Moghimi, N., Shamaee, S. H., Yu, H., Mei, Q., Gonzalez, V. A., Hamzeh, F. (2026). A proactive digital twin framework for dynamic pull scheduling in modular construction factories. In F. Hamzeh, M. Poshdar, & N. P. Garcia-Lopez (Eds.), *Proceedings of the 34th Annual Conference of the International Group for Lean Construction* (IGLC34) (pp. 837–848). https://doi.org/10.24928/2026/0246

A PROACTIVE DIGITAL TWIN FRAMEWORK FOR DYNAMIC PULL SCHEDULING IN MODULAR CONSTRUCTION FACTORIES

Nima Moghimi[1], Sahar H. Shamaee[2], Qipei Mei[3], Vicente A. Gonzalez[4], & Farook Hamzeh[5]

ABSTRACT

Achieving continuous flow in modular construction is challenging due to high-mix variability, where traditional static methods often fail, leading to excess Work-in-Process (WIP) and flow stagnation. This study proposes a Digital Twin framework utilizing a novel "Offline Learning, Online Control" architecture. The methodology employs Simulation-Based Optimization to derive optimal control policies offline, which are synthesized into a lightweight surrogate model. This model drives an autonomous Scheduler Agent that acts as a virtual pacemaker, regulating real-time flow to shift production from a 'push' to a dynamic 'pull' strategy. Specifically, the agent monitors the cumulative count of finished units at critical downstream constraints, throttling upstream releases until the cumulative count of finished units from the current project surpasses a dynamically calculated threshold. To validate this approach, a case study was conducted on a wall panel assembly line in Edmonton, Canada. For this specific implementation, a Genetic Algorithm was used to train a linear regression surrogate policy against stochastic demand scenarios. Experimental results demonstrate a 38.5% reduction in average Cycle Time (Lead Time) compared to the facility's baseline practice. These findings confirm that integrating simulation-based learning with real-time surrogate control successfully stabilizes flow efficiency, minimizes WIP accumulation, and prevents gridlock in constrained manufacturing environments.

KEYWORDS

Lean construction, pull scheduling, modular construction, digital twin, simulation-based optimization

INTRODUCTION

The construction industry continues to face productivity loss, schedule delay, and quality inconsistency due to the temporary and site-based nature of conventional practice (Jeong et al.,

[1] Corresponding Author, Ph.D. Student, Department of Civil and Environmental Engineering, University of Alberta, Canada, nmoghimi@ualberta.ca, orcid.org/0009-0008-4733-1276

[2] Graduate Research Assistant, Department of Civil and Environmental Engineering, University of Alberta, Canada, hamedsha@ualberta.ca

[3] Assistant Professor, Department of Civil and Environmental Engineering, University of Alberta, Canada, qipei@ualberta.ca

[4] Hal Kvisle Professor and Tier 1 Canada Research Chair in Digital Lean Construction, Infrastructure Human Tech (IHT) Lab, Strategic Projects Insight Centre in Engineering (SPICE), Department of Civil and Environmental Engineering, Faculty of Engineering, University of Alberta, Canada, vagonzal@ualberta.ca, orcid.org/0000-0003-3408-3863

[5] Professor, Department of Civil and Environmental Engineering, University of Alberta, Canada, hamzeh@ualberta.ca, orcid.org/0000-0002-3986-9534

2022; Pervez et al., 2022). Industrialised construction has emerged as a broader shift toward factory-based production, controlled workflows, and greater process repeatability. Within this paradigm, modular construction represents a particularly demanding context because large work units, rigid outbound logistics, severe floor-space constraints, and substantial project-specific variability make production planning and scheduling significantly more complex than in conventional manufacturing (Abdul Nabi & El-adaway, 2020; Yang et al., 2019). These conditions create dynamic bottlenecks, interrupted value streams, and fluctuating Work-in-Process (WIP) levels that traditional static scheduling methods often fail to manage effectively (Salama et al., 2019; Zaheraldeen et al., 2019). The lack of real-time production visibility further increases the risk of excess inventory and shortage-induced delays (Eltahan et al., 2025; Gerber et al., 2010).

To manage this complexity, scheduling approaches have evolved from traditional push systems toward pull-based control. Push systems such as CPM rely on predetermined plans and often obscure flow inefficiencies by prioritizing local resource utilization (Slack et al., 2010). In contrast, pull systems release work only when downstream conditions can support it (Womack & Jones, 1997). In construction, this logic is reflected in the Last Planner System (LPS), while in production theory it is closely related to WIP-based control concepts such as CONWIP (Ballard & Tommelein, 2016; Hopp & Spearman, 2004; Liker, 2020; Pourrahimian et al., 2023). However, the high-mix and space-constrained nature of modular production often exceeds the capacity of manual pull signals, motivating simulation-based and data-driven approaches that can digitize the release logic (Fani et al., 2023).

