Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

SIMULATION-DRIVEN REINFORCEMENT LEARNING VS. SEARCH-BASED OPTIMIZATION FOR REAL-TIME CONTROL IN CONSTRUCTION MANUFACTURING

Ian Flood[1] and Madison E Hill[1]

[1]Rinker School, University of Florida, Gainesville, Florida, USA

ABSTRACT

This paper presents a comparative study of AI-based decision agents for real-time control in construction manufacturing, a field traditionally ill-suited to mass production due to high customization and uncertainty. Three approaches are developed and evaluated: an experience-based agent trained through reinforcement learning (EDA), a search-based agent using local stochastic sampling (SDA), and a hybrid combining both strategies (HybridDA). The goal is to minimize component delivery delays. A simulation model of a real precast concrete factory helps generate training patterns and assess production performance. Results show that the EDA significantly outperforms the SDA, despite relying on generalized experience rather than task-specific knowledge. The HybridDA provides modest gains, especially as search effort increases. These findings highlight the unexpected effectiveness of experience-based agents in dynamic manufacturing environments and the potential of hybrid approaches. Future work will explore alternative learning strategies, extend the scope of manufacturing decisions, and evaluate latency performance across the agents.

1 INTRODUCTION

1.1 The Construction Manufacturing Control Problem

Factory production of construction components has the potential to reduce inefficiencies associated with traditional on-site construction, such as low productivity and inconsistent quality (Durdyev and Ismail 2019; Rocha et al. 2023). However, achieving production efficiency in construction factories is significantly more challenging than in most other manufacturing industries. Mass production techniques fail to accommodate the unpredictable work demands of construction, which are characterized by irregular order arrival times, variations in batch size, and the customization of components (Flood and Flood 2022). Moreover, component designs vary not only across batches but also within batches, often with little to no repetition. Consequently, production has to be on-demand, precluding inventory stockpiling and requiring constant adaptation to changing resource needs. These challenges make it difficult to establish effective work control policies for managing decisions such as component dispatching, task scheduling, and resource allocation. A promising solution is to incorporate artificial intelligence (AI) agents into operations management, as they can recognize complex dependencies among system variables. In this context, the agents can act as advisors in a human-in-the-loop system or as autonomous controllers in an automated setting, offering solutions whenever operational decisions are needed.

AI-based decision agents have seen some success in controlling operations within the construction industry. In a study by Shitole et al. (2019), an agent was developed using an artificial neural network (ANN) trained with reinforcement learning (RL) to optimize a simulated earth-moving operation. The agent outperformed previously published, manually designed heuristics. RL, a learning technique that has gained significant success in recent years, improves decision-making by discovering and reinforcing effective behaviors (Sutton and Barto 2018). The study's earth-moving system included two excavators and a fleet of dump trucks, with the agent responsible for directing trucks to one of the excavators at a junction on the return road, aiming to maximize the system's overall production rate. Hatami and Flood (2024) explored a similar approach, using an RL-trained ANN decision agent to optimize open-pit mining operations, where 40 thirty-ton trucks are dispatched to nine types of excavators in a cyclical system. The RL-trained ANN agent outperformed the established rule-of-thumb (RoT) approach by approximately a factor of ten in the system studied, with both approaches evaluated against a random selection policy as the baseline.

Other studies have demonstrated the effectiveness of the RL-trained ANN agent approach in optimizing factory-based manufacturing operations. For instance, Waschneck et al. (2018) applied the method to production scheduling in a simulated semiconductor manufacturing environment, achieving promising results compared to conventional dispatching rules. Similarly, Zhou et al. (2020) used this concept for job-shop scheduling in smart manufacturing, demonstrating improved scheduling performance over traditional methods. Xia et al. (2021) employed a digital twin environment to train RL agents for smart manufacturing plants, demonstrating enhanced control and adaptability in manufacturing processes. These studies highlight the potential of RL-trained ANN agents to enhance efficiency and adaptability in manufacturing settings. However, these applications have largely been outside the scope of construction manufacturing, and as such, do not address the unique challenges faced by this industry.

Flood and Flood (2022) demonstrated through a proof-of-concept study that an RL-trained deep ANN can significantly outperform a hand-crafted RoT approach for decision-making in construction factory control. Recent studies (Flood and Zhou, 2023; Zhou and Flood, 2024) further reinforce these findings by optimizing production control through systematic variation of key model parameters related to the structure of the ANN agent and the hyperparameters of the RL algorithm. These advancements build upon earlier work by Flood (1989) and Kim et al. (2022), which focus on offline methods limited to sequencing predefined sets of prefabricated reinforced concrete (PRC) components.

Heuristic search methods offer an alternative AI approach to ANN-based agents for optimizing construction factory processes. Researchers such as Benjaoran and Dawood (2005), Chan and Hu (2002), Leu and Hwang (2001), Dan et al. (2021), and Yang et al. (2016) have successfully employed genetic algorithms (GAs) for this purpose, demonstrating improvements in production efficiency and cost reduction compared to customary rules. Additional heuristic-based methods that have found application in construction manufacturing include: Ant Colony Optimization (ACO), which mimics ant behavior to find optimal paths and has been used for sequencing and balancing mixed production lines (Chen et al. 2022); Simulated Annealing (SA), a probabilistic technique applied to precast scheduling under multiple constraints (Podolski 2022); and Tabu Search (TS), which utilizes memory structures to explore complex solution spaces and has been shown effective for resource allocation and repetitive scheduling (Glover and Laguna 1999). Hybrid approaches combining ANNs with GAs have also been explored in construction manufacturing. For example, Haq et al. (2009) developed a hybrid ANN-GA model for permutation flow shop scheduling to minimize makespan, though it was limited to offline optimization of a fixed set of jobs.

1.2 Research Aim

In summary, decision agents can be broadly categorized based on their reasoning method: experience-based, search-based, or a hybrid of these two approaches (Flood and Flood 2022). Experience-based agents, which include RoTs and ANNs, are developed from past successes and failures observed in real or simulated systems, from which they attempt to generalize to new problem variants. In contrast, search-based agents, such as GAs and blind search methods, systematically explore the solution space within a model of the real system by directly testing potential candidates, without utilizing knowledge gained from previous problem experiences. Hybrid agents integrate elements of both approaches by, for example, drawing on prior experience as an initial approximation and refining it through systematic search within the model.

