Reinforcement Learning for Smart Mobile Factory Operation in Linear Infrastructure Projects

Cao, Jianpeng; Čustović, Irfan; Soman, Ranjith; Hall, Daniel

DOI
10.22260/ISARC2024/0096

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the 41st International Symposium on Automation and Robotics in Construction

Reinforcement Learning for Smart Mobile Factory Operation in Linear Infrastructure projects

Jianpeng Cao[1], Irfan Čustović[1], Ranjith Soman[2], and Daniel Hall[1]

[1]Faculty of Architecture & the Built Environment, Delft University of Technology, Netherlands
[2]Faculty of Civil Engineering and Geosciences, Delft University of Technology, Netherlands
C.J.P.Cao@tudelft.nl, I.Custovic@tudelft.nl, R.Soman@tudelft.nl, D.M.Hall@tudelft.nl

Abstract

Mobile factories promise an increased project efficiency with on-demand production and Just-in-Time delivery of prefabricated elements. However, traditional scheduling methods predominantly focus on either factory or site and neglect the factory mobility, often leading to suboptimal synchronization. To address this gap, this paper introduces a novel reinforcement learning (RL)-based model for optimizing the operational policy of mobile factories in infrastructure projects. The developed model simultaneously schedules on-site and off-site operations, effectively integrating the performance metrics at the project level. Utilizing RL, the factory's production management system continuously learns and adjusts in response to real-time project developments, ensuring optimal decision-making regarding scheduling and resource allocation.

Keywords –

Reinforcement Learning; Mobile Factory; Scheduling;

1 Introduction

1.1 Mobile Factories for Infrastructure Projects

The evolution of on-site and near-site prefabrication factories in construction and architecture is marked by notable milestones. Early 20th-century pioneers like Walter Gropius, Martin Wagner, and Adolf Meyer introduced systems such as the “Occident System” and the “Frankfurt Assembly Method,” which emphasized standardization [1]. The latter part of the century witnessed unique projects, including Moshe Safdie's Habitat ‘67 [2] in Montreal and Thomas Herzog's EXPO 2000 timber roof in Hannover, highlighting local production for specialized architecture [3]. Additionally, SKANSKA AB’s “Flying Factories” [4] and LiWood’s “Field Factories” [5] for near-site modular timber prefabrication represent systematic attempts to bring prefabrication closer to construction sites, focusing on flexibility through low levels of automation.

More recent research has shifted focus towards enhancing the mobility of mobile factories. Alix et al. [6] introduced a reconfigurable manufacturing system designed for frequent relocations, adept at accommodating fluctuating demand. Following this, Wagner et al. [7] unveiled a transportable and adaptable timber construction platform, specifically for carpentry. This innovation was validated through the construction of an intricate wooden pavilion, demonstrating its potential to elevate both the quality and efficiency of carpentry work.

Benefits of mobile factories include efficient manufacturing and pre-assembly operations near the building site, safer and cleaner working environments, and reduction in the number of transport kilometers between the factory and the building site [8]. Particularly, this concept of a mobile factory is suitable for situations with long distances and high logistics costs like the fabrication of components on the construction site.

Despite the benefits of mobile factories, existing research underscores the necessity for broader industrial testing across various domains, as noted by Alix et al [6]. Specifically, the application of mobile factories in large infrastructure projects like rail and road construction remains limited. This gap is noteworthy given the alignment between the intrinsic benefits of mobile factories and the demands of infrastructure projects. Therefore, it is crucial to urgently develop operational policies and decision support systems for scheduling mobile factories in infrastructure construction.

1.2 Integrated Project Scheduling

Project scheduling is a crucial aspect of project management, especially in dynamic and complex environments like factory production and site construction. However, these two areas are typically addressed separately [9]. This separation overlooks the potential efficiencies that could be realized through an integrated scheduling approach. In the realm of industrialized construction, this integration becomes increasingly important. Industrialized construction requires a more streamlined and coherent workflow, ensuring that the prefabrication process in factories aligns precisely with the timelines and demands of site construction, thereby optimizing resource utilization and reducing project delays.

