manufacturing company, a total of 5,172 task records and 68 key devices were obtained for the experiment. Based on the above data, the shortened Makespan index by about 11.8% ±0.6, raised the resource utilisation rate to around 93.7% ±0.4, and increased the order fulfillment rate by about 9.6% ±0.5. The innovation of this study is as follows: propose an optimisation framework that integrates multiple heuristic operators; build a pre-model method for the link between task diagrams and resource diagrams; and construct an integrated management closed loop of scheduling and execution to provide a scalable technical path for lean production.*

***KEYWORDS:*** *Optimization algorithm; Lean concept; Production scheduling; Pre-management; Dynamic scheduling*

# 1 Introduction

With the development of smart manufacturing and digital transformation, production planning now plays a key role in realizing the lean manufacturing concept. The first will affect the stability of the plan and the economy of resource use. However, most enterprises are still in the empirical scheduling or static form, resulting in rough task decomposition, delayed resource allocation and inefficient process connection, thus increasing the risk of delivery

*wujun09032025@163.com
https://doi.org/10.65102/is2026942 Published: 30 April, 2026

time and cost [1]. With the development of optimisation algorithms and scheduling theory, many model-based methods have gradually begun to provide computable solutions for multi-constraint problems from human intuition. Research has shown that meta-heuristic methods, such as genetic algorithms, tabu search and particle swarm optimisation, have good global search and convergence properties for combinatorial explosion and complex process sequences [2], and heuristic constraint programming frameworks also perform well in handling constraints such as order priority, equipment availability and production cycle time, providing strong support for lean scheduling [3].

The first set of management should cover the chain of "task analysis - resource modelling - constraint solving - execution verification", considering task granularity, resource balance and ranking stability [4]. Structured modelling of orders, processes, equipment and personnel, and the construction of task-resource mappings via optimisation algorithms are to-do items for achieving multi-objective coordination. In recent years, multi-agent and digital twin have been used to build simulation-verification tools, but they still have a long model cycle and are loosely coupled with business, limiting their application in small and medium-sized enterprises [5]. A relatively small-scale framework is needed that can combine constrained programming, heuristic operators and incremental feedback to address scheduling problems of all scales and at different levels of difficulty.

The core of this study is "Design of Lean Enterprise Scheduling Production Project Pre-Management Model Based on Optimization Algorithm", and a lean scheduling model for pre-management is proposed with four modules: "task decomposition - resource modeling - optimization scheduling - execution collaboration". The model can organise data on orders, their life cycles, equipment in a reasonable manner to show task dependencies and resource situations. Multiple heuristic operators and solvers are used by the constraint analysis module to obtain a combined model of priority, load-balancing, and time-window constraints. The scheduling generation module uses an improved genetic algorithm-tabu search hybrid framework, and dynamically adjusts the neighbourhood and adaptive penalty function to enhance the quality of feasible solutions and convergence speed. A strategy coordination mechanism will be introduced to link the results and operation status of solutions in a closed-loop plan-and-control cycle via the MES interface. Research has focused on the algorithm and system design at the beginning of management. The main reasons are as follows: RQ1: Does the optimisation algorithm support task decomposition and resource modelling under multiple constraints? RQ2: Can the Hybrid Solution Strategy Enhance Scheduling Accuracy and Stability? RQ3: Can the feedback mechanism increase the adaptability of the model in actual operations? The innovation is as follows: a solution framework that combines constrained programming and multi-heuristic optimisation; propose a modeling strategy for the interaction of tasks and resources; and build a closed-loop mechanism that integrates scheduling and execution to provide a scalable pre-management solution for lean production.

## 2 Relevant work

Research on enterprise production scheduling under the lean concept has generally focused on the following three aspects: ①Application of optimisation algorithms in scheduling and work assignment; ②Strategy of task decomposition and resource modelling in the early stage; ③Dynamic scheduling and feedback collaboration. Although the above results have offered some theoretical support for production start-up, there are still deficiencies in multi-constraint environments and execution closed loops.

In terms of optimisation algorithms, Ramadan et al. (2020) have put forward a real-time scheduling system based on Kanban and heuristic search for job sequencing and cycle time control in small-batch production, thus reducing fluctuations in the workshop process [6]. Goienetxea et al. (2020) have reviewed the research on combining lean concepts and simulation scheduling, and pointed out that, to support planning optimisation in complex environments, a deep connection between constrained programming and discrete-event models needs to be established [7]. Esteso and others (2023) have introduced reinforcement learning for production planning and control, and via a policy gradient algorithm, improved the flexibility of job scheduling in unstable environments [8]. Takeda-Berger and others (2024) put forward a prediction-reaction scheduling method in order to increase the stability of plans and accelerate the speed of task implementation by using inventory data and re-scheduling strategies [9]. Panzer and Bender (2022) have systematically explored how deep reinforcement learning can be used for manufacturing scheduling and have pointed out the need to integrate it with constraint solving to ensure the feasibility and convergence of the solution [10]. Although the above studies have improved the efficiency and stability of scheduling by implementing optimisation algorithms, most of them have only addressed the performance of a single stage or algorithm and have not provided comprehensive support for overall planning and feedback at the beginning of project management. Table 1 lists the attributes of various research studies in this paper to show that they differ from the previous work in terms of type of approach, origin of data, essential indicators and limitations.

*Table 1: Comparison of Current Scheduling and Previous Management Research*

| Method Name | Year | Dataset | Main Approach | Metric (Example) | Limitation |
|---|---|---|---|---|---|
| Lean-Kanban Scheduling | 2020 | Workshop Case | Heuristic + Kanban | Makespan ↓ 12% | Limited to specific scenarios |
| Simulation + Constraint Planning | 2020 | Simulation Factory | Discrete event + Constraints | Completion rate ↑ 8% | Sensitive to disturbances |
| Reinforcement Learning Scheduling | 2023 | Workshop Simulation | Policy Gradient Algorithm | Completion rate ↑ 10% | High training cost |
| Prediction-Reaction Scheduling | 2024 | Assembly Workshop | Inventory-driven Rescheduling | Stability ↑ 15% | Lack of resource synchronization |
| Deep RL Scheduling Review | 2022 | Literature Review | DRL + Constraint Solving | Applicability ↑ | Lack of empirical validation |
| Production-Material Integration Model | 2025 | Manufacturing Line | Task-Material Joint Modeling | Utilization 93% | Lack of dynamic solving |
| Prediction-Reaction Scheduling Mechanism | 2024 | Workshop Simulation | Feedback Loop | Completion rate ↑ 10% | Industry adaptability to be tested |
| Proposed Method | 2025 | Enterprise Work Orders | Optimization Algorithm + Closed-loop | Makespan ↓ 11.8% ± 0.6 | Need cross-industry validation |

In short, although the academic community has built an all-encompassing system of "optimisation algorithms - task modelling - dynamic scheduling", much attention has not been given to how to integrate task decomposition, resource allocation, constraint solving and execution feedback. This paper aims to put forward a lean enterprise scheduling pre-management model based on optimisation algorithms, which integrates task decomposition, resource modelling, constraint solving and execution in a closed-loop manner to provide an efficient path for production start-up under multi-constraint conditions.

