information

Article

Dynamic Scheduling of Crane by Embedding Deep Reinforcement Learning into a Digital Twin Framework

Zhenyu Xu [1], Daofang Chang [1], Miaomiao Sun [1] and Tian Luo [2,*]

[1] Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China; xuzhenyu0009@stu.shmtu.edu.cn (Z.X.); dfchang@shmtu.edu.cn (D.C.); mmsun@stu.shmtu.edu.cn (M.S.)
[2] School of Economics & Management, Shanghai Maritime University, Shanghai 201306, China
* Correspondence: luotian10@stu.shmtu.edu.cn

Abstract: This study proposes a digital twin (DT) application framework that integrates deep reinforcement learning (DRL) algorithms for the dynamic scheduling of crane transportation in workshops. DT is used to construct the connection between the workshop service system, logical simulation environment, 3D visualization model and physical workshop, and DRL is used to support the core decision in scheduling. First, the dynamic scheduling problem of crane transportation is constructed as a Markov decision process (MDP), and the corresponding double deep Q-network (DDQN) is designed to interact with the logic simulation environment to complete the offline training of the algorithm. Second, the trained DDQN is embedded into the DT framework, and then connected with the physical workshop and the workshop service system to realize online dynamic crane scheduling based on the real-time states of the workshop. Finally, case studies of crane scheduling under dynamic job arrival and equipment failure scenarios are presented to demonstrate the effectiveness of the proposed framework. The numerical analysis shows that the proposed method is superior to the traditional dynamic scheduling method, and it is also suitable for large-scale problems.

Keywords: digital twin; deep reinforcement learning; crane; dynamic scheduling

check for updates

Citation: Xu, Z.; Chang, D.; Sun, M.; Luo, T. Dynamic Scheduling of Crane by Embedding Deep Reinforcement Learning into a Digital Twin Framework. *Information* **2022**, *13*, 286. https://doi.org/10.3390/info13060286

Academic Editor: Zoran H. Peric

Received: 9 May 2022
Accepted: 2 June 2022
Published: 4 June 2022

1. Introduction

A general type of job shop scheduling problem (JSP) can be defined as n jobs being processed on m machines and how to sequence the jobs on each machine in such a way that some objective is optimal. Problems related to JSP have always been the focus of research. However, due to the lack of consideration of many constraints, the research results often have difficulty playing an efficient role in industrial applications [1]. Transportation between jobs is a topic that needs to be discussed, and it is an extremely important part of the entire manufacturing process [2]. On the one hand, the processing sequence of jobs affects the allocation of transport equipment and route selection, and then affects the overall transport efficiency. On the other hand, the transit time also affects the start time of job processing, which in turn affects the makespan. Therefore, there is a high correlation between the processing and transportation of jobs. The transportation equipment widely used in the workshop mainly includes cranes, automated guided vehicles (AGVs), manipulators and robots. Among them, the crane has the characteristics of high bearing capacity, which makes it indispensable and irreplaceable in the production field of shipbuilding, steel, heavy machinery, and other industries.

The structure of the crane determines that its transportation process is highly complex and dynamic. Tang et al. [3] developed a two-stage algorithm to solve the scheduling problem of a single crane in steel production to minimize the makespan. Liu et al. [4] proposed the integrated scheduling problem of crane transportation and flexible JSP, and developed a hybrid genetic and firefly swarm optimization algorithm to solve the energy consumption of manufacturing processing and transportation equipment. For the same

problem, Zhou et al. [5,6] developed two more competitive algorithms, including multi-objective evolution improvement and mixing with particle swarms, to faster find better solutions in the given solution space. Li et al. [7] designed a simulation-based solution to solve the scheduling problem of multiple cranes in a steelmaking workshop, and fully considered the physical interference and other constraints of cranes. Du et al. [8] established a distributed flexible JSP mathematical model with crane transportation, designed a hybrid algorithm and verified the performance through large-scale examples. These studies have greatly enriched the depth and breadth of crane transportation-related scheduling problems. However, these studies are focused on the optimization of static scheduling schemes, while enterprises usually face crane scheduling under dynamic job demand or real-time scheduling optimization.

The vigorous development of the Internet of Things, artificial intelligence and big data technology has strongly supported the transformation of Industry 4.0, and the resulting deep reinforcement learning (DRL) and digital twin (DT) technologies provide new solutions for dynamic scheduling optimization in the manufacturing field. DRL expresses the approximation function of reinforcement learning as a parameterized function form with a weight vector, usually to solve the Markov decision process (MDP). The essence of reinforcement learning is that the reward obtained by interaction with the environment guides the behaviour, and the goal is to maximize the return of the agent. Qu et al. [9] studied a manufacturing scheduling problem with multiskilled labour and multitype machines, constructed an MDP model for it, and developed a multi-agent-based Q-learning algorithm for dynamic scheduling updates. Shahrabi et al. [10] proposed a combination of Q-factor reinforcement learning and variable neighbourhood search to solve dynamic JSP. Aiming at the uncertain environment of the workshop, Wang et al. [11,12] realized the adaptive scheduling of the workshop through the improved Q-learning algorithm. With the amazing performance of AlphaGo in Go, the combination of deep learning and reinforcement learning shows better perception and decision-making ability. One of the methods is to use an artificial neural network to approximate the Q function. Lin et al. [13] studied JSP based on an edge computing intelligent manufacturing framework and further used this framework to adjust a deep Q network (DQN) to solve it to reduce the response time of production decisions. Shi et al. [14] applied discrete-time simulation in combination with DRL to the scheduling of linear, parallel, and reentrant automated production lines. For dynamic JSP, Liu et al. [15] proposed a DRL model including an actor network and critic network, which combines asynchronous update and a deep deterministic policy gradient to train the model. Improved algorithms based on DQN, including double DQN (DDQN), proximal policy optimization, and advantage actor criticism, have also been developed and used in production scheduling-related problems [16–20]. The application of DRL in scheduling problems has achieved good results. However, few studies have attempted to use DRL in problems related to crane scheduling.

DT efficiently connects the physical workshop with the virtual space and interacts with each other in real-time, and on this basis, it solves scheduling-related problems. Fang et al. [21] proposed a real-time JSP scheduling pattern based on DT, which adjusts parameters through dynamic events to reduce scheduling bias. Zhou et al. [22] studied a knowledge-driven DT manufacturing cell framework capable of autonomously learning scheduling rules. Wang et al. [23] built a DT scheduling model with a control mechanism and determined the rescheduling process through the system. Zhang et al. [24] divided the scheduling of the entity workshop into two levels: the whole workshop scheduling and the service unit scheduling, and the DT agent is used for hierarchical dynamic scheduling. For the JSP with limited transportation resources, Yan et al. [25] proposed a system framework combining the improved genetic algorithm with the five-dimensional DT to realize scheduling optimization. In summary, there is very little literature on the dynamic scheduling of cranes in the workshop, and there is also a lack of research on the effective application of DT or DRL in crane scheduling. However, the problem of dynamic scheduling optimization of cranes is extremely common in the industry, and enterprises also urgently need to apply

the industrial Internet of Things and artificial intelligence related technologies represented by DT and DRL to catch up with the manufacturing wave of Industry 4.0. Based on this, this paper mainly studies a dynamic scheduling problem with multiple cranes in the workshop, i.e., under the production situation of dynamic arrival of jobs in the workshop, to reasonably decide the processing sequence of jobs and assign cranes to transport these jobs, to achieve the goal of minimizing the makespan. It is important to note that this study considers multiple cranes transported on the same track, with identical crane sizes and parameters and physical interference between them, a type of crane system that is extremely common in the manufacturing industry. To solve this problem, a DT application framework is proposed, in which the DRL algorithm is embedded. DT can make the management of workshops transparent, digital, and intelligent, but needs effective decision algorithm support in the field of scheduling, and DRL's powerful intelligent learning, self-updating and efficient decision-making characteristics can just make up for its limitations. This study fills a gap in the research of dynamic crane scheduling in the manufacturing field and gives a template and technical route for enterprises to solve practical production logistics related scheduling problems through DT and DRL technology.

