

A State-Updated Constraint Propagation Method for BIM–Digital Twin-Driven Local Rescheduling in Prefabricated Component Manufacturing

Jiankang Wu^{1,*}, Xiaozhu Li¹

¹College of Urban Construction, Hainan Vocational University of Science and Technology, Haikou 571137, China

Abstract

INTRODUCTION: Prefabricated component manufacturing requires adaptive rescheduling when equipment failures, delays, resource shortages, or order changes invalidate original production plans. Existing approaches often rely on predefined disturbances or global re-optimization and do not sufficiently connect BIM/IFC-derived constraints with digital-twin state feedback.

OBJECTIVES: This study aims to develop a stable local rescheduling framework that continuously maps BIM/IFC component relationships and digital-twin state deviations into dynamic constraints, affected task sets, rescheduling triggers, and local repair boundaries.

METHODS: A BIM-DT state-updated constraint propagation mechanism was designed to transform floor sequences, assembly dependencies, operation precedence, mold compatibility, and resource capacity into initial constraints, and then update them using equipment-state deviations. A constraint-driven local rescheduling module was further developed using task locking, local repair windows, state-triggered activation, and adaptive schedule-stability control. Reconstructed experiments were conducted using OR-Library scheduling instances and NASA C-MAPSS equipment-state data.

RESULTS: Across five repeated experiments, the proposed method achieved lower total tardiness, rescheduling cost, schedule deviation, and adjusted-task ratios than most baseline methods under reconstructed disturbance scenarios. Compared with the best-performing dynamic baseline in this comparison, ND3QNP, total tardiness decreased from 101.8 to 96.4 on OR-Library and from 131.7 to 126.5 on C-MAPSS. Ablation and representation analyses provided supporting evidence for the roles of constraint propagation, state triggering, local repair, task locking, and adaptive stability control.

CONCLUSION: The framework provides a reproducible constraint-driven local repair method for disturbance-oriented rescheduling in prefabricated component manufacturing, while future validation with real factory BIM-DT data remains necessary.

Keywords: artificial intelligence in manufacturing, digital twin, BIM/IFC, state-updated constraint propagation, local rescheduling, schedule stability

Received on 06 July 2026, accepted on 10 August 2026, published on 12 October 2026

Copyright © 2026 Jiankang Wu *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetsis.13902

*Corresponding author. Email: wjklxz@126.com

1. Introduction

Artificial intelligence is shifting manufacturing from static production planning to state-aware adaptive decision-

making. In intelligent manufacturing, digital twin, sensing, information modeling, and computational optimization are increasingly used to perceive production states, update constraints, and support disturbance-oriented rescheduling. With construction industrialization, prefabricated buildings

have become important for improving production efficiency and engineering coordination [1,2]. Prefabricated component manufacturing directly affects component delivery, on-site assembly rhythm, and project duration [3]. However, equipment failures, process delays, order changes, and resource shortages can weaken the feasibility of the original plan [4]. Therefore, updating production constraints, identifying affected tasks, and generating stable local rescheduling schemes are critical for adaptive production planning [5]. This study investigates how BIM/IFC-derived constraints and digital twin state deviations jointly affect the rescheduling feasible region and local repair boundary.

Existing studies can be broadly categorized into traditional scheduling, BIM/IFC-supported scheduling, digital-twin-based state tracking, and dynamic rescheduling. Traditional scheduling methods can optimize production sequences, mold allocation, and delivery times, but they make limited use of BIM/IFC component dependencies and assembly relationships [6,7]. BIM/IFC-based methods can support component information extraction and initial plan generation, but they mainly focus on static modelling [8]. Digital twin, Internet of Things (IoT), Radio Frequency Identification (RFID), and Ultra-Wideband (UWB) methods can support state tracking and production monitoring, but the coupling between state feedback and the updating of the rescheduling feasible region remains insufficient [9,10]. Dynamic rescheduling methods can respond to disturbances, but local adjustment boundaries, task locking, and schedule stability still need to be further integrated with component dependencies and state deviations [11].

To address these issues, this paper proposes a stable local rescheduling framework based on BIM-DT state-updated constraint propagation, forming a closed-loop mechanism of “constraint generation–state updating–task identification–local repair.” The purpose of this study is not to replace general-purpose scheduling optimizers, but to establish a continuous mapping mechanism among BIM/IFC constraints, digital twin state deviations, affected task sets, and local repair boundaries after disturbances occur. The BIM-DT State-Updated Constraint Propagation module transforms floor sequences, assembly dependencies, mold compatibility, and operation precedence relationships into initial production constraints, and then updates the rescheduling feasible region by incorporating state deviations to generate the affected task set and the rescheduling trigger signal. The Constraint-Driven Stable Local Rescheduling module determines the local repair window according to the updated constraints and affected tasks, and generates an updated production plan through task locking, delivery constraints, and schedule deviation control.

The main contributions of this paper are as follows. First, a BIM-DT state-updated constraint propagation mechanism is proposed to transform BIM/IFC-derived constraints and digital twin state deviations into a dynamic constraint set, a rescheduling trigger signal, and an affected task set. Second, a constraint-driven stable local rescheduling method is designed to achieve stable rescheduling within the updated feasible region through a local repair window, task locking, and schedule deviation control, thereby avoiding unnecessary

global adjustments. Third, a state-triggered and adaptive stability control mechanism is developed to determine rescheduling activation conditions by jointly considering state deviations, constraint violations, and tardiness risk, while dynamically adjusting the schedule stability weight according to the scope of affected tasks.

2. Related Work

2.1 Traditional Production Scheduling Methods for Prefabricated Components

Traditional production scheduling methods for prefabricated components mainly rely on operations research optimization, heuristic algorithms, and multi-objective scheduling models [12,13]. These methods primarily address issues such as component production sequencing, mold allocation, resource occupation, delivery time, and inventory cost. Existing studies usually model prefabricated component production as a single-machine scheduling, multi-production-line scheduling, flow-shop scheduling, or resource-constrained scheduling problem, and generate production plans through genetic algorithms, particle swarm optimization, hybrid heuristic algorithms, or mathematical programming methods [14,15].

These methods can handle operation sequences, resource capacity, and delivery requirements, thereby providing a fundamental optimization framework for prefabricated component manufacturing planning [16]. However, their inputs are mostly derived from manually organized orders, processing times, and resource parameters, while the use of component geometry, floor relationships, and assembly dependencies in BIM/IFC remains relatively limited [17]. When equipment states, component progress, or order requirements change, the constraint set in the scheduling model usually needs to be redefined or manually adjusted. Therefore, the linkage between state changes and production constraint updating still requires further strengthening [18,19].

2.2 BIM/IFC-Based Component Information Modeling and Scheduling Support

With the development of Building Information Modeling (BIM) and Industry Foundation Classes (IFC) standards, related studies have begun to extract component types, geometric dimensions, spatial positions, and component relationships from building models, and apply this information to prefabricated component design, production, and supply chain management [20,21]. Some studies have further transformed IFC-based component information into production scheduling inputs to reduce the information gap between design models and manufacturing plans [22]. Semantic enrichment can also infer missing joints, aggregations, and connection relationships from precast IFC models, improving their support for downstream production decisions [23].

BIM/IFC-based methods provide structured data sources for prefabricated component production, enabling component attributes, spatial relationships, and assembly logic to be incorporated into the production planning process [24]. However, existing studies focus more on information extraction, static data transformation, or initial plan generation, while paying relatively limited attention to state changes and disturbance response during the production process. BIM/IFC information is usually used as an input condition before scheduling, but how it can be transformed into updatable dynamic constraints in response to production-state changes still needs to be further embedded into post-disturbance rescheduling decisions [25].

2.3 Digital-Twin- and IoT-Based Component State Tracking

Digital twin, RFID, UWB, and IoT technologies provide new means for state perception and process monitoring in prefabricated building manufacturing [26]. Related studies acquire information on component location, processing status, transportation status, and progress through sensing, identification, positioning, and data-integration platforms, and then map these states to BIM or digital twin models [27,28].

These studies have promoted the transition from static information management to dynamic state perception, enabling component tracking, progress monitoring, and process synchronization. Existing engineering platforms mainly support synchronization, anomaly identification, visual management, dynamic tracking, and platform-based decision support [29]. More recently, a BIM-enabled digital-twin platform has been applied to precast production and yard-stock management [30], further demonstrating the practical integration path from factory sensing to platform-based decision support. Compared with these broader digital twin-enabled decision support systems, existing platforms do not explicitly transform state deviations into updated constraints, affected task sets, and local–global repair decisions. Therefore, a clearer decision-linking mechanism is still needed between digital twin state data and production rescheduling.

2.4 Disturbance-Oriented Dynamic Rescheduling Methods

Disturbances such as order changes, equipment failures, material delays, operation timeouts, and delivery adjustments frequently occur during prefabricated component production [31]. Therefore, dynamic rescheduling has become an important research direction. Related studies usually regenerate production plans after disturbances through demand-change modeling, machine-failure modeling, multi-production-line rescheduling, local adjustment strategies, or hybrid optimization models, with the aim of reducing tardiness, cost, and the extent of schedule adjustment [32].

Recent dependency-aware rescheduling represents operation constraints through directed graphs and restricts schedule revision to affected operations [33].

These methods have shown that dynamic rescheduling can improve the responsiveness of production systems to disturbances. However, most rescheduling models mainly rely on predefined disturbance events and production parameters, and rarely use BIM/IFC component dependencies and digital twin state deviations jointly to determine rescheduling boundaries. In particular, during local rescheduling, which tasks should be locked, which tasks should belong to the affected set, and how the adjustment window should be determined still need to be further integrated with component constraints and production-state feedback.

2.5 Research Gap and Positioning of This Study

In summary, existing studies have established a foundation in prefabricated component production scheduling, BIM/IFC information modeling, digital twin state perception, and dynamic rescheduling. However, under disturbance scenarios, there remains a lack of a continuous mapping mechanism among BIM/IFC-derived constraints, digital twin state deviations, and local rescheduling decisions. A closed-loop process of “constraint generation–state updating–task identification–local repair” has not yet been fully established.

Specifically, semantic digital-twin scheduling [18] connects production information with schedule updating but does not explicitly derive affected-task and local-repair boundaries from state-updated IFC constraints. BIM–RFID monitoring [27] captures production states without converting them into rescheduling constraints, while proactive and DRL-based precast scheduling [31,32] optimize production decisions without jointly using IFC dependencies and digital-twin deviations. Therefore, local adjustment boundaries, task locking, and schedule stability remain insufficiently coupled with component dependencies and real-time state feedback. The closed loop is not claimed as an entirely new BIM-DT architecture; rather, it is a mechanism-level extension that explicitly transforms IFC dependencies and digital-twin state deviations into updated constraints, affected-task sets, and bounded repair decisions. It thus complements existing monitoring and platform-level decision support by providing an executable rescheduling layer.

Unlike these studies, this paper focuses on state-feedback-driven rescheduling under disturbance scenarios in prefabricated component manufacturing and establishes a continuous mapping mechanism among BIM/IFC constraints, digital twin state deviations, affected task sets, and local repair boundaries. This paper develops two modules: BIM-DT State-Updated Constraint Propagation and Constraint-Driven Stable Local Rescheduling. The former generates the dynamic constraint set, the rescheduling trigger signal, and the affected task set; the latter determines the local

repair window and generates a stable rescheduling scheme through task locking, schedule deviation control, and the state-triggering mechanism.

3. Methodology

3.1 Problem Definition

This study investigates the state-feedback-driven rescheduling problem under disturbance scenarios in prefabricated component manufacturing. Given the original production plan π^0 , the component task set $\mathcal{J} = \{j_1, j_2, \dots, j_n\}$, the equipment/mold resource set $\mathcal{R}$, the BIM/IFC model M_{IFC} , and the digital twin state stream S_t^{DT} , the objective is to generate an updated production plan π'_t after a disturbance event e_t occurs. The input information includes component types, geometric dimensions, floor sequences, assembly dependencies, operation precedence relationships, mold compatibility relationships, equipment availability, component processing progress, resource occupation status, and delivery deadlines.

In this paper, the component relationships and production rules derived from BIM/IFC are represented as the initial constraint set $\mathcal{C}^0$, while equipment failures, operation delays, resource shortages, and order changes obtained from digital twin feedback are represented as state deviations ΔS_t . Disturbance events may change equipment availability windows, task start times, resource capacities, and task dependency states. Therefore, $\mathcal{C}^0$, ΔS_t , and e_t need to be jointly mapped to the updated constraint set $\mathcal{C}_t$, the affected task set Ω_t , and the local repair window W_t .

The optimization objective is to locally reschedule the affected tasks while satisfying the updated constraints $\mathcal{C}_t$, so that the production plan can balance delivery timeliness, rescheduling cost, constraint feasibility, and schedule stability. Specifically, the model minimizes total tardiness $D(\pi'_t)$, rescheduling cost $R(\pi'_t)$, constraint violation $V(\pi'_t, \mathcal{C}_t)$, and schedule deviation $S(\pi'_t, \pi^0)$:

$$\min_{\pi'_t} \mathcal{L}_{res} = \alpha D(\pi'_t) + \beta R(\pi'_t) + \gamma V(\pi'_t, \mathcal{C}_t) + \lambda S(\pi'_t, \pi^0) \quad (1)$$

where α , β , γ , and λ are weight coefficients. The core of this problem is to search for a stable local repair plan within a dynamically updated feasible region while keeping the execution sequence and resource allocation of unaffected tasks unchanged as much as possible.