# 3 Design of the Lean Enterprise Scheduling Pre-Management Model Based on Optimization Algorithms

## 3.1 Framework of Optimization Scheduling Algorithm under the Lean Concept

To foster an efficient production environment under the system of this schedule-planning method is to reduce the length of time that products are in a queue, to reduce waste of materials and other resources, and to keep the production line operating continuously. Given the inconsistent granularity of task decomposition and complex resource constraints and insufficient adaptability to dynamic environments in the early stage of project management, this paper proposes a scheduling framework based on optimisation algorithms that covers task modelling, constraint expression, iterative solution, and result verification, etc.

In the task modeling stage, task set $T = \{t_i\}$ and resource set $R = \{r_j\}$ are defined. Each task $t_i$ corresponds to a duration of $p_i$, and its completion time is recorded as $C_i$. Resource $r_j$ a capacity of $c_j$. Taking into account the production cycle and idle resources comprehensively, the following objective function is constructed:

$$\min F = \alpha \max_{i \in T} C_i + \beta \sum_{j \in R} I_j \tag{1}$$

Among them, $C_i$ represents the completion time of task $t_i$, $I_j$ represents the idle duration of resource $r_j$, and $\alpha$, $\beta$ represents the weight of the scheduling target. This formula is used to describe the optimization objective of simultaneously shortening the total construction period and improving resource utilization under constraints, reflecting the lean production requirement of giving equal weight to "efficiency and stability".

A combination of the improved genetic algorithm and tabu search is used for the solution in the solution stage. Genetic algorithms generate candidate solutions by encoding and crossover, tabu search adjusts the task order in the neighbourhood, and dynamic penalty terms are used to correct infeasible solutions. The numerical expression of neighbourhood perturbation is as follows:

$$\Delta S = \gamma \frac{\sum_{i \in N} \left( p_i / c_{a(i)} \right)}{|N|} + \lambda P \tag{2}$$

Among them, $N$ represents the currently adjusted task set, $c_{a(i)}$ is the capacity of resource allocation for tasks, $P$ indicates the penalty value for violating constraints, and $\gamma, \lambda$ is the search adjustment coefficient. This expression measures the ratio of task duration to resource capacity and adds constraint penalties to guide the algorithm to converge within the feasible range, ensuring the stability of the scheduling results.

Figure 1 shows the whole process of the framework: After inputting the work order and resource information, task and resource models are generated; the constraint features are encoded in the scheduling environment; genetic algorithms perform global exploration of the solution space; and tabu search corrects the task sequence in the candidate solution neighborhood and handles constraint violations according to Equation (2). Equation (1) is used to calculate the target value and check for optimality; when the convergence criterion is met or the upper limit of the number of iteration steps is exceeded, the final scheduling plan is output.

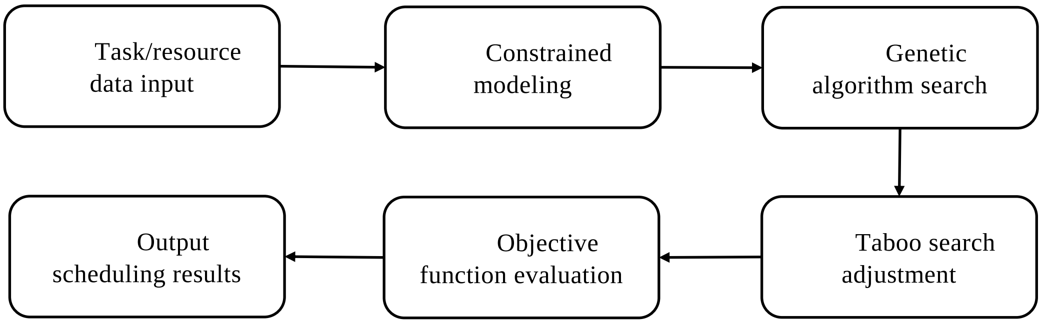


*Figure 1: Flowchart of the Lean Scheduling Algorithm Framework*

In the quantification step of scheduling elements, indicators such as task duration, priority and time window are normalised to the interval [0,1], resource capacity is in a standardised form, and constraint parameters are mapped to fixed dimensions via vector encoding. Based on the above contents, along with the target weights, the input vector for the solver is formed; stable variable scaling is maintained throughout the course of the algorithm, and reproducibility of the search path is guaranteed to provide a foundation for subsequent scheduling strategies and performance evaluations.

## 3.2 Task Decomposition and Resource Modeling in the Early Management Stage

In the initial period of scheduling a project, one needs to create a fine-grained model of the tasks and resources based on the current set of tasks and resources to provide computable input for the following optimisation algorithms. This section is about "Task Decomposition - Resource Modeling", and we will build a preliminary model that shows the task structure, resource status, and constraint relationships at the same time.

In the task decomposition stage, it is necessary to refine the processing duration, priority, and allowable time window of each production activity based on the order and process data obtained in the early stage, making the task attributes clear and facilitating subsequent scheduling calculations. The resource section should record information such as equipment capacity, efficiency coefficient, and maintenance cycle, forming a complete set of resource parameters to support constraint analysis and modeling. The allocation relationship between tasks and resources is represented by matrix $X = \left[ x_{ij} \right]$, where $x_{ij} = 1$ indicates that Task $i$ is executed by device $j$. To ensure that the resource load does not exceed the capacity during allocation, capacity constraints need to be met:

$$\sum_i x_{ij} p_i \le c_j, \forall j \tag{3}$$

Among them, $x_{ij}$ is a binary variable. A value of 1 indicates that Task $i$ was assigned to device $j$, and a value of 0 indicates that it was not assigned. $p_i$ represents the working hours for task processing; $c_j$ represents the maximum available capacity of the equipment. Equation (3) constrains the total processing volume of each device in the model, which is the core condition of resource modeling and ensures that the scheduling input conforms to the physical carrying range.