Prior research in modular construction has increasingly used simulation, BIM integration, and optimization to support planning and scheduling (Brown et al., 2019). Discrete Event Simulation (DES) has been used to model factory operations and assess line balance (Moghadam & Al-Hussein, 2014; Wu et al., 2016), while BIM-linked approaches improve data extraction for production planning (Bhatia et al., 2022; Mayouf et al., 2024). Optimization and meta-heuristic methods have also been used to improve static schedules before execution (Gundogar et al., 2016). Yet these approaches largely remain open-loop: they generate plans from pre-production assumptions but do not continuously adjust release decisions to real-time shop-floor conditions (Alsakka et al., 2024). Digital Twin (DT) technology has improved synchronization between physical and virtual systems, but most current DT implementations remain reactive, focusing on monitoring and deviation reporting rather than proactive release control (Baghdadi, 2025; Bahroun et al., 2024; Moghimi et al., 2026).

To address this gap, this paper proposes a proactive Digital Twin (PDT) framework for dynamic pull scheduling in modular construction factories. The framework uses an autonomous Scheduler Agent that monitors the cumulative count of finished units from the active project and regulates the release of the next project through a learned WIP gate threshold. To enable real-time decision-making without repeated online optimization, the framework adopts an "Offline Learning, Online Control" architecture: stochastic simulation and evolutionary optimization are used offline to identify effective release thresholds across diverse scenarios, and the resulting decisions are synthesized into a lightweight surrogate policy for deployment. The framework is evaluated through a simulated case study of a wall-panel assembly line in a modular prefabrication factory in Edmonton, Canada.

The contributions of this study are threefold. First, it proposes a proactive Digital Twin framework for dynamic pull scheduling in modular construction factories based on an "Offline Learning, Online Control" architecture. Second, it demonstrates how simulation-based optimization can be translated into a lightweight surrogate policy for real-time project-release control under stochastic factory conditions. Third, it validates the approach in a wall-panel assembly line case study, showing that dynamic release control can substantially reduce cycle time while maintaining near-baseline throughput. Conceptually, the framework extends lean

pull logic beyond LPS and static CONWIP-style control by learning context-sensitive release thresholds from offline optimization and executing them through real-time downstream progress signals. It also moves beyond reactive DT approaches by using the digital twin not only for visibility, but for proactive release regulation under stochastic modular production.

METHODOLOGY

This study adopts Design Science Research (DSR), which is suitable for developing and evaluating an artifact intended to solve a practical organizational problem (Hevner et al., 2004). the research is structured into four phases: (1) Problem Identification and Motivation, which establishes the limitations of static push scheduling and manual pull signals under high-mix modular production; (2) Definition of Objectives, which specifies the need for a proactive, computationally efficient framework for real-time pull scheduling under stochastic uncertainty; (3) Design and Development, which formulates the Proactive Digital Twin (PDT) framework using an "Offline Learning, Online Control" architecture that combines simulation, optimization, and surrogate modeling; and (4) Demonstration and Evaluation, in which the framework is tested on a wall panel assembly line case study in Edmonton, Canada, using Cycle Time and Throughput as the primary performance metrics.

To bridge the gap between static planning and dynamic execution, this research proposes a PDT framework. Unlike reactive monitoring systems, the PDT operates as an autonomous closed-loop control system following an "Offline Learning, Online Control" paradigm. As shown in Figure 1, this architecture decouples the computational cost of iterative optimization from the low-latency requirements of real-time control.

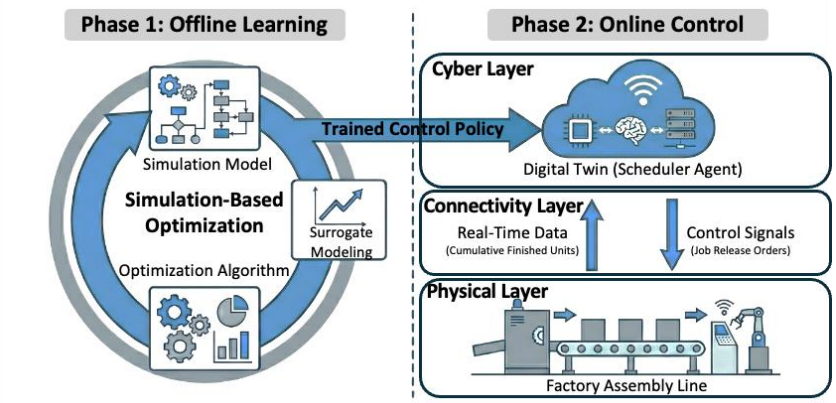

Figure 1: Structure of the Proactive Digital Twin (PDT) Framework.

Phase 1: Offline Learning (Strategic Planning): This phase generates system intelligence using a Simulation-Based Optimization (SBO) loop. High-fidelity simulations and optimization algorithms iteratively explore the solution space under stochastic uncertainty to identify optimal configurations (Moghimi & Kashani, 2026). The output is not a static schedule, but a comprehensive dataset of optimal states synthesized via Surrogate Modeling into a computable Trained Control Policy.