3.2 Overall Framework

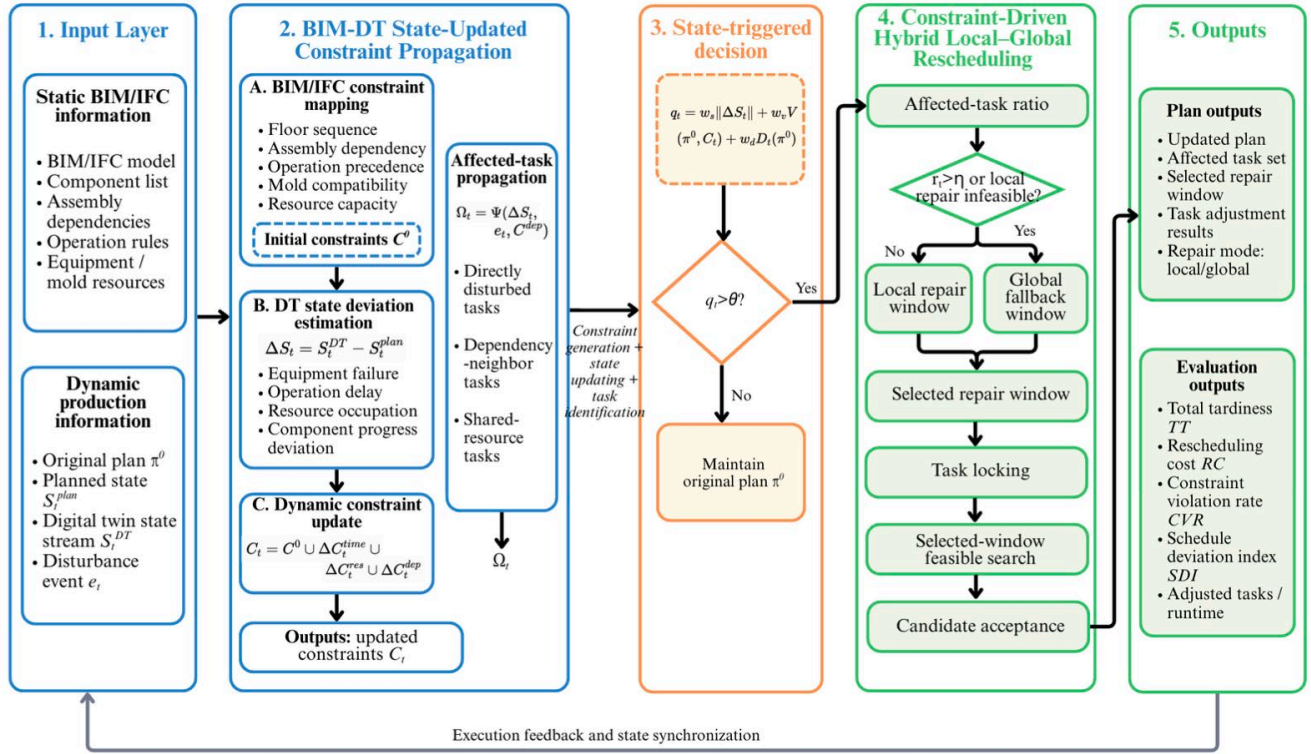
To address the insufficient coupling among BIM/IFC constraints, digital twin feedback, and disturbance-oriented rescheduling in prefabricated component manufacturing, this paper proposes a stable local rescheduling framework based on BIM-DT state-updated constraint propagation. Using BIM/IFC information, resource constraints, digital twin streams, and disturbance events, it establishes a closed loop of “constraint generation–state updating–task identification–local repair.” Figure 1(a) presents the computational workflow with conditional local–global switching, while Figure 1(b) shows the reference platform deployment architecture. Figures 2 and 3 further detail the two core modules.

The framework receives the BIM/IFC model, component list, assembly dependencies, operation rules, equipment/mold resources, original plan, and digital twin stream. The BIM-DT State-Updated Constraint Propagation module maps floor sequences, component dependencies, mold compatibility, and operation precedence into initial constraints. It then incorporates equipment failures, delays, resource occupation, and progress deviations to update the feasible region and generate the affected task set and trigger signal.

The Constraint-Driven Stable Local Rescheduling module determines the repair window from the updated constraints. Once rescheduling is triggered, the affected-task ratio and local feasibility determine whether the local window is retained or expanded to all unlocked tasks. The selected window then undergoes task locking, feasible search, and candidate acceptance. Outputs include the updated plan, affected tasks, selected window, repair mode, task adjustments, tardiness, rescheduling cost, constraint violations, schedule deviation, and runtime. Thus, BIM/IFC information and digital twin feedback are transformed into dynamic constraints, trigger conditions, and repair boundaries.

Figure 1(b) connects factory resources to the digital platform through RFID/UWB readers, PLCs, sensors, and an edge gateway. Synchronized streams enter the BIM-DT platform, where BIM/IFC services manage component relationships and resource attributes, while digital-twin services synchronize planned and actual states. The rescheduling service processes the updated constraints and returns the revised schedule to MES/ERP, workstations, and the monitoring dashboard. Execution feedback closes the loop. This figure represents a reference integration pathway rather than a completed factory deployment.

(a) Computational workflow with conditional local–global switching



(b) Reference platform deployment architecture

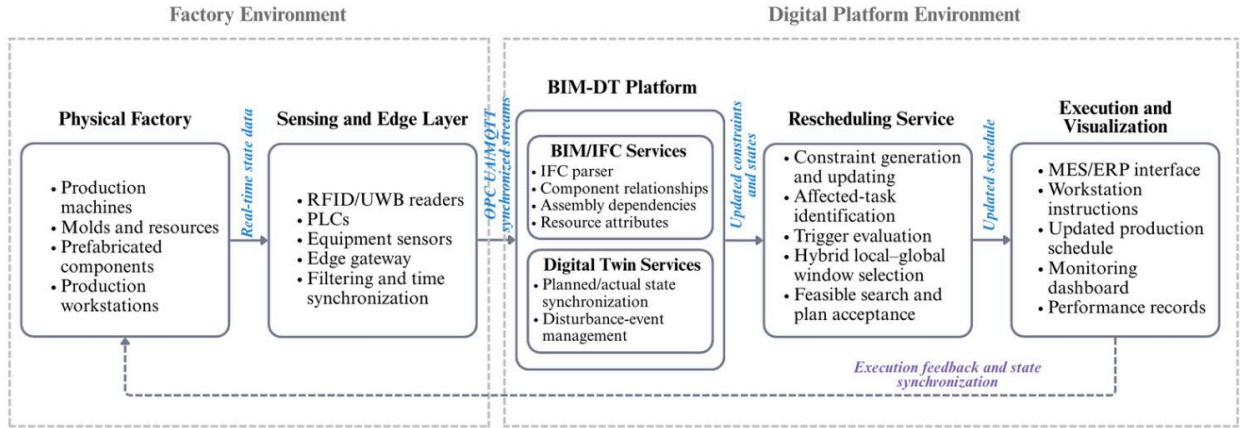


Figure 1. Computational workflow and reference deployment architecture of the BIM-DT rescheduling method:
 (a) state-updated constraint propagation with conditional local–global switching
 (b) integration of factory resources, sensing and edge devices, BIM-DT services, rescheduling services, and execution systems

3.3 BIM-DT State-Updated Constraint Propagation

The BIM-DT State-Updated Constraint Propagation module is designed to integrate BIM/IFC constraint mapping with digital twin state feedback, thereby enabling dynamic constraint updating, affected task identification, and

rescheduling trigger determination. As shown in Figure 2, the inputs of this module include the BIM/IFC model, component attributes, assembly dependencies, operation rules, equipment states, and disturbance events. The outputs include the updated constraint set C_t , the affected task set Ω_t , and the rescheduling trigger signal. The core idea is to transform state deviations into incremental terms of time-window

constraints, resource-capacity constraints, and dependency-propagation constraints.

First, the BIM/IFC model is mapped into the initial production constraint set:

$$\mathcal{C}^0 = \Phi_{\text{BIM}}(M_{\text{IFC}}) = \{C^{\text{floor}}, C^{\text{dep}}, C^{\text{proc}}, C^{\text{mold}}, C^{\text{res}}\} \quad (2)$$

where C^{floor} , C^{dep} , C^{proc} , C^{mold} , and C^{res} denote the constraints of floor sequence, assembly dependency, operation precedence, mold compatibility, and resource capacity, respectively.

Operationally, Φ_{BIM} traverses IFC entities and relationships to construct a task–dependency–resource graph. Each generated task i is assigned a start-time variable x_i , completion-time variable $c_i = x_i + p_i$, and resource-allocation variable $y_{i,r,t}$, while a component-to-task table maintains traceability to the original IFC entity. `IfcRelContainedInSpatialStructure` provides component–floor membership for C^{floor} ; `IfcRelAggregates` and `IfcRelConnectsElements` provide assembly relationships for C^{dep} ; `IfcTask` and `IfcRelSequence` define C^{proc} ; component types and manufacturing property sets determine C^{mold} ; and `IfcRelAssignsToResource` links tasks to molds or equipment for C^{res} . Manufacturing attributes absent from standard IFC fields are stored in user-defined property sets. For example, if component W_{17} belongs to Storey 3, precedes B_{06} in an assembly relation, and uses mold M_2 , the parser records its floor index, generates $x_{B_{06}} \geq c_{W_{17}}$, and imposes $\sum_i y_{i,M_2,t} \leq 1$.

The deviation between the digital twin state S_t^{DT} and the planned state S_t^{plan} is defined as:

$$\Delta S_t = S_t^{\text{DT}} - S_t^{\text{plan}} \quad (3)$$

Based on ΔS_t and the disturbance event e_t , the dynamic constraint set is updated as:

$$\mathcal{C}_t = \mathcal{U}(\mathcal{C}^0, \Delta S_t, e_t) = \mathcal{C}^0 \cup \Delta \mathcal{C}_t^{\text{time}} \cup \Delta \mathcal{C}_t^{\text{res}} \cup \Delta \mathcal{C}_t^{\text{dep}} \quad (4)$$

where $\Delta \mathcal{C}_t^{\text{time}}$, $\Delta \mathcal{C}_t^{\text{res}}$, and $\Delta \mathcal{C}_t^{\text{dep}}$ denote the update terms for time windows, resource capacity, and dependency relationships, respectively. Operation delays and delivery changes mainly update $\Delta \mathcal{C}_t^{\text{time}}$; equipment failures, mold occupation, and resource shortages mainly update $\Delta \mathcal{C}_t^{\text{res}}$; and component progress deviations form $\Delta \mathcal{C}_t^{\text{dep}}$ along BIM/IFC dependency relationships.

The affected task set is defined as:

$$\Omega_t = \Psi(\Delta S_t, e_t, C^{\text{dep}}) \quad (5)$$

where $\Psi(\cdot)$ first identifies directly disturbed tasks and then propagates along the BIM/IFC dependency graph to subsequent components, same-floor related components, and tasks sharing critical resources. In this way, state feedback is transformed into dynamic constraints, the affected task set, and rescheduling trigger conditions.

In the reconstructed experiments, BIM/IFC-derived constraints are represented by mapped scheduling variables from OR-Library, while state deviations are generated from normalized C-MAPSS sensor streams.

