www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

OPTIMIZATION AND EVALUATION OF A NEURAL NETWORK BASED POLICY FOR REAL-TIME CONTROL OF CONSTRUCTION FACTORY PROCESSES

SUBMITTED: October 2023
REVISED: January 2024
PUBLISHED: February 2024
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2024.005

Xiaoyan Zhou, PhD,
University of Florida;
zhouxiaoyan@ufl.edu

Ian Flood, Professor, PhD,
University of Florida;
flood@ufl.edu

SUMMARY: *This paper focuses on the development, optimization, and evaluation of an intelligent real-time control system for the fabrication of precast reinforced concrete components. The study addresses the unique challenges associated with real-time control in the construction manufacturing industry, including high customization, uncertain work demand, and limited stockpiling opportunities. A production system model is built based on a real construction manufacturing factory to simulate real-world precast reinforced concrete component fabrication, and acts as the basis for the development and validation of the control system. A review of alternative decision-making techniques is presented to identify the most suitable for the control of construction manufacturing factories. Ultimately, an artificial neural network approach trained using a reinforcement learning strategy is selected as a promising technique for effective real-time control. The controller is developed and validated, and its performance is optimized using sensitivity analysis, which takes into account both the structure of the artificial neural network and the parameters of the reinforcement learning algorithm. The ANN-based control policy is applied to the sequencing of precast reinforced concrete component production, while a rule-of-thumb policy is used as a benchmark for comparison. The study demonstrates that the optimized ANN-based control policy significantly outperforms the standard rule-of-thumb policy. The paper concludes by providing suggestions for further advancement of the ANN-based approach and potential avenues to increase the control policy's scope of application in construction manufacturing.*

KEYWORDS: *Artificial Neural Networks, Construction Manufacturing, Intelligent Control Policy, Machine Learning, Model Optimization, Precast Reinforced Concrete Components, Reinforcement Learning.*

REFERENCE: *Xiaoyan Zhou, Ian Flood (2024). Optimization and evaluation of a neural network based policy for real-time control of construction factory processes. Journal of Information Technology in Construction (ITcon), Vol. 29, pg. 84-98, DOI: 10.36680/j.itcon.2024.005*

1. INTRODUCTION

1.1 Background to the problem

The construction industry is a significant contributor to the world's economy having a market size of around $8.2 trillion in 2022 and is expected to double its market size in 2030 compared to 2020 (Statista, 2022). To reach this goal, unique challenges to traditional onsite construction have to be confronted, such as low productivity, a lack of skilled labour, and excessive material wastage (Deng et al., 2013, Zhu et al., 2022). Factory-based construction manufacturing is one promising approach that holds the potential to address the inefficiencies commonly associated with traditional onsite construction (Hong et al., 2018).

Achieving efficiency within a construction manufacturing factory is, however, more difficult to achieve than in other manufacturing industries. For example, job demand in construction is typically uncertain, job orders arrive irregularly, and products within and between job batches are often highly customized and rarely reproduced (Flood & Flood, 2022). As a result, construction manufacturing does not lend itself to mass production. Work has to be produced on request with limited or no opportunity for stockpiling. Moreover, production control decisions cannot be planned well in advance (offline) but rather must be made in real-time (online). These complexities necessitate the use of more advanced decision-making methods than traditional rule-of-thumb approaches to ensure efficiency in the production of construction components.

In the field of optimization theory, offline decision-making is concerned with situations where sufficient relevant information is available early enough to search for a near-optimal solution before the decision must be made. Online decision-making, however, is required where the opposite is true, that is, when the necessary information becomes available too late to use search-based methods to identify a near-optimal solution. In this case, decisions have to be made on-the-fly as the problem unfolds. Online decision-making is inherently difficult to solve and usually has to be based on experience-based methods such as pre-trained artificial neural networks (ANNs) to overcome the computational time constraint.

1.2 Past work and its limitations

The complex nature of construction processes and their resulting products make it impossible to formulate simple methods or manually craft rules that will produce near-optimal production control decisions. A more sophisticated approach is required. Genetic algorithms (GAs) are one such optimization approach that have been widely used in the field of construction manufacturing decision-making. GAs are a heuristic search algorithm inspired by the natural evolution mechanism (Vose, 1999). Researchers, such as Benjaoran & Dawood (2005), Chan & Hu (2002), Leu & Hwang (2001), Dan et al. (2021), and Yang et al. (2016), have applied GAs as an offline optimization approach to refine the production process of precast reinforce concrete (PRC) components. While this approach has proven effective, GAs, like all search methods, are unsuitable for online optimization problems where decisions must be made in real-time due to their substantial computational time demands.

A potential solution to this challenge involves the use of ANN-based agents, developed using machine learning techniques. These agents can act as advisors in a human-in-the-loop system or as controllers in an automated environment, making decisions dynamically with the intent of optimizing the performance of the operation. ANNs offer the potential to produce effective solutions to a decision-making problem in real-time. To achieve this, they must be pre-trained on a comprehensive set of near optimal solutions to the problem. Unfortunately, in the case of construction manufacturing, such solutions are not available in advance of training. This problem can be circumvented by using reinforcement learning (RL), a training technique based on discovery and rewards (Sutton and Barto, 2018), discussed in more detail in section 3.2. Outside the field of construction, several researchers have applied the RL approach to dynamic control of manufacturing (Waschneck, 2018, Zhou et al., 2020, Xia et al., 2021). In the Waschneck et al. (2018) study, a comparison was made between an RL-based approach and a more conventional solution involving human intervention or prior expert knowledge for solving factory operation control problems. The results were promising with the RL approach demonstrating superior performance in effectively controlling the system. In fact, the benefits of applying RL techniques in controlling manufacturing systems have been considered as far back as the 1990s (Zhang & Dietterrich, 1995, Kim & Lee, 1995, Riedmiller & Riedmiller, 1999), although, due to the relatively limited computational power available, they had limited success at that time. Machine learning breakthroughs made in the 2010s (Krizhevsky et al., 2012, Mnih et al., 2013, LeCun et al., 2015, Silver et al., 2017) have greatly inspired and facilitated the further application of RL in the manufacturing industry. A systematic review of deep-RL in production systems can be found in (Panzer &

Bender, 2022). However, these applications have been outside the field of construction manufacturing and, therefore, failed to address the numerous challenges within this industry as addressed in Section 1.1. In addition, as pointed out by Panzer and Bender (2022), although most RL applications outperform conventional solution methods within manufacturing and reduce the dependence on human expertise, more research is required to transfer the findings to nontrivial real-world systems.

The application of RL to the control of construction operations is very limited. Shitole et al. (2019) developed an ANN-based agent, trained using RL, to control a simulated earth-moving operation with the goal of optimizing the system production rate. While the agent outperformed previously published heuristic techniques, it suffers from an inherent lack of flexibility. That is, if the problem scope were to change (such as relocating the earth-moving operation to a different area of the site) then redevelopment of the agent would be necessary. There may not be sufficient time to collect the required performance data and redevelop each new version of the agent if the problem is changing frequently.