Phase 2: Online Control (Operational Execution): This phase executes the policy across three layers: the Physical Layer (factory assembly line); the Connectivity Layer (middleware transmitting finished unit counts upwards and release orders downwards); and the Cyber Layer (hosting the Scheduler Agent). Crucially, the Agent does not re-run simulations; instead, it uses the pre-trained policy to convert real-time data streams into instantaneous, optimized release decisions.

PHASE 1: OFFLINE SIMULATION-BASED POLICY LEARNING

The primary objective of the offline phase is to derive a control policy capable of autonomously regulating job releases to optimize system performance under stochastic conditions. While specific parameters may vary depending on the implementation, the framework adheres to a generalized three-step process: (1) modeling the factory dynamics, (2) optimizing decision variables via simulation-based search, and (3) synthesizing the results into a surrogate model.

Simulation Environment & Decision Variables

In the present implementation, the offline learning phase was executed using a hybrid Python/SimPy simulation of the wall panel assembly line described later in the case study. The model combines a Discrete Event Simulation (DES) layer for material flow and capacity constraints with an Agent-Based Simulation (ABS) layer for workforce behavior and task prioritization. This simulation environment was used to evaluate release-control decisions under stochastic processing times, finite buffers, and variable project compositions.

The key decision variable is the WIP gate threshold K_{ij} between two consecutive projects i and j. This threshold represents the number of finished downstream units from project i that must be completed before project j is released into production. Thus, the method does not optimize fixed project start times directly; instead, it optimizes pull-based release conditions that link upstream project release to downstream production progress.

In the case-study implementation, the project-transition state used for offline learning is described using a compact set of macroscopic predictors rather than a high-dimensional factory-state vector. These predictors are the difference in total small-panel counts between consecutive projects (ΔPanels), the difference in Multi-Wall counts (ΔMW), and the next-project density (ρ_{next}), which serves as a proxy for incoming batch complexity. During online execution, these project-specific features are combined with the real-time cumulative finished-unit count from the active project to evaluate whether the release condition has been satisfied.

Optimization Formulation

In the present study, the offline optimization phase was implemented using a Genetic Algorithm (GA) to search for release-threshold vectors K across simulated one-week production scenarios. For each scenario, the GA evaluates candidate threshold values by repeatedly running the hybrid stochastic simulation model and computing the following weighted objective function:

$$J(\mathrm{K}) = w_1 \cdot \left(\frac{CT(\mathrm{K}) - CT_{min}}{CT_{max} - CT_{min}}\right) - w_2 \cdot \left(\frac{TH(\mathrm{K}) - TH_{min}}{TH_{max} - TH_{min}}\right) \quad (1)$$

where *CT(K)* and *TH(K)* are the average cycle time and throughput obtained under threshold vector K, and w_1 and w_2 are user-defined weights representing the relative importance of cycle-time reduction and throughput preservation. In this implementation, equal weights were used. The normalization terms place both performance measures on a comparable scale despite their different units.

The GA therefore searches for the set of release thresholds that best balances lower cycle time with minimal throughput loss under stochastic production conditions. The resulting near-optimal threshold decisions from all simulated scenarios were then collected to form the training data used in the surrogate-learning stage.

Surrogate Policy Learning

Running the full simulation-optimization loop for every real-time decision is computationally infeasible. Therefore, the offline phase concludes by training a surrogate model that approximates the relationship between project-transition characteristics and the corresponding optimized release threshold.

Formally, let $\mathcal{D} = \{(S^{(1)}, K^{*(1)}), \dots, (S^{(N)}, K^{*(N)})\}$ be the dataset generated during the offline optimization phase, where $S^{(n)}$ is the state vector of the n-th scenario and $K^{*(n)}$ is the corresponding optimal threshold found by the optimization algorithm.

In practical terms, each training sample consists of a project-transition feature vector and the threshold value found by the GA to be most effective under stochastic simulation. The surrogate model is therefore trained to replace repeated simulation-based optimization with an instantaneous prediction rule that estimates the threshold directly from observable project characteristics.

The goal is to learn a function $\hat{\pi}: \mathcal{S} \to \kappa$ parameterized by θ that minimizes the prediction error over the training set:

$$\theta^* = \arg\min_{\theta} \sum_{n=1}^{N} \mathcal{L}\left(K^{*(n)}, \hat{\pi}(S^{(n)}; \theta)\right) \tag{2}$$

where $\mathcal{L}$ is a loss function (e.g., Mean Squared Error). The selection of an appropriate surrogate model follows a complexity spectrum tailored to the specific operational environment:

For scenarios characterized by high-dimensional, granular data, such as those involving complex component geometries or detailed material properties, the framework utilizes Non-Linear Approximators, including Deep Neural Networks (DNN) or Gaussian Process Regression (Kriging). These advanced architectures are necessary to accurately capture intricate, high-order interactions within the state space. Conversely, in environments defined by primary constraints or where model transparency is a priority, Linear Approximators provide a robust alternative that generalizes effectively under stochastic conditions.