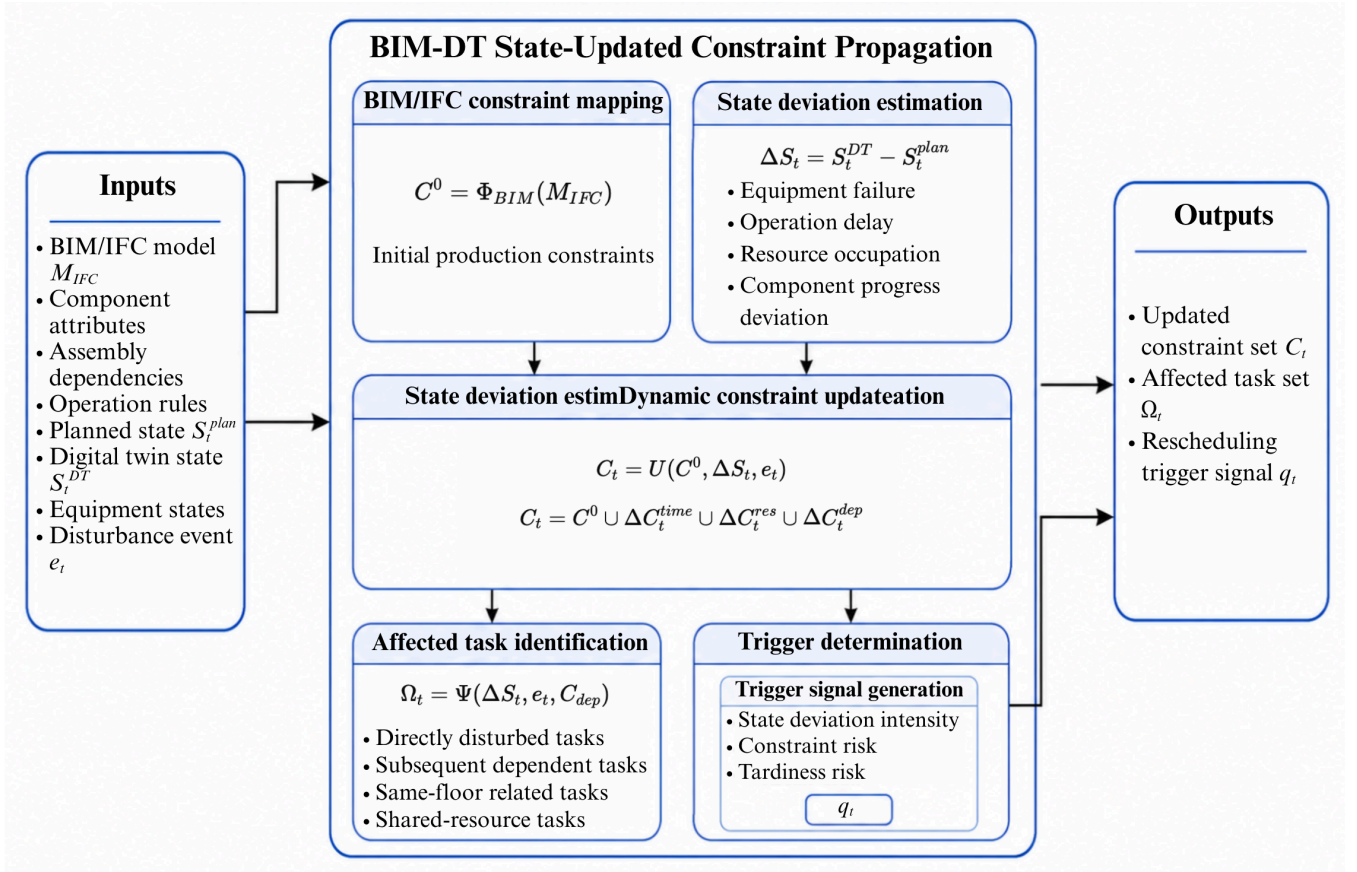


Figure 2. BIM-DT State-Updated Constraint Propagation Module

3.4 Constraint-Driven Stable Local Rescheduling

The Constraint-Driven Stable Local Rescheduling module performs stable rescheduling using updated constraints and affected tasks. As shown in Figure 3, it takes C_t , Ω_t , the original plan π^0 , delivery deadlines, and resource states as inputs, and outputs the updated plan π'_t , task adjustments, tardiness, rescheduling cost, and schedule deviation. The local boundary is jointly determined by BIM dependency propagation and state deviations.

The local repair window is defined as:

$$W_t = \Omega_t \cup N_{\text{dep}}^h(\Omega_t) - L_t \quad (6)$$

where $N_{\text{dep}}^h(\Omega_t)$ contains tasks within h hops of the affected tasks in the BIM/IFC dependency graph. L_t is the locked set, including completed or non-interruptible tasks and tasks occupying critical resources whose adjustment costs exceed the threshold.

For extreme disturbances, a hybrid local–global fallback is introduced. Let $r_t = |\Omega_t|/|J|$ denote the affected-task

ratio. The local window is retained when $r_t \leq \eta$ and local repair is feasible; otherwise, it expands to all unlocked tasks:

$$W_t^H = \begin{cases} W_t, & r_t \leq \eta \text{ and local repair is feasible,} \\ J \setminus L_t, & r_t > \eta \text{ or local repair is infeasible.} \end{cases} \quad (7)$$

Here, η is selected on the validation set.

For tasks outside the selected window W_t^H , the original plan remains unchanged:

$$x'_i = x_i^0, r'_i = r_i^0, \forall j_i \notin W_t^H \quad (8)$$

where x_i denotes the task start time and r_i denotes resource allocation. The objective within W_t^H is:

$$\min_{\pi'_t} L_{\text{res}} = \alpha D(\pi'_t) + \beta R(\pi'_t) + \gamma V(\pi'_t, C_t) + \lambda S(\pi'_t, \pi^0) \quad (9)$$

where D , R , V , and S denote normalized total tardiness, rescheduling cost, constraint violation, and schedule deviation. α , β , γ , and λ are weight coefficients. Through the local window, task locking, deviation control, and conditional global fallback, this module maintains stable rescheduling under local and extreme disturbances.

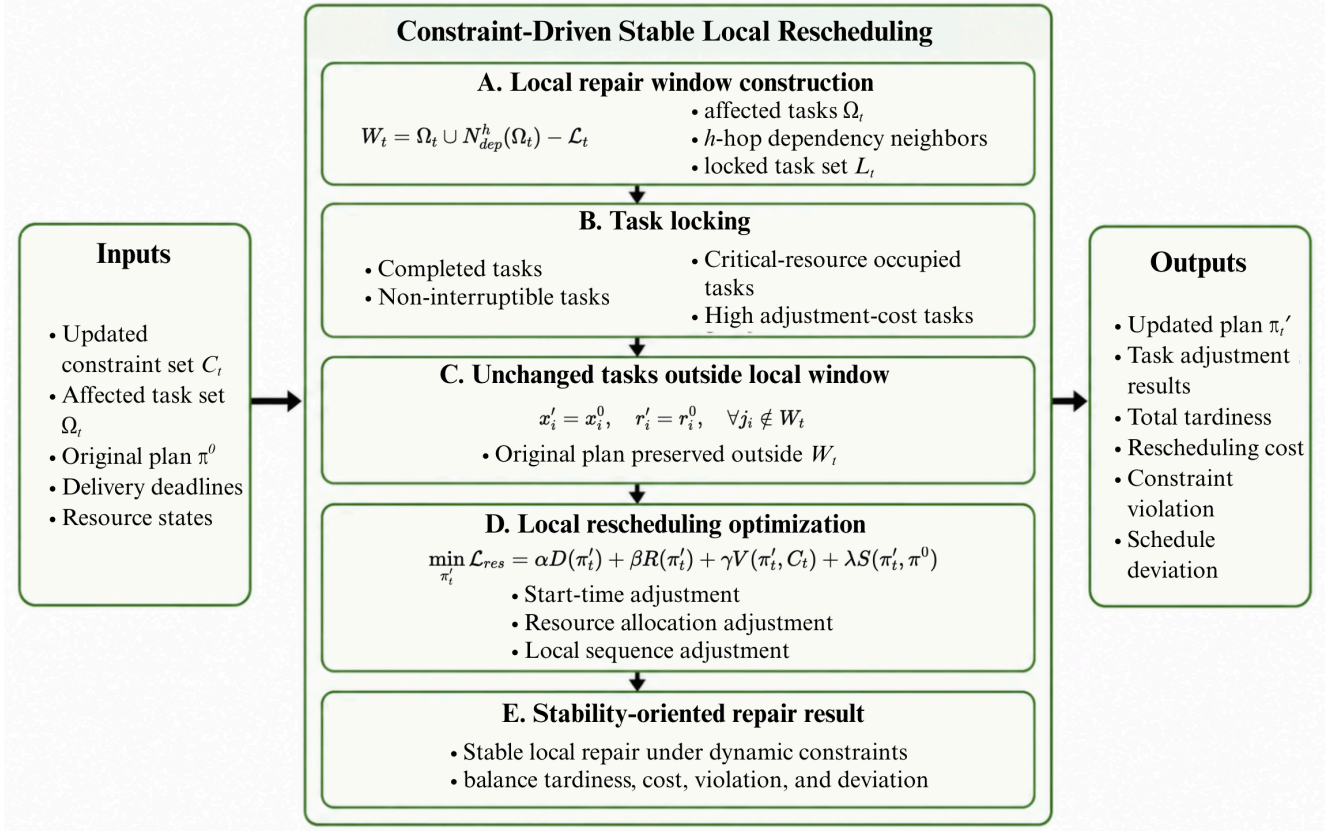


Figure 3. Constraint-Driven Stable Local Rescheduling Module

3.5 State-Triggered Stable Local Rescheduling Algorithm

To avoid unnecessary schedule disturbances caused by fixed-period rescheduling or global rescheduling, this paper designs a State-Triggered Stable Local Rescheduling algorithm. The algorithm takes the original plan π^0 , the dynamic constraint set C_t , the affected task set Ω_t , the initial local repair window W_t , and the state deviation ΔS_t as inputs. Optimization is activated only when state deviation, constraint risk, and tardiness risk reach the trigger condition. The core idea is that rescheduling is jointly driven by production states, constraint feasibility, and delivery risk, rather than being directly triggered by a single disturbance event.

First, the rescheduling trigger score is defined as:

$$q_t = w_s \parallel \Delta S_t \parallel + w_v V(\pi^0, C_t) + w_d D_t(\pi^0) \quad (10)$$

where $\parallel \Delta S_t \parallel$ denotes the deviation intensity between the digital twin state and the planned state, $V(\pi^0, C_t)$ denotes the violation degree of the original plan under the updated constraints, and $D_t(\pi^0)$ denotes the tardiness risk of the original plan under the current disturbance state. w_s , w_v , and w_d are weight coefficients. When $q_t \leq \theta$, the system maintains the original plan. When $q_t > \theta$, the algorithm

selects the hybrid repair window W_t^H according to the affected-task ratio and local feasibility condition defined in Equation (7). It then activates optimization within the selected feasible region:

$$\hat{\pi}_t = \arg \min_{\pi \in \mathcal{F}(C_t, W_t^H)} L_{\text{res}}^t(\pi) \quad (11)$$

where $\mathcal{F}(C_t, W_t^H)$ denotes the search space constrained by the dynamic constraint set and the selected hybrid repair window. Only the start times, resource allocations, and local sequences of tasks within W_t^H are allowed to be adjusted, while tasks outside W_t^H remain locked. When local repair is feasible and the affected-task ratio does not exceed η , $W_t^H = W_t$; otherwise, $W_t^H = \mathcal{J} \setminus L_t$.

The candidate plan acceptance rule is formulated as:

$$\pi'_t = \begin{cases} \hat{\pi}_t, & \text{if } V(\hat{\pi}_t, C_t) = 0 \text{ and } L_{\text{res}}^t(\hat{\pi}_t) < L_{\text{res}}^t(\pi^0), \\ \hat{\pi}_t, & \text{if } V(\hat{\pi}_t, C_t) < V(\pi^0, C_t) \text{ and } L_{\text{res}}^t(\hat{\pi}_t) < L_{\text{res}}^t(\pi^0), \\ \pi^0, & \text{otherwise.} \end{cases} \quad (12)$$

This rule gives priority to accepting plans that fully satisfy the dynamic constraints and reduce the objective value. When a fully feasible solution is temporarily unavailable, a soft repair plan is allowed to be accepted if it reduces constraint

violations and improves the objective value. In this way, the algorithm forms a closed-loop process of “state updating–trigger determination–hybrid window selection–bounded search–candidate acceptance.”

3.6 Adaptive Stability Control and Termination

To balance delivery timeliness and schedule stability, this paper introduces an adaptive stability mechanism. A fixed deviation penalty cannot adapt to different disturbance intensities. Minor disturbances require greater plan stability, whereas severe disturbances require a wider repair scope. Therefore, this paper extends the fixed stability weight λ into an adaptive weight that varies with the disturbance scope:

$$\lambda_t = \frac{\lambda_0}{1 + \rho \frac{|\Omega_t|}{|J|}} \quad (13)$$

where λ_0 is the initial weight, ρ is the adjustment coefficient, and $|\Omega_t|/|J|$ is the affected-task ratio. A small disturbance maintains a high λ_t to suppress unnecessary deviations, while a larger scope reduces λ_t to prioritize tardiness and constraint repair.

To characterize sensor uncertainty, let $\tilde{S}_t^{\text{DT}} = S_t^{\text{DT}} + \varepsilon_t$, where $\|\varepsilon_t\| \leq B_\varepsilon$, giving $\tilde{\Delta S}_t = \Delta S_t + \varepsilon_t$. If the update operator $\mathcal{U}$ and violation function V are locally Lipschitz with constants L_U and L_V , the trigger-score perturbation satisfies $|\tilde{q}_t - q_t| \leq (w_s + w_v L_V L_U) B_\varepsilon$. The trigger decision remains unchanged when $|q_t - \theta|$ exceeds this bound; otherwise, false or missed triggers may occur. Noise beyond B_ε falls outside this guarantee and is evaluated in Section 4.6.

Accordingly, the stable local rescheduling objective at time t is defined as:

$$\mathcal{L}_{\text{res}}^t = \alpha D(\pi_t') + \beta R(\pi_t') + \gamma V(\pi_t', \mathcal{C}_t) + \lambda_t S(\pi_t', \pi^0) \quad (14)$$

where $D(\pi_t')$, $R(\pi_t')$, $V(\pi_t', \mathcal{C}_t)$, and $S(\pi_t', \pi^0)$ denote total tardiness, rescheduling cost, constraint violation, and schedule deviation, respectively. The optimization process terminates when the change in the objective value falls below a threshold or when the maximum number of iterations is reached:

$$|\mathcal{L}_{\text{res}}^k - \mathcal{L}_{\text{res}}^{k-1}| < \varepsilon \text{ or } k = K_{\text{max}} \quad (15)$$

where ε is the objective-change threshold and K_{max} is the iteration limit. This mechanism links disturbance scope, repair intensity, and stability without using a fixed penalty.