Compared to onsite construction, factory-based construction manufacturing does not require frequent redefining of the optimization problem scope or redevelopment of the ANN, since the basic factory system configuration is typically long lived. A proof-of-concept study undertaken by Flood et al. (2022, 2023) indicated that an RL-trained deep ANN (an ANN with multiple hidden layers) could outperform a rule-of-thumb approach to controlling a construction manufacturing process in a simulated environment. The preliminary research showed that the ANN-based agent has the potential to conduct real-time optimization in a situation where the arrival of the job batches follows a Poisson process, the number of PRC components varies between batches, and all the PRC components can have different designs.

Other studies using ANN's to control factory-based construction processes have been limited in their depth of application. The ANN model developed by Flood (1989), for example, was essentially an offline method constrained to sequencing a pre-defined number of PRC components. The work reported by Kim et al. (2022) was similarly constrained, with the reported experiments limited to 36 PRC components with a maximum of 10 different component designs. Ideally, what is needed is an online method of dispatching PRC components, where orders arrive as a continual random stream, with no constraints on the extent of design customization.

There has been notable interest in applying ANN methods to construction manufacturing, but unfortunately, this work has been outside the realm of process control. Examples cover a diverse range of topics including component design (Navarro-Rubio et al., 2020), visual task identification (Rashid & Louis, 2020), component progress tracking (Martinez et al., 2010), and construction robotics coordination (Delgado & Oyedele, 2022).

1.3 Aim and objectives

The primary aim of this study is to achieve a significant advancement in optimizing production performance within the area of construction manufacturing, using state-of-the-art intelligent control of processes, building upon the proof-of-concept work reported by Flood & Zhou (2023). To achieve this, a comprehensive analysis is undertaken to understand how the ANN's structure and RL algorithm parameters affect the ability of the agent to learn decisions that are more effective. The goal of this analysis is to optimize the design of the ANN-based agent to boost its production control performance. The RL approach is used to train the ANN in a simulated environment based on real data, and its performance is compared with a rule-of-thumb approach that serves as a benchmark.

2. REAL-TIME CONTROL OF CONSTRUCTION MANUFACTURING PROCESSES

2.1 Intelligent agent control

The sequence of events experienced by a construction manufacturing process can be categorized in terms of those that are controllable (such as selecting equipment to be allocated to new jobs) and those that are uncontrollable (such as the failure of equipment). Planners can leverage the controllable events to direct the system's trajectory in a favourable direction with the intent of maximizing the performance of strategic metrics such as productivity and profit. As shown in Figure 1, control can be made dynamically by one or more intelligent decision agents over the lifetime of the manufacturing system. These agents monitor the relevant variables that define the current state (and relevant parts of earlier states) within the system including its environment and utilize this information to take appropriate actions. Although a decision agent's actions are generally focused on the immediate future, they may also extend to events further in time that have a long lead-time such as outsourced components that are engineered-

to-order. The agents are developed to make decisions that are intended to optimize the performance of the manufacturing system over time.

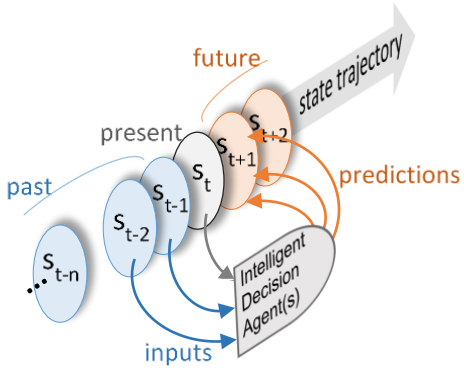

Figure 1: Intelligent Decision Agent Control of a Dynamic System.

Decision agents can be categorized into either search-based methods (such as blind and heuristic techniques) or experience-based methods (such as rules-of-thumb and ANNs) (Flood & Flood, 2022). Search-based methods utilize a systematic exploration of the solution space to find the best attainable action. They tailor each solution to the problem at hand and thus are arguably more likely to find a closer to optimum solution than an experience-based method. In addition, search-based methods can be easily adapted to new variants of a problem making them highly extensible. However, a shortcoming is that they can be computationally expensive making them unsuitable for online optimization problems where decisions must be made in real-time. Experience-based methods, on the other hand, rely on knowledge reaped from prior exposure to similar situations to make decisions. Once developed, they can make decisions rapidly. For this reason, they are well suited to making online decisions, although the solutions they provide may not be as close to optimal as those generated by search-based methods. Furthermore, experience-based methods usually lack extensibility so that each new iteration of the problem necessitates redevelopment of the agent. This process involves acquiring and assimilating a significant amount of new information about system behaviour, making it a time-consuming endeavour. A hybrid of the two types of decision methods is also possible. For example, search-based methods can be used to collect training patterns for the development of an ANN, or an experience-based method can be used as the first approximation of a solution that is then refined using a search-based method.

This paper focuses on the development, evaluation, and optimization of an RL trained ANN-based agent, which is an experience-based method utilizing a search technique to collect near optimal training examples. For this application, the search for training examples is conducted within a simulation of the manufacturing system. The agent is able to make real-time decisions, meeting the need for online control of the manufacturing production process.

2.2 Factory simulation model

Figure 2 shows a schematic of the manufacturing simulation model used for this study, representing the production of PRC components such as columns, wall panels, slabs, beams, and stairs. The processes and the duration parameters were obtained from the study published by Wang et al. (2018), chosen because it captures the unique and challenging features of construction manufacturing, namely:

- Job orders arrive randomly and thus must be made to order without the ability to stockpile;
- Each order comprises a batch of PRC components variable in quantity;
- PRC components have a customized design, both within and between batches, and thus may vary significantly in their handling times at each process;
- The handling times at each process for all PRC components include uncertainty;
- All components must be delivered by a given point in time determined by a site assembly schedule.

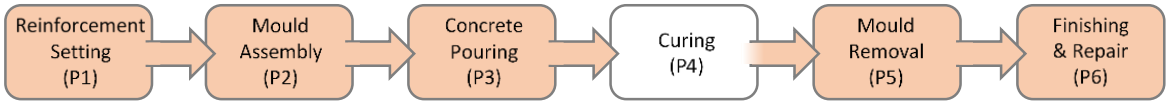

Figure 2: PRC Production Processes Sequence.

Additionally, the following assumptions were made about the operation of the system:

- All PRC components follow the same sequence of processes, as shown in Figure 2;
- Orders consist of a batch of PRC components. The number of PRC components in a batch is sampled from a triangular distribution, rounded to provide a positive integer;
- The batch orders arrive following a Poisson process. The average number of order arrivals per unit of time (the arrival rate, λ) was selected so that the work demand would slightly exceed the maximum throughput of the system before optimization of the decision agent;
- The required on-site delivery of a PRC component is measured as a contingency time beyond the sum of the PRC component's process durations. This is sampled from a triangular distribution;
- There are only sufficient productive resources at a process to serve one PRC component at a time, except the curing process which can handle an unlimited number of components;
- Curing is considered to have a constant duration for all PRC components. Given this, and the fact that the performance of the system was measured relative to a baseline decision policy, the impact of curing effectively cancels out and was thus not incorporated in the model.

The stochastic time related parameters used for this study, including the types of distribution considered, are listed in Table 1. The dynamics of the system are determined by the relative values of these parameters (not their absolute values) and thus units of time are not stipulated. The triangular distribution was adopted because it is computationally efficient while providing a flexible way of approximating a wide range of distribution forms, including those with skew.