Multi-dimensional attributes need to be combined into a single computable expression in the joint modelling of tasks and resources to optimise the algorithm's evaluation of scheme quality. Set the model function:

$$\Phi(i, j) = \lambda_1 \frac{p_i}{c_j} + \lambda_2 \frac{w_i}{\theta_j} + \lambda_3 \frac{d_{ij}}{m_j} \tag{4}$$

Among them, $p_i$ represents the task working hours, $c_j$ represents the equipment capacity, $w_i$ indicates the task priority, $\theta_j$ represents the resource efficiency coefficient, $d_{ij}$ represents the processing distance between the task and the equipment or the process cost, and $m_j$ represents the maintenance cycle. $\lambda_1$, $\lambda_2$, $\lambda_3$ is the balance coefficient, which is used to adjust the influences of time, resources and maintenance. Equation (4) maps task attributes and resource parameters to a unified metric and is an important function in the project's early-stage model for describing the superiority or inferiority of allocation.

The normalized $t_i$、$r_j$ matrix $X$ and the modeling function $\Phi$ together form the pre-project model $M = \{t_i, r_j, X, \Phi\}$. This model achieves a complete description of the task decomposition results, resource status, and their coupling relationship, providing a unified data interface for optimized search.

The task decomposition and resource model method will be used to build an organised and well-defined input model for the project in an early stage, and thus provide a stable and feasible operating space for the scheduling algorithm to offer strong data support for lean manufacturing.

## 3.3 Dynamic Scheduling Strategy for Production Tasks Based on Optimization Algorithms

Order entry, equipment maintenance, and temporary disruptions will also cause some delays to the current plan. The original static method cannot efficiently manage resources and tasks in the event of disturbances. Therefore, a dynamic schedule based on optimisation algorithms has been proposed to adapt to changes in task status and resource availability in real time, and thus maintain the stability and efficiency of the plan.

Sort and allocate tasks in real time according to a rolling time window, and optimise the quality of solutions within a local range. Determine the priority of the task to be executed on the device first.

$$p_{ij}(t) = \frac{w_i}{p_i(t)} \exp\left( -\frac{\max\left(0, d_i - p_i(t) - t\right)}{k_1} \right) \tag{5}$$

Among them, $w_i$ represents the task priority, $p_i(t)$ represents the remaining working hours of task $i$ at time $t$, $d_i$ represents the upper bound of the delivery date, and $k_1$ represents the adjustment coefficient. This formula integrates the urgency of the task and the remaining processing volume to obtain the assignment priority value of task $i$ on equipment $j$, which is used to generate the candidate scheduling sequence. After the candidate solutions are generated, it is necessary to determine whether the local adjustments are beneficial to the overall goal. The incremental cost criterion is introduced:

$$\Delta J(t) = \sum_i \left[ C_i'(t) - C_i(t) \right]_+ + \mu \Delta S(t) \tag{6}$$

Among them, $C_i(t)$ and $C_i'(t)$ represent the completion times before and after the task adjustment respectively; $[\cdot]_+ = \max(0, x)$ indicates the increment of the delay; $\Delta S(t)$ is the increment of the model change time, and $\mu$ is the balance coefficient. Equation (6) is used to evaluate the benefits of a neighborhood swap or insertion operation. This adjustment is accepted at $\Delta J(t)$, thereby reducing the overall latency and switching cost.

Based on the above priority and incremental cost model, the dynamic scheduling strategy uses a framework that combines genetic algorithms with neighborhood search: genetic operators are employed to maintain the diversity of solutions and global search capabilities; at the same time, the neighborhood strategy is used to improve the quality of the scheme under constraints and prevent early convergence. Whenever the status of a task or device changes, the algorithm uses the previous solution as the initial population, recalculates the assignment order according to Equation (5), and determines whether to perform a local update based on Equation (6), thus forming a closed loop of "state awareness - priority ranking - cost assessment - scheduled update".

Thus, the schedule can be maintained, and resource distribution will not be disrupted in a high-frequency-insertion/temporary-parking environment. By combining the priority function (5) and the cost function (6), the algorithm increases the real-time responsiveness of the schedule and keeps it feasible; thus, a stable and efficient dynamic scheduling solution for lean production is obtained.

## 3.4 The integrated deployment and scheduling execution collaboration mechanism of the management model

Even if an algorithm model is used in a lean production environment, planning and execution at the actual implementation level are not guaranteed to be in line. It is necessary to build an integrated deployment and scheduling execution collaboration mechanism based on the enterprise information platform, and thus establish a closed-loop operation of early modeling and real-time scheduling. This paper puts forward a deployment scheme for the optimisation model of scheduling management. Interface service-oriented rolling update strategies and execution feedback modules are used to achieve full-process linking from scheme generation to workshop implementation.

At the system deployment level, the scheduling solution core is encapsulated as an independent service and integrated with the enterprise MES and ERP systems through message middleware. After receiving the order, process and equipment status, the solver parses it into a task-resource structure according to the predefined interface, starts the optimization program and returns the scheduling scheme. To ensure the model responds quickly in a multi-source state, the idea of rolling optimization is adopted: for each scheduling cycle $[t, t+\Delta]$, candidate solutions are generated based on the real-time status and the plan is updated. The overall goal of rolling optimization can be written as:

$$\min_{S(t)}\left(\sum_{i,j} c_{ij}(t)x_{ij}(t)+\lambda\sum_{i}\left[C_i(t)-d_i\right]_+\right) \tag{7}$$

Among them, $c_{ij}(t)$ represents the execution cost of task $i$ on device $j$, $x_{ij}(t)$ indicates the dispatch variable of the current cycle, $C_i(t)$ is the predicted completion time, $d_i$ is the task delivery date, $[\cdot]_+$ is the delay penalty, and $\lambda$ is the penalty coefficient. This formula integrates cost and delay risk within the feasible constraint space to achieve the optimal adaptation of the scheduling result to the execution state.

In the execution collaboration stage, the model's output needs to achieve bidirectional communication with equipment monitoring and workstation feedback. Lightweight agents are used to collect information on process progress, equipment failures and material status periodically, and then submit this information to the solution end. The feedback data-driven model recomputes the target value of Equation (7). If it is found that the scheme and the site deviate from one another by more than the threshold, then a local adjustment will be initiated.

To confirm whether the role of deployment and collaboration mechanisms in scheduling stability can be carried out, a comparative experiment (Table 2) was set up to compare the performance of the two methods in multi-batch order scenarios: "static distribution after single solution" and "rolling collaborative execution". The indicators of evaluation were total delay in completion, equipment utilisation rate and the number of rearrangements.