3.7 Termination and Bounded Stability Analysis

To analyze the optimization stability of the proposed algorithm, this paper provides an explanation from three

aspects: the non-increasing property of the objective function, finite termination, and bounded disturbances. According to the candidate plan acceptance rule in Section 3.5, the algorithm accepts an updated plan only when the candidate plan fully satisfies the dynamic constraints and reduces the objective value, or when a fully feasible solution is unavailable but the candidate plan reduces constraint violations and improves the objective value. Therefore, if the candidate solution is accepted at the k -th iteration, the following condition holds:

$$\mathcal{L}_{\text{res}}^k \leq \mathcal{L}_{\text{res}}^{k-1} \quad (16)$$

Since total tardiness, rescheduling cost, constraint violation, and schedule deviation are all non-negative terms, and the weight coefficients α , β , γ , and λ_t are non-negative and finite, the objective function has a non-negative lower bound:

$$\mathcal{L}_{\text{res}}^t \geq 0 \quad (17)$$

Therefore, under the acceptance rule, the objective function follows a non-increasing trend and is constrained by a non-negative lower bound. Meanwhile, given the finite task set, finite resource set, and the maximum iteration limit K_{max} , the optimization process can terminate within a finite number of steps.

For disturbance uncertainty, this paper assumes that the digital twin state deviation remains within the observable production-state range:

$$\|\Delta S_t\| \leq B_s \quad (18)$$

where B_s is the state-deviation bound. Because the dynamic constraint updates are generated from bounded deviations and a finite task set, disturbance-induced objective fluctuations are also bounded.

For real-time execution, let T_{iter} be the per-iteration time bound and τ_{RT} the response-time budget. The iteration limit satisfies

$$T_{\text{online}} \leq K_{\text{max}} T_{\text{iter}}, K_{\text{max}} \leq \left\lfloor \frac{\tau_{\text{RT}}}{T_{\text{iter}}} \right\rfloor. \quad (19)$$

When the budget is reached, the best accepted plan is returned. This establishes monotonic acceptance, bounded response time, and finite termination, but not global-optimum convergence. Tables 3 and 4 provide empirical runtime evidence.

3.8 Online Rescheduling Procedure

This section summarizes the online rescheduling procedure of the proposed framework. It sequentially performs BIM/IFC constraint generation, digital twin state updating, affected-task identification, local-window construction, state-

trigger determination, hybrid local–global window selection, and candidate-plan acceptance.

Algorithm 1: State-Triggered Stable Local Rescheduling

Input: $M_{\text{IFC}}, \pi^0, \mathcal{J}, \mathcal{R}, S_t^{\text{DT}}, S_t^{\text{plan}}, e_t, \eta$
Output: π'_t, Ω_t, W_t^H
1: $C^0 \leftarrow \Phi_{\text{BIM}}(M_{\text{IFC}})$
2: $\Delta S_t \leftarrow S_t^{\text{DT}} - S_t^{\text{plan}}$
3: $C_t \leftarrow \mathcal{U}(C^0, \Delta S_t, e_t)$
4: $\Omega_t \leftarrow \Psi(\Delta S_t, e_t, C^{\text{dep}})$
5: $W_t \leftarrow \Omega_t \cup N_{\text{dep}}^h(\Omega_t) - L_t$
6: $W_t^H \leftarrow W_t$
7: $q_t \leftarrow w_s \parallel \Delta S_t \parallel + w_v V(\pi^0, C_t) + w_d D_t(\pi^0)$
8: if $q_t \leq \theta$ then return π^0, Ω_t, W_t^H
9: $r_t \leftarrow |\Omega_t| / |\mathcal{J}|$
10: if $r_t > \eta$ or $\mathcal{F}(C_t, W_t) = \emptyset$ then $W_t^H \leftarrow \mathcal{J} \setminus L_t$
11: $\hat{\pi}_t \leftarrow \arg \min_{\pi \in \mathcal{F}(C_t, W_t^H)} L_{\text{res}}^t(\pi)$
12: if $L_{\text{res}}^t(\hat{\pi}_t) < L_{\text{res}}^t(\pi^0)$ and
 $[V(\hat{\pi}_t, C_t) = 0 \text{ or } V(\hat{\pi}_t, C_t) < V(\pi^0, C_t)]$ then
13: $\pi'_t \leftarrow \hat{\pi}_t$
14: else $\pi'_t \leftarrow \pi^0$
15: end if
16: return π'_t, Ω_t, W_t^H

Steps 1–5 update the BIM-DT constraints and construct the initial local window. Steps 6–8 determine whether rescheduling is required. Steps 9–10 retain local repair when the affected scope is limited and expand the window to all unlocked tasks when the affected-task ratio exceeds η or no feasible local solution exists. Steps 11–15 optimize and accept the candidate plan according to feasibility and objective improvement. The procedure integrates state triggering, hybrid window selection, repair acceptance, and stability control into a closed-loop plan-repair process.

4. Experiments and Results

4.1 Experimental Setup

4.1.1 Dataset Setup

To validate the BIM-DT state-updated constraint propagation and stable local rescheduling method, this paper constructs reproducible experimental scenarios based on two public datasets. Since there is currently a lack of large-scale public datasets that simultaneously contain BIM/IFC constraints, digital twin state streams, and prefabricated component rescheduling records, this study conducts reconstructed experiments using public scheduling data and equipment-state data. Therefore, the experiments mainly evaluate the effectiveness of the rescheduling mechanism under public-data-based reconstructed scenarios, while its applicability in real factory BIM-DT platforms still requires further validation.

This study uses the Taillard Flow Shop Scheduling Instances from OR-Library to construct production scheduling scenarios. In the reconstruction process, jobs are mapped to component tasks, machines are mapped to workstations, molds, or equipment resources, and processing times are mapped to operation processing times. Operation sequences, delivery deadlines, and resource-capacity constraints are then generated accordingly. The NASA C-MAPSS Jet Engine Simulated Data are used to construct equipment state streams and disturbance events. After normalization, the sensor sequences are compared with the normal-state baseline to form state deviations ΔS_t , which are further mapped to equipment unavailability, processing delays, or resource-capacity reduction. The component tasks, resource constraints, and state disturbances used in the experiments are all reconstructed from public datasets. They are mainly used to validate the reproducibility of the proposed rescheduling mechanism and are not equivalent to data from a real prefabricated component factory BIM-DT platform. The OR-Library experiments use benchmark scheduling instances with reconstructed disturbance events, whereas the C-MAPSS experiments retain the reconstructed scheduling structure but derive equipment-state deviations and disturbance triggers from normalized C-MAPSS sensor sequences. Table 1 summarizes the data reconstruction process.

Table 1. Public datasets and reconstructed experimental variables

Data source	Original data content	Reconstructed variables in this study	Experimental use
OR-Library Taillard Flow Shop	Jobs, machines, processing times	Component tasks, operations, mold/workstation resources, delivery deadlines	Constructing the initial plan, the initial constraint set C^0 , and local rescheduling tasks
NASA C-MAPSS	Operating cycles, operating settings, sensor measurements	Equipment state streams, state deviations, disturbance events	Constructing digital twin state feedback, dynamic constraint updates, and rescheduling trigger events

4.1.2 Baseline Methods

To compare the proposed method with existing approaches, nine baseline methods are selected, covering rule-based scheduling, constraint optimization, metaheuristics, deep reinforcement learning, and recent dynamic scheduling

methods. All methods are run under the same task instances, disturbance events, resource capacities, and evaluation metrics. The proposed method is denoted as Proposed. Table 2 presents the method types and adaptation settings.

Table 2. Baseline and proposed methods

Method	Type	Adaptation in this study
PDR[34]	Rule-based scheduling	Reschedules affected tasks after disturbances according to rules such as SPT, EDD, and FIFO
CP-SAT[35]	Constraint optimization	Uses the same time, resource, operation, and delivery constraints
HGA-TS[36]	Metaheuristic	Optimizes TT, RC, CVR, and SDI within the same task set
DQN-DFJSP[37]	Reinforcement learning scheduling	Uses task states, resource states, and disturbance indicators as inputs
GNN-DRL[38]	Graph reinforcement learning scheduling	Constructs a graph of tasks, resources, and dependency relationships
MachineRank-D3QN[39]	Recent dynamic scheduling	Uses the dynamic disturbance events constructed in this study
ET-DQN[40]	Recent dynamic scheduling	Uses the same state space and action space
DDQN-APS[41]	Recent dynamic scheduling	Uses the same equipment-failure, processing-delay, and delivery-change scenarios
ND3QNP[42]	Recent dynamic scheduling	Performs rescheduling under the same disturbance samples and resource constraints
Proposed	Proposed method	Uses dynamic constraints, the affected task set, the local window, and the adaptive stability weight

4.1.3 Implementation Details

All methods are compared using the same task instances, disturbances, and resource constraints. Experiments are conducted on an Intel i7-12700K CPU, 32 GB RAM, and an NVIDIA RTX 3060 GPU. OR-Library provides the benchmark task structures used to construct initial plans, deadlines, and resource constraints, while normalized C-MAPSS sensor streams are mapped to equipment-state deviations and trigger signals. Disturbances include equipment failures, processing delays, capacity reductions, and deadline changes. Extreme tests additionally simulate simultaneous failures of two, three, and four critical resources. The hybrid strategy changes only the repair window. These reconstructed scenarios support preliminary algorithmic comparison rather than factory-level validation: OR-Library lacks precast-specific IFC semantics, while C-MAPSS degradation patterns may differ from the sensor and failure characteristics of precast production equipment.

The samples are divided into training, validation, and test sets. The validation set selects θ , h , ρ , and the local–global threshold η , which is fixed at 0.45 during testing. Each experiment is repeated five times with independent seeds and disturbance samples. Results are reported as mean $\pm$ standard deviation. Runtime excludes offline training and preprocessing.

4.1.4 Evaluation Metrics

Six metrics are used to evaluate rescheduling performance, and lower values indicate better performance for all metrics.

(1) $TT = \sum_{i=1}^{|J|} \max(0, C_i - d_i)$, where TT denotes total tardiness, C_i is the completion time, and d_i is the delivery deadline. (2) $RC = \sum_{i=1}^{|J|} (\omega_t |x'_i - x_i^0| + \omega_r \delta(r'_i \neq r_i^0))$, where RC denotes the rescheduling cost caused by changes in start times and resource allocations. (3) $CVR = \frac{N_{violated}}{N_{constraints}}$, where CVR denotes the dynamic constraint violation rate. (4)

$SDI = \frac{1}{|J|} \sum_{i=1}^{|J|} \frac{|x'_i - x_i^0|}{H}$, where SDI denotes the degree of schedule deviation, and H is the time span of the original plan. (5) $Runtime = T_{end} - T_{start}$, where Runtime denotes the online rescheduling time. (6) $ATR = \frac{N_{adjusted}}{|J|} \times 100\%$, where ATR denotes the adjusted task ratio, namely Adjusted tasks.

4.2 Main Experimental Results

Tables 3 and 4 present the main results under the OR-Library and C-MAPSS disturbance scenarios, respectively. All results are reported as mean $\pm$ standard deviation over five repeated experiments. Proposed achieves relatively low TT, SDI, and adjusted task ratio on both datasets. Compared with the strongest dynamic baseline, ND3QNP, Proposed reduces TT from 101.8 to 96.4 on OR-Library and from 131.7 to 126.5 on C-MAPSS, indicating its favorable tardiness control under reconstructed disturbance scenarios.

To quantify the operational economic implications, two relative indicators are calculated against ND3QNP:

$$RCSR = \frac{RC_{ND3QNP} - RC_{Proposed}}{RC_{ND3QNP}} \times 100\%, DDRR = \frac{TT_{ND3QNP} - TT_{Proposed}}{TT_{ND3QNP}} \times 100\%.$$

The rescheduling-cost saving rate (RCSR) and delivery-delay reduction rate (DDRR) are 4.2% and 5.3% on OR-Library and 4.1% and 3.9% on C-MAPSS, respectively. These results indicate lower adjustment burden and delivery delay. Because the reconstructed datasets lack monetary unit costs and actual factory delivery records, the indicators are operational proxies rather than realized financial savings or calendar-day delivery advances.

Different methods show different advantages. PDR achieves the lowest Runtime, but its TT and CVR are relatively high, indicating that simple dispatching rules respond quickly but have difficulty handling dynamic constraints. CP-SAT achieves the best CVR, but its RC, SDI, and Runtime are relatively high, suggesting stronger constraint feasibility but a larger repair scope and higher computational cost. DDQN-APS and ND3QNP perform close to Proposed, but their adjusted task ratios remain higher, which may be related to the lack of explicit dependency-propagation boundaries and task-locking design.

Overall, Proposed maintains low levels of TT, SDI, and adjusted task ratio, corresponding to better tardiness control and schedule stability. CP-SAT has an advantage in constraint violation control, whereas PDR has an advantage in computational time, indicating that different methods emphasize different optimization objectives. Figure 4 reports the normalized overall score calculated from TT, RC, CVR, SDI, Runtime, and adjusted task ratio on the two datasets, where a lower score indicates better combined performance. Figure 5 further illustrates the TT–SDI trade-off relationship to compare tardiness control and schedule stability.

