

## Article

# Intelligent Scheduling Method for Bulk Cargo Terminal Loading Process Based on Deep Reinforcement Learning

Changan Li <sup>1,2</sup>, Sirui Wu <sup>3</sup>, Zhan Li <sup>3,4,\*</sup> , Yuxiao Zhang <sup>3</sup>, Lijie Zhang <sup>1,\*</sup> and Luis Gomes <sup>5</sup> 

<sup>1</sup> Key Laboratory of Advanced Forging & Stamping Technology and Science of Ministry of Education of China, Yanshan University, Qinhuangdao 066004, China; changan.li.c@chnenergy.com.cn

<sup>2</sup> Chnenergy (Tianjin) Port Co., Ltd., Tianjin 300450, China

<sup>3</sup> Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China; wusr\_1234@163.com (S.W.); zyx04105@gmail.com (Y.Z.)

<sup>4</sup> Ningbo Institute of Intelligent Equipment Technology Co., Ltd., Ningbo 315201, China

<sup>5</sup> NOVA School of Sciences and Technology—Centre of Technology and Systems, NOVA University Lisbon, 2829-516 Monte de Caparica, Portugal; lugo@fct.unl.pt

\* Correspondence: zhanli@hit.edu.cn (Z.L.); ljzhang@ysu.edu.cn (L.Z.)

**Abstract:** Sea freight is one of the most important ways for the transportation and distribution of coal and other bulk cargo. This paper proposes a method for optimizing the scheduling efficiency of the bulk cargo loading process based on deep reinforcement learning. The process includes a large number of states and possible choices that need to be taken into account, which are currently performed by skillful scheduling engineers on site. In terms of modeling, we extracted important information based on actual working data of the terminal to form the state space of the model. The yard information and the demand information of the ship are also considered. The scheduling output of each convey path from the yard to the cabin is the action of the agent. To avoid conflicts of occupying one machine at same time, certain restrictions are placed on whether the action can be executed. Based on Double DQN, an improved deep reinforcement learning method is proposed with a fully connected network structure and selected action sets according to the value of the network and the occupancy status of environment. To make the network converge more quickly, an improved new epsilon-greedy exploration strategy is also proposed, which uses different exploration rates for completely random selection and feasible random selection of actions. After training, an improved scheduling result is obtained when the tasks arrive randomly and the yard state is random. An important contribution of this paper is to integrate the useful features of the working time of the bulk cargo terminal into a state set, divide the scheduling process into discrete actions, and then reduce the scheduling problem into simple inputs and outputs. Another major contribution of this article is the design of a reinforcement learning algorithm for the bulk cargo terminal scheduling problem, and the training efficiency of the proposed algorithm is improved, which provides a practical example for solving bulk cargo terminal scheduling problems using reinforcement learning.

**Keywords:** bulk cargo loading; MDP model; deep reinforcement learning; intelligent scheduling



**Citation:** Li, C.; Wu, S.; Li, Z.; Zhang, Y.; Zhang, L.; Gomes, L. Intelligent Scheduling Method for Bulk Cargo Terminal Loading Process Based on Deep Reinforcement Learning. *Electronics* **2022**, *11*, 1390. <https://doi.org/10.3390/electronics11091390>

Academic Editor: George A. Tsihrintzis

Received: 3 April 2022

Accepted: 24 April 2022

Published: 27 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The bulk cargo terminal works as the transit station for transportation; thus, the efficiency of its scheduling is very important. At present, its scheduling method is generally completed manually, which cannot meet the surge in throughput and the trend of using large-scale ships. Compared with container terminals, there are few studies on the scheduling problems of bulk cargo terminals, and the main research method is offline scheduling based on an operation model. Therefore, using up-to-date intelligent methods to scientifically dispatch the loading and unloading equipment of the terminal, further improving the efficiency of the terminal operation, reducing the cost of human resources, and improving the overall benefits is of great research significance.

At present, in most cases, the production scheduling problem of bulk cargo terminals is still considered to be a constrained optimization problem, and the deep reinforcement learning method for the production scheduling problem of bulk cargo ships still needs to be studied. There are similar studies on the intelligent optimization of bulk cargo terminal control systems [1], as well as bulk cargo terminal berth scheduling systems, based on machine learning [2]. These studies were mainly aimed at the berth scheduling problem. In these studies, aiming at the problems of lagging automation and low informatization of the bulk cargo terminal, the control system of the bulk cargo terminal was optimized, and the control efficiency and operability of the control system were improved. In addition, there are currently some reinforcement learning studies on container terminals. For example, Fateme Fotuhi proposed a yard crane scheduling model based on reinforcement learning [3], which was used to solve the scheduling optimization problem of yard cranes. However, compared with container terminal scheduling, the problem is that the scheduling model of bulk cargo terminals often has the characteristics of a large scale and complex situation. Research on using reinforcement learning to address these types of problems is still very preliminary.

Wharf scheduling problems can be summarized as a berth allocation model (BAP). The related research on this type of model has made some progress. Research on the berth allocation model is generally divided into two categories; one focuses on the research of the model itself, while the other focuses on the study of model solving methods. The research on the model itself mainly focuses on abstracting and establishing mathematical models from practical problems, improving the established universal models, and applying them to specific problems. For different practical problems, berth allocation models include static and dynamic, as well as continuous and discrete. The optimization objectives of the models are different, and there are different constraints in the models.

In 2007, Kobe University conducted a study on the berth allocation problem considering service time and delay time objectives [4]. The research took time as the optimization objective, which is also the most important indicator for most scheduling problems. In 2015, the authors of [5] proposed an optimization model for container terminal berth scheduling. The main focus of the study was to optimize the constraints in the model and reduce the constraint variables. Subsequent studies also include terminal loading studies with optimization objectives such as energy consumption and emissions [6]. In 2016, the authors of [7] proposed an interference model for berth allocation and shore crane allocation problems based on behavior perception. The study aimed to solve the problem that the actual situation does not match the plan due to accidental factors in the scheduling process. For the bulk cargo shipping problem, there are also many studies at present, mainly for the time optimization of bulk cargo tank allocation and fleet scheduling problems [8,9].

For the solution method of the terminal scheduling model, traditional planning methods, heuristic methods, and intelligent methods are generally used. The traditional planning method is generally to give the corresponding exact solution or approximate solution through numerical calculation for the multivariable mathematical model and a set optimization objective. In 2011, Barros et al. proposed a tidal bulk berth configuration model with stock constraints [10], which was solved by the integer linear programming method. In 2017, Menezes et al. studied the routing problem of port import and export orders for bulk cargo terminals in Brazil, taking into account the timeliness of cargo storage and transportation, and aiming to reduce operating costs [11]. Heuristic methods are generally aimed at larger-scale problems. Common solution methods include the genetic algorithm and particle swarm algorithm. In 2016, a hybrid particle swarm algorithm [12] was proposed that simultaneously solved the dynamic discrete berth allocation problem and dynamic shore crane allocation. The study showed that this method outperformed the basic genetic algorithm and the hybrid genetic algorithm. In addition, in order to solve the problem of model uncertainty, in addition to using meta-heuristic algorithms, some studies also used a fuzzy optimization model [13,14] for problems with a high degree of uncertainty.