In the present case-study implementation, a linear surrogate model was selected because the available predictors are macroscopic project descriptors and interpretability is important for operational deployment. The resulting control policy is expressed as:

$$\widehat{K} = \beta_0 + \sum_{j=1}^{m} \beta_j x_j + \epsilon \tag{3}$$

where, β_j represents the learned coefficients and x_j denotes the feature set derived from the state vector S_t.

The trained surrogate then functions as the computationally efficient decision engine for the online phase, enabling the Scheduler Agent to map the current project-transition state to a predicted threshold $\widehat{K}$ with negligible computational latency.

PHASE 2: ONLINE CONTROL VIA PROACTIVE DIGITAL TWIN

The online phase shifts the focus from learning to execution. In this phase, the Scheduler Agent operates within the cyber layer of the Proactive Digital Twin (PDT) as an autonomous release controller. The critical link between Phase 1 and Phase 2 is the transfer of the trained control policy. The optimization performed in Phase 1 is embedded into the Scheduler Agent through the learned model parameters (e.g., the regression coefficients $\beta_0, \dots, \beta_m$) As a result, the Agent does not need to run simulations or optimization routines in real time; instead, it evaluates a pre-trained mathematical policy using live production data.

Real-Time Data Ingestion and Validation

The foundation of the online control loop is the continuous data stream provided by the connectivity layer. In the present implementation, the key real-time input is the cumulative count of finished downstream units from the active project. This quantity can be obtained using several sensing technologies, ranging from RFID and computer vision to simple photoelectric sensors.

For the wall panel assembly line considered in this study, the cumulative finished-unit count can be captured at the loading stage. As panels are lifted from the final assembly station and

loaded onto transportation trailers, a blockage sensor such as a retro-reflective photoelectric beam installed at the bay door can increment the production count each time the beam is interrupted. This provides a discrete throughput signal without requiring complex image processing.

Because the effectiveness of the PDT depends on the reliability of this data stream, implementation requires signal filtering and validation procedures to ensure that the digital count reflects physical production progress. False positives or missed counts would distort the control signal and could lead to suboptimal release decisions.

The Closed-Loop Control Logic

The Scheduler Agent executes a continuous closed-loop algorithm to regulate the release of subsequent projects. In the present case-study implementation, the online control state is intentionally simple: the Agent uses the pre-trained surrogate model to predict the release threshold for the next project from its project-transition features, and it continuously compares that threshold against the cumulative finished-unit count of the active project.

For a linear surrogate policy, the predicted threshold is computed as:

$$K^*_{threshold} = \beta_0 + \sum_{j=1}^{m} \beta_j \cdot x_j \tag{4}$$

where x_j are the selected project-transition features for the pending release decision. The release decision is then evaluated as:

$$\text{Decision}_t = \begin{cases} \text{RELEASE } J_{i+1}, & \text{if } C_t \geq K^*_{threshold} \\ \text{HOLD}, & \text{otherwise} \end{cases} \tag{5}$$

Where C_t is the real-time cumulative finished-unit count from the active project J_i.

If the release condition is met, the Agent transmits a release order to the physical shop floor or ERP system to commence work on the next project upstream. Thus, K^* does not control downstream production directly; rather, it acts as a dynamic upstream release gate derived from downstream progress.

Operationally, the logic is straightforward: the Scheduler Agent continuously counts finished downstream units from the active project and compares this live count against the threshold predicted for the next project. If the predicted threshold is 6, for example, then the next project is withheld until 6 finished units from the current project have been completed. Once that condition is met, the next project is released upstream. In this way, the policy acts as a dynamic pull gate that spaces project releases according to actual downstream progress rather than fixed dates.

Importantly, this control rule does not delay downstream processing to accumulate inventory. Instead, it delays the release of the next upstream project until sufficient downstream progress has been achieved on the active project. In this sense, the completed-unit count serves as a digital pull signal: downstream completion authorizes upstream release. The threshold K^* therefore functions as a release gate rather than a downstream batching target.

FRAMEWORK IMPLEMENTATION

To validate the proposed PDT framework, a comprehensive case study was conducted within a real-world manufacturing environment. This section details the experimental setup, ranging from the physical facility characteristics to the stochastic operational constraints that necessitate a dynamic scheduling approach.

CASE STUDY DESCRIPTION

The validation domain is a flexible flow shop wall panel assembly line (Figure 2). Production flows sequentially from Framing through Sheeting and Nailing to Packaging. Subsequently, interior panels route to storage, while exterior panels traverse four parallel Door & Window

lanes before reaching the Loading buffer. The facility is characterized by severe spatial constraints, with inter-station buffers often limited to a single panel—a common challenge in volumetric construction. In this context, a "project" refers to one scheduled production job released into the wall-panel assembly line, while individual panels are the physical units processed through the stations shown in Figure 2. Thus, the control policy operates at the project-release level even though production flow occurs at the panel level. This site was selected due to the convergence of high process variability (heterogeneous projects) and rigid outbound logistics (limited to ~3 trailers/day). These constraints render traditional "push" strategies ineffective, as trailer unavailability frequently triggers cascading flow stagnation, making the line an ideal testbed for the proposed PDT.