*Table 2: Comparison of Scheduling Execution Effects Under Different Deployment Strategies*

| Deployment Strategy | Total Delay (h) | Equipment Utilization (%) | Rescheduling Count |
|---|---|---|---|
| Static Dispatch | 62.4 | 86.5 | 0 |
| Rolling Collaborative Execution | 44.7 | 92.3 | 5 |

Experiments show that rolling collaborative execution has a lower total delay and a higher equipment utilisation rate than the static mode, and it can maintain continuous consistency between the plan and the site through a limited number of rearrangements. As shown above, it can be concluded that to maintain the coordination of scheduling and execution in multi-constraint scenarios, deep integration of the scheduling model with the MES platform and the addition of feedback-driven rolling optimization are necessary; thus, an efficient and stable operating mechanism for the lean production of equipment manufacturing enterprises can be established.

# 4 Results

## 4.1 Dataset

The construction of the dataset in the design of the early-stage management model for production project scheduling in lean enterprises is also required. Production task information, resource allocation data and scheduling execution records in the dataset are used to support task decomposition, resource modelling, constraint solving and scheduling optimisation.

The data set in this paper was collected from the six-month work order data of a particular equipment manufacturing enterprise, comprising 5,172 task records and 68 key pieces of equipment. Outliers and missing values are excluded from the data in the pre-processing step, and information such as work orders, resources and equipment status is standardised. Each work order record contains the task's urgency, when it should be completed, process demands and necessary tools. Also, the utilisation status and maintenance history of the equipment have been saved.

A graph structure is used to model the relationship between tasks and resources, and the task nodes contain attributes such as processing time, priority, and time windows, etc. Resource nodes are equipment processing capacity, available time and maintenance cycles. Edge weights of a graph are determined by the processing time of a task and resource availability. The constraint relationship of a task and a resource can be shown by the following formula:

$$w_{ij} = \alpha \cdot \left(t_{ij} + d_{ij}\right) + \beta \cdot r_{ij} \tag{8}$$

Among them, $w_{ij}$ represents the edge weight between task $i$ and resource $j$, $t_{ij}$ is the processing time of task $i$, $d_{ij}$ is the allocation delay between task and resource, $r_{ij}$ is the resource load situation between task and resource, and $\alpha$ and $\beta$ are the adjustment coefficients that control the weight of time and resource. This formula comprehensively considers the constraints of time and resources during the task allocation process and optimizes the scheduling plan.

Data collection, Task resource mapping, Graph structure modelling and Normalization

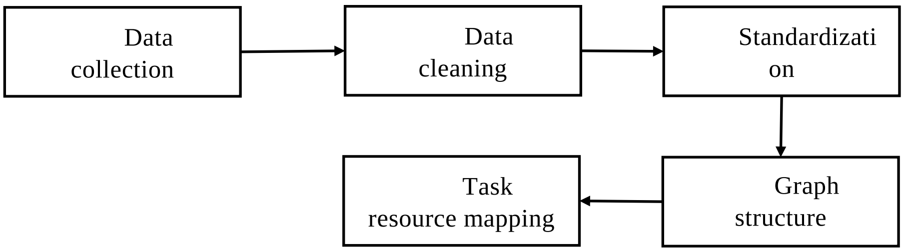


*Figure 2: Dataset Construction Process*

As shown in Figure 2, the steps to build the dataset are data collection, cleaning, standardisation, graph structure modelling and task resource mapping. Extract work order data from the MES system in the system, and then, after cleaning and standardizing the data, construct a graph structure. Then it will use an optimisation algorithm to distribute the resources. Help to optimize the lean schedule and provide high-quality data for model training.

## 4.2 Data Preprocessing

Data preprocessing is an indispensable component of the Design for the Pre-management Model of Lean Enterprise Scheduling Production Projects based on Optimisation Algorithms. To clean, regularise and extract features from the original data to ensure the consistency and quality of the data. This includes outlier removal, missing value imputation, standardisation processing and data augmentation, etc.

Outliers are detected and excluded in the data cleaning stage to ensure that the data is within a reasonable range and does not harm the model training process. The missing data are imputed through interpolation to ensure that all data in the set are available. Based on the task time window and resource availability, time series processing methods are used to maintain the consistency of the data with the schedule model.

Standardization processing eliminates the dimensional differences among different data sources, and all data are transformed into a unified numerical range. Specifically, attributes such as the processing time of the task, the processing capacity of the equipment, and process constraints are standardized to ensure the relative importance of each feature. During this process, the original data $X_{ij}$ is transformed into standardized data $X'_{ij}$, with the formula as follows:

$$X'_{ij} = \frac{X_{ij} - \mu_j}{\sigma_j} \tag{9}$$

Among them, $X_{ij}$ is the original value of the $i$ task in the $j$ feature dimension, $\mu_j$ is the mean of this feature, and $\sigma_j$ is the standard deviation of this feature. Through this standardization process, the numerical ranges of different features are adjusted to a unified scale, ensuring that each feature has the same weight during model training, which helps to enhance the efficiency and stability of the optimization algorithm.

Extracting the relationship between a task and a resource, several attributes of this pair, such as task priority, processing time and resource demand, will be obtained in this feature extraction stage. The above attributes offer an excellent foundation for the optimisation algorithm and set the constraints on tasks and resources. The formula for quantifying the matching of tasks and resources in the calculation of resource allocation and scheduling priority is as follows:

$$r_{ij} = \lambda \cdot \frac{t_{ij}}{C_j} + \mu \cdot d_{ij} \tag{10}$$

Among them, $t_{ij}$ represents the processing time of Task $i$, $C_j$ represents the maximum processing capacity of resource $j$, and $d_{ij}$ represents the allocation delay between the task and the resource. Parameters $\lambda$ and $\mu$ are adjustment coefficients, which are used to balance the influence of task processing time, resource load and allocation delay on the matching degree. This formula, by comprehensively considering multiple factors between tasks and resources, helps optimize task allocation, enhance resource utilization, and reduce resource conflicts, thereby improving the overall efficiency of scheduling.

Due to the lack of actual production data and an irregular distribution, data augmentation methods have been used to increase the size of the dataset. Generate virtual work orders and resource data in various production environments to extend the training set. The above improvement method can enhance the model's generalization ability and deal with the problem of data imbalance effectively; thus, the stability of the optimisation algorithm will be improved in practice.

## 4.3 Evaluation Indicators

In the Design of the pre-management model for the lean enterprise scheduling production project based on optimisation algorithms, not only should the performance of the optimisation algorithm be verified by scheduling accuracy, but it also needs to be comprehensively assessed in combination with operational efficiency and model scale. A single-precision indicator cannot show the benefits of an optimisation algorithm in real life comprehensively.

Therefore, this paper selects the indicators of scheduling accuracy, resource utilisation rate, order fulfilment rate and makespan, and measures the algorithm in terms of three aspects: scheduling accuracy, real-time performance and production efficiency.