With the continuous development of artificial intelligence, it has become a trend to use intelligent methods to solve production scheduling problems. In some studies, machine learning methods were applied to traditional search algorithms to improve the effect of the algorithm [15–17]. At present, there are also some studies that used deep reinforcement learning to solve production scheduling problems similar to bulk cargo terminal scheduling [18]. The AlphaDow project announced by Dow Chemical in 2020, based on the concept of intelligent manufacturing, uses deep reinforcement learning methods to carry out research on global factory production scheduling optimization problems.

The development of research on production scheduling, terminal production scheduling, and other issues using deep reinforcement learning methods depends on the development of deep reinforcement learning algorithms, on the one hand, and is affected by the actual production scheduling mode, on the other hand. Reinforcement learning algorithms can be improved according to actual task needs, and the development of algorithms will eventually promote the progress of industrial production models. At present, the research on deep reinforcement learning algorithms is relatively rich and in-depth [19–21]. Reinforcement learning algorithms are generally divided into model-free learning and model-based learning [22]. The former method learns the operation strategy of the agent in the environment directly according to the continuous acquisition of experience data, while the latter method learns the model according to the experience data, and then formulates the strategy according to the learned model. In addition, reinforcement learning methods can be divided into online learning and offline learning. Online learning refers to taking actions according to the currently learned strategies in the process of learning and exploration, while offline learning refers to adopting new strategies to explore during the learning process. The advantage of online learning is that the optimization is simple, but the disadvantage is that this method can easily fall into a local optimum. Offline learning is easier to jump out of the local optimum, but the choice of a “new strategy” needs to be reconsidered.

According to the different action selection methods in the optimization process, reinforcement learning algorithms can also be classified into value-based reinforcement learning methods and policy gradient-based reinforcement learning methods [22]. At present, the widely used value-based reinforcement learning method is the DQN method, which has been continuously improved, including DDQN, D3QN, and distributed DQN [23,24]. The recently proposed Rainbow algorithm [25] is a representative of this kind of algorithm with good effect. The Rainbow algorithm combines the advantages of different algorithm frameworks and is designed to take into account both algorithm performance and universality. For the policy gradient algorithm, the most commonly used algorithm for continuous action systems is the DDPG algorithm [26]. In addition, the actor–critic algorithm combines the advantages of value-based algorithms and policy gradient algorithms, which can not only handle continuous and discrete problems, but also perform single-step updates to improve learning efficiency. In the algorithm, the actor network is responsible for performing actions during the training process, and the critic network is responsible for scoring. The most commonly used method is the A3C algorithm [27], the essence of which is to put the actor–critic network into multiple threads for synchronous training. Because of this, the algorithm has the characteristics of fast speed and high efficiency. At present, the research on reinforcement learning is deepening, and the application background is becoming more and more complex. For example, the recently proposed multi-agent reinforcement learning method [28] is mainly aimed at the use of multiple decision-making units. This method can be applied directly to multi-agent problems, as well as to problems where a single decision-making agent can be split into multiple unrelated agents.

The scheduling problems of bulk cargo loading studied in this paper often have strong dynamics and uncertainties, as well as a large scale. Traditional optimization methods, heuristic methods, fuzzy optimization methods, and data-driven machine learning methods struggle to solve large-scale random and dynamic problems. Artificial intelligence methods have the advantage of solving the above problems. The continuous improvement of deep

learning allows better solving large-scale complex and uncertain problems. However, this study is based on the actual production scheduling process. Unlike “playing games”, errors in the actual production process will bring serious consequences; thus, some special situations need to be considered in advance. Therefore, to apply the method to the specific problem, the training part of the reinforcement learning algorithm needs to be modified in the research. Therefore, this paper proposes a research method based on a modified double deep Q learning algorithm to solve the problem of loading and scheduling of bulk cargo terminals, so as to achieve the purpose of solving the above problems at the same time.

The main contribution of this paper is the proposal of a scheduling method for the bulk cargo terminal loading process based on deep reinforcement learning. By analyzing the historical data of CHNENERGY (Tianjin) Port Co., the main processes involved in the production scheduling of the coal terminal are summarized and extracted, a reinforcement learning model that satisfies the Markov property is established, and its state space and action space are determined. After improving the double DQN algorithm for learning and training, a fast scheduling method is obtained when the tasks arrive randomly and the yard state is random.

The innovations and contributions of this paper are as follows:

- (1) An appropriate reward function is designed to satisfy the condition that the long-term reward and the optimization objective are completely equivalent, and to ensure the possibility of theoretically optimal training results.
- (2) This paper improves the  $\epsilon$ -greedy exploration strategy. During the training process, the agent sometimes performs random actions without considering whether the action is legal or not, and sometimes only selects one action from the legal actions to perform. This method is essentially a semi-masking strategy for illegal actions. This method combines the shielding and punishment of illegal actions, so that the training process can meet the convergence conditions and speed up the training progress. The improvement is mainly aimed at dealing with the models with illegal actions.
- (3) Special circumstances and accidental events in the production scheduling process are fully considered in the model, which makes the model more realistic.

The remainder of this paper is organized as follows: Section 2 establishes a Markov model for loading and scheduling of coal terminals, as well as designs a reward function according to the optimization objective in practical problems; Section 3 details the method of applying the deep Q network (DQN) to the scheduling optimization problem of coal terminal production scheduling; Section 4 focuses on improvements during reinforcement learning training; Section 5 designs simulation experiments to verify the model according to the actual production scheduling situation, as well as analyzes and summarizes the results.

## 2. Analysis and Modeling of Ship Loading and Production Scheduling Based on Condition Monitoring Data

### 2.1. Analysis of Port Operations and Storage Yard Environment

The production and loading process of the coal terminal means that the coal in the yard is transported to the ship through the reclaimer, belt conveyor, ship loader, and other large-scale mechanical equipment in the port. The arrangement, type, and quantity of coal in the yard, as well as the coal order demand of the ships to be loaded, are directly related to the choice of the production scheduling plan, and the operation of the equipment that transfers the coal from the coal pile to the ship needs to be considered. These factors affect coal delivery time.