Most researchers in this field have adopted a strategy of integrating site scheduling with storage, delivery, and other logistics processes. For example, Ahn et al. [9] streamline the synchronization of factory output with site demands by optimizing truck-dispatching schedules, and enhancing resource utilization and project timelines. Wang and Hu [10] integrate site scheduling into production scheduling by adding element storing and transportation processes to the traditional production model. This modification allows for simultaneous storage of different elements in the stockyard post-production, with the timing of the storing and transportation processes being closely aligned with site requirements and schedules. However, both works do not consider factory mobility, which is essential for infrastructure projects, where the factory is transportable in alignment with the project progression.

To address the identified limitations in current research on mobile factories, this paper proposes an innovative approach using a reinforcement learning-based model to optimize operational policies in infrastructure projects. Unlike traditional project scheduling methods, which typically segregate factory production from site construction, our approach focuses on integrating these two critical components. Consequently, this approach not only promises improved project efficiency but also marks a significant step in adaptive project management. Building upon this foundation, the following section reviews existing research in RL-based scheduling methodologies, setting the stage for a deeper understanding of the approach's context and significance.

2 Literature Review

Reinforcement learning (RL) has emerged as a powerful tool in this domain, offering adaptive and efficient solutions. The current literature on RL in project scheduling demonstrates significant advancements in site and factory production scheduling.

2.1 Site Scheduling

The application of RL in site scheduling is characterized by a variety of approaches aimed at addressing the dynamic and complex nature of construction environments. Kedir et al. [11] and Lee et al. [12] showcase how RL can be used to simulate and adapt to changing conditions on construction sites. The hybrid reinforcement learning–graph embedding network model proposed in [11] exemplifies an innovative approach to simulating complex construction planning environments. It shows the potential of RL in reducing computational burdens while establishing effective activity sequences and work breakdown structures. Similarly, [12] applies a digital twin-driven RL method for adaptive task allocation, indicating RL's capability to enhance real-time decision-making and efficiency in dynamic construction environments. This emphasis on adaptability and prompt responsiveness is similarly reflected in [13], which presents a novel method for generating Look-Ahead Schedules using RL. This method addresses the challenges of manual planning by offering a faster, more efficient approach to scheduling construction site activities.

2.2 Factory Production Scheduling

In the realm of factory production scheduling, RL is utilized to address the challenges of variability and the need for adaptability in manufacturing processes. Several studies highlight various aspects of how RL can improve efficiency and adaptability in factory settings [14–17]. Du and Li focus on automated assembly planning for robot-based construction, employing Deep Reinforcement Learning (DRL) in a re-configurable simulator to enhance assembly planning processes [14]. This approach aligns with [15] and [16], which also explore the dynamic nature of factory environments and how RL can be used to respond to changes in orders and resources. The comprehensive review in [17] of RL applications in production planning and control further underscores the versatility of RL in managing diverse aspects of manufacturing, including production scheduling, capacity planning, and inventory management.

2.3 Point of departure

While the reviewed literature on RL in project scheduling offers significant insights, it reveals a notable limitation: the lack of integration between site scheduling and factory production planning. To enhance overall operational efficiency and achieve the promised benefits of on-demand production and Just-in-time delivery, it is crucial to achieve a seamless integration of mobile factories and construction sites. By integrating these two components, RL can drive the evolution of project management towards more streamlined, efficient, and sustainable practices.

3 Methodology

The research methodology followed in this research is centered around the development and validation of a RL algorithm. RL is a branch of machine learning that draws inspiration from the natural learning process. In RL, the behavior of an entity, known as an agent, is shaped by the outcomes of its previous actions. Positive outcomes reinforce certain behaviors, lending greater importance to those actions and the decisions leading up to them. RL builds upon the foundations of Markov Decision Processes [18] and stands apart from supervised learning in that it does not rely on labeled input-output pairs for learning [19]. A RL model formally comprises:

- A discrete set of states, S;
- A discrete set of possible actions for the agent, A; and
- A set of scalar rewards.

In this framework (shown in Figure 1), an agent selects actions based on the rewards previously received in similar states. The ultimate objective is to devise a policy π that maps states to actions in a way that maximizes the overall reward.

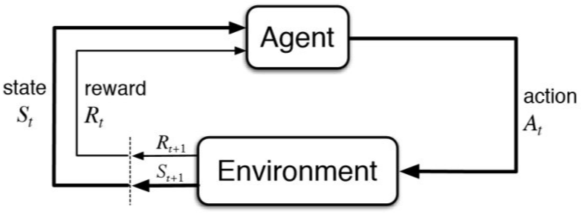

Figure 1. Reinforcement Learning Feedback Loop

The algorithm proposed in this research work aims to optimize operational policies in infrastructure projects that employ mobile factories for construction supply. The methodology uniquely combines logistical mobility with the complexities of production and assembly processes within the context of a mobile factory. This presents a novel and challenging environment for the application and exploration of RL techniques. In the following subsection, we will explain how the RL problem was formulated. These include, definition of environment, agent, agent's action space, reward and penalty.