Table 1: Process Duration Distribution Parameters (adapted from Wang et al. (2018)).

Production System Variables	Distribution Type	Related Parameters
Job Order Arrival	Poisson Process (Arrival Rate, λ)	1/7000
Batch Size	Discretized Triangular Distribution (Min, Mode, Max)	(10, 20, 30)
Reinforcement Setting	Triangular Distribution (Min, Mode, Max)	(180, 200, 250)
Mould Assembling	Triangular Distribution (Min, Mode, Max)	(80, 100, 120)
Concrete Pouring	Triangular Distribution (Min, Mode, Max)	(20, 40, 60)
Curing	Fixed	~
Mould Removing	Triangular Distribution (Min, Mode, Max)	(80, 100, 120)
Finishing & Repairing	Triangular Distribution (Min, Mode, Max)	(30, 50, 80)
Contingency (prior to delivery)	Triangular Distribution (Min, Mode, Max)	(10, 100, 200)

2.3 Decision policy types considered

This study compares the performances of two experience-based methods, namely, a decision policy based on an ANN-based agent trained using RL methods, and a rule-of-thumb decision policy. These two approaches represent two extremes in decision policy complexity, as detailed below. The ANN-based policy is the focus of this study, aimed at optimizing the production performance of the factory, whereas the rule-of-thumb policy is typical of what would be applied in industry and is used as a benchmark against which the performance of the ANN-based policy can be compared. In addition, a random PRC component selection policy was implemented as a baseline for measuring performance. This random policy represents the most performance-agnostic approach that can be implemented, and so all performance measurements are assessed in relation to this policy. Note that a decision agent is usually only needed for the first process in the PRC component system considered (detailed in Section 2.1) as it was the only one found to experience significant bottlenecks. For all other processes, the rule-of-thumb policy serves as the default. The following provides a more detailed description of these policies:

- An ANN-based policy developed and trained using the RL method described in Section 3. The ANN selects a PRC component from the first queue based on the predictions about the handling times for all the PRC components currently waiting at that queue.
- A rule-of-thumb policy that selects the PRC component with the least remaining contingency time from the queue.

A random policy that uses a uniformly distributed random variate to select the PRC component from the relevant queue.

3. ANN BASED REINFORCEMENT LEARNING

In a construction manufacturing environment, optimal solutions to decision problems cannot be easily obtained from experience with the real system. As a result, labelled training data will not be available for supervised learning of the ANN-based policy. However, adopting a hindsight strategy where the agent explores alternative decision paths using simulation, and chooses the most successful ones through trial and error, can address this problem (Flood & Flood, 2022). Specifically, the most successful decision paths found at any one stage can be used to provide the patterns to train (or further train) the ANN. The updated ANN can then be used to search for even better decision paths. This approach is the essence of RL. Inspired by behaviour psychology (Sutton, 1984), RL is designed to interact with the environment and learn the optimal policy that maximizes the cumulative rewards by trial and error (Sutton & Barto, 2018). The RL method has been applied where supervised learning is unavailable, or domain knowledge is costly to acquire (François-Lavet, 2018).

In construction, it is impractical to experiment with alternative decision-making methods for the control of construction production using a real system. Construction projects are typically unique in design and are rarely reproduced, making it almost impossible to compare the effectiveness of different decision-making strategies through direct experimentation. In addition, artificially reproducing the construction work is not realistic considering the required cost and time to manufacture a construction component (Flood & Flood, 2022).

An effective way to get around the problems listed above is to build a simulation model of the construction production system and then use it to explore and test alternative policies over a comprehensive range of scenarios (Flood & Flood, 2022). Related real-world information about the manufacturing factory and its behaviour are collected to develop and validate the simulation model. The simulator can then be used to generate data to train and evaluate the effectiveness of the ANN-based decision policy using RL, as described in Section 3.2.

3.1 ANN structure

The type of ANN used to implement the decision policy is a layered fully connected feedforward structure, where the number of hidden layers and the number of hidden neurons per layer are determined through optimization experiments as reported in Section 4.2. The inputs to the ANN specify the estimated process durations and the remaining contingencies for the PRC components waiting to be processed at the first queue in the production system. The inputs are normalized for each process and for the remaining contingency to have values within the range 0.0 and 1.0. The location of the values at the input to the ANN corresponds to the position of the PRC component within the queue.

An issue with this type of ANN is that they are structurally rigid in that the number of input neurons is fixed, yet the number of PRC components in a queue that need to be evaluated is variable. To overcome this problem, the number of PRC components inspected by the ANN was set to an upper limit, N. The optimum value for N was determined by experimentation as reported in Section 4.2. If the number of PRC components in the queue is less than N, then the spare input values are set to 0.0. If the number of PRC components in the queue is greater than N, then only the first N PRC components with the least remaining contingencies are evaluated. From this perspective, the ANN is a hybrid with the rule-of-thumb policy.

All the hidden neurons adopt the ReLU (rectified linear unit) activation function due to its computational efficiency and avoidance of the vanishing gradient problem (Glorot et al., 2011) which can otherwise impede training for ANNs with multiple hidden layers.

The outputs from the ANN indicate which PRC component is to be selected for processing at the first queue in the production system. There are N outputs from the ANN, each representing a different PRC component that can be selected from the queue. The output neurons employ the sigmoid activation function, constraining their output

values within the range 0.0 to 1.0. The values generated at all the output neurons are further normalized so that they sum to 1.0. These normalized values then act as the probabilities for selecting the corresponding PRC components from the queue.

The ANN-based policy has two modes of operation:

- *Exploration:* This mode is used to steer the production simulation through alternative partially stochastic paths using Monte Carlo (MC) sampling, with the aim of gathering training patterns. Stochastic sampling uses a uniformly distributed random variate to select a PRC component given the probabilities generated at the ANN output neurons. The higher the value at an output unit the more likely its corresponding PRC component will be selected. High-reward input-output pattern pairs (those that were found to improve the performance of the ANN-based policy) are collected for training. A training pattern will have one of its outputs set to 1.0 (the output representing the PRC component to be selected) while the rest of the outputs will be set to 0.0;

Validation and Implementation: This mode is used to select a PRC component from the queue deterministically. The selection is simply based on the output unit that generates the highest value. It is used to control the simulated system in non-training mode, to validate the performance of the current policy, and is adopted when using the policy to control the real system.

3.2 RL learning method

As shown in Figure 3, the RL learning strategy adopted for this study contains 3 phases: Phase I, the exploration and collection of the training patterns; Phase II, the training of the ANN; and Phase III, the validation of the ANN. An RL iteration is one cycle through these three phases, and is repeated until learning converges, with each iteration using the most recent version of the ANN to control the production simulation. Each time the system iterates back to Phase I, the simulation is reset to a new starting point to explore and collect new training patterns. These phases are described in detail in the following.