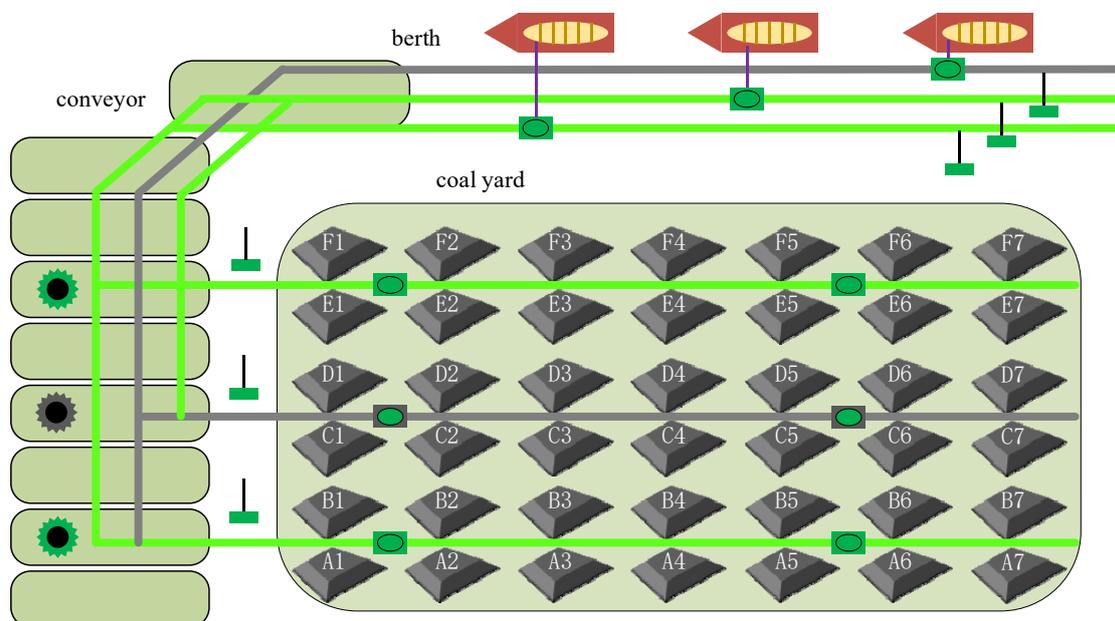
As shown in Table 1, various types of coal are transported through the port. Taking a ship loading process as an example, suppose a ship needs 10,000 tons of coal of type A and 20,000 tons of type B. It is necessary to distinguish the types of coal from the coal pile in the yard and extract the coal pile stock. At the same time, it is necessary to consider which transport machine will be used to transport the coal to the ship, the operation status

and availability of the machine, and when the occupied machine will be released from occupancy and other information.

**Table 1.** Statistics of coal output of the port within 3 months.

Type of Coal	Flag Number	Production (Tons)
Gao Shi 5000	1	324,818
Shen Hun 48	2	2,264,852
Shen Hun 52	3	3,096,472
Shen Hun 55	4	6,905,478
Shen You 1	5	215,788
Shen You 2	6	87,134
Wai Gou 55	7	2,791,898
Wai Shi 5000	8	2,344,022
Wai Shi 5500	9	303,024
Clearing coal	10	39,188
Shen Hun 45	11	610,350

The actual coal terminal yard distribution and coal transfer process are shown in Figure 1. There are six rows of coal piles, from A to F, in the coal yard, each row has seven coal piles, for a total of 42 coal piles are arranged in lattice form, which are A1 to F7 in the figure. There are three rows of reclaiming lines in the terminal, where each reclaiming line is located in the middle of the two rows of coal piles, and the reclaimers on the reclaiming line is responsible for grabbing the coal from the coal piles to the conveyor belt. The three ship loaders are responsible for transporting the coal from the storage yard. The coal near the berth is transferred into the hold of the moored vessel. The three berths are arranged in sequence, and each berth docks a corresponding ship according to the plan. Each ship to be loaded with coal has several cabins, and each cabin corresponds to the corresponding coal order demand.



**Figure 1.** Port coal transfer process.

*2.2. Establishment of the Model for Loading and Scheduling under the Port Operating Environment*

The plan for coal wharf scheduling refers to the allocation plan that uses different transportation equipment to transport coal from different coal piles to matching cabins according to the current working conditions of the yard, mechanical equipment, ships, and

the entire wharf system. The simplest scheme is a single-demand allocation scheme, i.e., when only a single cabin of a ship needs a certain type of coal, a single machine is allocated to transport a certain amount of coal from a certain coal pile to a designated cabin. The information contained in the scheme is the selection of transport machines (reclaimers, ship loaders, conveyor belts), the selection of coal piles, the setting of coal loadings, and the positions of loaded ships and cabins. A single-demand scheme is the smallest unit of all scheduling schemes, and all scheduling schemes can be expressed as a combination of single-demand schemes in time.

Due to the existence of limited transportation machines, limited coal transportation routes, limited coal volume in coal piles, and limited coal type requirements and coal type matching problems in coal piles, the production scheduling scheme cannot traverse all spaces; hence, choosing the appropriate production scheduling scheme is necessary. The constraints that need to be considered in coal terminal production scheduling include reclaim line constraints, ship loader constraints, coal pile type matching constraints, coal pile quantity sufficiency constraints, and ship loading and drainage constraints.

In the general production scheduling model, the total man-hours is a suitable indicator to effectively measure the production scheduling effect. The total working hours refers to the sum of the time when the machine is running. When the system runs uninterrupted, the time period between the time the system starts working and the ending time is the total working hours. If the system has intermittent or low-load operation, the total man-hours should exclude the system downtime.

To study the problem of shipment scheduling and scheduling, the reinforcement learning model of the scheduling problem must be established first. The keys to the reinforcement learning model are the state space, action space, reward function, and state transition method. The state space includes the model elements and some constraints that must be reflected as states. The action space gives the action, the action is the decomposition of the plan, and a series of actions constitute the plan. In reinforcement learning models, the system continuously learns interactively with the environment to form policies for taking actions on the basis of states. The reward function refers to the feedback obtained after taking an action, and the level of reward is positively correlated with the quality of the action.

The Markov decision process (MDP) is a discrete-time stochastic process used to model decisions. It can be used to solve problems that learn interactively from the environment and achieve some goals. In MDP, it is the agent that learns, and the other things it interacts with are called the environment. The mathematical representation of the Markov decision process is a quintuple  $(S, A, P, R, \gamma)$ , where  $S$  is the set of states (state space),  $A$  is the set of actions (action space),  $P$  is the state transition probability,  $R$  is the set of instantaneous rewards, and  $\gamma$  is the discount factor to reflect the loss coefficient of each downward propagation step of the instantaneous reward. As shown in Figure 2, the execution process of Markov is as follows: the initial state of the agent is  $s_0$ . First, an action  $a_0$  needs to be selected from the action set to execute. After execution, the action  $a_0$  and the instantaneous reward  $r_0$  are obtained, and the agent randomly transitions to the next state  $s_1$  according to  $P$  probability.

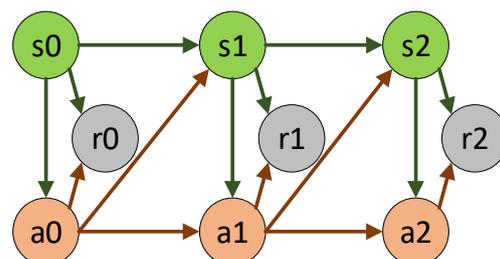


Figure 2. Markov chain.