3.1 Environment

The simulation divides a construction project into sections of uniform length, with on-site assembly of building components produced by the mobile factory. The assembly unfolds in a linear fashion, advancing from section to section until the project's completion is achieved.

3.2 Agent

The agent operates as the mobile factory, commencing at the first section and advancing toward the terminal section. It continuously tracks its position relative to the project and the quantities of both produced and assembled components.

3.3 Actions

The agent's operational choices are determined on a daily basis, introducing an element of strategic timing to the simulation. The agent has a repertoire of actions that directly influence the environmental state:

- **Production**: Engaging in this action, the factory commits to the fabrication of building elements at a set rate. While in production mode, the factory's status is updated to reflect the new production count, and its location remains unchanged.
- **Movement**: Opting to move prompts the factory to transition to the next section. This phase is characterized by a cessation of production, which realistically simulates operational downtime during relocation.
- **Idle**: By choosing to idle, the factory does not produce or move. This inaction provides an opportunity for strategic timing, potentially waiting for more favorable conditions or to better align with other segments of the project.

3.4 Reward

The reward function in this project environment is designed to incentivize optimal scheduling and resource allocation. It includes the following components:

- **Project Completion Reward**: This substantial reward is granted upon the successful completion of the entire construction project, i.e., when all sections have met their assembly requirements and the project has reached its final stage. This reward reflects the ultimate goal of project completion.
- **Milestone Reward**: Awarded each time the project successfully meets the assembly requirements for a current section and progresses to the next. This reward is a key driver for phase-wise project execution, encouraging the timely accomplishment of individual project segments. The milestone bonus not only acknowledges the completion of specific sections but also promotes a steady pace, ensuring that the project advances methodically from one stage to the next without unnecessary delays.

3.5 Penalty

The penalty function encapsulates various operational costs and risks, promoting efficient and strategic planning:

- **Duration Cost**: Emphasizes project time management, where shorter durations align with industry objectives. This cost accrues daily and escalates with extended project timelines.
- **Factory Running Cost**: Emphasizes the operational expenses associated with the daily functioning of the factory. This cost accrues continuously, reflecting the resource utilization and maintenance required to keep the factory operational.
- **Factory Movement Cost**: Underscores the expenditures associated with relocating the factory within the construction project area. This cost is incurred when the factory needs to be moved to a different location within the project site, often to align with the construction progress.
- **Shipment Cost**: Reflects the logistical complexity of material transportation from the factory to the construction site. This cost is quantified by the distance to the current project section and is enhanced by a predetermined factor, underscoring the value of logistical efficiency.
- **Inventory Cost**: Signals potential inventory management inefficiencies. This cost is activated when production outperforms assembly. The incurred cost, proportional to the imbalance and multiplied by a coefficient, advocates for a balance between production and assembly.
- **Underproduction Cost**: Underproduction Cost addresses the potential consequences of producing fewer components than required for the construction project. To mitigate this risk, additional resources may be needed, such as sourcing components from external suppliers or resorting to on-site production, often under urgent circumstances.

The penalty function complements the reward function by creating a balanced and comprehensive system of disincentives and incentives. This system encourages behaviors that are conducive to the overarching objectives of efficiency, cost-effectiveness, and timely delivery in construction project management.

4 Algorithm

Proximal Policy Optimization (PPO) [20] is selected for the mobile factory simulation. The PPO algorithm combines ideas from A2C (Advantage Actor-Critic) and TRPO (Trust Region Policy Optimization). It is well-regarded for its effective balance between exploration and exploitation, ensuring gradual improvements in decision-making. It operates by making incremental adjustments to its policies, which prevents drastic changes that could destabilize the learning process. This characteristic of PPO makes it particularly suitable for the mobile factory simulation, where decisions have a direct and significant impact on operational efficiency and project cost. The algorithm's ability to handle complex decision spaces and maintain steady progress is aligned with the requirements of coordinating production, assembly, and movement in the simulated environment.