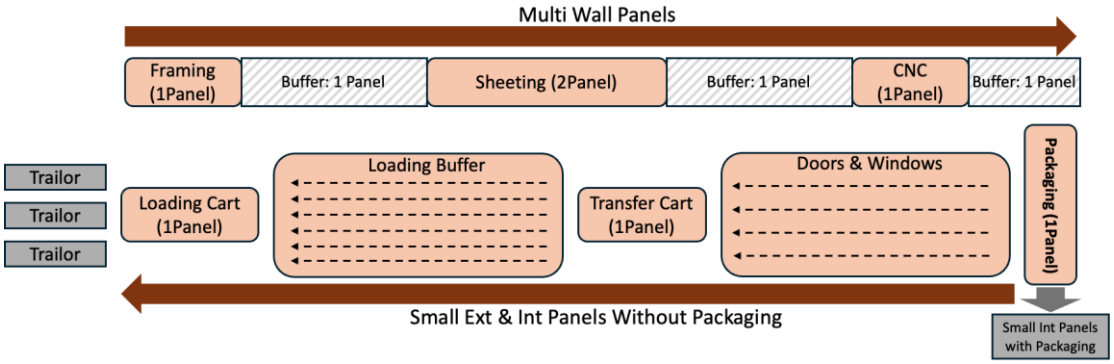

Figure 2: Schematic representation of the wall assembly line, illustrating station configurations, buffer capacities, and material flow paths.

SIMULATION MODEL DEVELOPMENT

To replicate the facility's stochastic dynamics, a hybrid Python/SimPy simulation was developed. A Discrete Event Simulation (DES) layer represents material flow, buffer capacities, and endogenous blocking, while an Agent-Based Simulation (ABS) layer governs workforce behavior, including multi-crew collaboration and task prioritization. The principal model inputs were: (1) the one-week validation schedule, (2) empirical processing-time distributions calibrated from plant observations, (3) workstation, crew, and inter-station buffer-capacity constraints, (4) project descriptors used in the release-threshold policy, and (5) outbound loading limitations reflecting trailer availability. The model was parameterized using 100–150 records per station. Because variability was attribute-driven, triangular distributions were fitted by panel type (interior vs. exterior).

Model validation compared simulated outputs against observed production behavior from the case facility, focusing on timing and performance patterns at key stations and at the overall production level. Agreement was assessed using R^2 and Mean Absolute Percentage Error (MAPE), yielding strong predictive accuracy (R^2: 0.88–0.95; MAPE: 16–18%), which is acceptable for construction simulation studies (Law et al., 2007).

In this case facility, electrical, plumbing, and similar service-integration activities are performed on site rather than on the wall-panel production line. Factory-side insulation work for exterior panels is captured implicitly through the modeled processing times, particularly at the Doors & Windows (DW) stage.

EXPERIMENTAL DESIGN AND FRAMEWORK CONFIGURATION

The offline learning phase was conducted on the wall panel assembly line shown in Figure 2. Thus, the "process" being simulated and optimized is the project-release control problem for this flexible flow-shop production line, where upstream releases must be coordinated with

downstream completion and loading constraints. The framework was configured through a structured three-stage process: generating realistic production scenarios, optimizing release thresholds via simulation, and synthesizing the control policy.

STOCHASTIC DEMAND GENERATION

In this study, the dataset used for offline learning was generated rather than directly observed as labeled training data. Specifically, 12 months of historical facility data were used to calibrate a stochastic scenario generator, which then produced 30 distinct one-week schedules representing low-, medium-, and high-workload conditions. These schedules define the production contexts on which the simulation-based optimization was performed. To ensure robustness against workload variability, a Hierarchical Stochastic Model was developed to generate 1-week look-ahead schedules, anchored in 12 months of historical facility data. This generator integrates four interdependent sub-models: (1) Project Occurrence modeled via a Poisson distribution $P(N_{proj} = k) = \frac{\lambda^k e^{-\lambda}}{k!}$ calibrated for Low ($\lambda = 8$), Medium ($\lambda = 15$), and High ($\lambda = 25$) workload scenarios; (2) Project Scope utilizing a uniform distribution for total panel counts; (3) Panel Mix employing conditional probabilities to determine exterior/interior ratios ; and (4) Deadline Assignment, which stochastically constrains loading windows to 0–2 days prior to delivery. Thirty distinct scenarios were synthesized using stratified sampling, yielding over 800 decision points for the training phase.

SIMULATION-BASED OPTIMIZATION CONFIGURATION

For each scenario, a Genetic Algorithm (GA) identified optimal WIP gate thresholds (K), dynamically constrained by the preceding project's small panel count ($0 \leq K_i \leq Q_{small}^{(i)}$) to ensure inventory feasibility. The GA minimized the weighted fitness function (Eq. 3) using equal weights for cycle time and throughput, normalizing against a "No-Gate" baseline simulation where projects were released immediately upon availability.