Table 3. Main results on OR-Library

Method	TT↓	RC↓	CVR (%)↓	SDI↓	Runtime (s)↓	Adjusted tasks (%)↓
PDR	186.4±7.1	49.8±2.4	8.9±0.6	0.141±0.010	0.18±0.03	41.2±2.6
CP-SAT	132.7±5.4	54.2±2.8	1.5±0.2	0.158±0.012	38.6±3.5	47.5±3.1
HGA-TS	119.8±4.9	46.3±2.1	5.3±0.4	0.124±0.009	21.4±2.2	35.4±2.4
DQN-DFJSP	126.5±5.6	43.2±2.0	5.7±0.5	0.119±0.008	7.8±0.7	33.8±2.0
GNN-DRL	113.9±4.1	40.4±1.9	4.1±0.4	0.111±0.007	9.6±0.9	31.5±1.8
MachineRank-D3QN	108.6±4.5	38.7±1.8	3.5±0.3	0.106±0.007	8.3±0.8	30.4±1.7
ET-DQN	106.9±4.2	37.6±1.7	3.2±0.3	0.103±0.006	8.0±0.8	29.8±1.6
DDQN-APS	103.5±3.9	36.2±1.8	3.0±0.3	0.101±0.006	7.3±0.8	28.9±1.6
ND3QNP	101.8±3.7	35.6±1.7	2.5±0.2	0.098±0.006	8.1±0.9	27.6±1.5
Proposed	96.4±3.5	34.1±1.6	1.8±0.2	0.085±0.005	7.0±0.7	24.7±1.3

Table 4. Main results under C-MAPSS-derived equipment-state disturbance scenarios

Method	TT↓	RC↓	CVR (%)↓	SDI↓	Runtime (s)↓	Adjusted tasks (%)↓
PDR	214.3±8.5	57.9±2.9	10.8±0.7	0.168±0.012	0.21±0.04	46.7±2.9
CP-SAT	163.5±6.7	64.1±3.2	2.3±0.3	0.184±0.014	45.7±4.1	52.9±3.5
HGA-TS	151.2±6.1	53.8±2.6	6.4±0.5	0.149±0.010	25.2±2.6	39.6±2.5
DQN-DFJSP	156.8±6.4	50.4±2.5	6.9±0.6	0.143±0.010	8.6±0.8	38.1±2.3
GNN-DRL	145.7±5.8	48.6±2.3	5.2±0.4	0.136±0.009	11.2±1.0	36.4±2.1
MachineRank-D3QN	139.6±5.4	45.8±2.1	4.6±0.4	0.128±0.008	9.4±0.9	34.8±2.0
ET-DQN	136.9±5.1	44.1±2.0	4.0±0.4	0.123±0.008	9.0±0.8	33.6±1.9
DDQN-APS	134.2±4.9	43.0±2.1	3.9±0.4	0.117±0.007	8.5±0.8	32.9±1.9
ND3QNP	131.7±4.6	41.8±2.0	3.4±0.3	0.112±0.007	9.3±0.9	31.1±1.8
Proposed	126.5±4.3	40.1±1.8	2.6±0.3	0.097±0.006	7.6±0.7	27.8±1.5

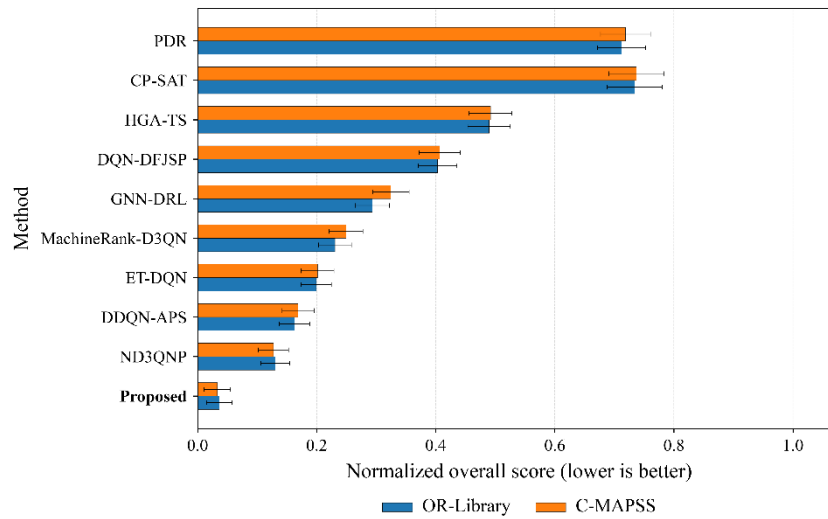


Figure 4. Normalized overall performance across six evaluation metrics on the two datasets

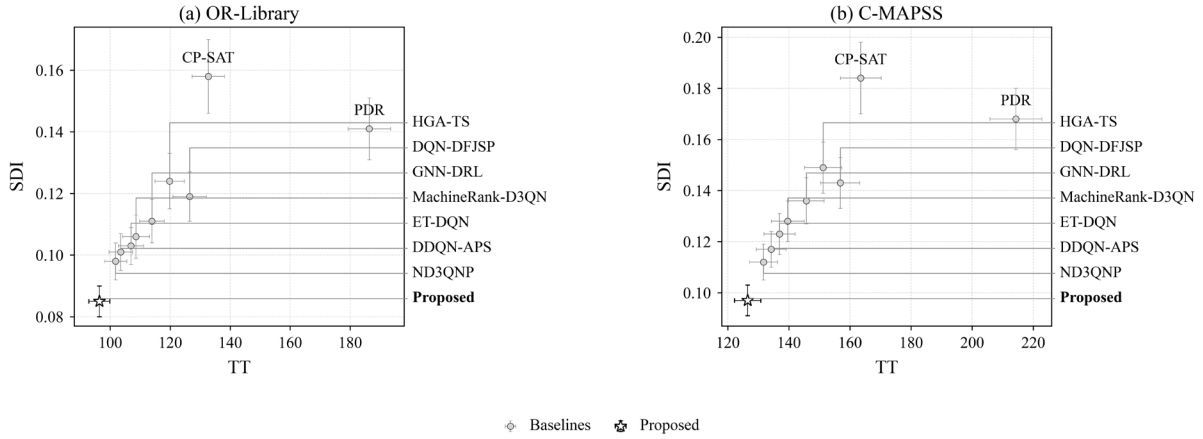


Figure 5. TT–SDI trade-off relationship:
(a) OR-Library and
(b) C-MAPSS

4.3 Statistical Significance Analysis

To provide an exploratory statistical assessment of the main results, this paper selects three strong dynamic scheduling baselines, namely ET-DQN, DDQN-APS, and ND3QNP, and conducts paired t-tests between these baselines and Proposed on the two datasets. The test object is the TT results from five repeated experiments, and the significance threshold is set to $p < 0.05$. As shown in Table 5, Proposed achieves positive TT improvements over the three baselines on both OR-Library and C-MAPSS, and all p-values are below 0.05. Most comparisons with ET-DQN and DDQN-APS reach $p <$

0.01. These results provide preliminary evidence of TT differences within the current experimental settings. None of the 95% confidence intervals crosses zero, showing a consistent improvement direction across the five paired runs. However, the limited number of runs restricts statistical power and the assessment of distributional assumptions; therefore, the p-values and confidence intervals should not be interpreted as definitive population-level inference. Mean differences and standard deviations are considered together with the test results. Figure 6 visualizes the 95% confidence intervals of TT improvement over the strong dynamic baselines.

Table 5. Paired t-test results between Proposed and strong baselines

Dataset	Comparison	$\Delta TT \downarrow$	t-value	p-value	95% CI of ΔTT
OR-Library	ET-DQN – Proposed	10.5	5.42	0.0056	[5.1, 15.9]
OR-Library	DDQN-APS – Proposed	7.1	4.38	0.0119	[2.6, 11.6]
OR-Library	ND3QNP – Proposed	5.4	3.71	0.0206	[1.3, 9.5]
C-MAPSS	ET-DQN – Proposed	10.4	5.16	0.0068	[4.8, 16.0]
C-MAPSS	DDQN-APS – Proposed	7.7	4.64	0.0097	[3.1, 12.3]
C-MAPSS	ND3QNP – Proposed	5.2	3.58	0.0232	[1.1, 9.3]

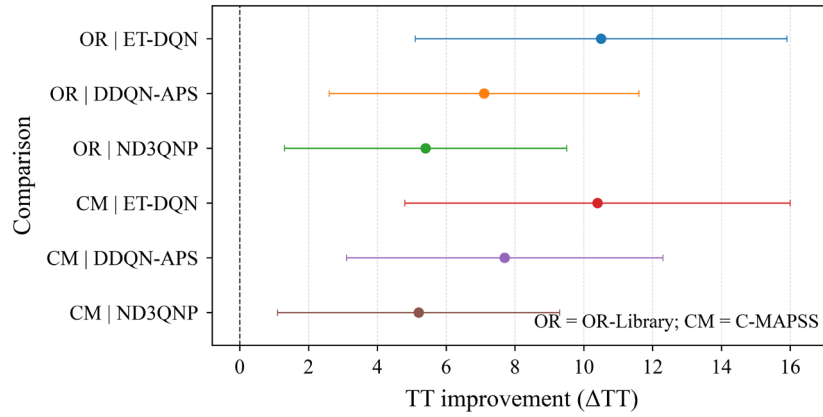


Figure 6. 95% confidence intervals of TT improvement over strong dynamic baselines

4.4 Representation and Feature Space Analysis

To examine the state–constraint–resource representation patterns, this paper conducts t-SNE, feature-distribution, and embedding-heatmap analyses. The representation vectors are extracted from the state–constraint joint encoding layer before rescheduling decisions and include state deviations, task dependencies, resource capacities, and time-window constraint features. Figure 7 shows that DQN-DFJSP and some dynamic scheduling methods exhibit representational overlap between normal and disturbed states. In contrast, the normal, mildly disturbed, and severely disturbed samples of Proposed form clearer local cluster structures.

Figure 8 presents the feature distributions of normal and disturbed states. Under disturbed states, Proposed shows

relatively concentrated responses and generally shorter distribution tails, which are consistent with a more compact disturbance-state representation. Figure 9 shows the task–constraint–resource embedding heatmap. Affected tasks have stronger responses to resource-capacity and time-window constraints, whereas unaffected tasks show lower embedding intensity. This phenomenon shows a certain correspondence with the local repair and task-locking design. Table 6 shows that Proposed achieves the best Silhouette and DBI values on both datasets, although it does not lead every CHI or inter-cluster-distance comparison. These analyses provide descriptive representation evidence rather than proof of causal mechanisms or generalization beyond the reconstructed scenarios. Therefore, the visualizations should be interpreted together with Table 6 and the main experimental results.

Table 6. Quantitative representation metrics

Method	Dataset	Silhouette ↑	DBI ↓	CHI ↑	Inter-cluster distance ↑
DQN-DFJSP	OR-Library	0.318	1.82	431.6	2.21
GNN-DRL	OR-Library	0.354	1.64	489.7	2.43
ET-DQN	OR-Library	0.389	1.47	528.4	2.58
DDQN-APS	OR-Library	0.397	1.39	556.2	2.79
ND3QNP	OR-Library	0.428	1.25	651.8	3.08
Proposed	OR-Library	0.466	1.11	638.5	3.19
DQN-DFJSP	C-MAPSS	0.291	1.93	402.7	2.06
GNN-DRL	C-MAPSS	0.329	1.75	458.9	2.34
ET-DQN	C-MAPSS	0.363	1.56	493.5	2.45

DDQN-APS	C-MAPSS	0.381	1.46	616.4	2.62
ND3QNP	C-MAPSS	0.405	1.35	552.8	3.12
Proposed	C-MAPSS	0.447	1.18	603.1	3.05

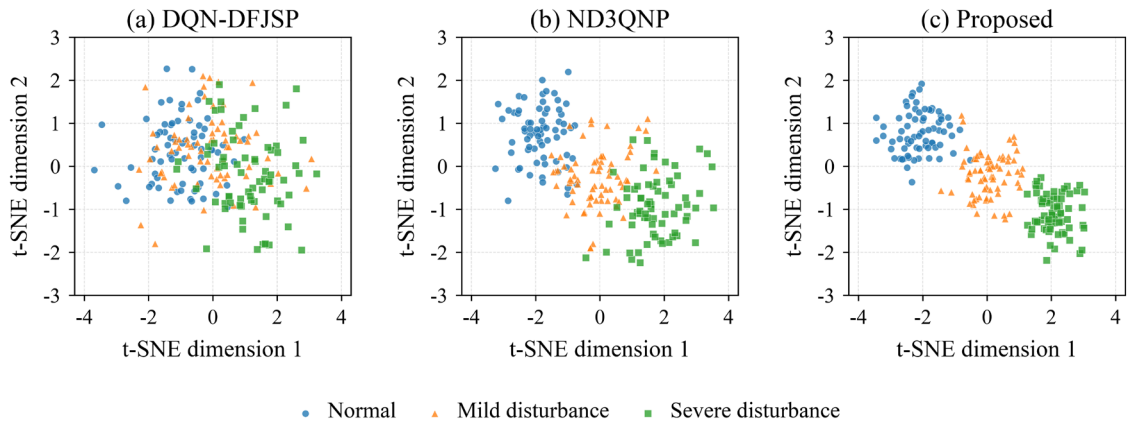


Figure 7. t-SNE visualization of the state-constraint-resource representation space