5 Implementation

The implementation for the RL problem described utilizes the OpenAI Gym framework to create a custom environment, **FactoryEnv**, which simulates a mobile factory moving through different sections of a construction project. It is important to note that the values of parameters used in this setup are for illustrative purposes only. In a real-world project setting, users have the flexibility to customize these values according to specific project requirements. This customization capability ensures that **FactoryEnv** can be adapted to various construction scenarios, allowing for more accurate simulations and effective training of RL models tailored to the unique dynamics of each project. The environment is characterized by parameters:

Table 1. Environment parameters

Parameter	Description	Value
num_stops	The total number of sections the infrastructure is divided in	20
parts_per_stop	The number of parts required at each section	10
assembly_rate	The rate at which the site assembles parts per day	5 parts per day
production_rate	The rate at which the factory produces parts per day	8 parts per day
movement_time	The time it takes to move from one section to the next	3 days

Along with these operational parameters, the environment's behavior and agent's performance are influenced by a set of reward and penalty parameters, defined as follows:

Table 2 Reward and penalty parameters

Parameter	Description	Value

DAILY_COST	Cost incurred for each day of the project duration.	-1
FACTORY_RUNNING_COST_PER_DAY	Cost incurred for daily functioning of the factory	-10
MOVE_COST_PER_DAY	Additional cost for each day the relocation of the factory	-20
SHIPMENT_COST	Cost per unit of distance between factory and assembly location	-0.5
INVENTORY_COST	Cost for parts produced in excess of the assembly requirements	-1
UNDERPRODUCTION_COST	Cost incurred for parts underproduced relative to the assembly requirements	-5
COMPLETION_REWARD	Reward granted upon the successful completion of the entire project.	1000
MILESTONE_REWARD	Reward for each project section completed	30

Employing PPO via the stable_baselines3 library, this study utilizes a multi-layer perceptron for simultaneous policy and value function approximation within a custom-defined **FactoryEnv** environment. The model's architecture and hyperparameters are meticulously calibrated: a linearly scheduled learning rate commencing at 1e-4, a discount factor at 0.99, a GAE (generalized advantage estimate) lambda at 0.95, and an entropy coefficient of 0.005. Additionally, the network architecture comprises dual-layered structures with 128 neurons each for both policy and value functions. Batch processing is implemented with 2048 steps per batch, balancing computational efficiency with learning efficacy. The model's initialization incorporates these parameters, while TensorBoard integration facilitates detailed progress monitoring.

6 Results

In the presented results, we observe the performance metrics of a RL model over the course of training, measured across one million timesteps. Figure 2 delineates the trajectory of the training loss, a key indicator of the model's prediction accuracy regarding future rewards. The plot reveals an initial phase with a high variance in loss, indicative of the model's exploratory learning and parameter optimization. As training progresses, a clear downward trend emerges, culminating in a stable, low loss value, which suggests that the model's predictions have become more reliable and consistent. Figure 3 showcases the evolution of the average reward during the model's evaluation phase. The initial negative values represent suboptimal decision-making by the model. However, an enhancement is noted as the average reward increases, eventually reaching a plateau, demonstrating significant learning and policy improvement throughout the training process.