The goal of reinforcement learning training is to learn a policy  $\pi$ , and the agent takes  $\gamma$  a series of actions  $a$  in different states to maximize the cumulative reward. After a given policy  $\pi$ , the value of state  $s_0$  under policy  $\pi$  is calculated by calculating  $V$ . When the expected return value obtained by the strategy is greater than or equal to other strategies, the strategy is optimal.

$$V^\pi(s_0) = E^\pi \left\{ \sum_{i=0}^n \gamma^i R(s_i, a_i) \right\}. \tag{1}$$

The following simplifications and assumptions about the actual shipping production process are made:

- (1) The wharf has three berths, and each ship has a maximum of 10 cabins;
- (2) The reclaiming speed of the reclaimer is a constant value of 3600 tons/h, and the moving speed is a constant value of 2 m/s;
- (3) The moving speed of the ship loader is a constant value of 1 m/s;
- (4) The size of the coal piles in the field is the same, and the distance between the centers of adjacent coal piles is 40 m.

In the decision making and scheduling of tasks, the bulk cargo terminal is related not only to the demand of arriving ships, but also to the current occupancy of the yard. Therefore, the description of the state space involves two parts: task characteristics and yard characteristics. Mission characteristics include the ship’s remaining time to drain, as well as the type and amount of coal required for each of the three ships’ cabins. On-site storage characteristics include the type and quantity of 42 coal piles, the occupancy and failure of reclaimers, the occupancy and failure of ship loaders, the coal piles taken by three reclaimers, and the occupancy time corresponding to the three reclaiming lines, the reclaimer corresponding to the three ship loaders, and the cabin corresponding to the ship loader. The state space designed in this way satisfies the Markov property. Since it contains all the field storage information and working condition information, the state at the next moment is only related to the current state and the action taken.

All model elements should be included in the state space, and the constraints that can affect the decision making should be considered. As shown in Table 2, the state space model of the optimization problem of loading and arranging ships is further explained below.

**Table 2.** Shipping scheduling state space.

State Space	Data Structure	Dimension	Types of	Meaning
$T_{wait}$	array	3	Int _	Displacement time for 3 boats
$M_{size}$	array	42	Int _	42 coal stockpiles
$M_{kind}$	array	42	Int _	42 types of coal piles
$X_{size}$	array	30	Int _	cabin demand
$X_{kind}$	array	30	Int _	Types of cabin requirements
$Q$	array	3	Bool _	Occupancy of the reclaiming line
$Z$	array	3	Bool _	Ship loader occupancy
$Q_i$	array	3	Int _	Coal piles corresponding to 3 reclaiming lines
$C_i$	array	3	Int _	3 ship loaders
$T_{occupy}^i$	array	3	Int _	The remaining occupied time of 3 reclaiming lines
$Q_z$	array	3	Int _	Corresponding relationship between reclaimer and ship loader

The ship information in the state space includes the remaining drain time of three ships, the type of coal required for 30 ships, and the expected amount of coal to be loaded in 30 ships.

The yard information in the state space includes 42 types of coal piles and 42 remaining quantities of coal piles.

The machine equipment status includes whether the reclaimer is occupied or faulty (0 idle, 1 occupied, 2 faulty) and whether the ship loader is occupied or faulty (0 idle, 1 occupied, 2 faulty), As well as the coal pile taken by the machine, the ship loader corresponding to the three reclaimers, the 30 cabins corresponding to the three ship loaders, and the remaining time of the current task of the three reclaim lines.

When choosing the dispatch action, the choices made include 42 coal piles corresponding to three reclaimers, three ship loaders, and up to 30 cabins. Therefore, when three berths have ships docked, the action options are the most, as there are  $42 \times 3 \times 30 = 3780$  choices in total. However, due to the limitation of conditions, most of the actions are unavailable. The available actions are also called “legal” or “correct” actions.

An intuitive way to decompose scenarios into actions is to directly equate a single requirement scenario with an action. It can be seen that the action space includes the position of the coal pile, the reclaiming line, the amount of coal taken, the type of coal taken, the loading, the berth, and the cabin.

In the actual situation, the coal type of the coal pile location is unique and can be displayed in the status. The amount of coal taken is repeated information. For the coal loading problem, it is assumed that as much coal as possible is taken each time, and the amount of coal taken is the minimum value of the coal stockpile and the demand for the ship’s cabin. At the same time, in the yard distribution, the position of the coal pile uniquely determines the reclaiming line (the reclaiming line corresponding to the coal pile is unique); thus, the reclaiming line is duplicate information.

To sum up, the information of action space is simplified as the coal pile position (total = 42), ship loader (total = 3), berth (total = 3), cabin (total = 10), which can be shown as Figure 3. All the information mentioned above is encoded to form an action space, and the total number of actions is  $42 \times 3 \times 3 \times 10 = 3780$ .

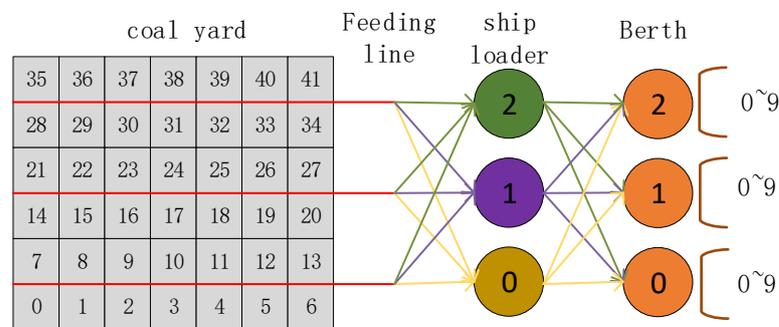


Figure 3. MDP modeling of port scheduling.

For this model, the total running time of the system is taken as the optimization target. When each action is executed, the total execution time of the system increases. Its negative value is taken, and it is considered that the time occupied by the action with a large amount of coal will be longer; therefore, the reward function of the single-step action is designed as  $r = k \times m - t_{increase}$  (where  $k$  is a proportional coefficient,  $m$  is the coal loading amount of the step action), and the long-term return at this time is  $R = k \times -T$ , where  $M$  is the total coal transportation amount, and  $T$  is the system total run time. It can be seen that, when the total demand is the same, a shorter running time leads to greater long-term return.

The setting of the reward function should reflect the measure of the pros and cons of the action. In the case where total man-hours is used as the measurement method for the production scheduling effect of the plan, the reward function can simply be set as the added value of production scheduling man-hours. The increase value is negatively related to the effect; hence, the increase value is negative, and a constant bias is added. When the

quantity of coal demanded by ships varies greatly, a function of quality can also be added. The form of the reward function is shown in Equation (2).

$$r = -t_{increase} + C. \quad (2)$$

When coal is transported from a coal pile to the cabin, the storage information and demand information changes, and the operation information of machinery and equipment also changes, as follows:

- (1) The corresponding coal stockpile is reduced;
- (2) The amount of coal to be loaded in the hold is reduced, and the reduction is the same as the reduction in stock;
- (3) For the corresponding reclaiming line, the ship loader is set to the occupied state;
- (4) The time of the reclaiming line is increased as the action consumption time;
- (5) The equivalent displacement time of the corresponding vessel changes;
- (6) The idle ship loaders are moved simultaneously to satisfy the sequence.