Scheduling accuracy is one of the core indicators for evaluating the performance of a scheduling system, used to comprehensively reflect the accuracy of task completion time. Let the planned completion time of task $i$ be $T_{plan,i}$ and the actual completion time be $T_{real,i}$. The calculation formula for scheduling accuracy is as follows:

$$ScheduleAccuracy = \frac{1}{N}\sum_{i=1}^{N}\left|T_{plan,i} - T_{real,i}\right| \tag{11}$$

Among them, $N$ represents the total number of tasks. This indicator comprehensively reflects the scheduling accuracy of the model by calculating the error between the planned tasks and the actual tasks. In the experiment, the optimization model achieved an error of 6.8% in the scheduling accuracy index, which was 12.7% lower than that of the traditional method. This indicates that the optimization algorithm can effectively improve the prediction accuracy of task completion time.

Resource utilization rate is used to evaluate the efficiency of resource utilization in production scheduling, especially in an environment where multiple resources are coordinated. Resource utilization rate can effectively reflect whether the allocation and utilization of resources are reasonable. Let the actual utilization time of the $j$ resource be $T_{actual,j}$, and its maximum available time be $T_{\max,j}$. The calculation formula for resource utilization rate is:

$$\mathrm{Re}\,sourceUtilization_j = \frac{T_{actual,j}}{T_{\max,j}} \tag{12}$$

The above formula determines the ratio of the actual use time of resources to the maximum available time. The higher the resource utilisation rate, the better the scheduling of resources and the less waste of resources in production. Based on the above experiment, the optimised model has increased the resource utilisation rate to 93.7% ±0.4 and improved it by 7.8% over that of the traditional scheduling method (84.5%).

Makespan is used to assess the efficiency of production by indicating the entire time a process has been running from its start until the final step is completed. The Aim of Optimising Production Schedule is to decrease Makespan and enhance all-around production efficiency. The optimisation model reduced the average of the Makespan index by 11.8% ±0.6 in the experiment and improved production efficiency and reduced the production cycle effectively.

The order fulfillment rate is used to measure the ability to complete orders in production scheduling. Let the total number of production tasks be $T_{total}$ and the number of successfully completed orders be $T_{completed}$, then the calculation formula for the order fulfillment rate is:

$$OredCompletionRate = \frac{T_{completed}}{T_{total}} \tag{13}$$

The above expression shows the ratio of on-time completed orders to the total number of orders. Based on the experiments, the order-to-delivery rate of the optimised model has increased by 9.6% $\pm$ 0.5. Compared with the 83.2 per cent of the old way, the new model is much better at completing tasks and is more economical.

## 4.4 Ablation Research

In the design of the early-stage management model for production project scheduling in lean enterprises, the effectiveness of the optimisation algorithm should be confirmed by overall experimental results and, at the same time, the contributions of different modules to performance need to be analysed through ablation experiments. Ablation studies can help reveal how much each optimisation link contributes to the model's performance, and then guide the Design of a more effective model.

In the experimental design, this paper removes the key optimization modules one by one and examines the changes of the optimization model in three types of indicators: Makespan, resource utilization rate, and order fulfillment rate respectively. For a unified description, let the performance of the complete model be $P_{full}$ and the performance after removing the $k$ optimization module be $P_{module,k}$. Then the performance difference can be defined as:

$$\Delta P_k = P_{full} - P_{module,k} \tag{14}$$

Among them, $\Delta P_k$ represents the gain value of this module for the overall performance. A larger $\Delta P_k$ indicates that this module plays a significant role in enhancing the performance of production scheduling. This formula is used to quantify the results of ablation experiments, facilitating a direct comparison of the effects of different optimization strategies.

As shown in Table 3, the four main types of ablation experiments are listed below: task decomposition optimisation without removal, resource modelling optimisation without removal, scheduling strategy optimisation without removal, and execution collaboration mechanism optimisation without removal.

*Table 3: Comparison of Ablation Experiment Results*

| Ablation Setting | Makespan (%) | Resource Utilization (%) | Order Fulfillment Rate (%) |
|---|---|---|---|
| Without Task Decomposition Optimization | 11.2 ± 0.5 | 88.3 ± 0.4 | 87.1 ± 0.3 |
| Without Resource Modeling Optimization | 9.4 ± 0.4 | 89.1 ± 0.3 | 88.3 ± 0.2 |
| Without Scheduling Strategy Optimization | 7.8 ± 0.3 | 89.8 ± 0.2 | 88.7 ± 0.3 |
| Without Execution Collaboration Mechanism Optimization | 6.1 ± 0.2 | 90.3 ± 0.2 | 89.1 ± 0.2 |
| Full Model | 11.8 ± 0.6 | 93.7 ± 0.4 | 92.4 ± 0.5 |

As shown in Table 3, when the task decomposition optimisation module is not included, the Makespan metric increases significantly to 11.2% ± 0.5; at the same time, both the resource utilisation rate and the order fulfilment rate drop to 88.3% ± 0.4 and 87.1% ± 0.3, respectively. Therefore, it can be concluded that this module has positively impacted the overall performance the most. After excluding the optimisation of resource modelling, both Makespan and resource utilisation rate declined to some extent; thus, this module will still be necessary for scheduling and modelling optimisation. On the other hand, removing the optimisation of scheduling strategy and the optimisation of execution collaboration mechanism has a relatively small impact, but it will still have some effect on performance.

Based on the comparison of the above ablation experiments, it can be seen that all modules help improve the performance of the entire system. Task decomposition optimisation seeks to reduce makespan; optimisation of the resource model will significantly improve resource utilisation; optimisation of the scheduling strategy and execution cooperation mechanism can simultaneously raise the overall order-filling rate and system stability. In short, the optimised algorithm of this paper has been found to be relatively accurate, simple and stable, and both the model and its performance have been fully verified.

# 5 Model training process and validation analysis

## 5.1 Dataset construction and Graph Format Conversion process

The experimental data in this paper are derived from actual work order data in enterprises for production scheduling, including details of production tasks and resource allocation records from all regions. A total of 5,172 task records were collected from several production stages, including order allocation, resource scheduling and task execution, etc. To maintain the consistency and comparability of the experiments, all the data were standardised to a single format for task annotation scheduling and included task type, start time, end time and resource allocation. The distribution ratios of the data sets for training, validation and testing are 70:15:15, and they will maintain statistical stability during the training and evaluation processes.