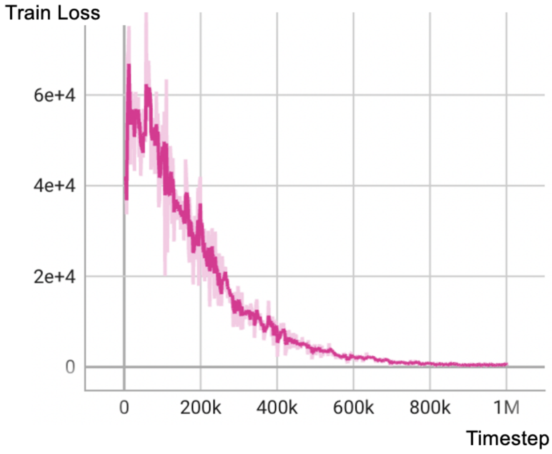

Figure 2. Training Loss Over Time

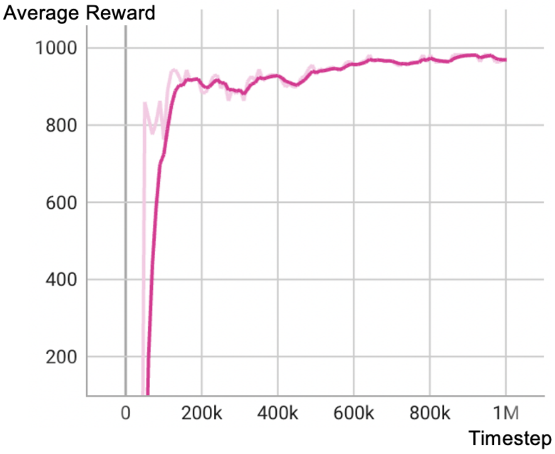

Figure 3. Evaluation of Average Reward

7 Validation

7.1 Single Environment Validation

In assessing the performance of our PPO-based RL model, we employed a quantitative validation strategy that entailed a comprehensive analysis of reward distributions. This strategy involved executing a random policy across 100,000 episodes within a consistent environmental setting of **FactoryEnv**. The objective was to establish a baseline distribution of rewards that could be leveraged as a comparative measure against the deterministic output of our trained RL model.

The histogram depicted in Figure 4 illustrates the frequency of total rewards obtained from the random policy across the 100,000 episodes. A dashed black line represents the reward achieved by our trained RL model, and a dashed red line denotes the maximum reward attained by the random policy throughout its trials. The RL model's reward, markedly higher than the random policy's mean and maximum reward, underscores the learning algorithm's success in optimizing decision-making to enhance reward outcomes.

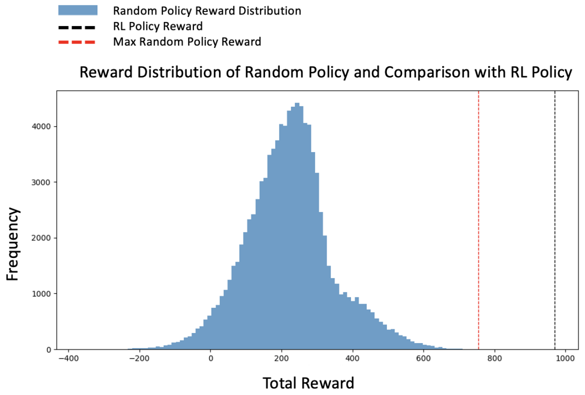

Figure 4. Reward Distribution of Random Policy and Comparison with RL Policy over the Same Setting

7.2 Diverse Environments Validation

To validate the robustness of the trained RL model, a comparative study was conducted against a baseline random policy. The comparison was done across a set of 100 diversified scenarios within a simulated environment, specifically designed to mimic a factory setting (**FactoryEnv**). Each scenario presented a unique configuration by varying the assembly rate, a critical parameter influencing the environment's dynamics. The assembly rates for each scenario were sampled from a normal distribution with a mean of 5 and a standard deviation of 1, ensuring a spectrum of operating conditions to challenge the robustness of the model. The trained RL model, developed using the PPO algorithm, was compared against the random policy in these scenarios to assess its adaptability and performance. The key metric for comparison was the total cumulative reward achieved by the end of each episode, serving as a proxy for the model's decision-making quality and efficiency.

The resulting performance, as shown in Figure 5, indicates a significant and consistent outperformance of the trained RL model over the random policy across all tested scenarios. The RL model achieved higher cumulative rewards in each individual case, demonstrating not only the ability to generalize across different settings but also the robustness of its learned policy.

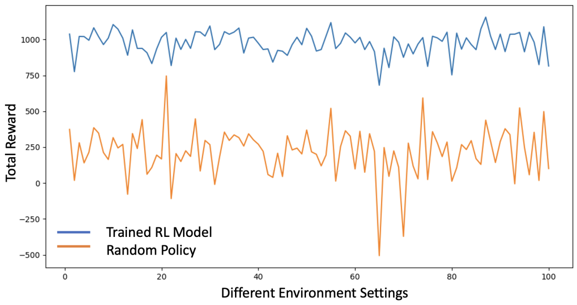

Figure 5. Comparison of RL Model versus Random Policy over Different Settings

8 Conclusion

This study contributes a novel RL-based scheduling model for optimizing the operation of mobile factories in infrastructure projects. It encompasses a comprehensive method for considering an array of performance indicators at the project level, including production and inventory costs, project duration, and shipping expenditures. As such, the reward and penalty parameters are designed to encourage cost-effectiveness and timely delivery of prefabricated elements. This aligns with the very motivation for applying mobile factories in construction projects – on-demand production and just-in-time delivery. Moreover, our approach exhibits remarkable flexibility, effectively adapting to a wide spectrum of production environments characterized by varying rates, the mobility of production facilities, and differing operational states such as idleness. Thus, the proposed method presents a holistic decision-making tool that can empower factory managers to optimize project execution strategies.