2. An Integrated DT Application Framework for Crane Transportation Dynamic Scheduling

The optimal crane transportation scheduling scheme can fully couple the operation process of the machine or station, thereby maximizing the productivity of the workshop with limited resources. Traditional scheduling methods face many difficulties. First, in the scenario of job dynamic arrival, the static scheduling method needs to start the rescheduling mechanism frequently, which leads to inefficiency. Second, the production process will produce massive data, from which how to accurately obtain the data related to crane scheduling is also an important issue to achieve accurate workshop scheduling. In addition, the interference of uncertain factors, such as equipment failure and temporary orders, often occurs in production, which will greatly reduce the efficiency of the original scheduling scheme. Therefore, embedding the DRL algorithm into DT for joint application has become the key to achieving the dynamic scheduling of cranes. Using the real-time interaction and virtual-real mapping characteristics of DT, as well as the intelligent perception and autonomous decision-making capabilities of DRL, the digital and transparent management and control of the workshop production process can be realized, and at the same time, it can better meet the needs of the new crane transportation scheduling mode. Based on ISO 23247, an architecture reference system for manufacturing DT provided by the International Organization for Standardization, this paper proposes a crane transportation dynamic scheduling DT integrated application framework. As shown in Figure 1, the DRL-DT consists of four parts: workshop physical space, twin data centre, cranes scheduling digital twin space and connection. It can be expressed as a four-dimensional model of Formula (1).

$$\text{DRL-DT} := \{\text{WPS}, \text{TDC}, \text{CSDTS}, \text{CON}\} \tag{1}$$

where := means defined as, WPS represents the workshop physical space, TDC represents the twin data centre, CSDTS represents the crane scheduling DT space, and CON represents the connection between different modules or elements. WPS is the cornerstone of the application framework, which includes manufacturing entities, logistics entities and their corresponding functional modules, and transmits workshop data to TDC in real-time through CON. TDC then stores, processes, and maps the data to provide data support services. Then, the logic simulation environment, algorithm library and 3D visualization model in CSDTS will actively obtain real-time workshop data or historical production data from TDC through CON according to scheduling instructions or functional needs. After calculating the scheduling instruction, CSDTS will issue the command through its workshop service system and control WPS to execute production. The four parts interact with each other, integrate virtual and real interactions, and realize the crane scheduling process of continuous iteration and dynamic optimization based on constantly updating data.

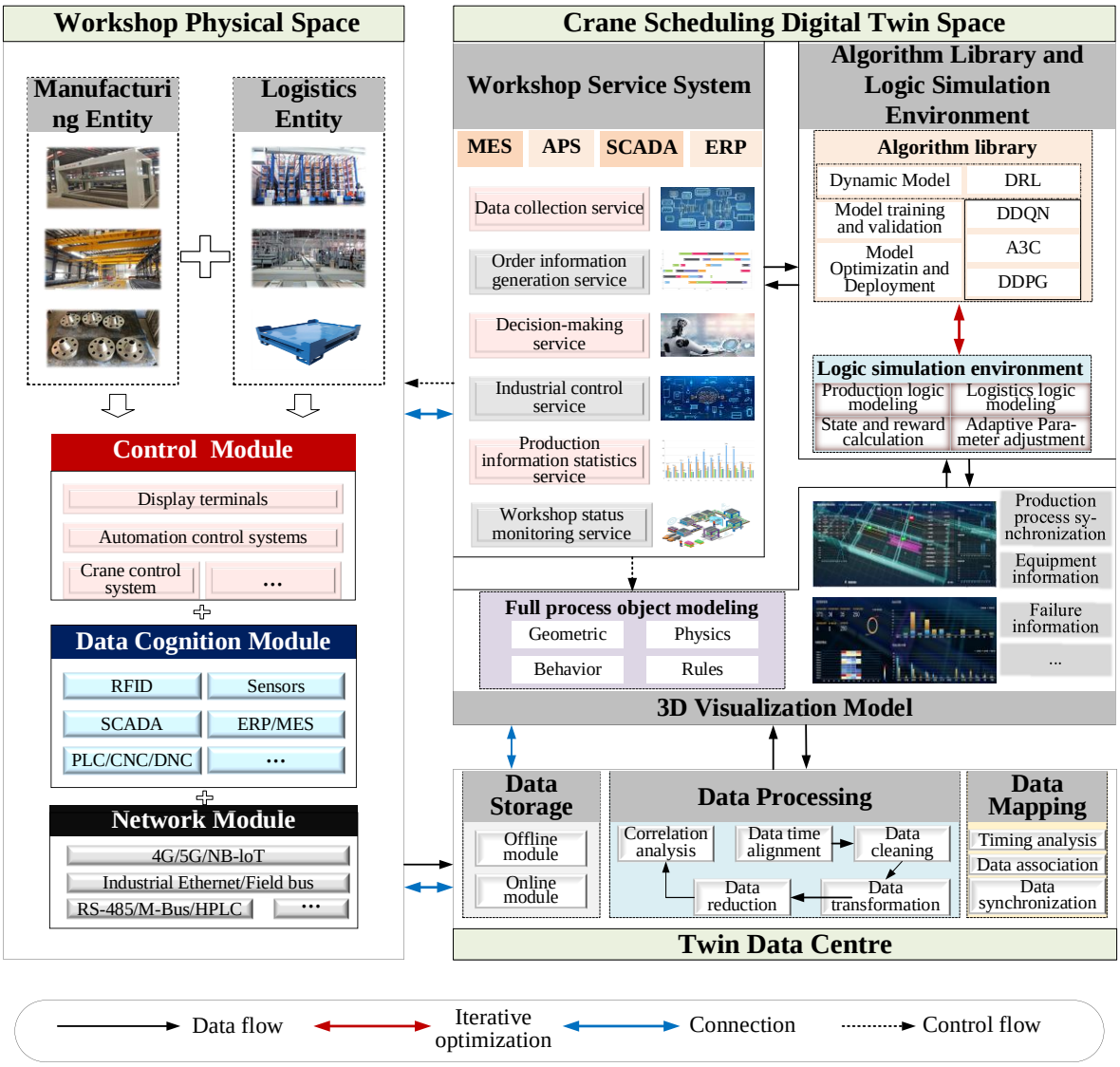


Figure 1. Crane transportation dynamic scheduling DT integrated application framework.

2.1. Workshop Physical Space

As shown in Figure 1, WPS mainly includes cranes, machines, workpieces, warehouses, buffers, and pallets. These entities are distributed in reasonable positions in the workshop and connected by IoT technology. The coupling of physical entities maintains the normal operation of the workshop and provides all-factor data information for the TDC. During data acquisition, cranes and machines mainly provide information such as equipment status (idle, working, blocking and failure), equipment location, job number, job destination and key parameters. Workpieces mainly provide information such as identification number, location, current processing status and destination. The warehouse and buffers provide the remaining capacity and stored material information, and pallets provide the information on workpiece occupancy and location.

The information perception and transmission of WPS is the basis of realizing the virtual-real interconnection and generating the optimal scheduling scheme. According to the entity of the physical workshop, three modules are defined: the control module, data cognition module and network module. The control module is used for controlling the automatic operation of the equipment and field management, and is composed of the display terminal, automatic control system, crane control system and the like. The data cognition module is used for multi-source data acquisition. On the one hand, process information such as workpiece and pallet location are acquired through hardware-based

deployments, such as RFID and sensors. On the other hand, production orders, equipment status and other information can be directly obtained from various systems or controllers in the workshop. The network module is responsible for ensuring the efficiency and quality of data transmission, which is mainly built by the cooperation of the cellular network, industrial Ethernet, and field bus. In addition, this study also defines the communication protocol based on the TCP/IP and OPC to realize high-frequency data transmission between the systems in DRL-DT.

2.2. Twin Data Centre

TDC is the medium that connects the physical workshop and the virtual workshop, and is the engine that drives the operation of DRL-DT. For data security, enterprises usually deploy TDC in private clouds. As shown in Figure 1, it mainly includes data storage, data processing and data mapping. The data storage contains an offline module and an online module. The offline module mainly stores the historical production data and related parameter information of the workshop for training the algorithm model and updating the virtual space; the online module covers the production operation, workshop parameters, scheduling scheme and other information for virtual and real interaction and performs dynamic scheduling based on real-time transmission.