The legitimacy of the action (whether it satisfies the constraints) needs to be judged by the program. The program judgment conditions are as follows:

- (1) The coal pile corresponding to the action is in stock;
- (2) The cabin corresponding to the action is in demand;
- (3) The coal type of the coal pile corresponding to the action matches the type of cabin demand;
- (4) The reclaiming line corresponding to the action is idle;
- (5) The ship loader corresponding to the action is idle;
- (6) The order of the ship loader remains unchanged after the action is executed.

For illegal actions, punitive measures should be added, and the reward should be set to a negative value.

### 3. Optimization of Shipment Scheduling Based on Deep Reinforcement Learning Algorithm

In this study, we used the value-based deep Q network (DQN) method for training and testing. The value-based reinforcement learning algorithm, whose strategy is defined as selecting the action that maximizes the state value, makes the result approximate a state value function. Mnih et al. (2013) [23] proposed policy-based algorithms to directly learn policies that maximize the reward function by increasing the probability of actions that yield higher rewards.

The deep Q network (DQN) is a deep reinforcement learning algorithm that applies neural networks to reinforcement Q learning. Scholars such as Mnih proposed the concept of DQN, which can be regarded as a process of approximating the neural network Q and the weight function. By directly taking raw data (state features) as input and the Q function value of each state–action pair as output, DQN can handle complex decision-making processes with large and continuous state spaces.

The optimal scheduling reinforcement learning model for port production scheduling involves many state variables, and the state space has extremely high modeling complexity. The DQN algorithm is used to approximate the high-dimensional state space and action space, which can greatly simplify the scheduling.

The DQN algorithm takes the minimum value of the mean square error of the current value function and the target value function as the parameter of the network update of the DQN algorithm, as shown in Equation (3).

$$L(\theta) = E((r + \gamma \max Q(s', a', \theta') - Q(s, a, \theta))^2). \quad (3)$$

The loss function can be differentiated to get the gradient formula in Equation (4).

$$\frac{\partial L(\theta)}{\partial \theta} = E((r + \gamma \max Q(s', a', \theta') - Q(s, a, \theta)) \frac{\partial Q(s, a, \theta)}{\partial \theta}). \quad (4)$$

In the early stage of training, the agent conducts more random explorations and packs the state, action, income, and the state of the next moment of the agent into the experience replay pool until the experience replay pool is full before learning. Compared with traditional Q learning, which directly discards samples after learning, the experience replay pool reduces the time it takes for the agent to interact with the environment to collect samples.

During the training process, the experience playback pool includes the process of collecting and sampling samples. The collection process is sorted according to the timeseries of agent interaction. If the storage is full, the later samples overwrite the first stored samples; during the sampling process, the experience playback pool randomly samples a batch of samples for training and learning, which can reduce the fluctuation of a single sample sequence, while a sample can also be used for multiple training iterations, which improves the utilization rate.

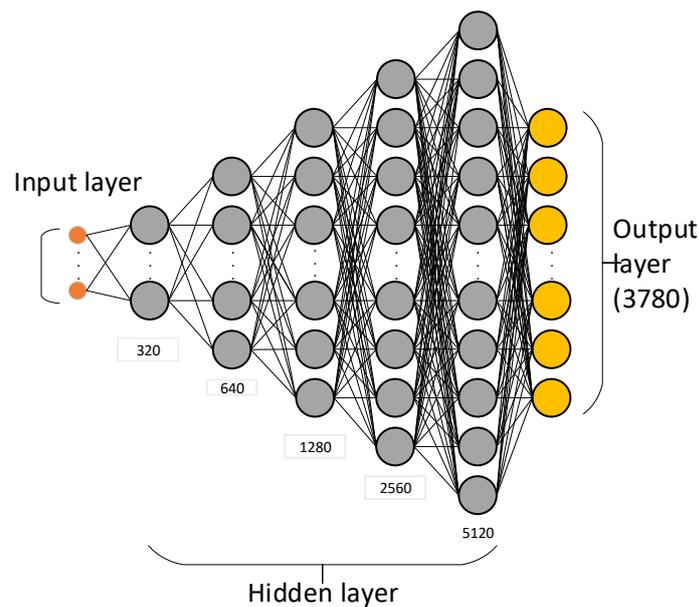
Due to the instability of the data itself, there will be some fluctuations in the process of training iterations, and these fluctuations may have an impact on the next iterative calculation. Therefore, two identical network structures are used during training, and the training process is as follows:

- (1) Two networks are initialized with the same parameters;
- (2) The network interacting with the environment in the training is recorded as the behavior network, and the interaction samples are obtained;
- (3) The target value is calculated through the target network, and it is compared with the estimated value calculated by the behavior network, before updating the network parameters;
- (4) After completing the specified number of iterations, the parameters of the target network are copied to the behavior network.

The double DQN algorithm in the deep reinforcement learning algorithm was used for training. First, the environment, including field storage information, occupancy information, and ship requirements, was initialized, and epsilon-greedy strategies were used to select actions according to the current state. Since most actions are in the illegal state when the action is selected, the return of the illegal action is 0, and the state of the system is not changed. Therefore, the updating method of the Q network is as follows:

$$Q(s, a) = \begin{cases} r(s, a) + \gamma * \max Q(s', a') & \text{action is legal} \\ \gamma * \max Q(s', a') & \text{action is illegal} \end{cases} \quad (5)$$

This design method satisfies that the Q value is gradually reduced to a non-maximum value when the action is illegal. At the beginning, a larger exploration rate was set, and, as the number of training iterations increased, it decayed according to the rate of  $\varepsilon = (1 - \text{decay}) \times \varepsilon$ , where  $\varepsilon = 1$ ,  $\text{decay} = 0.001$ , and the minimum value of  $\varepsilon$  is 0.02. The fully connected network used was a six-layer fully connected neural network. The input dimension of the network was the number of states, and the output was the Q value of each action, as shown in Figure 4.



**Figure 4.** The design using a fully connected neural network diagram.

#### 4. Optimization of Shipment Scheduling Based on Improved Deep Reinforcement Learning Algorithm

Traditional reinforcement learning algorithms usually use tables to store value functions; thus, they are also called table reinforcement learning. They require that the state space of the model should not be too large, while continuous states cannot be evaluated. Furthermore, it is necessary to use function approximation. The DQN algorithm is a reinforcement learning method that combines Q-learning and deep neural networks [22]. Compared with Q-learning, DQN uses a neural network to approximate the value function, establishes an experience playback pool to train the process of reinforcement learning, sets an independent target network, and handles deviations in the temporal difference process.