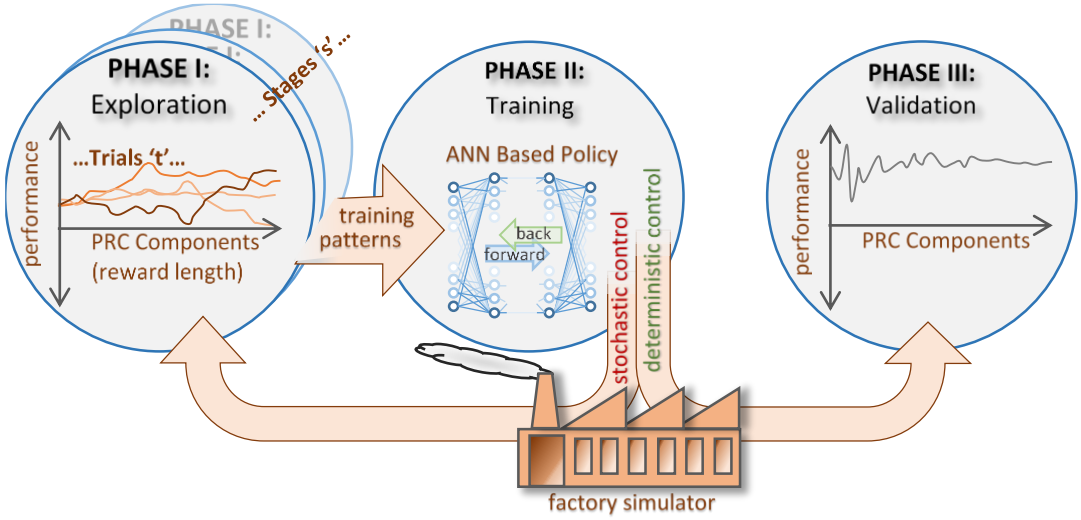

Figure 3: Three Phases of the RL Process. Phase I: select the best trial at each stage in the current RL cycle, and use these to generate a set of training patterns. Phase II: use the most recent set of training patterns to train (or further train) the ANN-based policy. Phase III: validate the most recently trained version of the ANN-based policy.

3.2.1 Phase I: Exploration to identify high-reward training patterns

Phase I, shown to the left of Figure 3, is designed to explore and collect training data patterns that have the intent of improving the performance of the ANN-based decision policy. Phase I is broken down into a number of stages *s*, each representing the production of a predefined number of PRC components, *n*, termed the reward length. The product of *s* and *n* gives the number of training patterns that are generated at each iteration of Phase I. The optimum value for the reward length *n* was determined by experimentation as reported in Section 4.2, and the corresponding value for *s* was set to generate 2,000 training patterns per RL iteration. Each stage is repeated for a number of trials

t, each occasion using a different sequence of random numbers for sampling from the probabilities generated at the output by the current ANN-based policy. The trial that results in the best delivery performance at a stage is stored for subsequent training of the ANN. Delivery performance is measured in terms of the delays to the delivery of the PRC components. To make sure that all PRC components register a cost, the delays are offset to a base value, given that some or all PRC components may be delivered early. The cost function is given as the root-mean-square (RMS) of these delays for the sequence of PRC components being considered, and the goal is to minimize this value:

$$\text{Cost} = \sqrt{\frac{\sum_{i=1}^{n}(d_i - b)^2}{n}} \quad (1)$$

where:

d_i is the i^{th} PRC component's delay at its completion;

b is the base value against which the delays 'd_i' are offset - this value is equal to the maximum contingency time possible for a PRC component;

n is the number of PRC components fabricated at the current trial (the reward length).

The number of trials t used at a stage was selected to exhaust the observed improvement in the above cost function; a value of $t = 100$ was found to satisfy this requirement for the experiments presented in Section 4. After completion of the exploration and collection of the training patterns, the system moves to Phase II, ANN training.

3.2.2 Phase II: Training the ANN based on the most recent set of high-reward training patterns

Phase II, shown in the middle of Figure 3, is the training of the ANN using the high-reward training patterns collected from the previous Phase I iteration. For the first RL iteration, the ANN is initialized with randomly selected weights and biases, whereas for subsequent RL iterations the weights and biases are carried over from the previous RL iteration.

The ANN software is implemented in Python (van Rossum & Drake, 1995) with PyTorch (Paszke, 2019). Root-mean-square propagation (RMSProp) is used as an optimizer for gradient descent calculation during training. Mean square error (MSE) is used to calculate loss, which is the average squared difference between the output value and target value. A data loader is used to feed the collected training patterns to the ANN with a mini-batch size of 64 with shuffling switched on. The learning rate is set to 0.001, which was found through trial-and-error to provide stable convergence on a solution. The training of the ANN is conducted until the output from the MSE function converges, which was found to happen within 1,000 epochs.

3.2.3 Phase III: Validation or implementation of the ANN-based decision policy

Phase III, shown in the right of Figure 3, is concerned with the validation and implementation of the ANN-based policy. For the experiments presented in Section 4, validation was run for a sequence of 8,000 PRC components, different from those used for training as determined by the seed of the random number generator used for sampling from the distributions given in Table 1. After Phase III is completed, the RL iteration returns to Phase I. The number of RL iterations (passing through Phases I to III) was set to 50 since the experiments considered in this study showed that little learning was observed beyond this point. The best performing ANN policy was considered the one trained at the RL iteration that was best at satisfying delivery deadlines, measured for the 8,000 validation PRC components.

4. RESULTS AND DISCUSSION

A series of experiments were conducted to evaluate and optimize the performance of the ANN-based policy relative to the rule-of-thumb policy for the factory set-up outlined in Section 2.2.

4.1 Initial model training and validation

The first experiment used 6 hidden layers and 64 hidden neurons per hidden layer for the ANN structure, a reward length of $n = 20$ PRC components (see Section 3.2.1), and a maximum number of PRC components sampled at a queue set to $N = 20$ (see Section 3.1). These values were derived from pilot experiments with the RL method, but an objective of this study was to optimize them through subsequent sensitivity analyses. Table 2 provides a summary of these initial modelling values as Experiment 1. Figure 4 shows the progress during training (Phase II)

for three of the 50 RL iterations considered in the development of this initial ANN-based policy. The vertical axis shows the mean absolute residual error for the training patterns plotted against the training epoch. The 1st, 2nd and 5th iterations were plotted to provide a representation of the range of behaviours observed during the RL development process. Interestingly, as can be seen in this figure, earlier iterations tend to be slower to converge.

Table 2: RL Model Parameters Considered for each Experiment.

Experiment	Number of Hidden Layers	Number of Neurons per Hidden Layer	Reward Length (*n*)	Number of PRC Components that can be Sampled (*N*)
1	6	64	20	20
2	Range: 1 to 8 Optimum = 1	64	20	20
3	1	Range: 2 to 256 Optimum = 64	20	20
4	1	64	Range: 2 to 2000 Optimum = 20	20
5	1	64	20	Range: 0 to 30 Optimum = 10
6	1	64	20	10