SURROGATE POLICY DERIVATION

For each simulated project transition within these generated schedules, the GA searched for the threshold value K that best balanced cycle-time reduction and throughput preservation under stochastic production dynamics. Each resulting training sample therefore consisted of: (1) a transition state between two consecutive projects, described by macroscopic project features, and (2) the corresponding threshold value identified as optimal for that transition. Collecting these samples across the 30 scenarios yielded the ~800 decision points. To translate offline optimization results into deployable real-time control logic, a surrogate model was trained on the given dataset. Linear Regression was selected for its interpretability and robustness given the available macroscopic project metrics. Through rigorous feature selection, three critical predictors were identified: the difference in total small panel counts between projects (ΔPanels), the difference in Multi-Wall counts (ΔMW), and the "Next Project Density" ($MW_{next}/Panel_{next}$), which serves as a proxy for incoming batch complexity. The “next project density” is defined as a compact proxy for the workload intensity of the incoming project, reflecting how much downstream congestion the next release is likely to create relative to its size.

The fitted linear model yields the following control equation for the Scheduler Agent, where K^* represents the optimal finished unit threshold:

$$K^* \approx 11.95 - 0.98(\Delta\text{Panels}) + 0.43(\Delta\text{MW}) - 2.62\left(\frac{MW_{next}}{Panel_{next}}\right) \quad (6)$$

Equation (10) is the linear surrogate policy obtained by fitting the GA-derived optimal thresholds from the offline scenarios. In practical terms, it predicts how many finished units from the current project should be completed before the next project is released, based on the relative size and complexity of the project transition.

Validation via Leave-One-Out Cross-Validation (LOOCV) yielded an (R^2) of 0.68. While this reflects the inherent limits of using macroscopic inputs to model complex stochastic dynamics, it establishes a strong baseline for preliminary control. The unexplained variance is likely attributable to granular geometric constraints not currently captured; future iterations leveraging detailed BIM data are expected to further enhance predictive power. Crucially for operational applicability, the model demonstrated a Mean Absolute Error (MAE) of 1.12 units. This indicates that predictions typically deviate from the theoretical optimum by only a single panel. Given the volumetric scale of factory operations, this margin is operationally negligible, confirming the linear surrogate provides a robust foundation for real-time control.

VALIDATION RESULTS: BASELINE VS. PROACTIVE DIGITAL TWIN

To quantify operational impact, the trained surrogate policy was validated using a real-world one-week production schedule comprising 14 projects and benchmarked against a Baseline simulation representing the factory's current static push practice. The PDT reduced average Cycle Time by 38.5% (202.22 min versus 328.65 min for the Baseline), with only a marginal difference in system throughput (3.519 vs. 3.668 panels/hour).

As shown in Figure 3, each bar represents one project from the validation week according to its simulated execution window on the wall-panel assembly line. Under the Baseline condition, projects are released using the facility's current static logic, resulting in elongated durations and substantial overlap between projects. Under the PDT condition, the surrogate-based threshold rule of Eq. (10) releases the next project only when downstream progress on the active project reaches the predicted threshold. This controlled staggering reduces overlap and downstream congestion, limits WIP buildup, and substantially improves average cycle time despite the slight reduction in gross throughput.

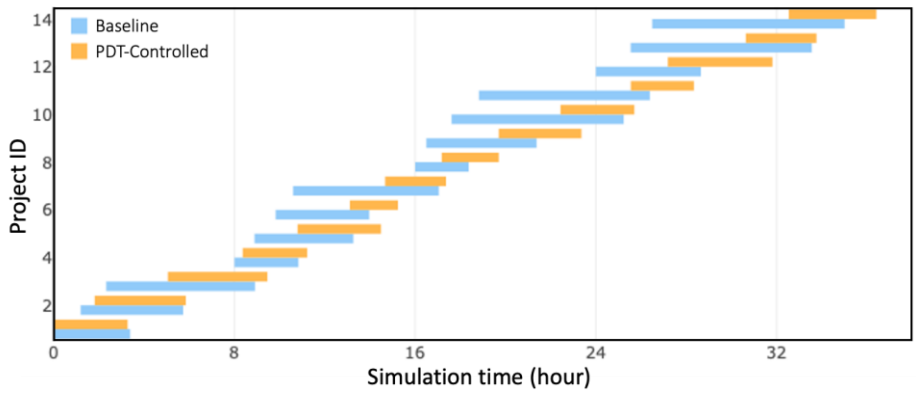

Figure 3: Comparative project execution timeline for the 14-project validation week: Baseline schedule vs. PDT-Controlled schedule.