Recognizing this categorization is important, as experience-based and search-based decision agents each offer distinct benefits and challenges in manufacturing control, differing in their ability to optimize production performance, ensure timely decision-making, and adapt to new problem variations. The relative advantages of these agents are not always as evident as they might initially seem. For example, search-based agents may be expected to outperform experience-based agents in optimizing production performance, as they tailor solutions to the specific problem at hand rather than generalizing from similar past situations. However, some research suggests the opposite, as demonstrated by Vala et al. (2011). Their study compared an experience-based method (an ANN) with a search-based method (a GA) for predicting truck types based on the dynamic strain response of steel bridges, finding that the former outperformed the latter in prediction accuracy. Likewise, the relative advantages of experience-based and search-based agents, in terms of timely decision-making and adaptability to new problem variations, remain underexplored. This study will focus on comparing the ability of different agent types to optimize production, specifically an RL-trained ANN agent, a search agent, and a hybrid of these two approaches. Other performance metrics, such as adaptability and decision-making speed, will be addressed in a follow-up study.

2 DESCRIPTION OF THE DECISION AGENTS

Figure 1 illustrates the processes and their sequence implemented at the PRC component factory, with detailed explanations of each element provided in Section 3. A decision agent manages production by selecting a PRC component from the appropriate queue whenever a process becomes available to start processing. For the case study outlined in Section 3, PRC component bottlenecks were consistently observed only at the Rebar process. Consequently, decision agents were deployed specifically at that process, as shown in Figure 1; however, if bottlenecks were to arise at other processes, decision agents would need to be developed for them and added accordingly. For all other processes not managed by a decision agent, the system defaults to selecting the PRC component at the front of the queue. Upon entering a queue, components are sorted using the RoT criteria, previously defined in Section 2.4, to determine their processing priority.

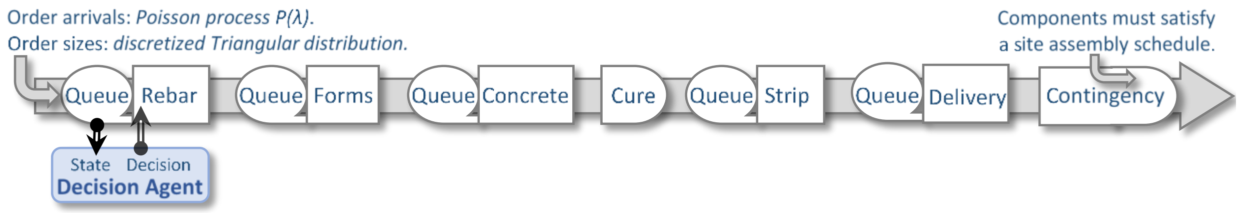

Order arrivals: *Poisson process P(λ).*
Order sizes: *discretized Triangular distribution.*
Components must satisfy a site assembly schedule.
Queue
Rebar
Queue
Forms
Queue
Concrete
Cure
Queue
Strip
Queue
Delivery
Contingency
State
Decision
Decision Agent

Figure 1: Simulation-based process model of PRC component production, used for exploring and generating training patterns and validation of decision agents (adapted from Flood and Zhou (2023)).

The performance of a decision agent is measured in terms of the delays to the delivery of individual PRC components within a sequence, with smaller delays indicating better performance. More specifically, the cost function used for this purpose is the root-mean-square (RMS) of the delivery delays, resulting from the fabrication process, computed across all PRC components in a specified sequence, as defined by Equation (1). Note that a PRC component could be delivered early, indicated by a negative delay. However, the square operation in the RMS calculation would cancel the negative sign, treating the early delivery as an equivalent delay. Therefore, the delays in this function are offset relative to a base value, *b*, to give emphasis to actual delays, rather than early deliveries:

$$cost = \sqrt{\sum_{i=1}^{n}(d_i + b)^2 / n} \tag{1}$$

where:

d is the delay for the *i* th PRC component on its completion;
n is the number of PRC components completed in the sequence being evaluated;
b is the base value against which the delays are offset - this value is the maximum contingency time (spare time) possible for a PRC component.

2.1 Experience-Based Decision Agents (EDA)

The experience-based decision agent (EDA) used in this study is based on a layered feedforward ANN trained with the RL algorithm developed by Flood and Flood (2022), with its operation summarized in Figure 2. When the EDA is required to make a decision, the ANN part of the agent receives inputs describing the state of the manufacturing system, specifically the remaining process durations and the remaining contingencies for the PRC components currently in the system. These data are normalized at the input for each process, with the position of each value indicating both the PRC's place in the queue and the specific process it relates to. All hidden units in the ANN use the ReLU (rectified linear unit) activation function, due to its computational efficiency and its ability to mitigate the vanishing gradient problem (Glorot et al., 2011). The number of layers and units per hidden layer in the ANN are established by experimentation in Section 4.1. The ANN's output layer is responsible for selecting PRC components from the target queue for processing. All output units use a sigmoid activation function, which constrains their outputs to values between 0.0 and 1.0. The number of output units is limited to N, representing the maximum number of PRC components that the EDA is designed to evaluate. For any given decision, the number of active output units is determined by the smaller of *N* or the current queue length. The outputs from these active units are normalized to sum to 1.0, allowing them to be interpreted as probabilities for selecting PRC components from the queue. The EDA has two modes of operation, illustrated to the right side of Figure 2, which are defined by how the outputs of the ANN are interpreted:

- Stochastic Decision Making for Exploration. This mode steers the simulation along alternative and partially random decision paths to collect high-reward input-output pattern pairs used to train the ANN during the reinforcement learning process. Monte Carlo sampling is used to select PRC components based on the values produced by the relevant output units; the higher the value, the greater the likelihood that the corresponding PRC component will be chosen.
- Deterministic Decision Making for Implementation. This mode directs the system by selecting the PRC component associated with the ANN output unit that generates the highest value. The operation is entirely deterministic and is used to control the simulated system during non-training runs to validate the performance of the current EDA. This is also the mode intended for deployment when using the EDA to control the real system.