In practical applications, it has been found that the DQN algorithm has the problem of over-estimation, i.e., the estimated value function is larger than the real value function, which is caused by taking the maximum Q value in the Q learning algorithm, as shown in Equation (6).

$$y_i = R_j + \gamma \max_{a'} Q(s'_j, a'_j, \omega). \quad (6)$$

Using the DQN algorithm, the problem of overestimation is eliminated by decoupling the selection and calculation of the target Q value action [23]. It no longer looks for the largest Q value in each action in the target net, but starts with the behavior net's Q network to select the action with the largest value, before using it to calculate the Q value in the target network, as shown in Equation (7).

$$y_i = \gamma Q'(s'_j, \arg \max_{a'} Q(s'_j, a'_j, \omega), \omega'). \quad (7)$$

In the standard model, the rewards of illegal actions are set to 0; therefore, the training process is not designed to make a wrong decision. However, in a real system, it is inevitable that there will be program errors. For example, some processes generate failures, but the corresponding failure information is not obtained in the program. This is where the system needs to be able to analyze itself and avoid bad decisions. The neural network trained for the DQN problem is the reference for decision making. The output of the network reflects the value of the action. Therefore, we need to reflect the negative value of illegal actions in the training process. The most straightforward way is to give a penalized reward function (e.g., constant  $-100$ ) for wrong actions.

At the same time, the influence of illegal movements should be reflected in the training process. Illegal actions cannot be completely blocked. At this time, the  $\varepsilon$ -greedy strategy used in the training process needs to be improved. Setting different exploration rates to perform completely random selection and feasible random actions enables the network to converge quickly while more accurately reflecting the value of illegal actions.

The specific scheme is that, during training, the adopted strategy is expressed as selecting random legal actions, completely random actions, and legal action according to the probability with the largest  $Q$  value.

$$action = \begin{cases} \text{completely random} : P = \varepsilon_1 \\ \text{correctly random} : P = \varepsilon_2 \\ \text{greedy} : P = 1 - \varepsilon_1 - \varepsilon_2 \end{cases} . \quad (8)$$

### 5. Algorithm Testing and Analysis

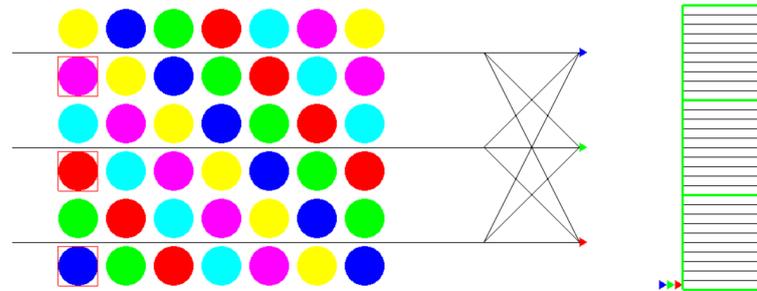
Using the port scheduling model, the reinforcement learning scheduling algorithm and its improved algorithm were verified. The experimental environment was as follows: CPU: Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz 2.10 GHz (two processors); GPU: NVIDIA GeForce RTX 2080Ti; memory: 32 GB. By setting the yard status information of the model and the demand information of arriving ships, the validity of the model under different actual conditions was verified.

During the training process, a fixed order sequence of 35 ships was used, each holding 1000 tons of coal. The order sequence was a single demand order, which could cover most of the order demands through its combination, with a certain representativeness and generalization effect. The six types of coal commonly used in the orders obtained by statistics were replaced by the numbers 0–5. The specific orders are shown in Table 3. On the basis of the order sequence, the reinforcement learning scheduling algorithm was trained, and the trained network was further verified.

**Table 3.** Order number for training.

Order Number	Hatches 0–9
1	2134211333
2	5222454203
3	3530043540
4	3400324050
5	2513541432
6	1111355354
7	4023541402
8	3150110402
9	3550512501
10	2052120005
11	0505124454
12	4121505345
13	0024220311
14	3523435110
15	3300554032
16	2335124515
17	0521225515
18	0254311420
19	1123253321
20	5100233230
21	5553354231
22	3244220324
23	4153115551
24	0241541540
25	0352255314
...	...

According to the modeling process of port production scheduling, a visual simulation environment was built as shown in Figure 5. In order to reflect the optimization of the ship loader, the information of the yard was set ideally, the types of coal piles were arranged according to the rules, and the number of coal piles was constantly replenished. In this case, the ship loader could be displayed. The initial number of the yard was fixed, and the current episode ended when the remaining coal in the yard was insufficient.



**Figure 5.** Visualization of port scheduling model.

The simulation results are given as Figures 6–9. It can be seen that, during the training process, the reward value gradually increased, and the reward value was positively correlated with the effect of labor scheduling, indicating that the effect of labor scheduling was gradually improved during the training process. The reward function mainly reflects the index of production scheduling time in production scheduling. In the simulation experiment, an increase in reward value represents the shortening of production scheduling time, i.e., in the case of the same task volume (including the number of orders, the total coal loading in the order), the scheduling task can be completed faster. For general reinforcement learning DQN problems, the goal is to optimize performance indicators so that the main agent is more adaptable to the environment. Therefore, other factors were added to the reward function design, e.g., the average degree of occupancy of the machine. Then, in the decision-making scheme after training, this can also reflect the feature that the allocation of machines is more even.

In the validation phase, this study compared three different scheduling strategies: (1) the artificially selected production scheduling strategy, which simulates the artificial plan, which is shown as Figure 7; (2) the random scheduling strategy, which simulates the machine automatically selecting the corresponding process to complete the task (this strategy is equivalent to the initial strategy before the neural network is trained), which is shown as Figure 8; (3) the post-training policy, which is the policy obtained after training the model with the methods used in the study, which is shown as Figure 9. In this study, the terminal loading system used three strategies to complete a set of randomly generated task order sequences, obtaining three sets of data. The validity of the training results and the superiority of the intelligent system over the artificial system were verified by comparing the three sets of data.

As shown in Figure 9, the production scheduling strategy maintained a good working efficiency and had an excellent production scheduling effect. The optimization of the production scheduling results can also be seen from the production scheduling Gantt chart and the data comparison. At the same time, the production scheduling time was shortened, and the production scheduling process was more closely connected, verifying the effectiveness of the reinforcement learning method. As shown in Table 4, compared to randomly selecting actions, the total man-hours of the trained strategy were shortened by 23.1%.

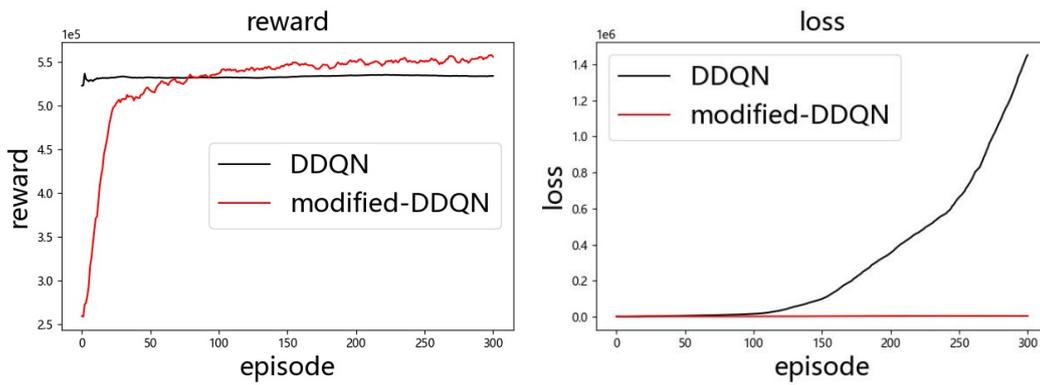


Figure 6. Production scheduling simulation reward function curve and simulation loss function curve.