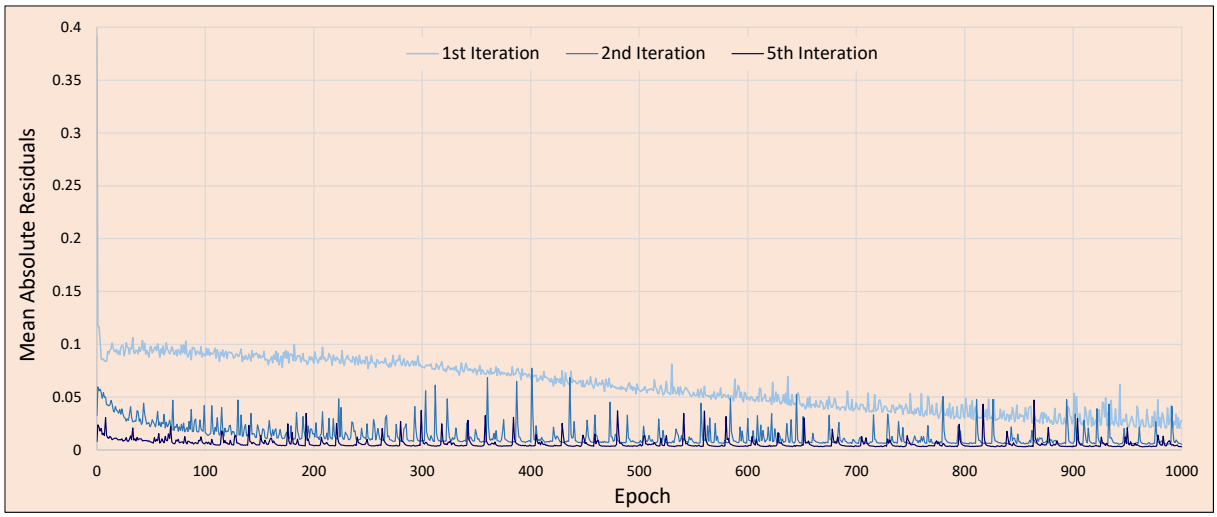

Figure 4: Phase II Training of the Initial ANN Configuration for Various RL Iterations.

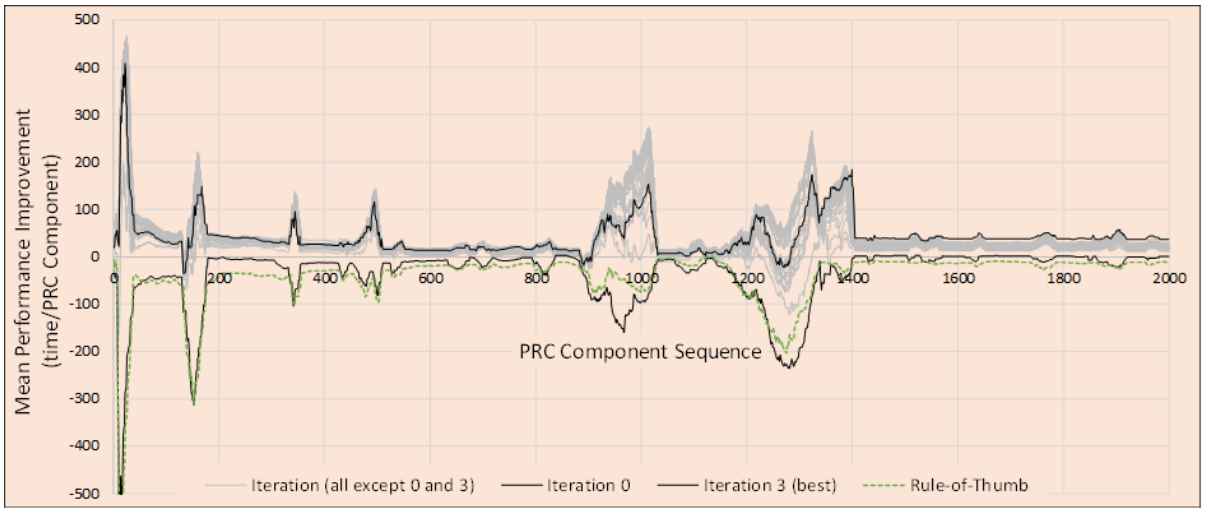

Figure 5: Training of the Initial ANN-Based Policy: progress for the first 50 RL iterations for the ANN-based policy compared to the rule-of-thumb policy.

A clear understanding of the performance of this initial ANN-based policy can be gained by inspecting Figure 5 and Figure 6. Both figures show performance as the mean improvement in the production time for a sequence of PRC components plotted against the PRC component number in the sequence. Figure 5 shows performance for the 2,000 PRC component sequence used for training while Figure 6 shows performance for the 8,000 PRC component sequence used for validation. Performance is measured relative to the random policy (see Section 2.3). For example, a mean performance improvement of 100 indicates that the ANN-based policy requires 100 time units less to produce each PRC component compared to the random policy. The grey and black lines represent the performance of the ANN-based policy at different iterations in the RL process, with the black lines bracketing the first and best iterations. A total of 50 RL iterations were considered, at which point it was apparent no further improvement in performance was likely. The 14th iteration was found to be the best for this experiment, being measured at the completion of the 8,000 PRC component sequence used for validation. The green curve in each figure shows performance using the rule-of-thumb policy described in Section 2.3.

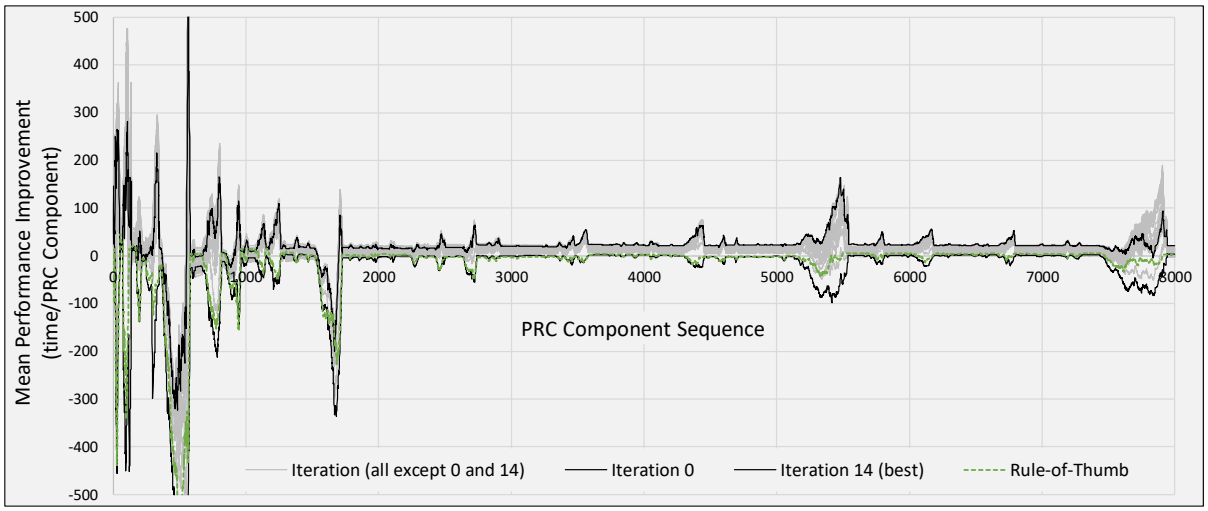

Figure 6: Validation of the Initial ANN-Based Policy: progress for the first 50 RL iterations for the ANN-based policy compared to the rule-of-thumb policy.

For this first experiment, the rule-of-thumb policy did not outperform the random policy, converging on a mean performance improvement of close to zero. In other runs of the random policy, the rule-of-thumb would sometimes start-off better than the random policy but as the PRC component sequence proceeded, it was found to converge on a similar conclusion. The ANN-based policy, on the other hand, significantly outperformed the random policy achieving over 21 time units of improvement per PRC component by the end of the 8,000 validation sequence. That is around 4.2% of the expected production time for the PRC components.