Data processing is the core guarantee for DRL-DT to play an effective role. The original data generated by physical space and virtual space are mainly time series data, which require data time alignment, data cleaning, data transformation, data reduction and correlation analysis. The purpose of data time alignment is to make the data collected from different devices or systems have the same sampling frequency, i.e., to ensure the validity of the obtained combined data. Data cleaning is a necessary part of data processing. Its purpose is to screen and delete duplicate, invalid and redundant data, insert missing data, repair noise data, and finally form valid data for further processing. Based on the characteristics of scheduling data in this study, the general process of data cleaning is defined as follows. First, the data are preprocessed by backup and unified data format. Second, the missing data of the dataset are supplemented with the Newton interpolation method given by Equation (2). Finally, the filtered noise data are repaired, deleted, or corrected according to the average value of the abnormal degree.

$$N_n(x) = f_0 + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k] \prod_{j=0}^{k-1} (x - x_j) \tag{2}$$

where x_i is the independent variable of function $f(x_i)$, $\prod_{j=0}^{k-1} (x - x_j)$ is the polynomial, and $N_n(x)$ is the Newton basic interpolation polynomial of degree n. The advantage of Newton's interpolation method is that if one more node is added, there is no need to recalculate, only one more difference quotient needs to be calculated, and one more term can be added to the polynomial. Data transformation mainly refers to the functional transformation of data, the purpose of which is to unify the dimension of data and standardize it to facilitate subsequent processing and information mining when needed. Data reduction refers to the reduction of data dimensions and data volume to reduce the data scale on the premise of keeping the original appearance of the data as much as possible. This paper uses principal component analysis to deal with discrete data in the scheduling process. Correlation analysis was used to test the relationship between data.

Data mapping is used to support virtual-real interconnection and virtual-real synchronization and establish a mapping mechanism through data structure information. It includes timing analysis, data association and data synchronization [26].

2.3. Crane Scheduling Digital Twin Space

The CSDTS is mainly composed of a workshop service system, algorithm library, logic simulation environment and 3D visualization model. The workshop service system mainly

includes the manufacturing execution system (MES), advanced planning and scheduling (APS), supervisory control and data acquisition (SCADA) and enterprise resource planning (ERP). Through these systems, it provides production scheduling and management-related services for the workshop. These services run through the whole crane scheduling process. The algorithm library contains various DRL algorithms, which is the brain that directs the scheduling. In the following, this study designs a DDQN algorithm to implement crane dynamic scheduling decisions. DDQN needs to obtain accurate environmental feedback as a guide; thus, a logic simulation environment needs to be developed separately. The logic environment is used to simulate the operation and control mechanism of the complex crane joint workshop, calculate the corresponding state, and reward, and realize the dynamic interaction with the algorithm to ensure that the algorithm obtains an efficient scheduling scheme. The 3D visualization model based on full-process object modelling is directly deployed in the production field. On the one hand, it can display the realistic production and logistics process of the virtual and real synchronous operation effect, to facilitate the process visualization management of the workshop; on the other hand, it is used to display all types of statistical analysis and management operation data of the workshop to assist managers making decisions.

3. Problem Description and MDP Modelling

3.1. Crane Dynamic Scheduling Problem Description

The general scenario of crane dynamic scheduling studied is shown in Figure 2. Jobs arrive at the workshop dynamically during the production process. Each job has a known number of operations, and each operation needs to be processed on its matching machine. After finishing the processing of an operation on a machine, the job needs to be transported by a crane to the machine where the next operation is located, or to the exit of the workshop to finish the processing. The goal is to arrange a reasonable scheduling scheme to complete the specified task in the shortest time, i.e., to minimize the makespan. It is difficult to achieve the global optimal solution only by considering the allocation of cranes, and the processing sequence of operations needs to be considered synchronously.

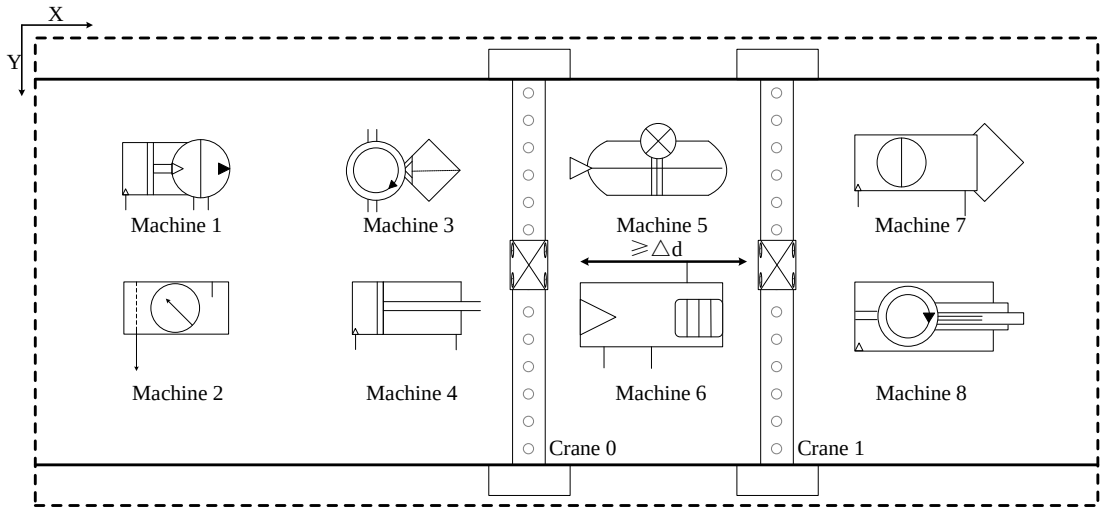


Figure 2. General workshop layout with crane transportation.

This study makes the following reasonable assumptions about the problem. Considering the safety factor, the movement of the crane in three directions needs to be performed step by step in turn; the machine can only handle one job at a time; the crane can only transport one job at a time; stealing is not allowed during the transportation of the crane; physical interference shall be considered among multiple cranes, but each crane can reach the position of all machines in the workshop for transportation; the multiple operations of the job shall be processed one by one in strict accordance with the established sequence.

To facilitate modelling, this study treats each operation to be processed as an independent task and uniformly puts it into the optional task set.

3.2. Problem as MDP

3.2.1. Basic Knowledge of MDP

DRL uses the formal framework of MDP to define the interaction process between the learning agent and the environment using state, action, and reward. Generally, the finite MDP can be described as a five-tuple (S, A, P, γ, R), and the agent and the environment interact according to the strategy π at every time. At discrete time t, the agent observes a certain characteristic expression $s_t \in S$ of the environment state and chooses an action $a_t \in A$ on this basis. In the next time step, the environment feeds back to the agent a numerical reward $r_t \in R$ and with the transition probability $p(s_{t+1}|s_t, a_t) \in P$ to enter a new state s_{t+1} and repeats this process until the final state is obtained.

3.2.2. State Definition

The state is used to describe the environment and guide decisions [27]. Considering the jobs, machines and cranes, the state at time t can be defined as vector $s_t = \{U_{mt}, MC_t, P_{ct}, J_t, R_{jt}, D_{rt}\}$.

(1) U_{mt} represents the average utilization of machines at time t.

$$U_{mt} = \frac{\sum_{k=1}^{N_m} u_{kt}}{N_m} \tag{3}$$

where u_{kt} is the utilization rate of machine k at time t, and N_m is the total number of machines. This state reflects the working efficiency of the machine.

(2) MC_t indicates the state of all devices at time t. $MC_t = \left\{\{z_{kt}\}_{k=1}^{N_m}, \{z_{ct}\}_{c=1}^{N_c}\right\}$, where z_{kt} represents the state of machine k at time t, z_{ct} represents the state of crane c at time t, and N_c is the total number of cranes. Whether it is a machine or a crane, there are four states, including idle, working, blocking, and failure, which are represented with 1, 2, 3, and 4, respectively. For example, $N_m = 3$, $N_c = 2$, and $MC_t = \{2, 1, 4, 3, 2\}$ means that at time t, machine 1 is working, machine 2 is idle, machine 3 is failed, crane 1 is blocking and crane 2 is transporting jobs. By defining MC_t, DRL can intelligently make optimal decisions under uncertain disturbances.