DISCUSSION

The distinct temporal spacing observed under the PDT confirms the agent's ability to enforce a dynamic pull strategy. Downstream completion acts as the authorization signal for upstream release, replacing static schedules or manual pull cues with a learned real-time release gate. By synchronizing releases with downstream trailer availability, the framework reduces Mura (unevenness) and prevents the congestion typical of static push systems. Although the Baseline achieves slightly higher gross throughput (~4%), this comes at the cost of severe overlap and WIP buildup. The PDT's small throughput reduction is therefore a strategic trade-off for more reliable system-level flow and substantially lower cycle time.

The study advances prior work on simulation-based planning, modular production scheduling, and reactive digital twins by closing the control loop. Rather than using simulation

only for pre-production evaluation or digital twins only for visibility, the proposed PDT converts offline optimization results into a lightweight surrogate policy that actively regulates project release during production. This "Offline Learning, Online Control" architecture shows how computationally intensive optimization can be operationalized for real-time decision-making under stochastic factory conditions.

For modular factories, where floor space is often the dominant constraint, the 38.5% reduction in Cycle Time is strategically significant. The controlled release logic reduces project overlap, limits congestion in downstream constrained areas, and improves flow reliability. Because the surrogate policy was derived from optimization across multiple stochastic scenarios, these results should be interpreted as robust to workload variation rather than as an artifact of a single deterministic schedule.

From a practitioner perspective, implementation does not require real-time re-optimization. Instead, it depends on three enabling elements: (1) a calibrated simulation environment for offline learning, (2) a reliable method for tracking downstream completion progress, and (3) a lightweight decision rule that can be embedded into planning, ERP, or shop-floor workflows. Practical adoption may nevertheless face barriers, including sensing reliability, data integration, and organizational willingness to shift from static release logic to data-driven pull control.

While the framework is developed and validated in a modular construction context, its core logic may also be relevant to other factory-based construction systems, although adaptation would likely be required for settings with smaller, more standardized work units and different routing or spatial constraints. Relative to Digital Twin applications in manufacturing, which often focus on monitoring, prediction, maintenance, or line-level optimization, the present study addresses a different problem: proactive project-release regulation in a high-mix, space-constrained, project-based production environment.

CONCLUSION AND FUTURE WORK

Shifting construction activities to the factory floor creates scheduling challenges under high-mix variability and space-constrained conditions. This study presented a Proactive Digital Twin (PDT) framework that bridges static planning and stochastic execution through an "Offline Learning, Online Control" architecture. Using simulation-based optimization, the framework distills complex release decisions into a lightweight surrogate policy that enables dynamic pull scheduling in real time. The Scheduler Agent monitors finished-unit counts and regulates upstream release based on downstream progress, thereby controlling WIP and reducing congestion. The framework was validated on a wall panel assembly line in Edmonton, Canada. A Genetic Algorithm optimized release thresholds across stochastic scenarios, and the resulting policy was synthesized into a linear surrogate model for deployment. Compared with the facility's baseline static scheduling practice over a one-week production window, the PDT reduced average Cycle Time by 38.5%, with only a marginal reduction in gross throughput (4%). This trade-off improves flow reliability and helps prevent excess WIP accumulation and cascading blocking effects. Theoretically, the study contributes a proactive DT-based pull-scheduling logic that extends prior work on static planning and reactive digitalization. Practically, it offers a lightweight pathway for factories to regulate project release using downstream progress signals rather than computationally intensive online optimization. The findings should nevertheless be interpreted in light of the study's limitation to a single facility validation. Future work will extend testing to other factory contexts and explore integration with high-fidelity BIM data and more robust IoT infrastructure for real-time implementation.

ACKNOWLEDGEMENT

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Alliance grant ALLRP 581074–22, Discovery Grant RGPIN-2020-04420, and NSERC Alliance—Alberta Innovates Advance Program 232403286. All findings and conclusions expressed in this paper are those of the authors and do not reflect those of the contributors.

REFERENCES

Abdul Nabi, M., & El-adaway, I. H. (2020). Modular construction: Determining decision-making factors and future research needs. *Journal of Management in Engineering*, *36*(6), 04020085.

Alsakka, F., Yu, H., El-Chami, I., Hamzeh, F., & Al-Hussein, M. (2024). Digital twin for production estimation, scheduling and real-time monitoring in offsite construction. *Computers & Industrial Engineering*, *191*, 110173. https://doi.org/10.1016/j.cie.2024.110173

Baghdadi, A. (2025). A comprehensive review of digital twin implementation in construction: current trends and future directions. *Journal of Asian Architecture and Building Engineering*, 1–15.

Bahroun, Z., Shamayleh, A., As' ad, R., & Zakaria, R. (2024). Integrated proactive-reactive tool for dynamic scheduling of parallel machine operations. *International Journal of Engineering Business Management*, *16*, 18479790241301164.

Ballard, G., & Tommelein, I. (2016). Current process benchmark for the last planner system. *Lean Construction Journal*, *89*, 57–89.

Bhatia, A. P. S., Han, S., & Moselhi, O. (2022). A simulation-based statistical method for planning modular construction manufacturing. *Journal of Information Technology in Construction*, *27*.