Comparing between the plots of Figure 5 and Figure 6, it can be seen that the ANN-based policy performed better for the training rather than the validation sequence. This is to be expected as the ANN is likely to develop an inherent bias towards the training examples, and for this reason, the validation results are taken to provide an independent and more accurate measure of performance.

4.2 Optimization of the ANN-based policy for production performance

After assessing the performance of the initial ANN-based policy, the goal was to optimize its effectiveness through a series of experiments tuning the following four modelling parameters: the number of hidden layers, the number of hidden neurons per layer, the reward length (n), and the number of PRC components that can be sampled from a queue by the ANN (N). The values for these parameters considered for optimization are summarized under Experiments 2 to 5 in Table 2. All variants of this ANN-based policy used the same development procedure described above in Section 4.1, that is, running the RL method for 50 iterations, using a training sequence of 2,000 PRC components and a validation sequence of 8,000 PRC components, and training the ANN at each RL iteration for 1,000 epochs.

Experiment 2 was concerned with ranging the number of hidden layers in the ANN from 1 to 8 in single layer increments (see Table 2), and measured the impact of this on the performance of the ANN-based policy. The results

of this analysis are presented in Figure 7 part (a), showing the relationship between validation performance and the number of hidden layers in the ANN. There is a clear upward trend in performance with a reduction in the number of hidden layers, with the optimum being a single hidden layer. This was not expected as it was thought that an effective decision policy for this application would be composite in form and thus would benefit from a deeply layered ANN structure. However, the smoothness of the performance curve in Figure 7 part (a) indicates that this is a sound conclusion rather than a stochastic anomaly. Reducing the number of hidden layers from 6 to 1 increased the mean performance improvement from 21.3 to 105.4 time units per PRC component.

Experiment 3 adopted the optimum value of 1 hidden layer determined in Experiment 2 and ranged the number of hidden neurons in the layer from 2 to 256 (see Table 2). The effect of this on the validation performance of the ANN is shown in Figure 7 part (b). Again, the trend of the curve is relatively well behaved apart from a slight irregularity around the optimum value. The initial estimate of the optimal value was found to be correct at 64 hidden neurons.

Experiment 4 adopted the previously derived optimal values for the structure of the ANN and ranged the reward length, *n*, from 2 to 2,000 PRC components (see Table 2). Figure 7 part (c) shows the effect of this on the validation performance of the ANN. There is an obvious peak in the performance curve close to the centre of the plot, with an optimal reward length measured to be 20 PRC components, the same as that estimated in the initial experiment.

Experiment 5 completed the optimization tests by ranging the number of PRC components in a queue that can be sampled by the ANN-based policy, *N*, from 0 to 30. The effect of this on validation performance is shown in Figure 7 part (d). There is a distinct peak in performance when N is set to a maximum of 10 PRC components, increasing the mean performance improvement to 168.9 time units per PRC component, approximately 34.4% of the expected production time for the PRC components. This is approximately 2.27 times better than the mean performance improvement achieved in the proof of concept study by Flood & Zhou (2023).

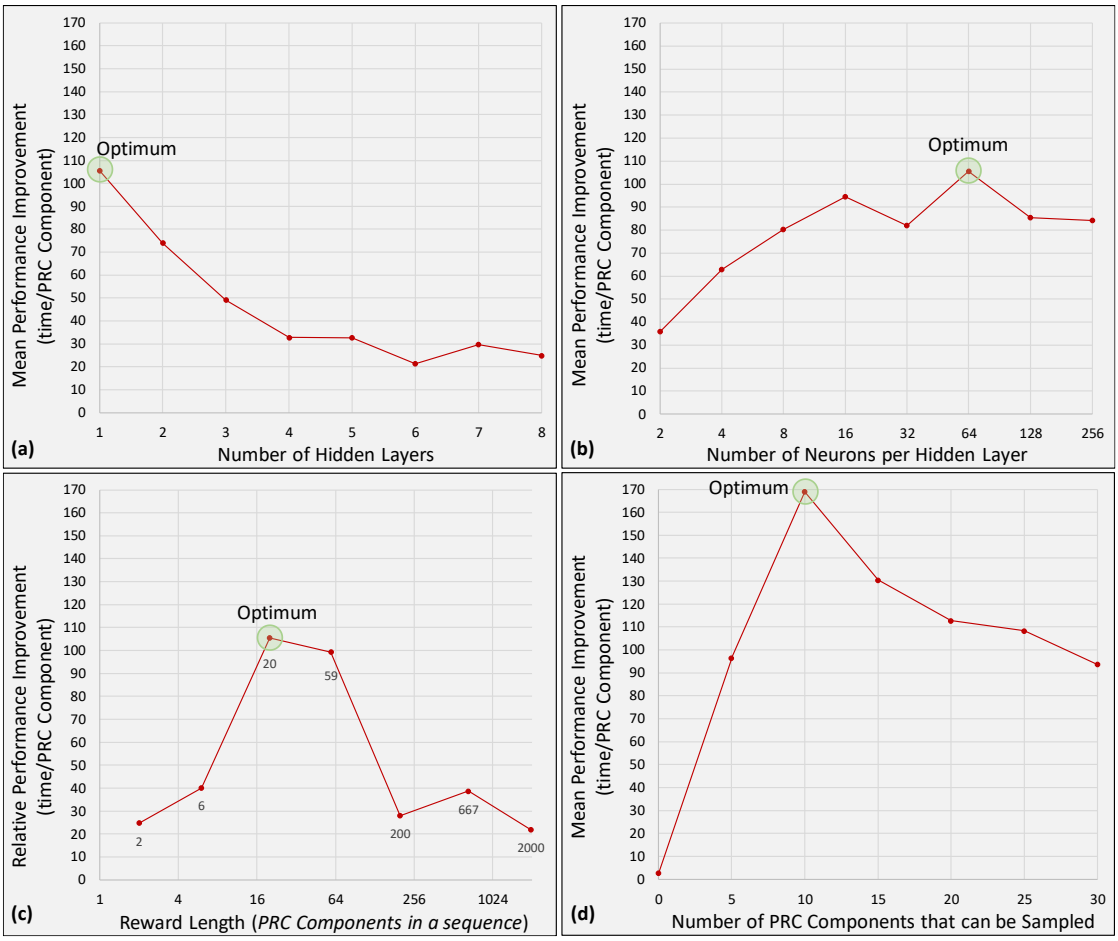

Figure 7: Summary of the Optimization of the ANN-Based Policy for Production Performance (results based on the PRC component sequences used for validation).

A summary of the performance progress of the RL-based policy for the optimization experiments is provided in Figure 8. Each curve represents the mean improvement in the production time for a sequence of PRC components, plotted against the first 50 iterations in the RL procedure. The solid curves represent progress for the sequence of 2,000 PRC components used for training, and the dashed curves represent progress for the sequence of 8,000 PRC components used for validation.