(3) P_{ct} represents the position of crane c at time t. $P_{ct} = \left\{\{x_{ct}\}_{c=1}^{N_c}, \{y_{ct}\}_{c=1}^{N_c}\right\}$, where x_{ct} and y_{ct} represent the position of crane c on the x- and y-axis, respectively.

(4) J_t represents the number of jobs to be transported, which reflects the scale of the current task set.

(5) R_{jt} represents the average processing time of the task set at time t.

$$R_{jt} = \frac{\sum_{i=1}^{J_t} t_{ijt}}{J_t} \tag{4}$$

where t_{ijt} represents the processing time of job i in the task set. This state is mainly used to feed back the time occupied by the job.

(6) D_{rt} represents the average transport distance of jobs.

$$D_{rt} = \frac{\sum_{i=1}^{J_t} d_{ijt}}{J_t} \tag{5}$$

where d_{ijt} represents the transport distance of job i. This state mainly reflects the average transportation volume of jobs. Of note, the transport distance is defined as the Euclidean distance from the machine where the job is currently located to the target machine.

3.2.3. Action Space

Actions directly determine the specific scheduling scheme. This study considers the scheduling policy itself as an Agent. Each time there is a scheduling request, a decision needs to be made about which crane to select and which job to transport. Therefore, this study designs an action space composed of eight heuristic rules shown in Table 1 and the optional set of cranes to serve the above decisions. The action space can be defined as vector $a_t = \{R_{ut}, C_{nt}\}$. Here, $R_{ut} = \{FIFO = 1, SPT = 2, STD = 3, SRPT = 4, NVF = 5, LWT = 6, FRO = 7, MPI = 8\}$. C_{nt} is the number set of cranes, $C_{nt} = \{Crane1 = 1, Crane2 = 2, \ldots, Cranei = i\}$. The selected action contains a heuristic rule and a crane. In the process of action execution, on the one hand, the crane selected in the action can be directly confirmed. On the other hand, the heuristic rule selected in the action is used to determine which job to transport. The number of optional actions is the product of the number of rules and the number of cranes. For example, if there are two cranes in the workshop, the total number of action combinations is 16. It is important to note that the simulation environment will first identify the crane to perform the task because some rules are strongly related to the selected crane, such as NVF.

Table 1. Heuristic rules for job selection.

Rule	Description
FIFO	Select the job with the first coming
SPT	Select the job with the shortest processing time
STD	Select the job with the shortest transportation distance
SRPT	Select the job with the shortest remaining processing time
NVF	The crane will select the job with the nearest load machine (Euclidean distance)
LWT	Select the job with the longest waiting time
FRO	Select the job with the fewest remaining operations
MPI	The crane will select the job with minimal physical interference from other cranes

3.2.4. Reward Function

In the DRL task, the agent continuously improves the strategy according to the reward from the environment during exploration. In fact, the essence of the DRL process is the deep processing of input state information by neural networks under the guidance of a reward function. The objective of the crane dynamic scheduling problem proposed in this study is to minimize the makespan, which is closely related to machine utilization and machine idle time. To achieve this, a function $\eta_k(t)$ representing the state of the machine is defined as:

$$\eta_k(t) = \begin{cases} 0 & \text{if machine } k \text{ is processing at time } t \\ -1 & \text{if machine } k \text{ is idle at time } t \end{cases} \tag{6}$$

Next, the reward function is defined as follows:

$$r_n = \frac{1}{N_m} \sum_{k \in N_m} \int_{\sigma=t_n}^{t_{n+1}} \eta_k(\sigma) d\sigma \tag{7}$$

where r_n represents the reward received by the agent after moving from state s_n to state s_{n+1}. Maximizing cumulative reward G is equal to minimizing makespan $C_{\max}$. The proof process is as follows [28]. Let N represent the number of trajectory states and T_k represent the idle time of machine k in the interval [0,$C_{\max}$], then,

$$\begin{aligned} G &= \sum_{n=1}^{N} r_n = \frac{1}{N_m} \sum_{n=0}^{N-1} \sum_{k=1}^{N_m} \int_{\sigma=t_n}^{t_{n+1}} \eta_k(\sigma) d\sigma = \frac{1}{N_m} \sum_{k=1}^{N_m} \sum_{n=0}^{N-1} \int_{\sigma=t_n}^{t_{n+1}} \eta_k(\sigma) d\sigma \\ &= \frac{1}{N_m} \sum_{k=1}^{N_m} \int_{\sigma=0}^{C_{\max}} \eta_k(\sigma) d\sigma = -\frac{1}{N_m} \sum_{k=1}^{N_m} T_k \end{aligned} \tag{8}$$

Since $T_k = C_{\max} - \sum_{i=1}^{I} \sum_{j=1}^{O_i} p_{ik}^j$, where p_{ik}^j denotes the processing time of operation j of job i on machine k, thus,

$$\begin{aligned} G &= -\frac{1}{N_m} \sum_{k=1}^{N_m} (C_{\max} - \sum_{i=1}^{I} \sum_{j=1}^{O_i} p_{ik}^j) = -\frac{1}{N_m} N_m C_{\max} + \frac{1}{N_m} \sum_{k=1}^{N_m} \sum_{i=1}^{I} \sum_{j=1}^{O_i} p_{ik}^j \\ &= \frac{1}{N_m} \sum_{k=1}^{N_m} \sum_{i=1}^{I} \sum_{j=1}^{O_i} p_{ik}^j - C_{\max} \end{aligned} \tag{9}$$

Since $\frac{1}{N_m} \sum_{k=1}^{N_m} \sum_{i=1}^{I} \sum_{j=1}^{O_i} p_{ik}^j$ is a constant, so that minimizing $C_{\max}$ is equal to maximizing G, i.e., the goal of DRL and the objective of the problem are identical.

4. Methodology

4.1. Double DQN Algorithm

Finding an optimal policy π to maximize the expected return is the essence of solving reinforcement learning problems. The optimal action-value function can be defined as:

$$Q_*(s,a) = \max_\pi Q_\pi(s,a) = \mathbb{E}[r_t + \gamma \max_{a'} Q_*(s',a') \big| s,a] \tag{10}$$

where $Q_*(s,a)$ is the optimal action-value function and $\gamma \in (0,1]$ is the discount factor. This formula is the Bellman optimality equation, which can be converged to Q_* using the tabular method. However, in practical engineering applications, the scale of state space and action space may be very large, which will lead to the problem of dimensional disaster. Using an artificial neural network $Q(s,a;\theta)$ with parameter θ to approximate the optimal value function $Q_*(s,a) \approx Q(s,a;\theta)$ can practically solve the above computationally difficult problems, which is the Deep Q Network. In addition, DQN also introduces a target network $\hat{Q}(s,a;\theta^-)$ and experience replay to collect training data and break the correlation of sequences, making the training of the algorithm more efficient. Standard DQN evaluates and selects actions with the same value, which can easily lead to overestimation problems. Therefore, the concept of double DQN was proposed. Without adding additional networks, the main network $Q(s,a;\theta)$ is used to select actions, and the target network $\hat{Q}(s,a;\theta^-)$ is used to evaluate actions. The target value is updated as follows:

$$y_t^{DDQN} = r_t + \gamma \hat{Q}(s_{t+1}, \text{argmax}_a Q(s_{t+1},a;\theta);\theta^-) \tag{11}$$

The purpose of training is to update the neural network to reduce the estimation error between it and the optimal action-value function. Then, the loss function is defined as:

$$L(\theta) = ((y_t^{DDQN} - Q(s_t,a_t;\theta))^2 \tag{12}$$

4.2. Crane Dynamic Scheduling Based on DRL-DT

This study uses the DRL algorithm DDQN to integrate with the DT framework. The implementation of crane dynamic scheduling by embedding DDQN into the DT framework consists of two steps: offline training and online scheduling. Due to the relatively long offline training time of the algorithm, this process is usually performed before the first deployment of DRL-DT. With the progress of production, the effective data will be accumulated all the time; thus, a fixed cycle, such as one month, will be set to regularly train the network model offline, so that the scheme obtained in online scheduling can maintain a high-quality level.