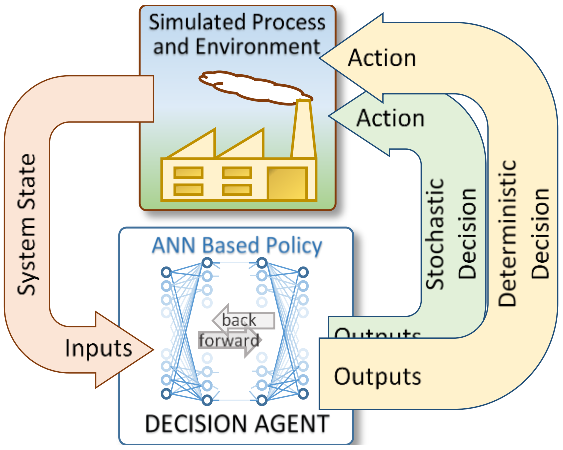


Figure 2: Operation modes of the EDA: stochastic for exploration; and deterministic for implementation.

Figure 3 illustrates the overall RL strategy used to train the ANN component of the EPA. The process cycles through three phases: Phase I involves collecting training patterns by exploring alternative decision paths and their associated delivery performances; Phase II focuses on training the ANN using the patterns gathered during the most recent Phase I; and Phase III validates the EDA by running the system through a fixed sequence of PRC component batch orders that were different from those used during training. These phases are described in more detail in the following subsections:

- Phase I: Exploration. This phase generates training patterns by exploring different decision paths within a double-nested loop of simulated production scenarios, as shown on the left side of Figure 3. Each trial simulates the fabrication of a set of *n* PRC components, referred to as the *reward length*. A trial is repeated *t* times using the same sequence of PRC components, but with slightly varied decisions determined stochastically, as described in reference to Figure 2. The trial with the best delivery performance (see Equation 1) is selected for later training of the ANN and serves as the lead-in for the next stage of the simulation, contributing *n* training patterns. These patterns represent the mappings from input to output for each state transition in the selected trial. This trial-testing process, called a stage, is repeated *s* times, each with a different sequence of PRC components and continuing from the previous best trial. As a result, a total of $p=n\times s$ training patterns are produced at each RL training iteration.
- Phase II: Training. This phase uses the training patterns generated in the exploration phase to train the ANN, or to fine-tune train it in repeated iterations through the three phases, as shown in the middle of Figure 3. The ANN and its trainer were implemented in Python (Van Rossum, 1995) and PyTorch (Paszke et al., 2019), using the RMSProp (root-mean-square propagation) optimizer and the MSELoss (mean-squared-error) loss function, with reduction set to 'mean'. Data loading used a mini-batch size of 64 with shuffling enabled. The learning rate was set to 0.001. Training was conducted until the output from the loss function had converged, which was typically within 1,000 epochs.
- Phase III: Validation. After training, the system moves to the validation phase, as shown on the right side of Figure 3. This phase involves running the simulation in deterministic mode (see Section 2.1), using a new sequence of *m* PRC components that were not used during training. The validation sequence is used to assess the delivery performance of the ANN (see Equation 1), helping to determine whether it is undertrained or overfitted, and ultimately to evaluate its decision-making effectiveness compared to alternative decision agents. After validation, the RL process returns to Phase I. Iterations through the three phases continue until the delivery performance during validation either plateaus or begins to decline. The ANN with the best validation performance across all iterations is then adopted.

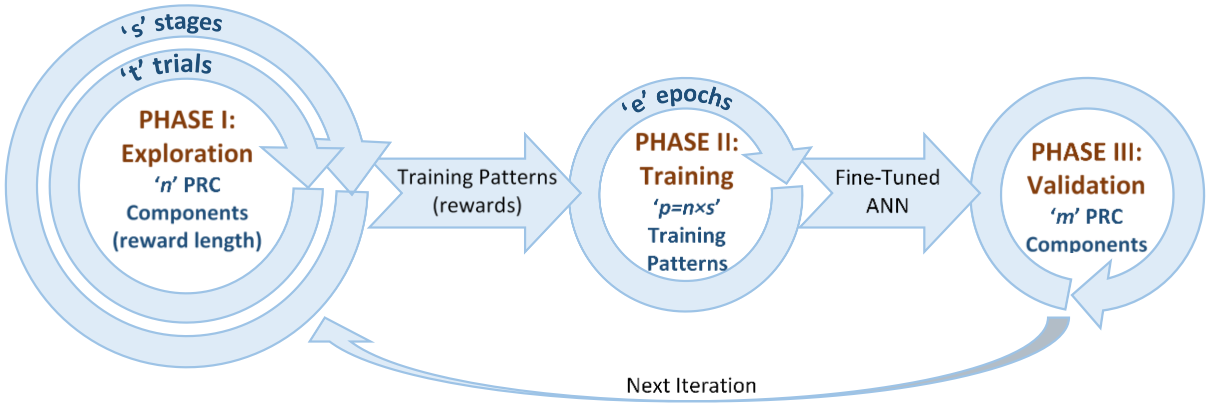


Figure 3: The three phases of the EDA's RL development algorithm.

2.2 Search-Based Decision Agents (SDA)

The search-based decision agent (SDA) adopted for this study performs a localized search over the decision space to select the next PRC component to process, based specifically on the currently known component orders. This approach contrasts with the EDA method, which considers a broad range of hypothetical orders and seeks a generalized decision-making strategy. At each decision point, the SDA starts with an initial decision, randomly generated for exploration. It then applies small random perturbations to the decision sequence, enabling a local search within the decision space without relying on a trained model. The search considers all outstanding current PRC component orders, ensuring that decisions remain focused on the full scope of the known problem. The search can be repeated for any specified number of steps *s*, with additional steps generally expected to improve delivery performance, albeit with diminishing returns relative to the computational time invested. This method was chosen for the SDA because it aligns with the exploration strategy used in the EDA's RL training approach. Its compatibility ensures it can be seamlessly integrated into the hybrid decision agent system described next.