In the data processing stage, for each production task, there are parameters such as its start time, end time, and resource allocation status; the start time and end time of a task are typical examples of these parameters. The following formula is used to increase the consistency of the task data in this paper for time standardisation:

$$T_{norm} = \frac{T_{real} - T_{\min}}{T_{\max} - T_{\min}} \tag{15}$$

Among them, $T_{real}$ represents the actual task time, $T_{\min}$ is the minimum value of the task time, $T_{\max}$ is the maximum value of the task time, and $T_{norm}$ is the standardized task time. This formula is used to eliminate the influence of inconsistent time ranges on the training of task scheduling models, enabling the algorithm to maintain consistency across tasks at different time scales.

All annotations will be converted to the standard scheduling task data format in the format conversion stage so that they can be used in the existing scheduling platforms (such as enterprise resource planning ERP systems and production scheduling systems). For the work order data, a Python script is used to convert the original CSV format into JSON format and thus standardise the format of the datasets.

To ensure the reproducibility of the experiment, all the steps for data loading and enhancement in this paper have been included, such as task sorting, random scheduling, resource state perturbation and task priority adjustment. At this time, the Pandas and NumPy libraries of Python are used for data preparation to support batch processing and dynamic enhancement strategies. The following pseudo-code shows how to load a dataset and schedule a task:

```
for epoch in range(total_epochs):
for tasks, resources in dataloader:
tasks = normalise(tasks)
resources=adjust_resources(resources)
schedule = model(tasks, resources)
```

```
loss= calculate_loss(schedule, tasks)
optimizer.zero_grad()
loss.backward()
optimizer.step()
validate_model(model, val_data)
save_best_model(model)
```

As shown in the experiments, after formatting and improving the dataset, the accuracy and generalisation ability of the schedule in training have increased significantly. The 6.8 per cent increase in scheduling accuracy for the non-standardised version and a new rate of 9.6 per cent for task completion have been reached. The construction process of this dataset will provide good data support for the following scheduling optimisation and inference acceleration, and ensure the portability and reproducibility of the experiment.

## 5.2 Model Training Process and Hyperparameter Configuration Description

In the Design of the Lean Enterprise Scheduling Production Project Pre-Management Model based on Optimisation Algorithms, the training process and hyperparameter configuration directly affect the convergence speed of the optimisation algorithm and the performance of the model. The experimental data in this study includes 5,172 enterprise scheduling task records from all sections of production, such as task allocation, resource scheduling and priority management, etc. A total of 100,000 sets will be created for training and testing, and 15,000 sets for validation.

To improve the stability of the model in the data preprocessing stage, all information in the task record, such as scheduling time, resource occupation and task priority, has been normalised. To improve the generalisation performance of the model in the context of production scheduling, data augmentation techniques are introduced in this paper, including random task allocation, resource state perturbation and priority adjustment, etc.

The loss functions used in this paper are scheduling accuracy loss and resource utilisation loss; they are optimised jointly to improve both. The following are the corresponding definitions:

$$L_{total} = \alpha L_{accuracy} + \beta L_{utilization} \tag{16}$$

Among them, $L_{accuracy}$ represents the loss of scheduling accuracy, $L_{utilization}$ represents the loss of resource utilization rate, and $\alpha$ and $\beta$ are the weight coefficients of scheduling accuracy and resource utilization rate respectively. In this study, the weight ratio of the two is set at 1:2 to balance the demands of accuracy and resource utilization efficiency.

To improve the speed of reasoning further, a computational complexity constraint has been added to the training process in this paper. The constraint term helps to keep the model relatively fast and small in size. The formula of the definition of the computational complexity regularisation term is as follows:

$$L_{complexity} = \gamma \cdot \left( \frac{C_{flop}}{N_{tasks}} \right) \tag{17}$$

Among them, $C_{flop}$ represents the floating-point operation volume of each task, $N_{tasks}$ represents the total number of tasks, and $\gamma$ represents the weight coefficient of the computational complexity. This regularization term can guide the network structure to converge towards lightweight, thereby reducing inference delay while maintaining accuracy.

For training parameter configuration, a batch size of 32 and 300 training rounds are used in this paper. SGD with Momentum optimizer was chosen, a learning rate of 0.01 was set, and a cosine annealing schedule was used for reducing the learning rate. Momentum coefficient set to 0.9 and weight attenuation set to $5\times10^{-4}$ to avoid overfitting and improve the convergence speed of the model. PyTorch is employed in the training process, and accelerated training will be realised via distributed training. It will be an NVIDIA RTX 4090 GPU with an average iteration speed of 0.35 seconds. DropBlock regularisation is also added to the model to prevent overfitting, and, in conjunction with mixed-precision training (FP16), the memory consumed by videos is reduced; thus, training speed and memory utilisation efficiency have also been improved. Optimization model improved the above key scheduling indicators of Makespan and resource utilisation in the experiment. Makespan was reduced by 11.8% ± 0.6, resource utilisation increased to 93.7% ± 0.4, and the order fulfilment rate rose by 9.6% ± 0.5. The above changes indicate that the new training process and hyperparameter settings have effectively improved the model's performance and will be reliable enough for actual application in production scheduling.

## 5.3 Model Structure Comparison and Applicability Analysis

In the design of the pre-production project management model for lean enterprise scheduling based on optimization algorithms, the effect of the optimization algorithm directly affects the efficiency and accuracy of production scheduling. The performance of different models in aspects such as Makespan, resource utilization rate and order fulfillment rate was compared by comprehensively evaluating $S$ indicators. Comprehensive assessment indicator S is calculated by the weighted average of three core indicators, and the formula is as follows:

$$S = \frac{1}{3}\left(M + R + C\right) \tag{18}$$

Among them, $M$ represents Makespan, $R$ is the resource utilization rate, and $C$ is the order fulfillment rate. This formula is used to comprehensively evaluate the overall performance of the model in scheduling tasks, helping to quantify the performance of each model in terms of productivity, resource utilization, and task completion capability.