As shown in Figure 3, during offline training, the virtual space does not need to interact with the physical workshop in real-time, but it needs to obtain historical production data through the workshop service system to drive the logical simulation environment to run. These data mainly include information such as production orders and faults. At the

beginning of training, the algorithm hyperparameters are initialized, and the historical data are accessed for the logic simulation environment. The completion of a complete production order task is called an episode. When the number of training episodes reaches the preset value, the training is over, and the trained network is saved and applied to online scheduling. Otherwise, the logic simulation environment is reset at the beginning of each episode, and a new historical training dataset is selected. When a new operation is triggered in the logic simulation environment or the crane completes transport, the state s_t at this time is calculated and an action selection request is sent to the DDQN. This study uses the epsilon decreasing strategy to balance exploration and exploitation and then select action a_t with probability ε [18]. The update formula is:

$$\varepsilon = \varepsilon_{\max} - (\varepsilon_{\max} - \varepsilon_{\min}) * \min(1, n/N_E) \tag{13}$$

where $\varepsilon_{\max}$ and $\varepsilon_{\min}$ are the maximum and minimum values of the epsilon given at the beginning of training, respectively; n is the current step counter; and N_E is the total exploring steps. After receiving the action, the logic simulation environment completes the scheduling and gives the reward r_t and the new state s_{t+1} to the DDQN. This forms a transition (s_t, a_t, r_t, s_{t+1}) that is stored in the reply buffer D. If the buffer is full, the oldest transition is replaced. In each reply period, a random minibatch of transitions D is sampled based on Equation (11), and then calculate the target value y_t^{DDQN}. Then, the loss function is calculated by Equation (12), and the main network parameters are updated. To maintain the stability of the training process, the target network is updated every C steps.

In the online scheduling application, the physical workshop, workshop service system, logical simulation environment and 3D visualization model keep running synchronously at all times, and the four are interconnected through the TDC. When a new job is triggered in the physical workshop or a crane completes a transportation task, the workshop service system will transmit the scheduling request to the 3D visualization model and display it on the workshop field terminal and then send it to the TDC. On the premise that the algorithm network has been trained, the logic simulation environment will calculate the state of the workshop at this time and send it to the algorithm library. Next, the DDQN selects the optimal action based on the state, that is, it selects the crane and the job to be transported. When the crane is unavailable, the state needs to be recalculated; otherwise, the scheduling scheme is directly accepted and sent to the workshop service system, mainly to the crane control system and MES. Finally, the crane control system controls the crane in the physical workshop to perform the corresponding transportation tasks, and the MES is responsible for transmitting the scheduling information and production information to the 3D visualization model for real-time presentation. In addition, both offline training and online scheduling of the algorithm require high-frequency data transmission, and to ensure efficiency, this study uses the socket interface for data communication.

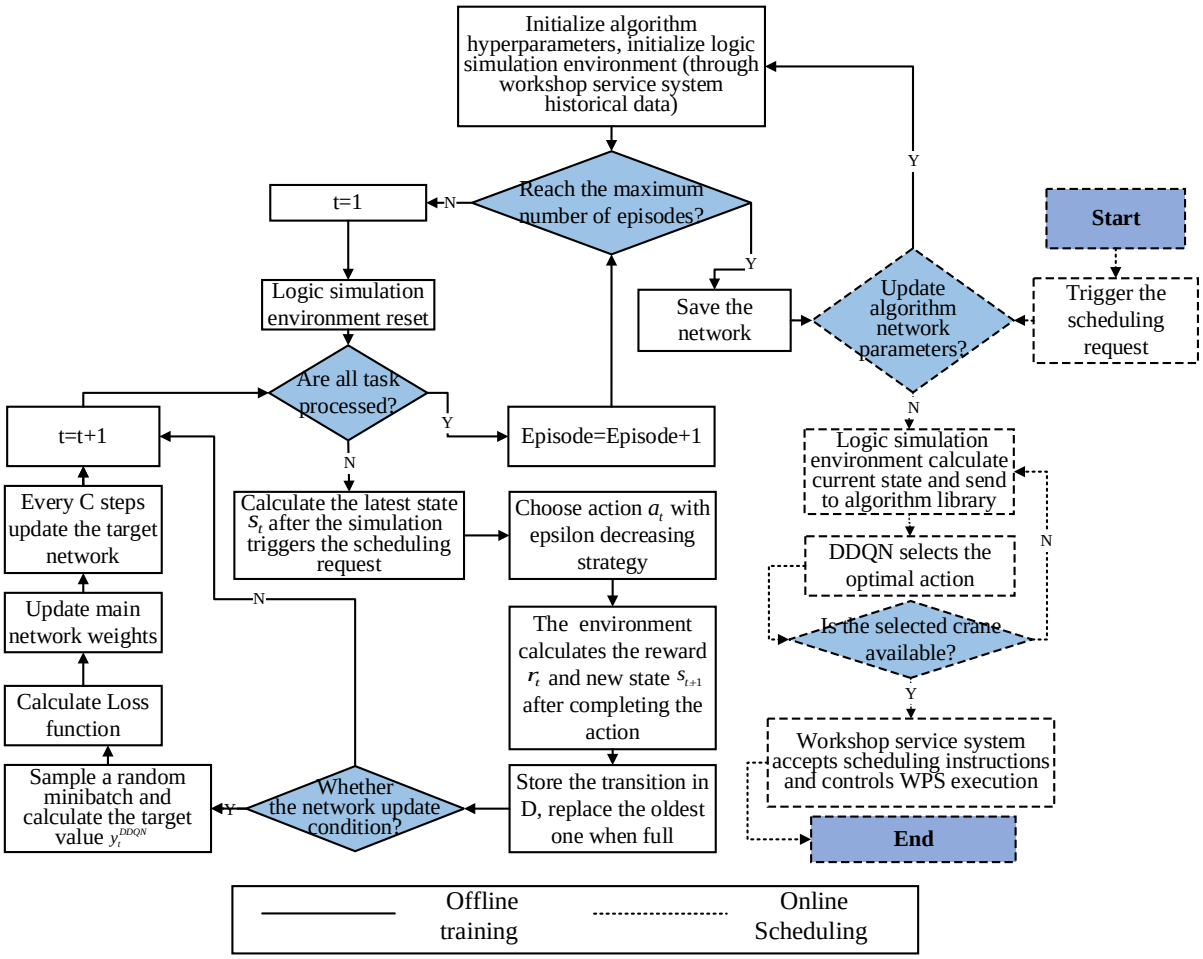


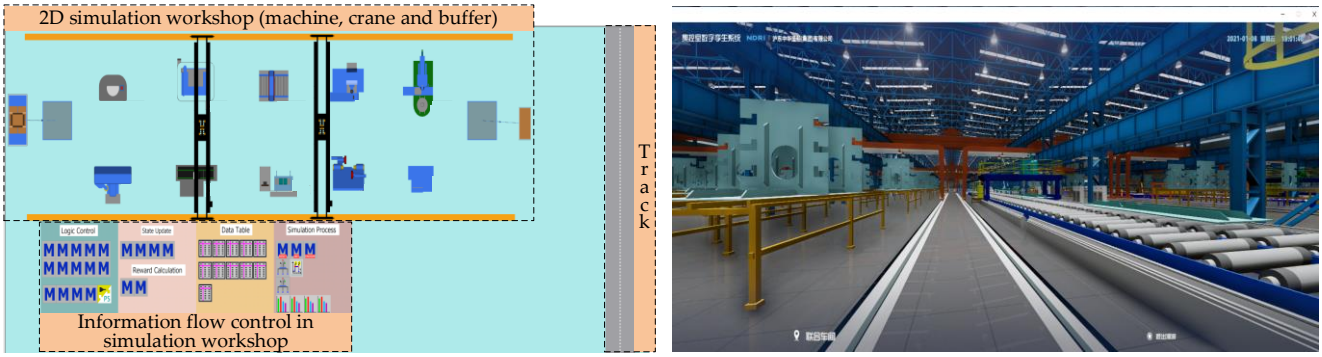
Figure 3. Crane scheduling under DRL-DT.