2.3 Hybrid of Experienced-Based and Search-Based Decision Agents (HybridDA)

The hybrid decision agent (HybridDA) developed for this study integrates elements from both the EDA and the SDA, aiming to combine the advantages of each approach. In the HybridDA, the EDA component is first used to generate an initial decision based on the current state of the manufacturing system, as described in Section 2.1. The SDA component then refines this decision using the method outlined in Section 2.2, performing a localized search for more effective solutions based on the currently known PRC component orders. The search process is repeated for a specified number of steps, with the goal of improving the delivery performance, albeit at the cost of increased computational time. This hybrid approach merges the structured learning of the EDA with the focused search capabilities of the SDA, facilitating the agent's ability to make informed decisions while dynamically responding to the evolving workload.

2.4 Rule-of-Thumb (RoT) Decision Policy Adopted for this Study

In this study, the RoT prioritizes PRC components within a queue for processing based on their least remaining contingency time. PRC components are sorted according to this criterion upon arrival at a queue. Processes not controlled by a decision agent select the PRC component at the head of the queue, thereby favoring components with the least remaining contingency. The use of the RoT policy for all processes provides the baseline for evaluating the performance of the EDA, SDA, and Hybrid decision agents.

3 FACTORY CASE STUDY SIMULATION MODEL

To assess the delivery performance of the EDA, SDA, and HybridDA, a case study was conducted using the construction manufacturing system described by Wang et al. (2018). This system specializes in the delivery of PRC components, such as columns, wall panels, slabs, beams, and stairwells. A simulation model was developed to represent the system's operational processes and workflows, and was implemented in Python (Van Rossum, 1995) to provide compatibility with the RL trainer described in Section 2.1. Figure 1 presents a schematic of this model, which serves as the basis for assessing how each decision agent performs under realistic manufacturing conditions. The process flows and duration parameters within the model are adapted from the study by Wang et al. (2018), selected for their ability to capture the unique and challenging features of construction manufacturing, namely:

- Orders arrive randomly, requiring PRC components to be produced on demand, without the ability to stockpile.
- Each order comprises a batch of PRC components, with the batch size varying between orders.
- PRC components have customized designs, both within and between batches, leading to significant variation in their handling times at each process.
- The handling times for all PRC components at each process are subject to uncertainty.
- All components must be delivered by a specific time, as determined by the site assembly schedule.

Furthermore, the following assumptions were considered for the operation of the system:

- All PRC components follow the same sequence of processes shown in Figure 1.

- Orders consist of a batch of PRC components, with the number of components in each batch sampled from a triangular distribution and rounded to yield a positive integer.
- Batch orders arrive according to a Poisson process, with the average arrival rate (λ) set so that the work demand slightly exceeds the system's maximum throughput before optimizing the EDA.
- The required on-site delivery time for a PRC component is measured as the contingency time for the component (sampled from a triangular distribution) added to the sum of its process durations.
- Each process can handle only one PRC component at a time due to limited resources, except for the curing process, which can accommodate an unlimited number of components.
- Curing is considered to have a constant duration for all PRC components. Given this, and since system performance was measured relative to a baseline RoT (see Section 2.4), the impact of curing on performance effectively cancels out and was thus excluded from the model.

The stochastic time and batch order parameters used in this study, along with their corresponding distribution types, are summarized in Table 1. Note, units of time are not specified, as the system's behavior in this study is evaluated based on delivery time performance relative to a baseline decision rule, making the actual time units unnecessary for comparison purposes. For all arrivals, batch sizes, and handling times of each component, values are sampled independently using the Monte Carlo method. The triangular distribution was chosen for its computational efficiency and its ability to approximate a broad range of distribution shapes, including skewed forms.

Table 1: Modeling sampling parameters (adapted from Wang et al. (2018), and further adapted from Flood and Zhou (2023)).

System Variable	**Distribution**	**Parameters**
Batch order arrival time	Poisson process	Arrival rate (λ) = $^1/_{7,000}$
Batch size	Discretized triangular distribution	Minimum, Mode, Maximum: 10, 20, 100
Rebar duration	Triangular distribution	Minimum, Mode, Maximum: 120, 200, 250
Forms duration	Triangular distribution	Minimum, Mode, Maximum: 130, 150, 170
Concrete duration	Triangular distribution	Minimum, Mode, Maximum: 0, 50, 70
Cure of concrete duration	Fixed	~
Strip molds duration	Triangular distribution	Minimum, Mode, Maximum: 80, 100, 120
Delivery duration	Triangular distribution	Minimum, Mode, Maximum: 30, 50, 70
Contingency relative to site assembly time	Triangular distribution	Minimum, Mode, Maximum: 10, 100, 200

4 RESULTS AND ANALYSIS

The following experiments evaluate and compare the delivery performance of the EDA, SDA, and HybridDA using the case study described in Section 3.

4.1 EDA Training, Optimization, and Validation

The initial experiments focused on developing the EDA, beginning with a verification of the optimal RL training hyperparameters reported by Zhou and Flood (2024). To ensure independence from the original study, new random number generator seeds were used to initialize the ANN's weights and biases, as well as the sequence and customization of PRC components used in training and validation runs. Additionally, the performance of the EDA was measured relative to the RoT decision policy (see Section 2.4), instead of using the random selection approach of Zhou and Flood (2024) as the baseline. This study replicated the original hyperparameter tuning process using the same four key variables and, despite the new initialization, arrived at the same optimal values, as listed below (see Section 2.1 for descriptions of each variable):

1. Reward Length (see Figure 3): optimal at n=20 PRC components.
2. Number of PRC Components Sampled at the ANN Input: optimal at N=10 PRC components.

3. Number of Layers in the ANN: optimal at L=3.
4. Number of Hidden Units per Hidden Layer: optimal at h=64.

A validation run of this EDA, using a sequence of 2,000 PRC components, demonstrated a mean delivery performance improvement of 149.2 (time/PRC component). In other words, using the EDA reduced the order-to-delivery duration by an average of 149.2 time units per PRC component, compared to using the RoT. This is slightly lower than the 168.9 reported by Zhou and Flood (2024). The difference can be attributed to stochastic effects arising from the use of different random number sequences and to performance being evaluated relative to the RoT decision policy baseline, rather than random decisions.