(a) DQN-DFJSP
(b) ND3QNP
(c) Proposed

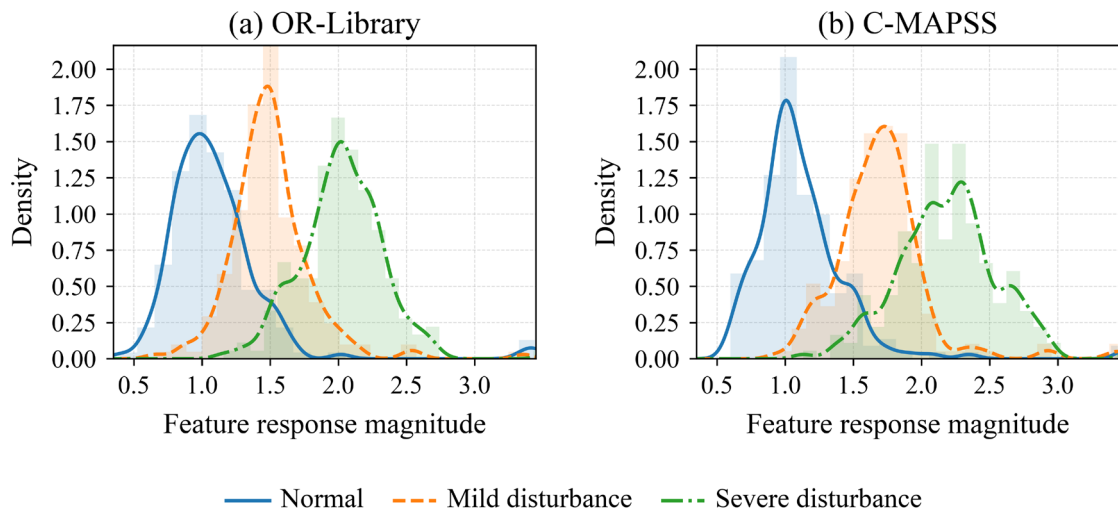


Figure 8. Feature distributions of normal and disturbed states:

(a) OR-Library and
(b) C-MAPSS

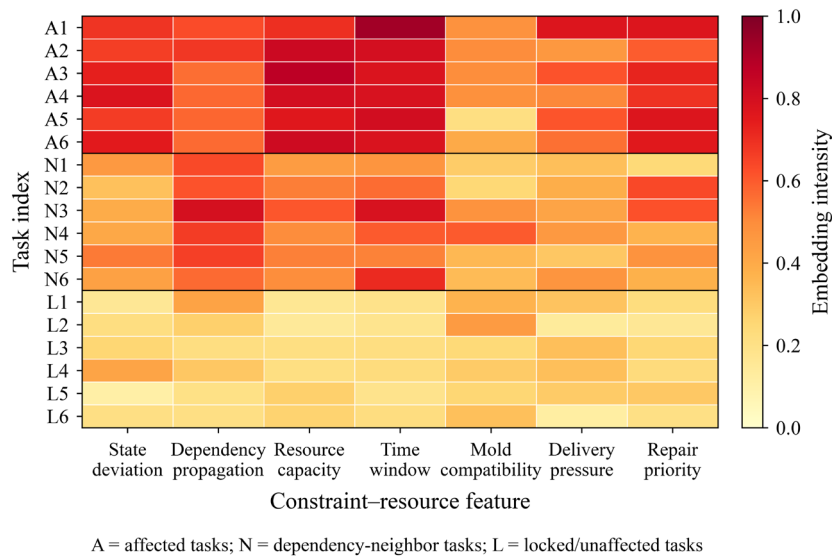


Figure 9. Task–constraint–resource embedding heatmap

4.5 Interpretability and Attention Analysis

To examine the observed decision patterns of Proposed, this paper visualizes the constraint contribution weights and trigger scores. The constraint contribution weights are calculated from the normalized contributions of time-window constraints, resource-capacity constraints, dependency propagation, mold compatibility, and delivery constraints to the trigger score and the rescheduling objective. Figure 10 shows the constraint contribution heatmaps under different disturbance scenarios. It can be observed that resource-capacity constraints receive higher weights under equipment-unavailability scenarios, time-window constraints account for a larger proportion under processing-delay scenarios, and delivery-deadline changes mainly activate delivery constraints. These results show descriptive differences in constraint-weight distributions across disturbance types.

Figure 11 shows the decomposition results of the trigger score q_t and its cross-run distribution. Under minor disturbances, state deviations and tardiness risks are relatively low, and the system mostly maintains the original plan. Under medium- and high-intensity disturbances, the constraint violation and tardiness-risk terms increase, thereby triggering local rescheduling. The trigger-score distributions are relatively concentrated, suggesting limited variation within the five repeated runs rather than broader statistical stability. Combining Figures 10 and 11, Proposed shows constraint responses that vary with disturbance type and focus rescheduling on affected tasks and their dependency neighborhoods. These results provide auxiliary descriptive evidence for interpreting constraint contribution identification and state triggering, but do not establish causal mechanisms or generalization beyond the reconstructed scenarios.

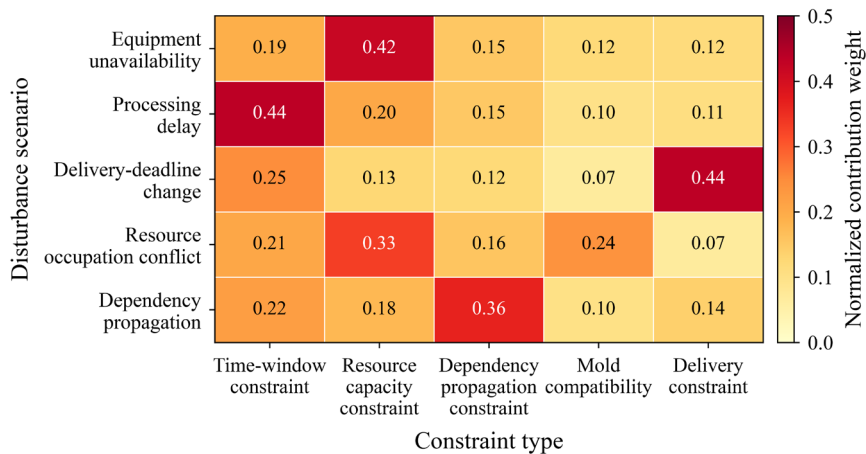


Figure 10. Constraint contribution heatmap under different disturbance scenarios

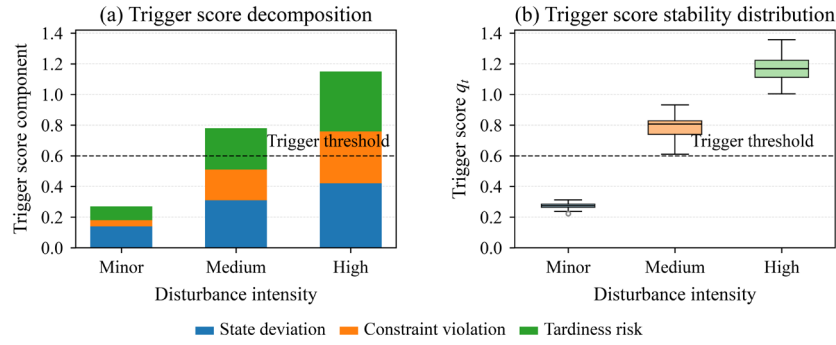


Figure 11. Trigger score decomposition and stability distribution:
(a) trigger score decomposition and
(b) trigger-score cross-run distribution

4.6 Error and Failure Case Analysis

To avoid presenting only positive results, this paper analyzes the failure cases of Proposed under extreme disturbances. Figure 12 presents four scenarios: simultaneous unavailability of multiple critical pieces of equipment, substantially advanced delivery deadlines, insufficient dependency propagation order, and severe sensor noise or overly strong state-deviation mapping. For the first scenario, two, three, and four critical resources are simultaneously unavailable, and the local-only and hybrid strategies are compared. With two unavailable resources, the affected-task ratio remains below $\eta = 0.45$, and local repair is retained. With three or four unavailable resources, the threshold is exceeded or no feasible local solution exists, activating global rescheduling of unlocked tasks. In the representative reconstructed cases, the hybrid fallback alleviates residual tardiness and constraint violations by expanding the adjustment scope, although it is accompanied by higher RC, SDI, and runtime. This comparison is descriptive because complete factory-level cost and execution records are unavailable. TT may remain high because multiple failures substantially reduce resource capacity.

When deadlines are substantially advanced, some tasks remain unable to satisfy delivery constraints because the feasible region becomes too narrow. A small BIM/IFC dependency order h may omit indirectly affected tasks and cause secondary tardiness. Sensor-noise sensitivity is

analyzed at 5%, 10%, and 20% of the normalized state range. Let $K_q = w_s + w_v L_V L_U$. Based on Section 3.6, these levels produce trigger-score uncertainty bounds of $0.05K_q$, $0.10K_q$, and $0.20K_q$, respectively. The decision remains unchanged when $|q_t - \theta|$ exceeds the corresponding bound. Severe noise may reverse decisions near θ , causing false or missed triggers and increasing SDI. This is a bounded sensitivity interpretation rather than physical-sensor evidence.

Table 7 summarizes the failure types and responses. The reconstructed failure cases suggest that the hybrid strategy can alleviate local-repair failure when the affected-task ratio exceeds 45%, but it cannot fully resolve severe resource shortages or near-infeasible deadlines. The trigger remains stable outside the noise-dependent uncertainty interval, while near-threshold samples require calibration or confirmation. These extensions can be incorporated without changing the closed-loop structure. Backup resource substitution would add alternative equipment or mold sets to ΔC_t^{res} , which would be tested before global fallback when local repair is infeasible. Adaptive propagation would replace fixed h with a disturbance-dependent h_t to update Ω_t and W_t . Feasibility pre-assessment would screen the repair window before optimization, while state-deviation calibration would precede constraint updating and trigger evaluation. Figure 12(a)–(c) visualize representative reconstructed test instances, whereas Figure 12(d) presents analytical uncertainty bounds rather than physical-sensor measurements.

Table 7. Failure types, causes, and future directions

Failure type	Observed result	Main cause	Response or future decision
Simultaneous unavailability of multiple critical pieces of equipment	Local-only repair retains high TT; the hybrid strategy expands the repair scope	Resource capacity is overly tightened	Activate global fallback and investigate backup-resource substitution

Substantially advanced delivery deadlines	CVR decreases, but tardiness cannot be eliminated	The original problem is close to infeasible	Add feasibility pre-assessment and risk classification
Insufficient dependency propagation order	Secondary tardiness occurs in subsequent tasks	The propagation order omits indirect dependencies	Design adaptive propagation depth
Sensor noise or overly strong state-deviation mapping	Trigger-score uncertainty increases from $0.05K_q$ to $0.20K_q$; near-threshold decisions may change and increase SDI	Measurement errors propagate through state deviations and constraint updating	Introduce noise filtering, uncertainty intervals, and state-deviation calibration
Excessively high affected-task ratio	Global fallback is activated when $(r_t > 0.45)$	The local repair scope is too large	Refine the switching threshold for different production scales