The light blue curves in Figure 8 represent RL performance progress for the ANN-based policy before any of its parameters were optimized, that is, the ANN evaluated in Section 4.1 (Experiment 1 in Table 2). The validation progress curve clearly has a significantly lower performance than the corresponding training progress curve. In addition, the validation curve dropped-down to around zero improvement at the 25 RL iteration and beyond. Both of these observations indicate overfitting probably caused by having too many hidden layers in the ANN, which in turn provided too many degrees of freedom for learning – this ANN had 6 hidden layers and the optimal was found to be just 1 hidden layer.

The dark blue curves in the centre of Figure 8 represent RL performance progress for the partially optimized ANN-based policy where the number of hidden layers was set to 1. In this case, the validation and training progress curves are close and level off without reducing, indicating no apparent overfitting.

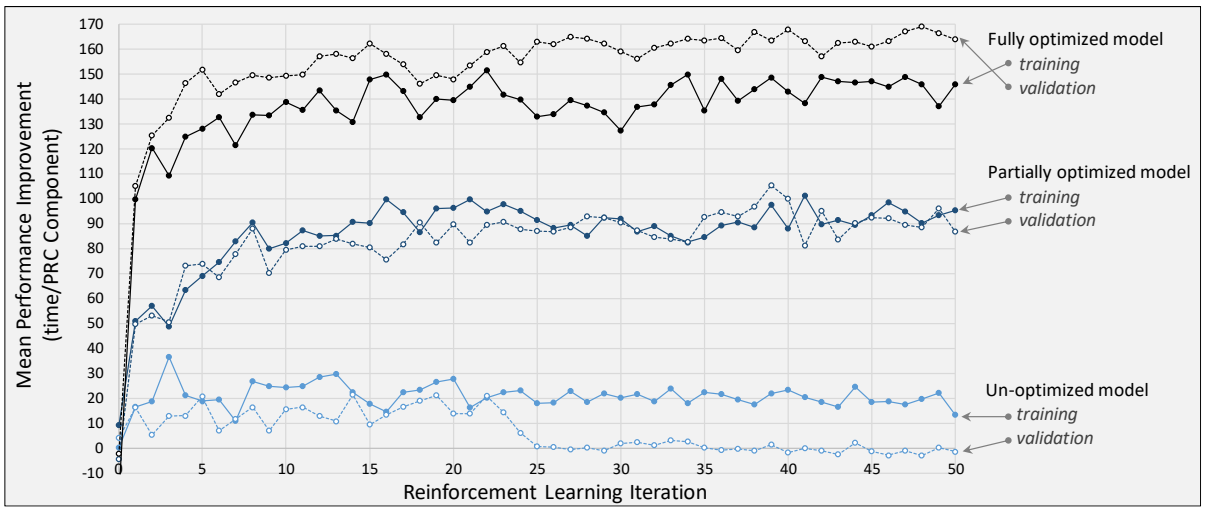

Figure 8: RL Performance Progress for Different Stages in the Optimization of the ANN-Based Policy.

The grey curves at the top of Figure 8 represent RL performance progress for the fully optimized ANN-based policy, that is, using the model parameters summarized under Experiment 6 in Table 2. Surprisingly, the validation curve outperformed the training curve. This is most likely due to chance but will be further investigated in future studies. Optimum performance was measured at the 48th RL iteration, but the trends of these curves suggest that further improvement is possible if the RL procedure were allowed to continue.

4.3 Summary of final model performance

Figure 9 shows the performance of the optimized ANN-based policy plotted against the 8,000 PRC component validation sequence. It is included for comparison with the same plot for the un-optimized ANN-based policy given in Figure 6. The difference in the extent of performance improvement is easily visualized by comparing these two figures, with the optimized ANN tending to outperform the un-optimized ANN by a factor of approximately 8. What is also apparent, although more subtle, is that perturbations in the performance curves (for example, around the 4,400 or 5,400 PRC components) are typically less extreme for the optimized ANN.

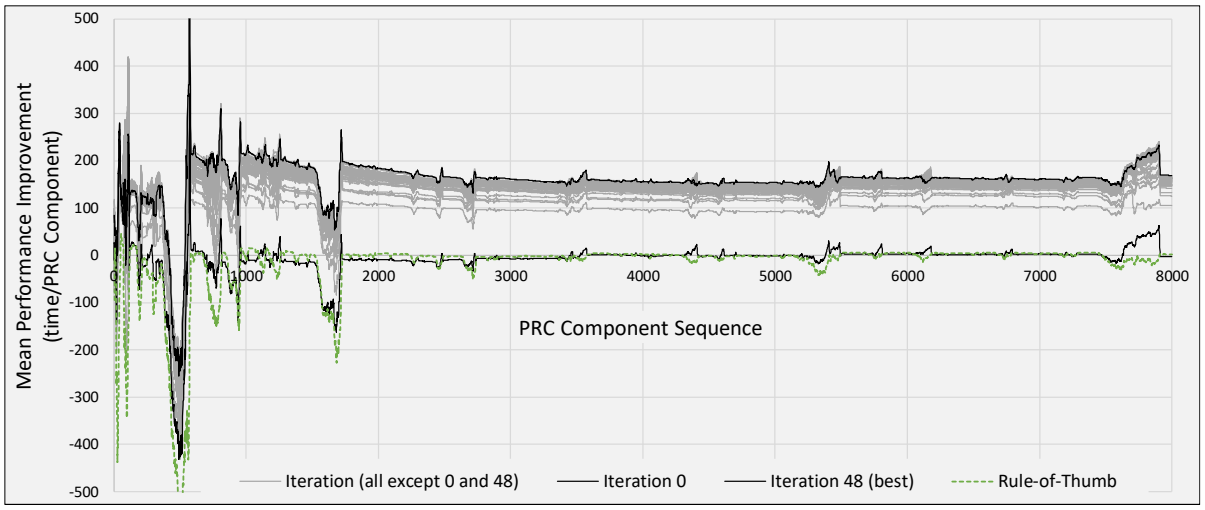

Figure 9: Validation of the Fully Optimized ANN-Based Policy: progress for the first 50 RL iterations for the ANN-based policy compared to the rule-of-thumb policy.

5. CONCLUSION AND FUTURE WORK

This study's primary objective was to develop and analyse a real-time control method for factory-based construction manufacturing, with the goal of optimizing its performance. The adopted method utilized an ANN-based decision policy, trained with RL techniques, to manage effectively the manufacturing processes to meet delivery deadlines. The study focused on real-time fabrication process optimization, addressing unique challenges specific to the construction manufacturing industry, such as demand uncertainty, high customization, and limited stockpiling opportunities.

A series of experiments were conducted using real factory data to optimize the performance of the ANN-based decision policy through sensitivity analysis techniques. The results demonstrated the significant advantage of this real-time decision-making approach over a standard rule-of-thumb policy for controlling PRC component production. It was further established that selecting an appropriate ANN structure and fine-tuning the RL development parameters are critical for enhancing the policy's production performance. For instance, the study revealed that employing a shallow ANN model resulted in approximately a fivefold performance improvement compared to a deep ANN model.

Future work will concentrate on refining the ANN-based policy's performance and broadening its scope of application to various other construction manufacturing operation problems. For example, the use of multi-agent systems to deal with multiple decisions simultaneously would expand the system's range of application considerably. In addition, the inclusion of an ensemble model, adopting alternative RL algorithms, and experimenting with a mix of reward lengths, could lead to further gains in performance. Additional case studies and onsite interviews should be undertaken to extend the range of applicable constraints taken into account. Continual refinement and expansion of this approach holds the promise of substantially enhancing the economic efficiency and viability of factory-based manufacturing processes within the construction industry.