An additional experiment was conducted to further optimize delivery performance by tuning the number of training patterns $p=n\times s$ (see Figure 3) used per RL training iteration. Zhou & Flood (2024) had arbitrarily fixed this at p=2,000. Given that n (the reward length) had already been optimized and was therefore fixed, the value of p (the number of training patterns) was varied by adjusting s (the number of stages in Phase I of the RL process - see Figure 3). The number of training patterns, p, used at each RL training iteration was varied from 1,000 to 4,000 in steps of 1,000. Increasing p from 2,000 to 3,000 led to a slight increase in delivery performance, from 149.2 to 151.0 (time/PRC Component). The smoothness of the sensitivity curve from this analysis suggests that 3,000 training patterns represents a stable optimal value, rather than one arising by chance, and was therefore adopted for the remainder of this study.

Figure 4 shows the delivery performance of the EDA for the 2,000 PRC component validation run, for iterations 0 to 21 the in the RL training process. Performance is represented on the vertical axis as the mean improvement in delivery time for PRC components compared to the RoT decision policy. For instance, a value of 151 on the vertical axis indicates that using the EDA for control results in delivering the PRC components 151 time units earlier on average than using the RoT. These values are the cumulative averages measured from the start of the PRC component validation run.

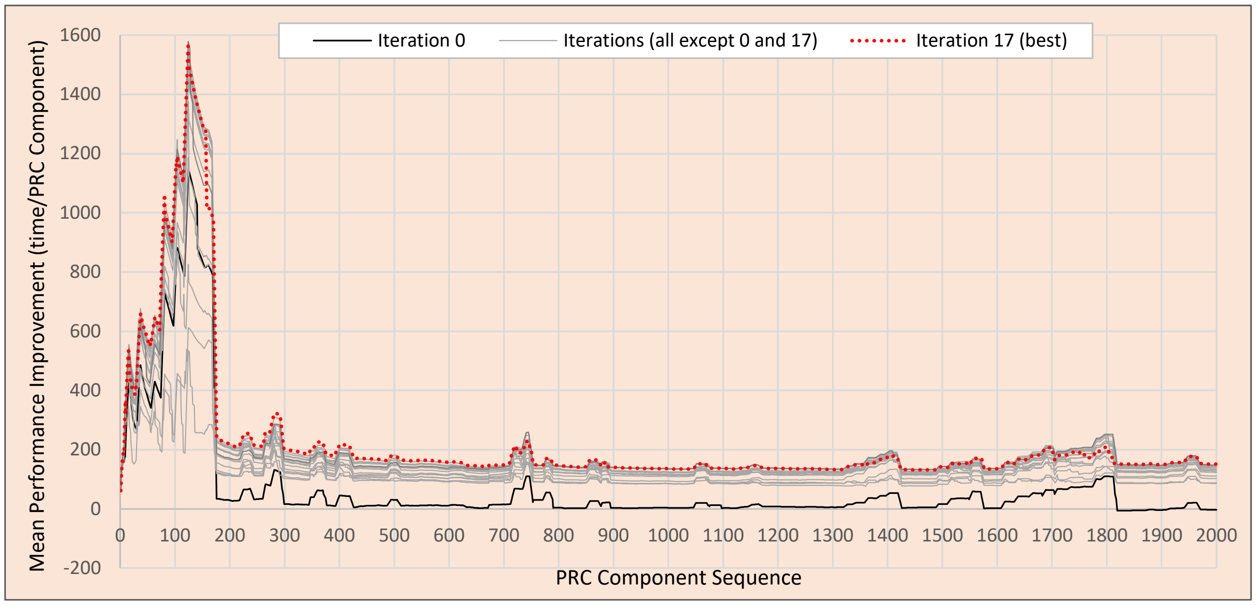


Figure 4: Delivery performance of the optimized EDA measured across the validation sequence of PRC components, for iterations 0 to 21 of the RL training process.

Each grey line in the figure represents the EDA's delivery performance at different iterations in the RL process. The dashed red line represents the iteration that gave the best performance (iteration 17) measured at the end of the 2,000 PRC component validation run. Going beyond iteration 17 led to overfitting in the ANN component of the EDA, causing a reduction in performance measured at the end of the PRC component validation run. The black line shows the performance of the EDA before any training, that is, with the weights and biases of the ANN set to their initial values. Given that Figure 4 measures performance using the RoT as a baseline (denoted by the horizontal line at the origin), it is evident that the best EDA (iteration 17) consistently delivers superior results throughout the sequence of validation PRC components.

4.2 Delivery Performance Comparisons: EDA, SDA, and HybridDA

Figure 5 summarizes the delivery performance of the EDA and HybridDA across iterations 0 to 21 of the RL training process for the validation run of PRC components. At each iteration, the HybridDA uses the EDA-generated decision as its initial approximation and performs 100 search steps in an attempt to improve overall delivery performance. As previously discussed in Section 2.3, the search mechanism of the HybridDA operates directly on the currently known PRC component orders. This direct use of current information explains the HybridDA’s improved delivery performance across most training iterations, except at iteration 17 where the EDA reached its optimal performance. Interestingly, the HybridDA's peak performance did not coincide with that of the EDA, occurring instead at iteration 19. These observations may be due to chance, and the performance of HybridDA could, in any case, improve with more search steps. Therefore, a sensitivity analysis will be conducted in the next set of experiments by varying the number of search steps and repeating each configuration to assess statistical significance.