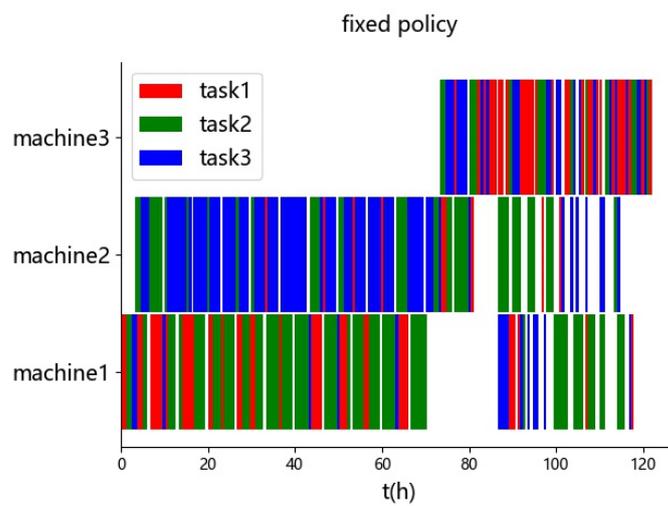


Figure 7. Gantt chart of fixed strategy scheduling method.

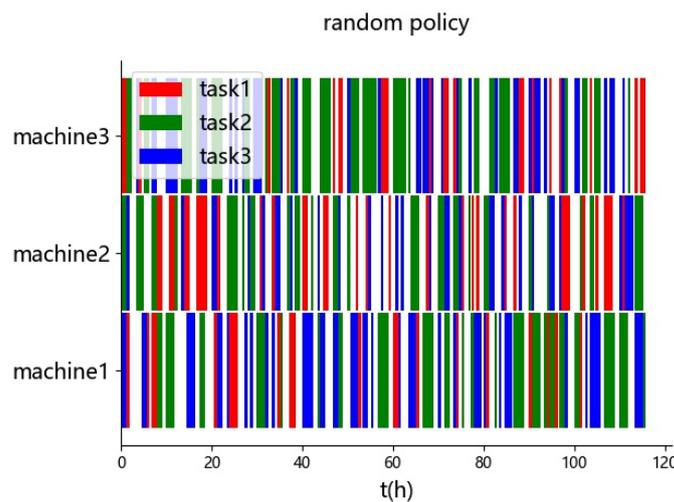
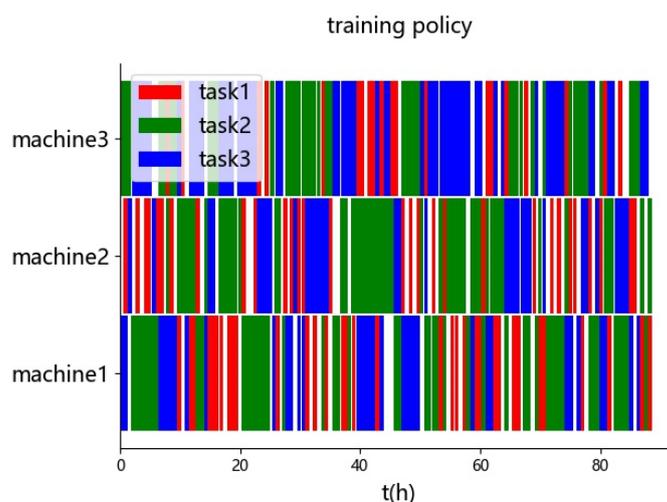


Figure 8. Gantt chart of random strategy scheduling method.



**Figure 9.** Gantt chart of training end strategy scheduling method.

**Table 4.** Comparison of scheduling results of loading and platooning.

Comparative Results	Production Scheduling Plan	Order Completion	Total Working Hours
Fixed scheduling strategy	Ship loader and berth number correspond	35 orders were all completed, with a total coal loading of 350,000 tons	121.3 h
Random scheduling strategy	Legal scheduling actions are randomly selected	35 orders were all completed, with a total coal loading of 350,000 tons	115.2 h
Strategies for scheduling labor after training	The legal action with the largest network output is chosen	35 orders were all completed, with a total coal loading of 350,000 tons	88.6 h

In reinforcement learning, the network structure and related parameters (such as learning rate) affect the learning effect. The network structure was introduced in the previous section. The main parameters in the simulation experiment were as follows (in the DDQN algorithm,  $\epsilon_0$  is the exploration rate; in the modified DDQN algorithm,  $\epsilon_0$  is the sum of  $\epsilon_1$  and  $\epsilon_2$ ): learning rate =  $5 \times 10^{-3}$ ; Batch size = 32;  $\epsilon_0 = 0.02$ ;  $\epsilon_1 = \epsilon_0 \times 0.6$ ;  $\epsilon_2 = \epsilon_0 \times 0.4$ .

## 6. Discussion and Remarks

The following points need to be noted:

- (1) A completely random policy means that the agent chooses an action randomly, whether it is legal or not. A correctly random policy means that the agent randomly chooses an action from all the correct actions (legal actions). A greedy policy means that the agent chooses the action with the highest value from all the correct actions.
- (2) The reward curve graph of the improved DDQN method shows that the reward value was lower in the early stage of training. This is because the improved DDQN method does not completely shield the wrong actions in the early stage of training, resulting in more penalties for the agent during the training process.
- (3) The reward curve graph of original DDQN method shows that the reward stopped rising and there was a bottleneck. This is because it got stuck in a local optimum during training.
- (4) The loss curve graph shows that the DDQN method before improvement diverged exponentially in the training process, which brought the loss to a very large value (magnitude of  $1 \times 10^6$ ). The loss of the improved DDQN method converged to a

relatively small value around 4000, which was very close to the horizontal axis when plotting it together with the diverging loss curve.

- (5) In the three Gantt charts, it can be seen that the last Gantt chart representing the final training effect was the most compact. This means that working machines tend to run continuously. This indirectly reflects the improvement of production scheduling efficiency. In addition, the total length of the Gantt chart represents the total scheduling time, which directly reflects the scheduling efficiency.

The improvement of the efficiency of loading and scheduling was mainly due to three reasons, as discussed below.

Firstly, the arrangement of the ship loader was more reasonable. Because it was in the actual coal terminal scheduling system, the behavior of the ship loader was constrained. Ship loaders load ships side by side; thus, the order cannot be changed, and there is a certain distance. Therefore, when loading a ship, each ship loader is restricted by other ship loaders. If the position of the ship loader that is loaded first is not properly arranged, the later ship loaders cannot find suitable space. At this time, the production scheduling efficiency is reduced, and the production scheduling time is increased.