5. Case Study

5.1. Test Platform and Case Data

To verify the effectiveness of DRL-DT in crane dynamic scheduling, the relevant models, systems, and platforms are built based on the process flow of a hull block manufacturing workshop, and the advantages and disadvantages of using the digital twin framework and traditional heuristic rule scheduling mode are compared through numerical experiments. In this study, 3D Max and Unity were used to build a 3D visualization model (Figure 4a) with a 1:1 reduction of the real world, and the workshop consisted of 10 machines and 2 cranes. PyTorch is used to develop the DDQN program, which interacts with the logic simulation environment (Figure 4b) developed by Siemens industrial software Plant Simulation through TCP/IP protocol to form network connections for offline training and online scheduling.

The DDQN is trained offline before online scheduling. Set the number of training episodes to 5000, each episode contains 100 job tasks, ε decreases from 0.9 to 0.05, the discount factor $\gamma = 0.9$, the replay buffer capacity is 100,000, and the minibatch size of samples $K = 256$. The number of target network update steps $C = 200$. The speed of the crane is 0.5 m/s in each direction. Trained and tested on a PC with Intel Core i7-10875H @ 2.30 GHz CPU and 16 GB of RAM, the training took a total of 19 h and 34 min to complete. The online scheduling tasks of the test are 5 randomly generated jobs, each job contains 5 operations, and there are 25 operations in total. The arrival time of jobs follows the normal distribution $N(40\text{ s}, 10\text{ s})$. The test task data are shown in Table 2. For example, Operation 2 of Job 2 needs to be processed on machine 3 for 54 s.



(a) (b)

Figure 4. (**a**) Logic simulation environment. (**b**) 3D visualization model.

Table 2. Case test data.

	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5
Job 1	1:37	3:54	5:57	8:36	9:31
Job 2	1:56	3:66	5:33	7:25	9:68
Job 3	1:43	4:48	6:58	8:72	9:22
Job 4	2:31	3:75	6:48	7:74	9:63
Job 5	1:74	4:41	5:80	8:43	9:44

5.2. Results Analysis and Discussion

In traditional crane dynamic scheduling, it is very common to use heuristic rules when selecting jobs, because these methods are generally very fast and easy to implement. This study separately uses the eight basic job selection rules contained in the abovementioned action space to compare with DRL-DT. Considering the characteristics of cranes, this study combines NVF with FRO and MPI to form two new scheduling rules, namely, NVF-FRO and NVF-MPI. The specific combination method normalizes the dimension of the value under a single rule and then calculates the priority of the new rule with 50% of the weight of each rule. Based on case test data in Table 2, the makespan for each of the five jobs is shown in Figure 5.

Figure 5 shows that DRL-DT outperforms any other heuristic rule. It is 32.8% better than the worst performing STD. This mainly occurs because when scheduling according to the STD, the selected jobs of the crane are unreasonable in many cases, which leads to low transportation efficiency, coupled with the physical interference between the two cranes, which means the crane has a longer, avoidable waiting time. In addition, the NVF-FRO performed best among the heuristic rules; thus, this study calculates the detailed scheduling process under the DRL-DT and NVF-FRO methods, as shown in Figures 6 and 7, where "0" and "1" are the respective numbers of the two cranes. DRL-DT takes 8% less time than NVF-FRO to complete the whole task, which verifies the effectiveness and efficiency of the proposed method. By observing the scheduling process of the two, it can be found that the allocation of cranes and the sequence of operations have been reasonably adjusted; for example, in operations 1-5, NVF-FRO uses crane 0, while DRL-DT allocates crane 1 for transportation. As another example, on machine 9, DRL-DT chooses to perform the 5-5 job first instead of 4-5. These adjustments make the scheduling scheme given by DRL-DT more efficient.

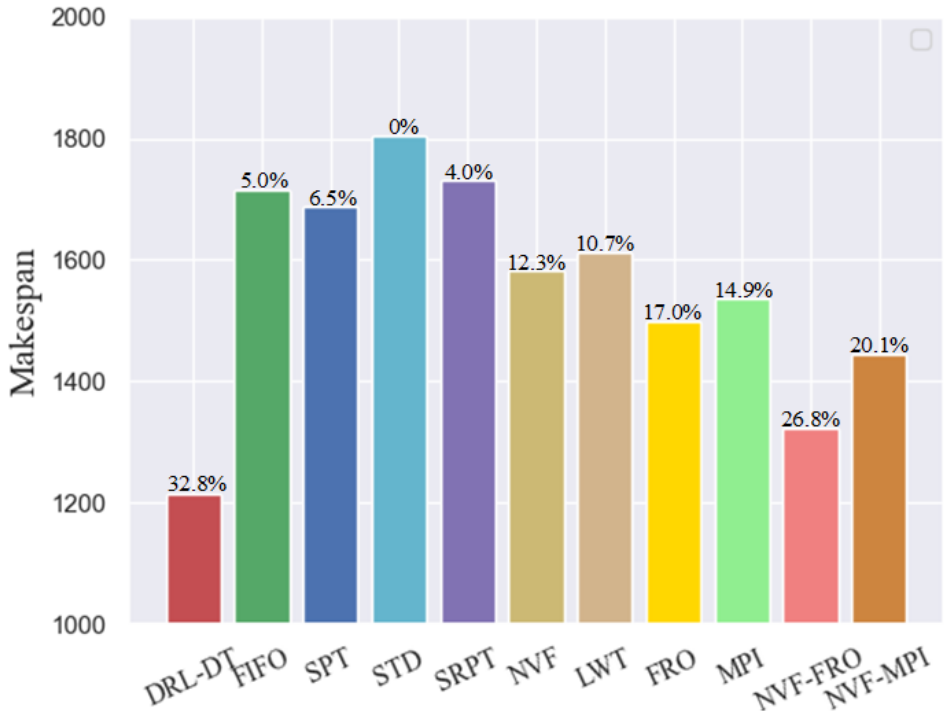


Figure 5. Comparison of the improvement of makespan depending on the method.

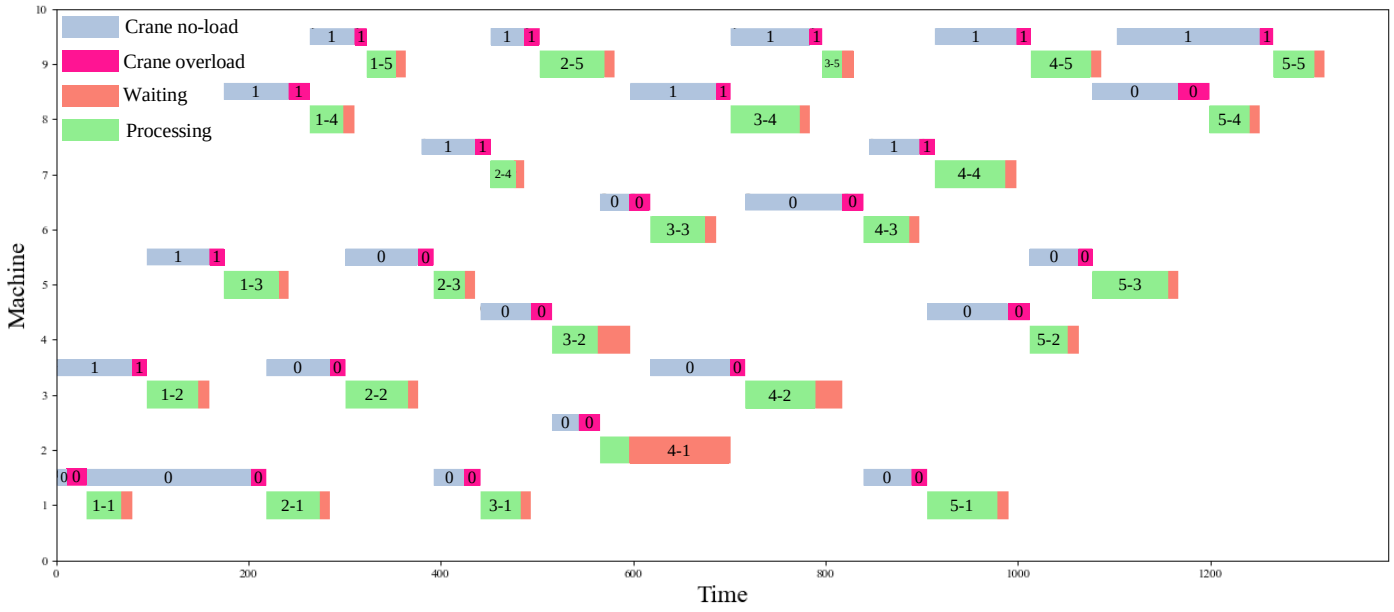


Figure 6. Scheduling Gantt chart (NVF-FRO).