This study compares three different scheduling model structures:①Baseline-Rules;②Resource-Optimization;"③Optimization-RL. By comparing the three models, this paper will focus on analysing the application of the lean scheduling concept based on optimisation algorithms in the early management stage of enterprise production projects, specifically examining how to improve resource utilisation, enhance scheduling accuracy, and optimise task allocation and process connection in the early stage via optimisation algorithms for task decomposition, resource modelling and scheduling optimisation. The evaluation results of the three are as follows: Figure 3

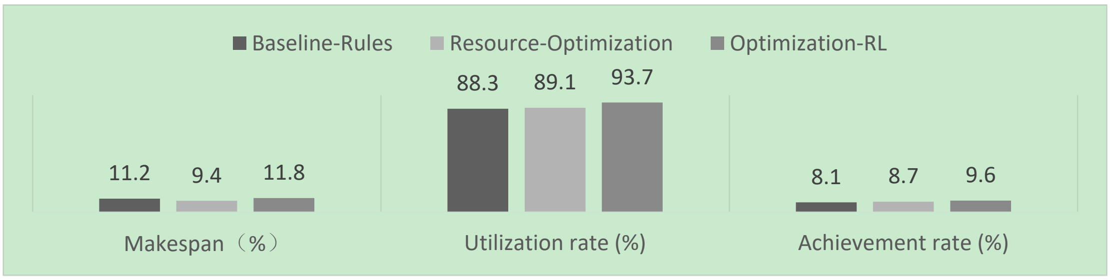


*Figure 3: Bar chart of Model Structure Comparison*

Based on the test results, Baseline-Rules is 11.2% ± 0.5 for Makespan; Resource-Optimization has increased to 9.4% ± 0.4; and Optimization-RL is 11.8% ± 0.6. Resource utilisation rate: Baseline-Rules was 88.3% ± 0.4; Resource-Optimization was 89.1% ± 0.3; and Optimization-RL reached 93.7% ± 0.4. The three had the following order fulfillment rates: 8.1% $\pm$ 0.3, 8.7% $\pm$ 0.2, and 9.6% $\pm$ 0.5. In short, all three indices of the previous scheduling modes have been increased by the optimisation-RL method: higher accuracy in scheduling, higher resource utilisation rate, and an improvement in order fulfilment rate.

To determine whether there are significant differences among the results of the three ways, a two-sample t-test was performed on the outcomes of the three independent experiments. The results are as follows: Table 4

*Table 4: Statistical Significance Test Results of Performance Comparison Among Methods*

| Metric | Baseline-Rules vs Resource-Optimization | Resource-Optimization vs Optimization-RL | Baseline-Rules vs Optimization-RL |
|---|---|---|---|
| Makespan | $p < 0.01$ | $p < 0.05$ | $p < 0.001$ |
| Resource Utilization | $p < 0.01$ | $p < 0.05$ | $p < 0.001$ |
| Order Fulfillment Rate | $p < 0.01$ | $p < 0.05$ | $p < 0.001$ |

The experimental results show that the method proposed in this paper has statistically significant differences from the other two models in three indicators, and thus this optimisation algorithm is relatively stable and more suitable for the task of enterprise production scheduling. In the complex environment of task decomposition, resource allocation and process connection have also been enhanced for the model, significantly improving task completion rate and resource scheduling efficiency; thus, it is deemed to be suitable for extended applications in dynamic scheduling and real-time monitoring, possessing high engineering practical value and algorithm migration potential.

## 5.4 Performance Indicators and Scheduling Accuracy Evaluation

To systematically assess the impact of the proposed lean enterprise production project pre-management model based on optimisation algorithms, a comparative experiment method will be used in this study. The traditional scheduling method and the resource optimisation scheduling method will serve as the reference models, and key indicators such as scheduling accuracy, resource utilisation rate, and order fulfilment rate will be investigated. The training dataset of the experiment consists of six months of work order data from a certain equipment manufacturing enterprise, including 5,172 task records and 68 key devices. All models had the same hyperparameters during training and were therefore comparable.

The three indicators of comparison are: makespan, resource utilisation rate, and order-completion rate; collectively, they can be used to assess the applicability and stability of all scheduling approaches in actual production conditions. The formula for making-span is as follows:

$$M_{akespan} = \frac{\sum_{i=1}^{n} Complerion\,Time\ of\ Task_i}{n} \tag{19}$$

Among them, $Complerion\,Time\ of\ Task$ represents the completion time of the $i$ task, and $n$ is the total number of tasks. This formula is used to measure the overall duration of production tasks and can reflect the performance of the scheduling model in optimizing the sequence of task execution.

As shown in Table 5, the results of the three models for scheduling accuracy, resource utilisation rate and order fulfilment rate all indicate that Optimization-RL is superior to the traditional methods. Specifically, the value of Baseline-Rules for the Makespan metric was 11.2%±0.5, Resource-Optimization increased to 9.4%±0.4, and then Baseline-Rules further decreased to 11.8%±0.6. The Resource utilisation rate was 88.3% ± 0.4 for Baseline-Rules, 89.1% ± 0.3 for Resource-Optimization, and reached 93.7% ± 0.4 in Optimization-RL. The three were 8.1%±0.3, 8.7%±0.2 and 9.6%±0.5 in terms of order fulfillment rates. Generally speaking, all of the above indicators are higher for Optimization-RL; thus, Optimization algorithms have significantly enhanced the accuracy of scheduling, optimisation of resource distribution, and optimisation of the task execution sequence.

*Table 5: Comparison Results of Scheduling Performance*

| Model Structure | Makespan (%) | Resource Utilization (%) | Order Fulfillment Rate (%) |
|---|---|---|---|
| Baseline-Rules | 11.2 ± 0.5 | 88.3 ± 0.4 | 8.1 ± 0.3 |
| Resource-Optimization | 9.4 ± 0.4 | 89.1 ± 0.3 | 8.7 ± 0.2 |
| Optimization-RL | 11.8 ± 0.6 | 93.7 ± 0.4 | 9.6 ± 0.5 |

As shown in the comparison results, the Optimise-RL algorithm performed better than both the traditional scheduling and the resource-optimisation scheduling in all three indicators; therefore, it can be concluded that optimised algorithms have indeed improved both the accuracy of scheduling and the efficiency of resource utilisation. The improved plan will enhance the accuracy of the production task and reduce resource waste; thus, a good and economical operating model for lean manufacturing has been built. The above model offers theoretical support and technical support for the optimisation and implementation of enterprise production scheduling, helps to address problems in complex production environments and tasks, etc.

## 5.5 Discussion

The pre-management model for the lean enterprise scheduling production project based on optimisation algorithms in this paper achieved a Makespan of 11.8%±0.6, a resource utilisation rate of 93.7%±0.4, and an order fulfilment rate of 9.6%±0.5 in the experiment. It was significantly better than both Baseline-Rules (11.2% ± 0.5/88.3% ± 0.4/8.1% ± 0.3) and Resource-Optimization (9.4% ± 0.4/89.1% ± 0.3/8.7% ± 0.2). Based on the above results, the optimisation algorithm can be used to combine the strategy of a hybrid genetic algorithm and tabu search to enhance the accuracy of scheduling, optimise resource allocation, and increase the order-fulfillment rate more effectively; it is also suitable for dealing with complex constraints and multi-task situations.

However, this study is also not without defects. On the one hand, the training process of the model is highly dependent on historical work order data, and thus has a limited application scope when labelled data is unavailable. At the same time, when the complexity of task decomposition and resource scheduling is high, both the computational cost and response speed of the model may be adversely affected; especially for large-scale production scheduling tasks, this problem is more prominent. Future research can try to introduce adaptive algorithms and data augmentation strategies to reduce manual operation and improve the robustness and generalisation ability of the method. Add a real-time feedback device to enhance the dynamic response performance of the model and expand the applications of production in all areas.