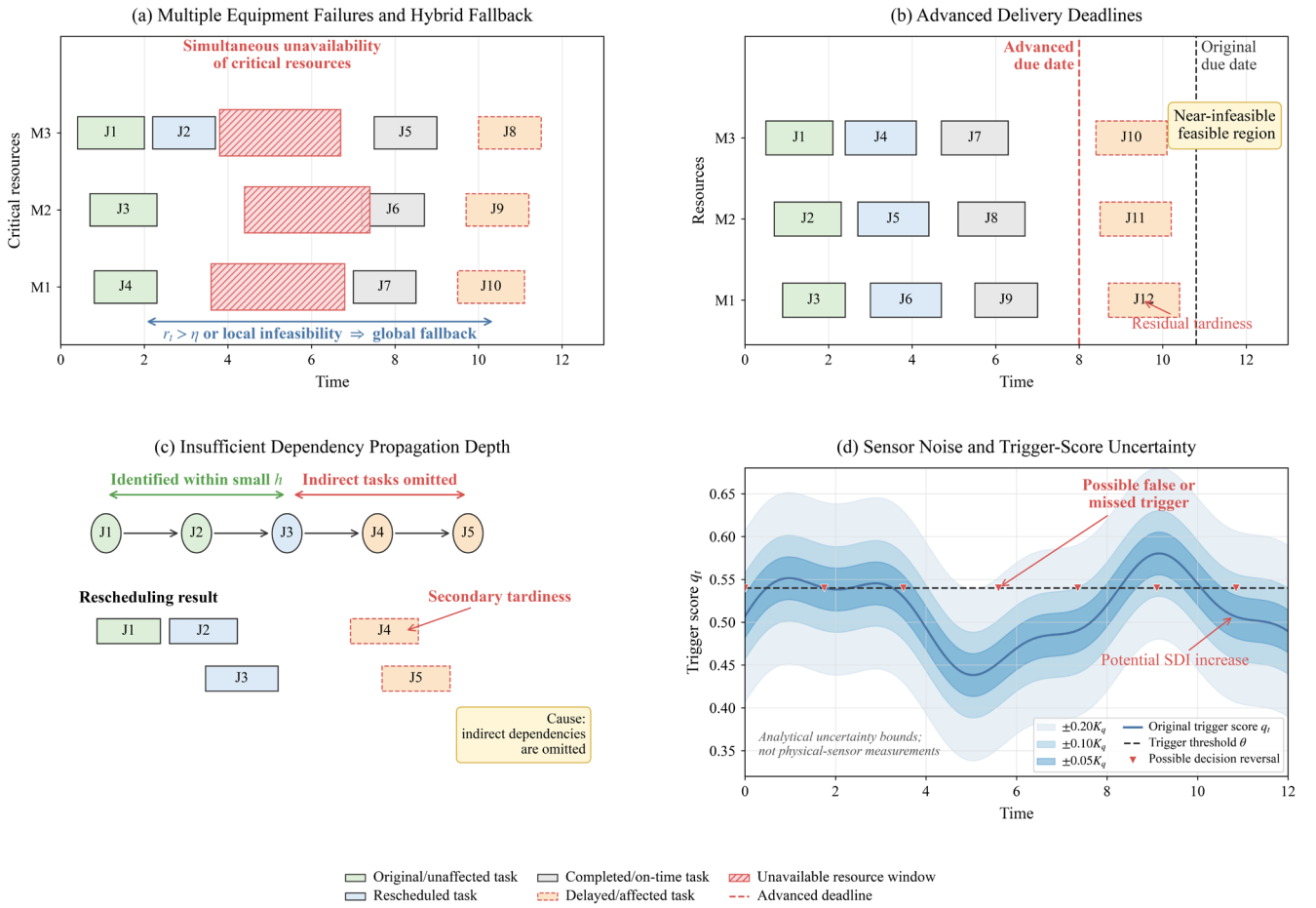


Figure 12. Visualization of failure cases:
 (a) multiple equipment failures;
 (b) advanced delivery deadlines;
 (c) insufficient dependency propagation depth;
 (d) severe sensor noise or overly strong state-deviation mapping

4.7 Ablation Study

To analyze each module, this paper conducts ablation experiments on OR-Library and C-MAPSS. Proposed uses BIM-DT state-updated constraint propagation, state triggering, dependency-based affected-task identification, the local window, task locking, and adaptive stability weighting. The variants include w/o BIM-DT, Static Ω_t , w/o Trigger, w/o Window, w/o Locking, and w/o Adaptive. In w/o Adaptive, the deviation term is retained, but λ_t is replaced with a validation-selected fixed weight λ_{fix} . Results are reported as mean $\pm$ standard deviation over five runs.

As shown in Table 8, removing BIM-DT propagation prevents equipment and progress deviations from updating time, resource, and dependency constraints, increasing TT and CVR. Static Ω_t includes only directly disturbed tasks and omits downstream dependencies, causing unresolved violations and secondary tardiness. Without triggering, minor deviations initiate unnecessary repairs, increasing RC, SDI, and the adjusted-task ratio. Removing the local window produces the broadest adjustment scope, whereas removing

task locking allows completed, non-interruptible, and high-cost tasks to change; both increase SDI and the adjusted-task ratio.

The simultaneous increase in TT and SDI under w/o Adaptive results from applying one fixed weight across different disturbance intensities. Under mild disturbances, it can be too weak to suppress unnecessary adjustments, increasing SDI; under severe disturbances, it can be too strong to permit sufficient tardiness recovery, increasing TT. Thus, the aggregate deterioration reflects different disturbance subsets rather than identical behavior in every instance. Overall, constraint propagation mainly supports feasibility and tardiness control, while triggering, window construction, task locking, and adaptive weighting regulate adjustment scope and stability.

Figure 13 shows the percentage degradation after ablation, calculated as $(M_{ablation} - M_{Proposed})/M_{Proposed} \times 100\%$, where all metrics are lower-is-better. In Figure 14, h varies within $\{1, 2, 3\}$, while θ and ρ are perturbed around their validation-set optimal values to evaluate parameter sensitivity.

Table 8. Complete numerical results of the ablation experiments

Dataset	Variant	TT↓	RC↓	CVR (%)↓	SDI↓	Adjusted tasks (%)↓
OR-Library	w/o BIM-DT	113.6 $\pm$ 5.1	36.9 $\pm$ 2.2	4.8 $\pm$ 0.5	0.101 $\pm$ 0.008	28.7 $\pm$ 1.9
OR-Library	Static Ω_t	110.8 $\pm$ 4.9	35.8 $\pm$ 2.0	4.2 $\pm$ 0.4	0.096 $\pm$ 0.007	26.1 $\pm$ 1.7
OR-Library	w/o Trigger	101.9 $\pm$ 4.4	39.5 $\pm$ 2.1	2.3 $\pm$ 0.3	0.112 $\pm$ 0.008	34.2 $\pm$ 2.1
OR-Library	w/o Window	103.2 $\pm$ 4.6	44.8 $\pm$ 2.5	2.7 $\pm$ 0.4	0.124 $\pm$ 0.010	37.6 $\pm$ 2.4
OR-Library	w/o Locking	105.4 $\pm$ 4.2	41.7 $\pm$ 2.1	2.4 $\pm$ 0.3	0.117 $\pm$ 0.009	35.1 $\pm$ 2.2
OR-Library	w/o Adaptive	107.1 $\pm$ 4.8	38.2 $\pm$ 1.9	2.9 $\pm$ 0.4	0.106 $\pm$ 0.007	30.8 $\pm$ 1.8
OR-Library	Proposed	96.4 $\pm$ 3.5	34.1 $\pm$ 1.6	1.8 $\pm$ 0.2	0.085 $\pm$ 0.005	24.7 $\pm$ 1.3
C-MAPSS	w/o BIM-DT	146.8 $\pm$ 5.9	43.5 $\pm$ 2.3	5.9 $\pm$ 0.6	0.121 $\pm$ 0.009	33.8 $\pm$ 2.1
C-MAPSS	Static Ω_t	143.5 $\pm$ 5.6	42.4 $\pm$ 2.2	5.1 $\pm$ 0.5	0.116 $\pm$ 0.008	31.6 $\pm$ 1.9
C-MAPSS	w/o Trigger	132.8 $\pm$ 5.0	46.7 $\pm$ 2.5	3.0 $\pm$ 0.4	0.128 $\pm$ 0.009	37.4 $\pm$ 2.3
C-MAPSS	w/o Window	136.1 $\pm$ 5.2	51.4 $\pm$ 2.8	3.6 $\pm$ 0.4	0.139 $\pm$ 0.011	41.5 $\pm$ 2.7
C-MAPSS	w/o Locking	138.7 $\pm$ 5.0	48.6 $\pm$ 2.5	3.1 $\pm$ 0.4	0.132 $\pm$ 0.009	39.3 $\pm$ 2.4
C-MAPSS	w/o Adaptive	140.2 $\pm$ 5.6	44.9 $\pm$ 2.2	3.8 $\pm$ 0.5	0.123 $\pm$ 0.008	35.2 $\pm$ 2.0
C-MAPSS	Proposed	126.5 $\pm$ 4.3	40.1 $\pm$ 1.8	2.6 $\pm$ 0.3	0.097 $\pm$ 0.006	27.8 $\pm$ 1.5

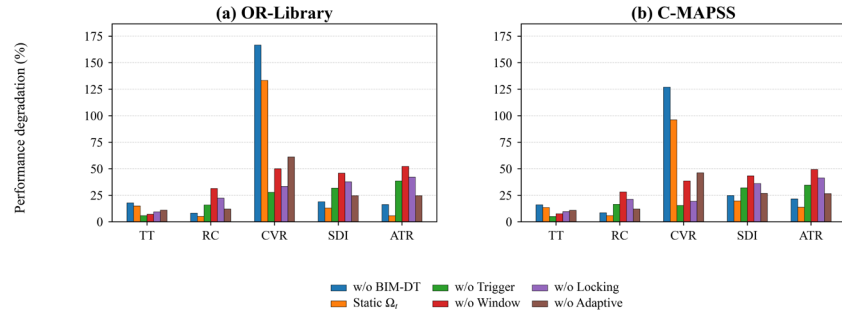


Figure 13. Ablation-induced Performance Degradation on Two Reconstructed Disturbance Scenarios:
(a) OR-Library;
(b) C-MAPSS

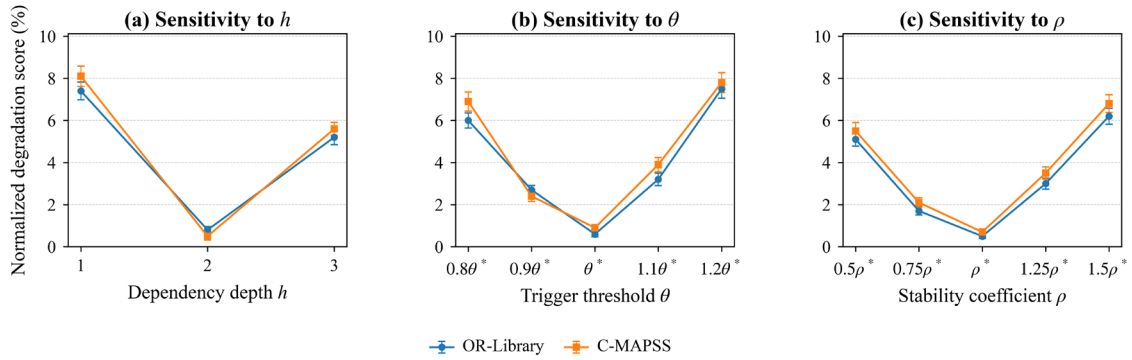


Figure 14. Sensitivity Analysis of Key Parameters:
(a) Dependency Depth h ;
(b) Trigger Threshold θ ;
(c) Stability Coefficient ρ

5. Discussion

This paper proposes a stable local rescheduling framework based on BIM-DT state-updated constraint propagation to address the insufficient coupling among BIM/IFC constraints, digital-twin feedback, and post-disturbance rescheduling in prefabricated component manufacturing. Unlike rule-based, constraint-optimization, metaheuristic, and reinforcement-learning approaches, the method does not treat disturbances as isolated trigger labels and avoids global re-optimization under locally repairable disturbances. Instead, its closed-loop process of “constraint generation–state updating–task identification–local repair” transforms BIM/IFC relationships and equipment-state deviations into dynamic constraints, affected tasks, trigger signals, and repair boundaries.

The BIM-DT State-Updated Constraint Propagation module transforms floor sequences, assembly dependencies, operation precedence, mold compatibility, and resource

capacity into initial constraints. It then updates the feasible region according to equipment failures, processing delays, capacity changes, and deadline adjustments. Compared with semantic digital-twin scheduling [18], BIM–RFID monitoring [27], due-date-uncertainty scheduling [31], and DRL-based precast scheduling [32], the proposed method specifically connects BIM/IFC dependencies and digital-twin deviations with constraint updating, affected-task identification, and repair-boundary determination. This represents a difference in the decision pathway rather than universal superiority. The local rescheduling module limits adjustments to affected tasks and their dependency neighborhoods, while task locking protects completed, non-interruptible, or high-cost tasks. When the affected-task ratio exceeds $\eta = 0.45$ or local repair is infeasible, the hybrid fallback expands the window to all unlocked tasks. The adaptive stability weight further adjusts the deviation penalty according to disturbance scope.

In current factory practice, disturbances are often handled through periodic schedule review or manually initiated large-

scale adjustment. The proposed workflow would instead convert real-time state deviations into event-triggered constraint updates and bounded repair recommendations delivered through MES/ERP. Production planners would retain final approval through the candidate-acceptance step, allowing gradual integration without replacing existing execution systems. Because the present validation uses reconstructed data, this operational impact remains prospective rather than empirically demonstrated.

Experiments reconstructed from OR-Library and NASA C-MAPSS show that the proposed method generally achieves lower tardiness, rescheduling cost, schedule deviation, and adjusted-task ratio than most baselines. Compared with ND3QNP, it reduces TT from 101.8 to 96.4 on OR-Library and from 131.7 to 126.5 on C-MAPSS. Paired t-tests provide preliminary within-setting evidence for these differences, although the five repeated runs limit statistical power and population-level inference. Ablation results indicate that constraint propagation, state triggering, local-window construction, task locking, and adaptive stability control contribute differently to the evaluated metrics. Representation analyses provide descriptive correspondences among state deviations, constraint contributions, trigger scores, and rescheduling decisions. Relative to ND3QNP, the proposed method achieves rescheduling-cost saving rates of 4.2% and 4.1% and delivery-delay reduction rates of 5.3% and 3.9% on OR-Library and C-MAPSS, respectively.

The framework provides a reproducible constraint-driven repair method that reduces unnecessary schedule changes under locally repairable scenarios, with conditional global fallback for extreme disturbances. Nevertheless, the experiments use reconstructed public data rather than complete factory BIM-DT data. BIM/IFC constraints are represented mainly through scheduling mappings and do not fully capture complex geometric, spatial, and assembly semantics. C-MAPSS state patterns may also differ from those of actual production equipment, while uniformly adapted baselines may not reflect their optimal performance in their original settings. Moreover, the economic indicators are operational proxies because monetary unit costs and actual factory delivery records are unavailable.