DRL will directly consider uncertain disturbance factors such as equipment failure. When a fault is observed, the algorithm adaptively adjusts its choice of action. It is assumed that machine 1 and crane 0 failures occur at 400 s and 800 s respectively, during the scheduling execution process, and the failure time lasts for 100 s. At this time, the final time for each method to complete the scheduling is shown in Table 3. The obtained results show that the makespan of DRL-DT is still the smallest, which indicates that DRL-DT has the strongest ability to cope with failures.

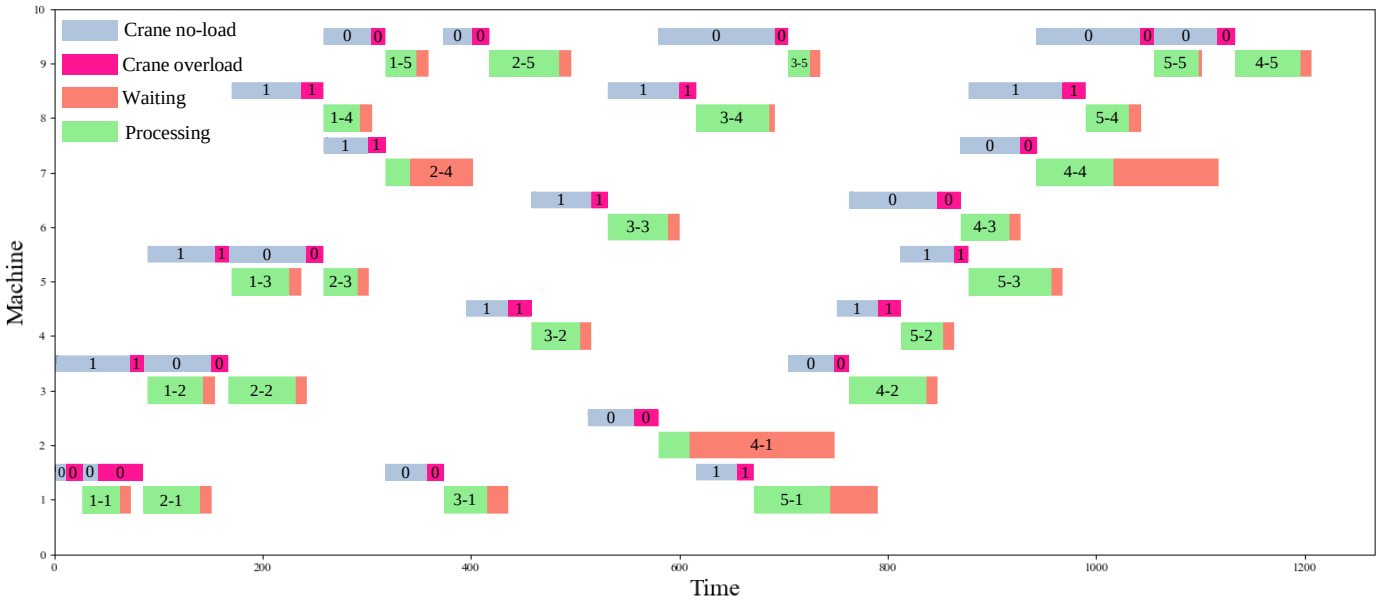


Figure 7. Scheduling Gantt chart (DRL-DT).

Table 3. Makespan (s) comparison of each method.

Methods	Makespan (No Failure)	Makespan (Failure)	Gap
DRL-DT	1213	1533	320
FIFO	1715	2420	705
SPT	1688	2312	624
STD	1805	2587	782
SRPT	1732	2644	912
NVF	1583	2331	748
LWT	1611	2170	559
FRO	1499	2198	699
MPI	1536	2161	625
NVF-FRO	1322	1785	463
NVF-MPI	1443	1838	395

The performance of DRL-DT in large-scale examples is the premise to determine whether it can be applied in practical engineering. For this reason, this study selected NVF-FRO and NVF-MPI, which have excellent performance, to compare with DRL-DT in four groups of job scenarios of different scales. The processing time of each operation of the job is set as a random number int (20, 80), and each group of experiments is performed ten times. As shown in Figure 8, DRL-DT still maintains sufficient superiority in large-scale examples. In addition, with an increase in the number of operations, the three methods have good stability and, basically, maintain a constant level of efficiency.

Through numerical experiments, it can be concluded that, compared with the scheduling mode under the traditional heuristic rule method, the production scheduling scheme obtained by DRL-DT is more efficient, especially more adaptable in the face of uncertain factors such as equipment failure, and shows a stable advantage in the face of large-scale production scenarios. More importantly, with the gradual use of DRL-DT after deployment, enterprises will get more and more production data. Therefore, since the powerful self-learning ability of the DRL algorithm, the obtained scheduling scheme will be better and better, and can adapt to the changes of various production factors in the workshop. In addition, the visualization technology given by the digital twin makes workshop production management transparent and digital, which is convenient for personnel to make

on-site decisions. At the same time, the production data analysis and mining capabilities brought by DT will also allow enterprises to maintain their core competitiveness.

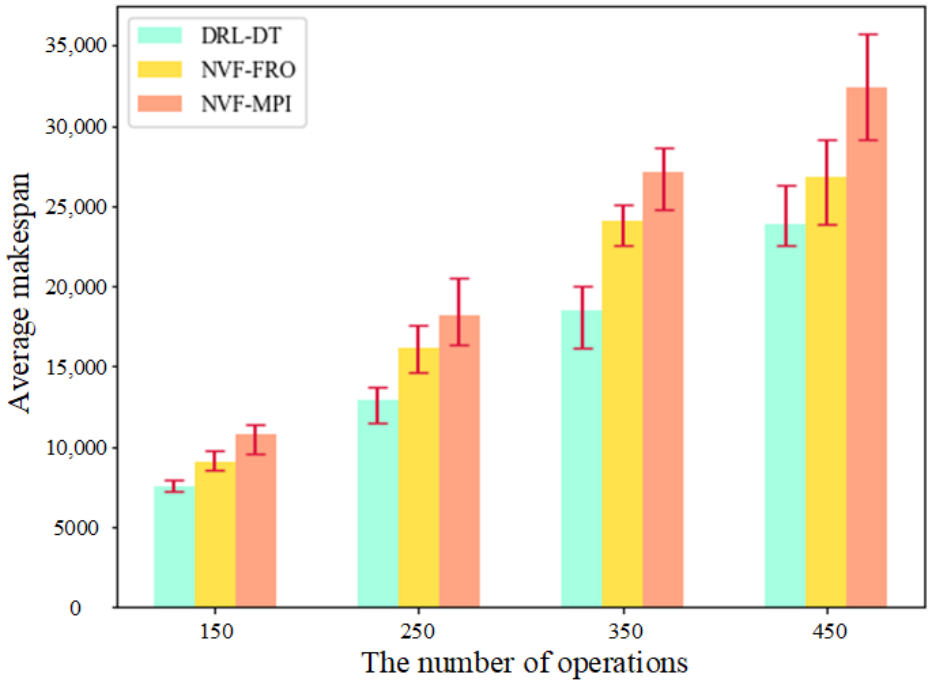


Figure 8. Performance comparison under different numbers of operations.