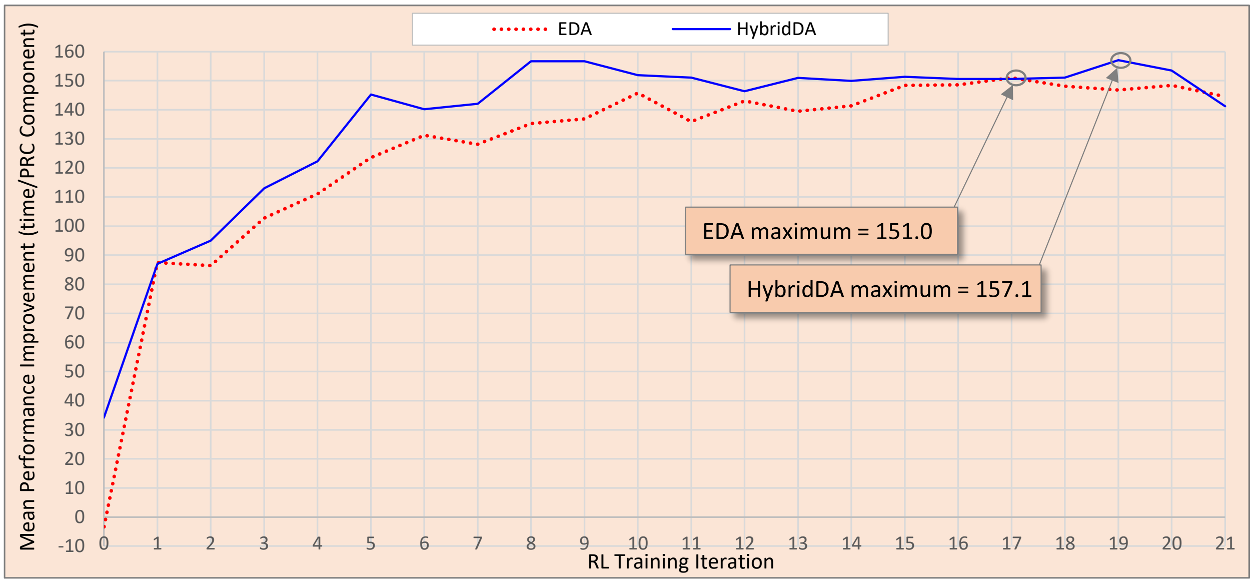
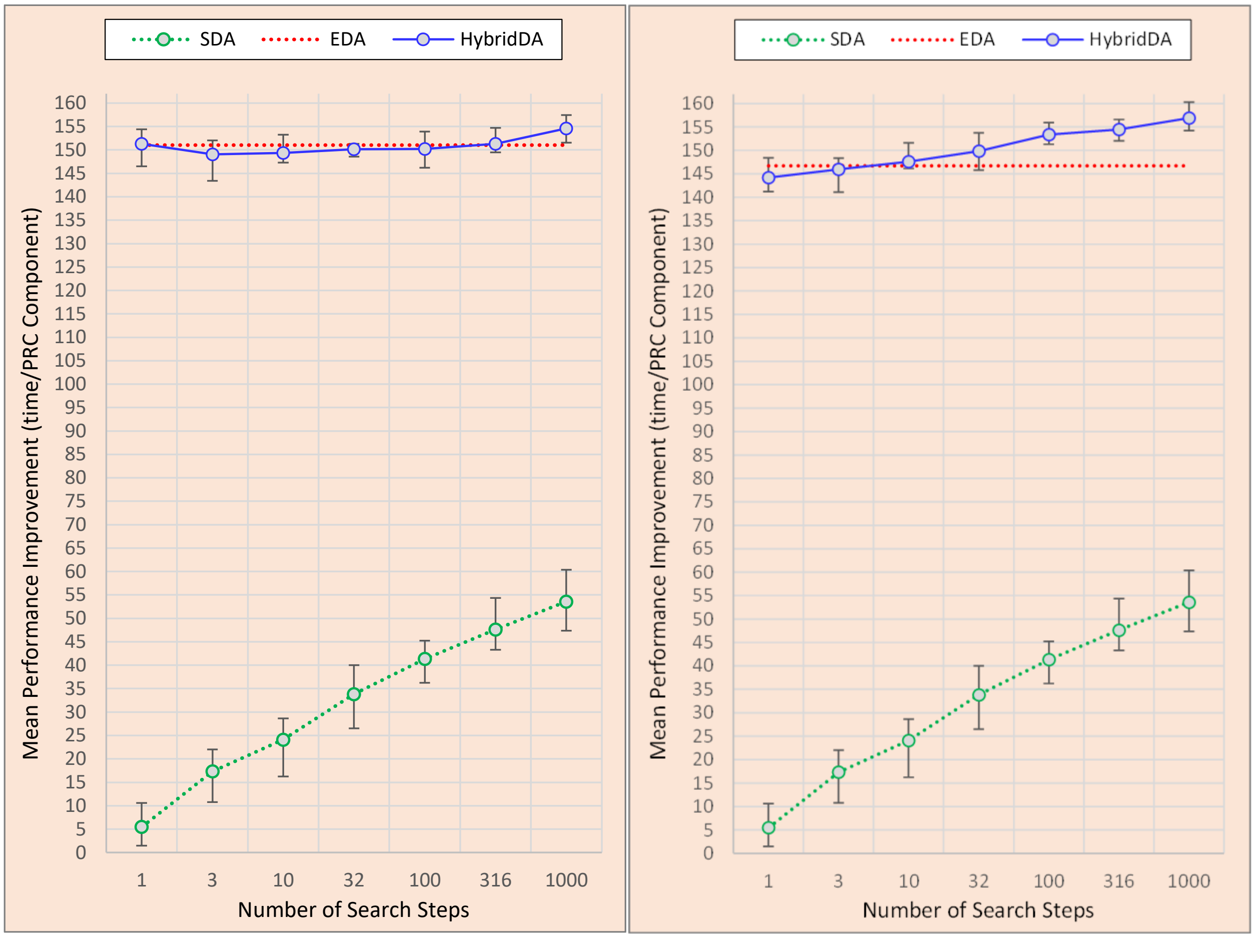


Figure 5: Delivery performance of the EDA and HybridDA, measured at iterations 0 to 21 of the RL training process, based on the validation sequence of PRC components.