Brown, R., Barkokebas, B., Ritter, C., & Al-Hussein, M. (2019). Predicting performance indicators using BIM and simulation for a wall assembly line. *Proceedings of the 27th Annual Conference of the International Group for Lean Construction (IGLC 2019), Dublin, Ireland*, 3–5.

Eltahan, A., Lee, G., & Hamzeh, F. (2025). Exploring Cognitive Load and Task Complexity in Dynamic Tracking Tasks: Insights for Construction Workflows. *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, *42*, 532–539.

Fani, V., Antomarioni, S., Bandinelli, R., & Bevilacqua, M. (2023). Data-driven decision support tool for production planning: a framework combining association rules and simulation. *Computers in Industry*, *144*, 103800.

Gerber, D. J., Becerik-Gerber, B., & Kunz, A. (2010). Building information modeling and lean construction: Technology, methodology and advances from practice. *Proceedings IGLC 18, Technion*, 683–693.

Gundogar, E., Sari, M., & Kokcam, A. H. (2016). Dynamic bottleneck elimination in mattress manufacturing line using theory of constraints. *SpringerPlus*, *5*(1), 1276.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 75–105.

Hopp, W. J., & Spearman, M. L. (2004). To pull or not to pull: what is the question? *Manufacturing & Service Operations Management*, *6*(2), 133–148.

Jeong, G., Kim, H., Lee, H.-S., Park, M., & Hyun, H. (2022). Analysis of safety risk factors of modular construction to identify accident trends. *Journal of Asian Architecture and Building Engineering*, *21*(3), 1040–1052.

Law, A. M., Kelton, W. D., & Kelton, W. D. (2007). *Simulation modeling and analysis* (Vol. 3). Mcgraw-hill New York.

Liker, J. (2020). *The Toyota way: 14 management principles from the world's greatest manufacturer*. McGraw-Hill.

Mayouf, M., Jones, J., Elghaish, F., Emam, H., Ekanayake, E., & Ashayeri, I. (2024). Revolutionising the 4D BIM process to support scheduling requirements in modular construction. *Sustainability*, *16*(2), 476.

Moghadam, M., & Al-Hussein, M. (2014). An enhanced scheduling technique for modular construction manufacturing. *Proceedings of the International Group for Lean Construction (IGLC-22)*.

Moghimi, N., Arshang, O., & Hamzeh, F. (2026). Digital twins in offsite construction: Current implementations, challenges, and future pathways. *Automation in Construction*, *183*, 106820.

Moghimi, N., & Kashani, H. (2025). A probabilistic framework to design the financing scheme for resilience enhancement of infrastructure systems. *Reliability Engineering & System Safety*, *262*, 111179. https://doi.org/10.1016/j.ress.2025.111179

Moghimi, N., & Kashani, H. (2026). Probabilistic Modeling to Develop the Optimal Seismic Resilience Enhancement Strategy for Industrial Facilities. *Journal of Infrastructure Systems*, *32*(1), 04025041. https://doi.org/10.1061/JITSE4.ISENG-2754

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77.

Pervez, H., Ali, Y., Pamucar, D., Garai-Fodor, M., & Csiszárik-Kocsir, Á. (2022). Evaluation of critical risk factors in the implementation of modular construction. *Plos One*, *17*(8), e0272448.

Pourrahimian, E., Shehab, L., & Hamzeh, F. (2023). Investigating and Simulating Collaboration among the Lps Phases. *31st Annual Conference of the International Group for Lean Construction (IGLC31)*, 929–941.

Salama, T., Salah, A., & Moselhi, O. (2019). INTEGRATED SCHEDULING OF MODULAR CONSTRUCTION USING "MOD-SCHEDULER." *Proceedings of the 7th International Canadian Society for Civil Engineering Construction Conference (CSCE 2019), Montreal, Canada*.

Sargent, R. G. (2010). Verification and validation of simulation models. *Proceedings of the 2010 Winter Simulation Conference*, 166–183.

Slack, N., Chambers, S., & Johnston, R. (2010). *Operations management*. Pearson education.

Song, L., & Eldin, N. N. (2012). Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling. *Automation in Construction*, *27*, 32–39.

Viana, D. D. (2015). *Integrated production planning and control model for engineer-to-order prefabricated building systems*.

Womack, J. P., & Jones, D. T. (1997). Lean thinking—banish waste and create wealth in your corporation. *Journal of the Operational Research Society*, *48*(11), 1148.

Wu, C., Jiang, R., & Li, X. (2016). Integration of BIM and computer simulations in modular construction, A case study. *Modular and Offsite Construction (MOC) Summit Proceedings*.

Yang, Y., Pan, M., & Pan, W. (2019). 'Co-evolution through interaction'of innovative building technologies: The case of modular integrated construction and robotics. *Automation in Construction*, *107*, 102932.

Zaheraldeen, H., Khoury, H., & Hamzeh, F. (2019). An AHP Approach for Selecting and Identifying Offsite Construction Systems. *Logistics and Supply Chain Management*, 73–82.