6. Conclusions

In this study, a DT framework for crane dynamic scheduling is proposed, and DRL methods are embedded in it to make real-time optimization decisions for uncertain scenarios such as dynamic job arrival and equipment failure. The experimental results show that, compared with the traditional heuristic scheduling rules, the scheduling mode of DRL-DT has higher production and transportation efficiency. Contributions from this work are summarized as follows:

1. An integrated application framework of crane dynamic scheduling by deep reinforcement learning and digital twins (DRL-DT) is designed.
2. The crane dynamic scheduling problem in the workshop is modelled as an MDP, and detailed definitions of state, action and reward are given.
3. The DDQN joint logic simulation environment method under DRL-DT is developed to realize the dynamic scheduling of crane transportation, and its effectiveness is proven by a case study.

On the one hand, the application of DRL based on a discrete event simulation environment can easily deal with the uncertainty of the workshop and make the scheduling decision better. On the other hand, the DT application based on real-time data integrates the virtual and real data synchronously, which makes the scheduling management process more transparent and efficient. Because of these two advantages, the research results in practice can both significantly improve the level of digital manufacturing in companies and genuinely increase productivity for them. However, this work also has certain limitations. First, only studying crane scheduling is not comprehensive enough for the overall production application of the factory. Second, the DDQN algorithm still has room for further improvement in efficiency. Therefore, in future research, we will try to apply the proposed framework to a wider range of workshop production and logistics areas, such as warehousing and distribution, as the DT framework can easily serve as a base to extend to these scenarios and integrate. At the same time, more efficient DRL methods will be explored for better scheduling performance and higher compatibility.

Author Contributions: Methodology, software, validation, formal analysis, writing, Z.X.; conceptualization, resources and review, D.C.; supervision, review, and editing, T.L. and M.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Industry and Information Technology of China for Cruise Program [No. 2018-473]; the National Key Research and Development Program of China [No. 2019YFB1704403].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Serrano-Ruiz, J.C.; Mula, J.; Poler, R. Smart manufacturing scheduling: A literature review. *J. Manuf. Syst.* **2021**, *61*, 265–287. [CrossRef]
2. Peterson, B.; Harjunkoski, I.; Hoda, S.; Hooker, J.N. Scheduling multiple factory cranes on a common track. *Comput. Oper. Res.* **2014**, *48*, 102–112. [CrossRef]
3. Tang, L.; Xie, X.; Liu, J. Scheduling of a single crane in batch annealing process. *Comput. Oper. Res.* **2009**, *36*, 2853–2865. [CrossRef]
4. Liu, Z.; Guo, S.; Wang, L. Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption. *J. Clean. Prod.* **2019**, *211*, 765–786. [CrossRef]
5. Zhou, B.; Liao, X. Decomposition-based 2-echelon multi-objective evolutionary algorithm with energy-efficient local search strategies for shop floor multi-crane scheduling problems. *Neural Comput. Appl.* **2020**, *32*, 10719–10739. [CrossRef]
6. Zhou, B.; Liao, X. Particle filter and Levy flight-based decomposed multi-objective evolution hybridized particle swarm for flexible job shop greening scheduling with crane transportation. *Appl. Soft Comput.* **2020**, *91*, 106217. [CrossRef]
7. Li, J.; Xu, A.; Zang, X. Simulation-based solution for a dynamic multi-crane-scheduling problem in a steelmaking shop. *Int. J. Prod. Res.* **2020**, *58*, 6970–6984. [CrossRef]
8. Du, Y.; Li, J.; Luo, C.; Meng, L. A hybrid estimation of distribution algorithm for distributed flexible job shop scheduling with crane transportations. *Swarm Evol. Comput.* **2021**, *62*, 100861. [CrossRef]
9. Qu, S.; Wang, J.; Govil, S.; Leckie, J.O. Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-skill Workforce and Multiple Machine Types: An Ontology-based, Multi-agent Reinforcement Learning Approach. *Procedia CIRP* **2016**, *57*, 55–60. [CrossRef]
10. Shahrabi, J.; Adibi, M.A.; Mahootchi, M. A reinforcement learning approach to parameter estimation in dynamic job shop scheduling. *Comput. Ind. Eng.* **2017**, *110*, 75–82. [CrossRef]
11. Wang, Y.-F. Adaptive job shop scheduling strategy based on weighted Q-learning algorithm. *J. Intell. Manuf.* **2020**, *31*, 417–432. [CrossRef]
12. Wang, H.; Sarker, B.R.; Li, J.; Li, J. Adaptive scheduling for assembly job shop with uncertain assembly times based on dual Q-learning. *Int. J. Prod. Res.* **2021**, *59*, 5867–5883. [CrossRef]
13. Lin, C.-C.; Deng, D.-J.; Chih, Y.-L.; Chiu, H.-T. Smart Manufacturing Scheduling With Edge Computing Using Multiclass Deep Q Network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4276–4284. [CrossRef]
14. Shi, D.; Fan, W.; Xiao, Y.; Lin, T.; Xing, C. Intelligent scheduling of discrete automated production line via deep reinforcement learning. *Int. J. Prod. Res.* **2020**, *58*, 3362–3380. [CrossRef]
15. Liu, C.-L.; Chang, C.-C.; Tseng, C.-J. Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. *IEEE Access* **2020**, *8*, 71752–71762. [CrossRef]
16. Luo, S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Appl. Soft Comput.* **2020**, *91*, 106208. [CrossRef]
17. Hu, L.; Liu, Z.; Hu, W.; Wang, Y.; Tan, J.; Wu, F. Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *J. Manuf. Syst.* **2020**, *55*, 1–14. [CrossRef]
18. Han, B.-A.; Yang, J.-J. Research on Adaptive Job Shop Scheduling Problems Based on Dueling Double DQN. *IEEE Access* **2020**, *8*, 186474–186495. [CrossRef]
19. Wang, L.; Hu, X.; Wang, Y.; Xu, S.; Ma, S.; Yang, K.; Liu, Z.; Wang, W. Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. *Comput. Netw.* **2021**, *190*, 107969. [CrossRef]
20. Yang, S.; Xu, Z.; Wang, J. Intelligent Decision-Making of Scheduling for Dynamic Permutation Flowshop via Deep Reinforcement Learning. *Sensors* **2021**, *21*, 1019. [CrossRef]
21. Fang, Y.; Peng, C.; Lou, P.; Zhou, Z.; Hu, J.; Yan, J. Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6425–6435. [CrossRef]

22. Zhou, G.; Zhang, C.; Li, Z.; Ding, K.; Wang, C. Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing. *Int. J. Prod. Res.* **2020**, *58*, 1034–1051. [CrossRef]
23. Wang, Y.; Wu, Z. Model construction of planning and scheduling system based on digital twin. *Int. J. Adv. Manuf. Technol.* **2020**, *109*, 2189–2203. [CrossRef]
24. Zhang, J.; Deng, T.; Jiang, H.; Chen, H.; Qin, S.; Ding, G. Bi-level dynamic scheduling architecture based on service unit digital twin agents. *J. Manuf. Syst.* **2021**, *60*, 59–79. [CrossRef]
25. Yan, J.; Liu, Z.; Zhang, C.; Zhang, T.; Zhang, Y.; Yang, C. Research on flexible job shop scheduling under finite transportation conditions for digital twin workshop. *Robot. Comput. -Integr. Manuf.* **2021**, *72*, 102198. [CrossRef]
26. Esposito, C.; Castiglione, A.; Palmieri, F.; Ficco, M.; Dobre, C.; Iordache, G.V.; Pop, F. Event-based sensor data exchange and fusion in the Internet of Things environments. *J. Parallel Distrib. Comput.* **2018**, *118*, 328–343. [CrossRef]
27. Hu, H.; Jia, X.; He, Q.; Fu, S.; Liu, K. Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0. *Comput. Ind. Eng.* **2020**, *149*, 106749. [CrossRef]
28. Zhang, Z.; Wang, W.; Zhong, S.; Hu, K. Flow Shop Scheduling with Reinforcement Learning. *Asia Pac. J. Oper. Res.* **2013**, *30*, 1350014. [CrossRef]