Figure 6 shows the delivery performance of the SDA, EDA, and HybridDA during the validation run of the PRC components, across an exponentially scaled range of search steps from 1 to 1,000. This scaling was chosen to reflect the expected diminishing returns between performance and the number of search steps used by the SDA and HybridDA. For the SDA and HybridDA, each point represents the median of twelve independently repeated experiments using different random seeds, with error bars indicating the spread between the 10^{th} and 90^{th} percentiles. The dashed red line shows the delivery performance of the EDA used to generate the initial decisions for the HybridDA. Panel (a) of the figure corresponds to iteration 17 of the RL training process where the EDA achieved its optimal performance, while panel (b) corresponds to iteration 19 of the same process where the HybridDA reached its peak performance using 100 search steps, as considered previously.

As a first observation from Figure 6(a), the EDA outperformed the median SDA by a factor of 3.55 at the 1,000 search step mark. The trends in the performance lines also suggest that, in practical terms, the SDA is unlikely to surpass the EDA's performance, even with a significant increase in the number of search steps. Consequently, the performance comparison in the remaining results section will focus on a detailed analysis of the EDA and the HybridDA. Interestingly, it was unexpected that the EDA outperformed the SDA so significantly by building a decision model based solely on hypothetical past experience, without considering the current component order situation. This suggests that, at least for the manufacturing system considered here, optimizing for generalized long-term trends may be more effective than relying on short-term precise knowledge.



(a) Iteration 17 of the RL training process (optimum for the EDA - see Figure 5).

(b) Iteration 19 of the RL training process (optimum for the HybridDA - see Figure 5).

Figure 6: Impact of search steps on delivery performance for SDA and HybridDA, with EDA included for reference, based on the validation sequence of PRC components.

Figure 6 shows that as the number of search steps increases, the performance of the HybridDA gradually improves, eventually surpassing that of the EDA. The trends in these HybridDA performance lines also indicate that further improvement is plausible for search steps beyond 1,000. The performances of the HybridDAs in panels (a) and (b) are comparable, with both falling within each other's 10th and 90th percentiles, suggesting no definitive difference between starting with initial decisions based on the EDA trained to the 17th or 19th RL training iteration. However, since the 17th and 19th RL training iterations are close in terms of EDA delivery performance, a more thorough analysis across a broader range of RL training iterations and EDA performances is needed. In summary, for the best-performing EDA (at iteration 17 of the RL training process), the median HybridDA achieved a validation delivery performance of 154.6 (time/PRC Component), representing approximately a 2.4% improvement over the EDA.

5 CONCLUSIONS AND FUTURE WORK

The work presented in this paper investigates the potential of AI-based decision agents for real-time control in construction manufacturing, an area where traditional mass production strategies are constrained by highly variable and customized production demands. This research advances the field by developing and systematically comparing three functionally distinct AI approaches: an experience-based agent that uses RL-trained ANNs to generate decisions based on prior experience (the EDA), a search-based agent that utilizes stochastic methods to explore the decision space for the actual known work to be controlled (the SDA), and a hybrid method that fully aggregates both approaches (the HybridDA). These decision agents are applied to optimize production in dynamic construction manufacturing environments. Their effectiveness is evaluated in a real-world construction factory case study relative to a customary RoT decision policy, with delivery delays serving as the primary performance metric. The findings provide valuable insights into the relative advantages of each AI decision agent for improving production efficiency in complex and variable manufacturing settings.

This study first extended the EDA method of Zhou and Flood (2024) by optimizing the number of training patterns used per RL training iteration. The number of patterns was varied from 1,000 to 4,000, with 3,000 patterns proving optimal, resulting in a slight 1.2% improvement in delivery performance. Following this optimization, the main set of experiments compared the EDA, SDA, and HybridDA methods. The EDA consistently outperformed the SDA, achieving superior delivery performance by a factor of 3.55 at the 1,000 search step mark. The HybridDA achieved a modest improvement over the EDA, enhancing delivery performance by 2.4% at RL training iteration 17. Moreover, the HybridDA's performance continued to improve as the number of search steps increased, suggesting potential for further optimization. These results highlight the clear advantage of the EDA over the SDA and demonstrate the additional promise of the HybridDA approach for further improving delivery performance.

Future work will be aimed at improving the performance of the different agents, and increasing their applicability to a more diverse range of construction manufacturing problems. This will include:

- The use of alternative RL algorithms for training the ANN component of these decision agents.
- The use of alternative heuristic search methods for the HybridDA, such as GAs, ACO, SA, and TS.
- Expanding the range of decision input data and the decision types considered, including multiple decision points, predictive maintenance scheduling, and dynamic inventory control.
- An assessment of the decision agents' ability to generate solutions within a computationally acceptable time frame, based on measured performance latency across a range of scenarios.

REFERENCES

Benjaoran, V., and N. Dawood. 2005. "An Application of Artificial Intelligence Planner for Bespoke Precast Concrete Production Planning: A Case Study." *ITcon*. http://itc.scix.net/paper/w78-2005-a11-5-benjaoran.

Chan, W. T., and H. Hu. 2002. "Production Scheduling for Precast Plants Using a Flow Shop Sequencing Model." *Journal of Computing in Civil Engineering* 16(3):165–174 https://doi.org/10.1061/(ASCE)0887-3801(2002)16:3(165).

Chen, X., Y. Wu, and M. Zhou. 2022. "Mixed Production Line Optimization of Industrialized Building Based on ACO Algorithm." *Advances in Civil Engineering* 2022:1–10 https://doi.org/10.1155/2022/2411458.

Dan, Y., G. Liu, and Y. Fu. 2021. "Optimized Flowshop Scheduling for Precast Production Considering Process Connection and Blocking." *Automation in Construction* 125:103575 https://doi.org/10.1016/j.autcon.2021.103575.

Durdyev, S., and S. Ismail. 2019. "Offsite Manufacturing in the Construction Industry for Productivity Improvement." *Engineering Management Journal* 31(1):35–46 https://doi.org/10.1080/10429247.2018.1522566.

Flood, I. 1989. "A Neural Network Approach to the Sequencing of Construction Tasks." In *Proceedings of the Sixth International Symposium on Automation and Robotics in Construction*, San Francisco, CA, 204–211 https://doi.org/10.22260/ISARC1989/0026.