However, this research has some limitations. The environmental and reward parameters employed within the simulated setting may not entirely capture the complexity of real-world projects. For example, the COMPLETION_REWARD value could include various dimensions like the effort required, the time to completion, resources needed, and the complexity of tasks. As a next step, the practical application and validation of our approach in a real-world project setting will be imperative to ascertain its effectiveness and to fine-tune the model parameters for enhanced realism and applicability. By bridging the gap between theoretical modeling and practical implementation, we anticipate that our RL-based approach will offer tangible benefits in the management of factory and construction operations.

Acknowledgements

This work was carried out within the research project “Smart Mobile Factory for Infrastructure Projects (SMF4INFRA)” and supported by the Swiss National

Science Foundation [grant no. 204852].

References

[1] Seelow A. The Construction Kit and the Assembly Line—Walter Gropius' Concepts for Rationalizing Architecture. In Arts, pages 95, 2018.

[2] Safdie M. Beyond Habitat, volume 978-0262690362. MIT Press, 1973.

[3] Herzog T. Expodach: Roof Structure at the World Exhibition, Hanover 2000, volume 978-3791323824. Prestel Pub, 2000.

[4] Haukka S. and Lindqvist M. Modern Flying Factories in the Construction Industry, Master's Thesis, Lulea University of Technology, Lulea, Sweden. 2015.

[5] Alvarez M. et al. The BUGA Wood Pavilion - Integrative Interdisciplinary Advancements of Digital Timber Architecture. In Proceedings of the 39th ACADIA Conference 2019, pages 490–499, Austin, USA, 2019.

[6] Alix T., Benama Y., and Perry N. A framework for the design of a Reconfigurable and Mobile Manufacturing System. Procedia manufacturing, 35:304–309, 2019.

[7] Wagner H. J., Alvarez M., Kyjanek O., Bhiri Z., Buck M., and Menges A. Flexible and transportable robotic timber construction platform–TIM. Automation in Construction, 120:103400, 2020.

[8] Martínez S., Jardón A., Victores J. G., and Balaguer C. Flexible field factory for construction industry. Assembly Automation, 33(2):175–183, 2013.

[9] Ahn S. J. et al. Integrating off-site and on-site panelized construction schedules using fleet dispatching. Automation in Construction, 137:104201, 2022.

[10] Wang Z. and Hu H. Improved precast production–scheduling model considering the whole supply chain. Journal of Computing in Civil Engineering, 31(4):04017013, 2017.

[11] Kedir N. S., Somi S., Robinson Fayek A., and Nguyen P. H. D. Hybridization of reinforcement learning and agent-based modeling to optimize construction planning and scheduling. Automation in Construction, 142:104498, 2022.

[12] Lee D., Lee S., Masoud N., Krishnan M. S., and Li V. C. Digital twin-driven deep reinforcement learning for adaptive task allocation in robotic construction. Advanced Engineering Informatics, 53:101710, 2022.

[13] Soman R. K., and Molina-Solana M. Automating look-ahead schedule generation for construction using linked-data based constraint checking and reinforcement learning. Automation in Construction, 134:104069, 2022.

[14] Du Y., and Li J.-q. A deep reinforcement learning based algorithm for a distributed precast concrete production scheduling. International Journal of Production Economics, (2023): 109102.

[15] Shiue Y.-R., Lee K.-C., and Su C.-T. Real-time scheduling for a smart factory using a reinforcement learning approach. Computers & Industrial Engineering, 125:604-614, 2018.

[16] Shi D., Fan W., Xiao Y., Lin T., and Xing C. Intelligent scheduling of discrete automated production line via deep reinforcement learning. International Journal of Production Research, 58(11):3362-3380, 2020.

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

[18] Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction, 2nd ed. MIT Press, Cambridge, Massachusetts, 2018.

[19] Kaelbling L. P., Littman M. L., and Moore A. W. Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4:237–285, 1996.

[20] Schulman J., Wolski F., Dhariwal P., Radford A., and Klimov O. Proximal policy optimization algorithms. arXiv preprint:1707.06347, 2017