Consideration will also be given to evaluating alternative policy development methods suitable for real-time control of manufacturing processes that have yet to be explored in depth. Potential approaches include evolutionary algorithms (Slowik & Kwasnicka, 2020) used to develop ANN-based policies similar to that used by Flood (1989) but using state-of-the-art methods, and model predictive control (MPC) (Schwenzer et al., 2021) which has had success in a wide range of control system applications including robotics and motion control.

REFERENCES

Benjaoran, V., Dawood, N., and Hobbs, B. (2005). Flowshop scheduling model for bespoke precast concrete production planning, *Construction Management and Economics*, 23(1), 93-105.

Chan, W. T., and Hu, H. (2002). Production scheduling for precast plants using a flow shop sequencing model, *Journal of Computing in Civil Engineering*, 16(3), 165-174.

Dan, Y., Liu, G., and Fu, Y. (2021). Optimized flowshop scheduling for precast production considering process connection and blocking, *Automation in Construction*, 125, 103575.

Delgado, J. M. D. and Oyedele, L. (2022). Robotics in construction: A critical review of the reinforcement learning and imitation learning paradigms, *Advanced Engineering Informatics*, 54, 101787.

Deng, F., Liu, G., and Jin, Z. (2013). Factors formulating the competitiveness of the Chinese construction industry: Empirical investigation, *Journal of Management in Engineering*, 29(4), 435-445.

Flood, I. (1989). A neural network approach to the sequencing of construction tasks, *Proceedings of the Sixth International Symposium on Automation and Robotics in Construction, Construction Industry Institute*, San Francisco, CA, 204-211

Flood, I. and Flood, P.D.L., (2022). Intelligent Control of Construction Manufacturing Processes Using Deep Reinforcement Learning, *Proceedings of the 12th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2022*, Lisbon, Portugal, 112-122. DOI: 10.5220/0011309600003274.

Flood, I. and Zhou, X., (2023). Improving Delivery Performance of Construction Manufacturing Using Machine Learning, *Journal of Simulation Engineering, JSimE*, 3, 17 pp.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., and Pineau, J. (2018). An introduction to deep reinforcement learning, *Foundations and Trends in Machine Learning*, 11(3-4), 219-354.

Glorot, X., Bordes, A., and Bengio, Y. (2011, June). Deep sparse rectifier neural networks, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings*, 315-323.

Hong, J., Shen, G. Q., Li, Z., Zhang, B., and Zhang, W. (2018). Barriers to promoting prefabricated construction in China: A cost–benefit analysis, *Journal of Cleaner Production*, 172, 649-660.

Kim, G. H., and Lee, C. G. (1995). Genetic reinforcement learning approach to the machine scheduling problem, *Proceedings of 1995 IEEE International Conference on Robotics and Automation, IEEE*, 1, 196-201.

Kim, T., Kim, Y. W., Lee, D., and Kim, M. (2022). Reinforcement learning approach to scheduling of precast concrete production. *Journal of Cleaner Production*, 336, 130419.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems 25*, NIPS 2012, 9 pp.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning, *Nature*, 521(7553), 436-444.

Leu, S. S., and Hwang, S. T. (2001). Optimal repetitive scheduling model with shareable resource constraint, *Journal of Construction Engineering and Management*, 127(4), 270-280.

Martinez, P., Barkokebas, B., Hamzeh, F., Al-Hussein, M. and Ahmad, R., (2021). A vision-based approach for automatic progress tracking of floor paneling in offsite construction facilities, *Automation in Construction*, 125, 103620.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning, *arXiv preprint*, arXiv:1312.5602.

Navarro-Rubio J., Pineda P., and Navarro-Rubio R. (2020). Efficient structural design of a prefab concrete connection by using artificial neural networks, *Sustainability*, *MDPI*, 12 (19), 8226.

Panzer, M., and Bender, B. (2022). Deep reinforcement learning in production systems: a systematic literature review, *International Journal of Production Research*, 60(13), 4316-4341.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems,* 32.

Rashid, K. M., and Louis J., (2020). Activity identification in modular construction using audio signals and machine learning, *Automation in Construction*, 119, 103361.

Riedmiller, S., and Riedmiller, M. (1999). A neural reinforcement learning approach to learn local dispatching policies in production scheduling, *IJCAI*, 2, 764-771.

Schwenzer, M., Ay, M., Bergs, T., and Abel, D. (2021). Review on model predictive control: an engineering perspective, *International Journal of Advanced Manufacturing Technology*, 117, 1327–1349.

Shitole, V., Louis, J., and Tadepalli, P. (2019, December). Optimizing earth moving operations via reinforcement learning, *In 2019 Winter Simulation Conference, WSC*, 2954-2965.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... and Hassabis, D. (2017), Mastering the game of go without human knowledge, *Nature*, 550(7676), 354-359.

Slowik, A., and Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems, *Neural Computing and Applications*, 32, 12363–12379.

Statista (2022) Size of the global construction market in 2021, with forecasts from 2021 to 2030, https://www.statista.com/statistics/1290105/global-construction-market-size-with-forecasts/ (last visit 2022/9/25).

Sutton, R. S. (1984). Temporal credit assignment in reinforcement learning, *PhD Dissertation, University of Massachusetts Amherst*.

Sutton, R. S., and Barto, A. G. (2018). Reinforcement Learning: An Introduction, *MIT Press*.

van Rossum, G., and Drake, F. L. (1995). Python reference manual, *Amsterdam: Centrum voor Wiskunde en Informatica,* 111, 52 pp.

Vose, M. D. (1999). The simple genetic algorithm: foundations and theory, *MIT press*.

Wang, Z., Hu, H., and Gong, J. (2018). Framework for modeling operational uncertainty to optimize offsite production scheduling of precast components, *Automation in Construction*, 86, 69-80.

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., and Kyek, A. (2018). Optimization of global production scheduling with deep reinforcement learning, *Proceedings of 51st Conference on Manufacturing Systems, CIRP*, 72, pp. 1264-1269.

Xia, K., Sacco, C., Kirkpatrick, M., Saidy, C., Nguyen, L., Kircaliali, A., and Harik, R., (2021). A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence, *Journal of Manufacturing Systems, Elsevier*, 58, 210-230.

Yang, Z., Ma, Z., and Wu, S. (2016). Optimized flowshop scheduling of multiple production lines for precast production, *Automation in Construction,* 72, 321-329.

Zhang, W., and Dietterich, T. G. (1995, August). A reinforcement learning approach to job-shop scheduling, *International Joint Conferences on Artificial Intelligence, IJCAI,* 95, 1114-1120.

Zhou, L., Zhang, L., and Horn, BKP. (2020). Deep reinforcement learning-based dynamic scheduling in smart manufacturing, *Proceedings of 53rd Conference on Manufacturing Systems, CIRP,* 93, 383-388.

Zhu, H., Hwang, B. G., Ngo, J., and Tan, J. P. S. (2022). Applications of smart technologies in construction project management. *Journal of Construction Engineering and Management, ASCE*, 148 (4), 12 pp.