Flood, I., and P. D. L. Flood. 2022. "Intelligent Control of Construction Manufacturing Processes Using Deep Reinforcement Learning." In *Proceedings of the International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2022)*, 112–122 https://doi.org/10.5220/0011309600003274.

Flood, I., and X. Zhou. 2023. "Improving Delivery Performance of Construction Manufacturing Using Machine Learning." *Journal of Simulation Engineering* 3:8:1–8:14 https://articles.jsime.org/3/2/.

Glorot, X., A. Bordes, and Y. Bengio. 2011. "Deep Sparse Rectifier Neural Networks." In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 15:315–323 https://proceedings.mlr.press/v15/glorot11a.html.

Glover, F., and M. Laguna. 1999. "Tabu Search I." *INFORMS Journal on Computing* 1(3):190–206 https://doi.org/10.1287/ijoc.1.3.190.

Haq, A. N., T. R. Ramanan, K. S. Shashikant, and R. Sridharan. 2009. "A Hybrid Neural Network–Genetic Algorithm Approach for Permutation Flow Shop Scheduling." *International Journal of Production Research* 48(14):4217–4231 https://doi.org/10.1080/00207540802404364.

Hatami, M., and I. Flood. 2024. "Optimizing Truck Allocation in Open-Pit Mining Using a Deep Reinforcement Learning Policy." In *International Conference on Computing in Civil and Building Engineering (ICCCBE)*, Montreal, Canada, 11 pp. https://doi.org/10.1007/978-3-031-87364-5_11.

Kim, T., Y. W. Kim, D. Lee, and M. Kim. 2022. "Reinforcement Learning Approach to Scheduling of Precast Concrete Production." *Journal of Cleaner Production* 336:130419 https://doi.org/10.1016/j.jclepro.2022.130419.

Leu, S., and S. Hwang. 2001. "Optimal Repetitive Scheduling Model with Sharable Resource Constraint." *Journal of Construction Engineering and Management* 127(4):270–280 https://doi.org/10.1061/(ASCE)0733-9364(2001)127:4(270).

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." *Advances in Neural Information Processing Systems* 32:8024–8035 https://doi.org/10.5555/3454287.3455008.

Podolski, M. 2022. "Effective Allocation of Manpower in Precast Production Using Metaheuristics." *Journal of Civil Engineering and Management* 28(4):247–260 https://doi.org/10.3846/jcem.2022.16453.

Rocha, P. F., N. O. Ferreira, F. Pimenta, and N. B. Pereira. 2023. "Impacts of Prefabrication in the Building Construction Industry." *Encyclopedia* 3:28–45 https://doi.org/10.3390/encyclopedia3010003.

Shitole, V., J. Louis, and P. Tadepalli. 2019. "Optimizing Earth Moving Operations via Reinforcement Learning." In *2019 Winter Simulation Conference (WSC)*, 2954–2965 https://doi.org/10.1109/WSC40007.2019.9004785.

Sutton, R., and A. Barto. 2018. *Reinforcement Learning: An Introduction*. 2nd ed. London: The MIT Press.

Vala, G., I. Flood, and E. Obonyo. 2011. "Truck Weigh-in-Motion Using Reverse Modeling and Genetic Algorithms." In *2011 ASCE International Workshop on Computing in Civil Engineering*, ASCE, Miami, FL, 219–226 https://doi.org/10.1061/41182(416)27.

Van Rossum, G., and F. L. Drake Jr. 1995. *Python Reference Manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Wang, Z., H. Hu, and J. Gong. 2018. "Framework for Modeling Operational Uncertainty to Optimize Offsite Production Scheduling of Precast Components." *Automation in Construction* 86:69–80 https://doi.org/10.1016/j.autcon.2017.10.026.

Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek. 2018. "Optimization of Global Production Scheduling with Deep Reinforcement Learning." *CIRP Annals - Manufacturing Technology* 72:1264–1269 https://doi.org/10.1016/j.procir.2018.03.212.

Xia, K., C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik. 2021. "A Digital Twin to Train Deep Reinforcement Learning Agent for Smart Manufacturing Plants: Environment, Interfaces and Intelligence." *Journal of Manufacturing Systems* 58:210–230 https://doi.org/10.1016/j.jmsy.2020.06.012.

Yang, Z., Z. Ma, and S. Wu. 2016. "Optimized Flowshop Scheduling of Multiple Production Lines for Precast Production." *Automation in Construction* 72:321–329 https://doi.org/10.1016/j.autcon.2016.08.021.

Zhou, L., L. Zhang, and B. K. P. Horn. 2020. "Deep Reinforcement Learning-Based Dynamic Scheduling in Smart Manufacturing." *CIRP Annals - Manufacturing Technology* 93:383–388 https://doi.org/10.1016/j.procir.2020.05.163.

Zhou, X., and I. Flood. 2024. "Optimization and Evaluation of a Neural Network Based Policy for Real-Time Control of Construction Factory Processes." *ITcon* 29:84–98 https://www.itcon.org/paper/2024/5.

AUTHOR BIOGRAPHIES

IAN FLOOD is a Professor in the Rinker School at the University of Florida. He received his Ph.D. from the University of Manchester, UK, on parallel computing techniques applied to the simulation of construction processes. He has held academic positions at the National University of Singapore, and the Department of Civil Engineering at the University of Maryland. He has published extensively on the subjects of simulation modeling and intelligent computing applied to the built environment, has received five best paper awards, and was the 2015 recipient of the ASCE Computing in Civil Engineering Award. His research is concerned with the development of new methods of simulating and optimizing engineered systems, and improving the performance of these systems using artificial intelligence techniques. Email: flood@ufl.edu.

MADISON E HILL is a designer and established pietra dura mosaicist with a career spanning both the U.S. and India. She holds a Master's degree in Construction Management and a Bachelor of Design in Architecture from the University of Florida. Her research focuses on integrating artificial intelligence and automation with construction sequencing and advanced material workflows, particularly in digital fabrication and design automation. She is the author of Natura Omnia Artes Magistra (Sillabe, 2024), the only comprehensive compendium of lapidary materials for pietra dura in print. Email: madisonhill@ufl.edu.