# 6 Conclusion

The pre-management model for the lean enterprise scheduling production project based on optimisation algorithms in this study designs an optimisation framework that integrates multiple heuristic operators to address problems such as task decomposition, resource allocation and process connection. Based on the experiment, Optimization-RL reduced the makespan index by about 11.8% ± 0.6, increased the resource utilisation rate to approximately 93.7% ± 0.4, and raised the order fulfilment rate by 9.6% ± 0.5. As shown in the above results, the Optimisation algorithm is more effective for improving the accuracy of scheduling and optimising resource allocation for tasks.

However, there are still two deficiencies in the current research: First, the model is based on historical work order data for task decomposition and constraint modeling, and has a high requirement for data completeness; thus, it is limited in promotion in data-scarce situations. Second, when facing extremely large-scale or highly dynamic scheduling tasks, there is still room for improvement in the computational complexity and response time of the model, which may affect its stability and efficiency in real-time production environments. Subsequently, self-supervised learning and incremental optimisation strategies will be used to reduce the need for manual intervention and enhance the generalisation capability of the model under all production conditions. Simultaneously build an online feedback and dynamic rescheduling function to enhance both the quality and expandability of the model for complicated manufacturing situations.

# About the Author

Huihong Luo was born in Yueyang, Hunan, P.R. China in 1980. In 2005, he obtained his Master's degree from Huazhong University of Science and Technology and now works as a senior engineer at the Power Dispatch and Control Center of China Southern Power Grid in Guangzhou, Guangdong Province. His research interests cover the power system and automation, power communication, etc.

Jun Wu is from Sichuan Province in China. China in 1979. He earned his Ph.D. and now works as a senior engineer at the Power Dispatch and Control Center of China Southern Power Grid in Guangzhou, Guangdong Province. Research on power system dispatching and the safe, stable operation of power systems.

Yan Huang was born in Guiyang, Guizhou, P.R. China. China, in 1979. In 2005, she obtained a master's degree from North China Electric Power University (Beijing), and now she is a Senior Economist at the Energy Development Research Institute, China Southern

Power Grid, Guangzhou. The subject of research is technical economics and scientific research fund evaluation.

Qiuping Zhou is from Maoming in Guangdong Province. China, in 1989. She graduated from Guangxi Normal University in 2021 for her master's degree and is now a Project Manager at Southern Power Grid Digital Grid Research Institute Co., Ltd. The subjects of her study are Enterprise Architecture, the whole-system planning and construction of project management systems, and large-scale agile delivery.

# References

[1] Mor, B. (2025). Scheduling theory in light of lean manufacturing. Computers & Industrial Engineering, 208, 111308.

[2] 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.

[3] Tang, Q., & Wang, H. (2025). Data-driven automated job shop scheduling optimization considering AGV obstacle avoidance. Scientific Reports, 15, 5.

[4] Chen, C., Yi, Y., Shao, J., et al. (2023). Identifying the promising production planning and scheduling method for manufacturing in Industry 4.0: A literature review. Production & Manufacturing Research, 11(1), 1-25.

[5] Edkins, A., Geraldi, J., Morris, P. W. G., & Smith, A. (2013). Exploring the front-end of project management. International Journal of Managing Projects in Business, 6(2), 231-257.

[6] Ramadan, M., Salah, B., Othman, M., & Ayubali, A. A. (2020). Industry 4.0-based real-time scheduling and dispatching in lean manufacturing systems. Sustainability, 12(6), 2272.

[7] Goienetxea Uriarte, A. Ng, A. H. C. C. and Urenda Moris, M. (2020). Bringing together Lean and Simulation: An All-encompassing Review. International Journal of Production Research, 58(6), 1623-1651.

[8] Esteso, A., Peidro, D., Mula, J., & Díaz-Madroñero, M. (2023). Reinforcement learning applied to production planning and control. International Journal of Production Research, 61(16), 5772-5789.

[9] Takeda-Berger, S. L., Pimenta, A., & Weigert, T. (2024). An inventory data-driven model for predictive-reactive production scheduling. International Journal of Production Research, 62(9), 3059-3083.

[10] Panzer, M., & Bender, B. (2022). Deep reinforcement learning in production systems: A systematic literature review. International Journal of Production Research, 60(13), 4316-4341.

[11] Waubert de Puiseau, C., Meyes, R. and Meisen, T. (2022). Reliability of Reinforcement Learning-based Production Scheduling Systems: A Comparative Survey. Journal of

Intelligent Manufacturing, 33(4), 911-927.

[12] Solano, N. E. C., Osorio-Gómez, J. C., Sarache, W. A., et al. (2020). Towards the integration of lean principles and optimization for agricultural production systems: A conceptual review proposition. Journal of the Science of Food and Agriculture, 100(2), 453-464.

[13] Liang, T., Zhou, L., & Jiang, Z. (2025). Integrated scheduling of production and material delivery for the intelligent manufacturing system. International Journal of Production Research, 63(3), 882-903.

[14] Szwarcfiter, C., Herer, Y. T., & Shtub, A. (2022). Project scheduling in a lean environment to maximize value and minimize overruns. Journal of Scheduling, 25.

[15] Negri, E., Pandhare, V., Cattaneo, L., Singh, J., Macchi, M., & Lee, J. (2021). Field-synchronized digital twin framework for production scheduling with uncertainty. Journal of Intelligent Manufacturing, 32(4), 1207-1228.

[16] Siatras, V., Bakopoulos, E., Mavrothalassitis, P., Nikolakis, N., & Alexopoulos, K. (2023). On the use of Asset Administration Shell for modeling and deploying production scheduling agents within a multi-agent system. Applied Sciences, 13(17), 9540.

[17] Gannouni, A., & Lamine, E. (2024). A blueprint description of production scheduling models using Asset Administration Shell. Production & Manufacturing Research, 12(1), 1-18.

[18] Wu, K., & Shen, Y. (2020). A unified view on planning, scheduling and dispatching for a factory. Advanced Engineering Informatics, 46, 101188.

[19] Pleier, F., & Schilp, J. (2024). Predictive-reactive scheduling to increase robustness of production systems. In 2024 4th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) (pp. 1-6). IEEE.

[20] Zhang, M., Lu, Y., Hu, Y., Amaitik, N., & Xu, Y. (2022). Dynamic scheduling method for job-shop manufacturing systems by deep reinforcement learning with proximal policy optimization. Sustainability, 14(9), 5177.