For industrial deployment, the framework can be implemented progressively through factory sensing and edge synchronization, BIM-DT integration, rescheduling-service connection, and MES/ERP-assisted execution, as shown in Figure 1(b). Implementation requires mapping heterogeneous sensor, IFC, and production data to a synchronized state model and connecting the rescheduling module through standard interfaces. Key barriers include incomplete IFC attributes, legacy-system compatibility, delayed sensor streams, and communication latency. These issues can be addressed through configurable data adapters, edge filtering and synchronization, latency-aware triggering, and planner approval before execution.

Although recent studies have reported BIM-, IoT-, and digital-twin-enabled platforms for precast production, published evidence based on complete real-factory BIM-DT scheduling records remains limited. Future research will combine real BIM/IFC models, production records, sensor

streams, disturbance logs, and factory schedules to validate practical applicability. The data will be aligned through component and equipment identifiers and timestamps, used to calibrate state-constraint mappings and trigger thresholds, and evaluated through historical replay followed by shadow-mode deployment. Disturbance logs will be used to reconstruct equipment failures, delays, resource shortages, and deadline changes, while factory schedules will provide references for comparing the proposed method with manual rescheduling decisions. Performance will be compared with factory plans and manual rescheduling using TT, RC, CVR, SDI, adjusted-task ratio, and runtime. Further work will examine state-deviation calibration, feasibility pre-assessment, adaptive dependency propagation, backup-resource substitution, and adaptive local-global switching thresholds under large-scale disturbances, extreme deadline compression, and uncertain production environments.

6. Conclusion

This study developed a BIM-DT state-updated constraint propagation framework for disturbance-oriented rescheduling in prefabricated component manufacturing. The framework maps floor sequences, assembly dependencies, operation precedence, mold compatibility, and resource capacity derived from BIM/IFC into production constraints and updates the feasible region according to digital-twin state deviations. State-triggered activation, dependency-based repair-window construction, task locking, adaptive stability weighting, and conditional local-global switching are integrated to regulate the timing and scope of schedule adjustment.

Under the reconstructed OR-Library and C-MAPSS scenarios, the proposed method achieved lower total tardiness, rescheduling cost, schedule deviation, and adjusted-task ratios than most comparison methods. Compared with ND3QNP, total tardiness decreased from 101.8 to 96.4 on OR-Library and from 131.7 to 126.5 on C-MAPSS. The ablation results further indicate that constraint propagation mainly supports feasibility and tardiness control, whereas state triggering, local-window construction, task locking, and adaptive stability weighting regulate adjustment scope and schedule stability. The hybrid fallback also provides a wider repair option when the affected-task ratio is high or local repair becomes infeasible.

These findings support the proposed mechanism under reconstructed disturbance scenarios rather than demonstrate factory-level effectiveness. The principal limitations include the use of reconstructed public data, simplified BIM/IFC semantic mappings, possible differences between C-MAPSS state patterns and actual production equipment, uniformly adapted baseline settings, and proxy-based economic indicators. Further validation should use real factory BIM/IFC models, production records, sensor streams, disturbance logs, and schedule data through historical replay and shadow-mode deployment.

Acknowledgements

This study is supported by The 2025 Ministry of Education Industry School Cooperation Collaborative Education Project, Research on Multi level Collaboration System of Intelligent Robot Prefabricated Construction Operation Integration Platform Based on Intelligent Prefabricated Components (Project Number: 2511130149)

References

- [1] Liu, H., & Zainul Abidin, N. (2024). A review on research of prefabricated building costs: exploring collaborations, Intellectual Basis, and research trends. *Sustainability*, 16(22), 9823.
- [2] Wang, L., Teng, D., Jin, R., Guo, X., Meng, X., Liu, G., ... & Su, Z. (2025). The development of prefabricated buildings and intelligent construction based on digital twins. *Journal of Intelligent Construction*, 3(1), 1-25.
- [3] Yang, Y., Yu, Y., Yu, C., & Zhong, R. Y. (2024). Data-driven logistics collaboration for prefabricated supply chain with multiple factories. *Automation in Construction*, 168, 105802.
- [4] Wang, Z., & Xu, H. (2024). Scheduling redundancy optimization for precast production to enhance disruption management in prefabricated construction. *Computers & Industrial Engineering*, 192, 110251.
- [5] Ho, C., Zabinsky, Z. B., & Kim, Y. W. (2026). Scheduling the production of prefabrication construction supply chains considering variable delivery times. *Journal of Civil Engineering and Management*, 32(3), 415-432.
- [6] Liang, J., Cao, Z., Zu, Q., Huang, H., & Guo, S. (2024). Production sequencing and layout optimization of precast concrete components under mold resource constraints. *Buildings*, 14(10), 3173.
- [7] Zhang, C., Jiang, J., Xia, C., Fu, Y., Liu, J., & Duan, P. (2024). Dual-objective optimization of prefabricated component logistics based on JIT strategy. *Scientific Reports*, 14(1), 31267.
- [8] Amarkhil, Q., & Elwakil, E. (2026). A novel framework for structuring and automating schedule-ready data from BIM for advanced construction scheduling. *J. Inf. Technol. Constr.*, 31, 651-673.
- [9] Gao, M. Y., Han, J., Yang, Y., Tiong, R. L., Zhao, C., & Han, C. (2024). BIM-based and IoT-driven smart tracking for precast construction dynamic scheduling. *Journal of Construction Engineering and Management*, 150(9), 04024117.
- [10] Chacón, R., Posada, H., Ramonell, C., Jungmann, M., Hartmann, T., Khan, R., & Tomar, R. (2024). Digital twinning of building construction processes. Case study: A reinforced concrete cast-in structure. *Journal of Building Engineering*, 84, 108522.
- [11] Li, S., Peng, C., Chen, G., Liang, Y., & Xu, Z. (2025). Swarm-intelligence collaboration based regular scheduling and dynamic rescheduling of precast component production: in prefabricated building project management. *Smart Construction*, 2(1), 1-30.
- [12] Zheng, R., Li, Z., Li, L., Dou, Y., Yuan, M., & Yin, X. (2024). Proposing a lean-optimized scheduling model of mixed-flow prefabricated component production in off-site construction. *Journal of Construction Engineering and Management*, 150(8), 04024088.
- [13] Zhao, Y., Du, S., Tu, M., Ma, H., Shang, J., & Xiang, X. (2025). Multi-Objective Scheduling Optimization of Prefabricated Components Production Using Improved Non-Dominated Sorting Generic Algorithm II. *Buildings*, 15(5), 742.
- [14] Dan, Y., Sun, C., & Luo, Q. (2025). Integrated scheduling optimization for multi-line production and transportation of prefabricated components considering shared resources. *Buildings*, 15(2), 187.
- [15] Yu, M., Ruan, W., Zhou, Y., & Zhao, Y. (2025). Flow shop scheduling for prefabricated components production considering parallel machines and buffer constraints. *Journal of Construction Engineering and Management*, 151(4), 04025010.
- [16] Qian, Y., Mao, N., Yu, J., & Shi, Q. (2025). Optimization of Precast Concrete Production with a Differential Evolutionary Algorithm. *Buildings*, 15(23), 4226.
- [17] Zhang, D., Zhang, H., Chen, D., Zhang, Y., & Wu, X. (2024). A review of production scheduling for prefabricated building components. *Voice of the Publisher*, 10(1), 11.
- [18] Kosse, S., Betker, V., Hagedorn, P., König, M., & Schmidt, T. (2024). A semantic digital twin for the dynamic scheduling of industry 4.0-based production of precast concrete elements. *Advanced Engineering Informatics*, 62, 102677.
- [19] Yu, M., Wang, L., Ruan, W., & Zheng, W. (2025). Robust Optimization for Precast Component Production Scheduling Considering Component Transfer. *Journal of Construction Engineering and Management*, 151(10), 04025156.
- [20] Li, Y., Zhao, Q., Yang, M., Ma, Z., & Hei, X. (2025). Advancements and Applications of Industry Foundation Classes Standards in Engineering: A Comprehensive Review. *Buildings*, 15(16), 2927.
- [21] Doan, D. T., Atencio, E., Muñoz La Rivera, F., & Alnajjar, O. (2025). A Systematic Literature Review of Building Information Modelling (BIM) and Offsite Construction (OSC) Integration: Emerging Technologies and Future Trends. *Applied Sciences*, 15(18), 9981.
- [22] Aghajamali, K., Rahimi, A., Metvaei, S., Lei, Z., Chen, Q., Suliman, A., & Pirzad, A. (2026). Production planning with an enriched BIM-Based heuristic construction order model to improve the efficiency of complex prefabricated projects. *Journal of Construction Engineering and Management*, 152(6), 04026057.
- [23] Belsky, M., Sacks, R., & Brilakis, I. (2016). Semantic enrichment for building information modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(4), 261-274.
- [24] Liu, Z., Li, X., Gao, Z., Zhang, Y., Teng, Y., & Wu, C. (2025). A hybrid and intuitive work packaging approach with multiple task relations, general work package precedence, and BIM in modular construction. *Journal of Construction Engineering and Management*, 151(4), 04025019.
- [25] Chong, H. Y., Yang, X., Goh, C. S., & Luo, Y. (2025). BIM and AI Integration for Dynamic Schedule Management: A Practical Framework and Case Study. *Buildings*, 15(14), 2451.
- [26] Osadcha, I., Jurelionis, A., & Fokaides, P. (2024). Patterns and trends in the use of RFID within the construction industry and Digital Twin architecture: A Latent Semantic Analysis. *International Journal of Sustainable Energy*, 43(1), 2421281.
- [27] Gao, M. Y., Han, C., Yang, Y., & Tiong, R. L. K. (2024). Integration of BIM and RFID-sensing technology for automated progress monitoring in modular construction. In *Proceedings of the 41st International Symposium on Automation and Robotics in Construction (ISARC 2024)* (pp. 1033–1039). International Association for Automation and Robotics in Construction.
- [28] Sheenan, S., Elghaish, F., Brooks, T., & Rahimian, F. (2026). IoT-Enabled Digital Twin for Autonomous Modular Construction Progress Monitoring. *ABC2: Journal of*

- Architecture, Building, Construction, and Cities, 2026(02), 23-37.
- [29] Elghaish, F., Mohandes, S. R., Rahimian, F., Abrishami, S., & Hosseini, M. R. (2026). Predictive digital monitoring of construction resources: An integrated digital twin solution. *Engineering, Construction and Architectural Management*, 33(8), 6094-6118.
- [30] Park, J., Kim, S., & Lim, J. (2025). Integrated Digital Twin and BIM Approach to Minimize Environmental Loads for In-Situ Production and Yard-Stock Management of Precast Concrete Components. *Applied Sciences*, 15(17), 9846.
- [31] Kim, T., & Kim, Y. W. (2024). Proactive Production Scheduling Approach for Off-Site Construction with Due Date Uncertainty. *Applied Sciences*, 14(23), 11017.
- [32] Yao, Y., Tam, V. W., Wang, J., Le, K. N., & Butera, A. (2025). Deep reinforcement learning-based schedule optimization for parallel precast production. *Engineering, Construction and Architectural Management*, 32(13), 285-318.
- [33] Zheng, T., Li, D., & Li, J. (2025). A Rescheduling Strategy for Multipurpose Batch Processes with Processing Time Variation and Demand Uncertainty. *Processes*, 13(2), 312.
- [34] Panwalkar, S. S., & Iskander, W. (1977). A survey of scheduling rules. *Operations research*, 25(1), 45-61.
- [35] Perron, L., Didier, F., & Gay, S. (2023). The CP-SAT-LP solver (invited talk). In 29th International Conference on Principles and Practice of Constraint Programming (CP 2023) (Vol. 280, pp. 3:1–3:2). Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
- [36] Li, X., & Gao, L. (2016). An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *International Journal of Production Economics*, 174, 93-110.
- [37] Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Applied Soft Computing*, 91, 106208.
- [38] Song, W., Chen, X., Li, Q., & Cao, Z. (2022). Flexible job-shop scheduling via graph neural network and deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, 19(2), 1600-1610.
- [39] Ren, F., & Liu, H. (2024). Dynamic scheduling for flexible job shop based on MachineRank algorithm and reinforcement learning. *Scientific reports*, 14(1), 29741.
- [40] Liu, Y., Zhang, F., Sun, Y., & Zhang, M. (2025). Evolutionary trainer-based deep Q-network for dynamic flexible job-shop scheduling. *IEEE Transactions on Evolutionary Computation*, 29(3), 749–763.
- [41] Yi, W., Deng, Q., Wang, B., Chen, Y., & Pei, Z. (2025). Dynamic flexible job shop scheduling problem considering multiple types of dynamic events. *International Journal of Production Research*, 63(24), 10291-10308.
- [42] Zheng, L., Chen, X., Zhuang, C., Liu, J., Zhang, Y., & Lai, L. (2025). Dynamic scheduling for flexible job-shop with reconfigurable manufacturing cells considering dynamic job arrivals based on deep reinforcement learning. *International Journal of Production Research*, 63(20), 7427-7459.