Secondly, the moving time of the machine was shortened. For reclaimers, the reclaim time includes the time from one coal pile to another. For ship loaders, the loading time includes the time to move from one hold to another. Through the reasonable selection of the line, this time is shortened, and the production scheduling time of the system is further shortened.

Thirdly, the overall reclaiming and loading sequence was optimized. Due to the complexity of the coal terminal loading model, it is difficult to directly analyze the overall optimization of the time. The influencing factors of this part include the learning of order rules, with three identical machines, the coordination of the reclaimer and the ship loader, and the treatment of waiting time for drainage. There are more complex factors in the scheduling time. Using optimization through analysis often cannot obtain good results. At the same time, coal production scheduling is a long-term process, and short-term greedy strategies often cannot obtain long-term advantages. Therefore, for this type of problem, the reinforcement learning method is suitable, as long-term optimization goals can be specified. Then, the system can independently learn production scheduling strategies, thereby improving the long-term benefits.

## 7. Conclusions

In this study, a reinforcement learning model of the bulk cargo terminal loading process was established, and a scheduling strategy algorithm was proposed. The reward function and neural network structure for the port scheduling optimization problem were designed, and the improved double DQN algorithm was applied for training. During the training process, the  $\epsilon$ -greedy strategy was improved for the limitation of action legitimacy, such that there was no need to shield the unfeasible ones during training. At the same time, a test to simulate the actual production scheduling was carried out, and a good optimization effect was achieved. Compared with the random strategy, the moving time of the ship loader and the reclaimer was significantly reduced, and the reward curve had an obvious upward trend, indicating a reduction in the total production scheduling time.

In future research, the neural network structure can be further improved to achieve better scheduling results. The model and algorithm design proposed in this work can be applied to upstream processes in a bulk cargo port after proper modification, such as the dynamic yard configuration problem or the cargo train car dumping schedule problem. Lastly, we hope that the overall scheduling problem of a bulk cargo port from car dumping to ship loading can be combined and solved using the reinforcement learning method.

**Author Contributions:** Writing—original draft preparation, C.L., S.W., Z.L. and Y.Z.; writing—review and editing, L.G.; supervision, C.L. and L.G.; project administration, C.L. and L.Z. All authors read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant U1964201 and Grant U21B6001, the Major Scientific and Technological Special Project of Heilongjiang Province under Grant 2021ZX05A01, the Heilongjiang Natural Science Foundation under Grant LH2019F020, and the Major Scientific and Technological Research Project of Ningbo under Grant 2021Z040.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wang, X.; Shi, H. Research on Intelligent Optimization of Bulk Cargo Terminal Control System. *J. Phys. Conf. Ser.* **2020**, *1601*, 052044. [[CrossRef](#)]
2. De León, A.D.; Lalla-Ruiz, E.; Melián-Batista, B.; Moreno-Vega, J.M. A Machine Learning-based system for berth scheduling at bulk terminals. *Expert Syst. Appl.* **2017**, *87*, 170–182. [[CrossRef](#)]
3. Fotuhi, F.; Huynh, N.; Vidal, J.M.; Xie, Y. Modeling yard crane operators as reinforcement learning agents. *Res. Transp. Econ.* **2013**, *42*, 3–12. [[CrossRef](#)]
4. Imai, A.; Zhang, J.T.; Nishimura, E.; Papadimitriou, S. The berth allocation problem with service time and delay time objectives. *Marit. Econ. Logist.* **2007**, *9*, 269–290. [[CrossRef](#)]
5. Iris, Ç.; Pacino, D.; Ropke, S.; Larsen, A. Integrated berth allocation and quay crane assignment problem: Set partitioning models and computational results. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *81*, 75–97. [[CrossRef](#)]
6. Venturini, G.; Iris, Ç.; Kontovas, C.A.; Larsen, A. The multi-port berth allocation problem with speed optimization and emission considerations. *Transp. Res. Part D Transp. Environ.* **2017**, *54*, 142–159. [[CrossRef](#)]
7. Liu, C.; Zheng, L.; Zhang, C. Behavior perception-based disruption models for berth allocation and quay crane assignment problems. *Comput. Ind. Eng.* **2016**, *97*, 258–275. [[CrossRef](#)]
8. Fisher, M.L.; Rosenwein, M.B. An interactive optimization system for bulk-cargo ship scheduling. *Nav. Res. Logist.* **1989**, *36*, 27–42. [[CrossRef](#)]
9. Fagerholt, K.; Christiansen, M. A combined ship scheduling and allocation problem. *J. Oper. Res. Soc.* **2000**, *51*, 834–842. [[CrossRef](#)]
10. Barros, V.H.; Costa, T.S.; Oliveira, A.C.; Lorena, L.A. Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Comput. Ind. Eng.* **2011**, *60*, 606–613. [[CrossRef](#)]
11. Menezes, G.C.; Mateus, G.R.; Ravetti, M.G. A branch and price algorithm to solve the integrated production planning and scheduling in bulk ports. *Eur. J. Oper. Res.* **2017**, *258*, 926–937. [[CrossRef](#)]
12. Hsu, H.P. A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously. *Swarm Evol. Comput.* **2016**, *27*, 156–168. [[CrossRef](#)]
13. Zhen, L.; Lee, L.H.; Chew, E.P. A decision model for berth allocation under uncertainty. *Eur. J. Oper. Res.* **2011**, *212*, 54–68. [[CrossRef](#)]
14. Lujan, E.; Vergara, E.; Rodriguez-Melquiades, J.; Jiménez-Carrión, M.; Sabino-Escobar, C.; Gutierrez, F. A Fuzzy Optimization Model for the Berth Allocation Problem and Quay Crane Allocation Problem (BAP + QCAP) with n Quays. *J. Mar. Sci. Eng.* **2021**, *9*, 152. [[CrossRef](#)]
15. Cheimanoff, N.; Fontane, F.; Kitri, M.N.; Tchernev, N. A reduced vns based approach for the dynamic continuous berth allocation problem in bulk terminals with tidal constraints. *Expert Syst. Appl.* **2021**, *168*, 114215. [[CrossRef](#)]
16. Sezer, A.; Altan, A. Optimization of deep learning model parameters in classification of solder paste defects. In Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 11–13 June 2021; pp. 1–6.
17. Tran, M.Q.; Liu, M.K.; Tran, Q.V.; Nguyen, T.K. Effective Fault Diagnosis Based on Wavelet and Convolutional Attention Neural Network for Induction Motors. *IEEE Trans. Instrum. Meas.* **2021**, *71*, 3501613. [[CrossRef](#)]
18. Tassel, P.; Gebser, M.; Schekotihin, K. A reinforcement learning environment for job-shop scheduling. *arXiv* **2021**, arXiv:2104.03760.
19. Tran, M.Q.; Liu, M.K.; Tran, Q.V.; Nguyen, T.K. Effective IoT-based Deep Learning Platform for Online Fault Diagnosis of Power Transformers Against Cyberattack and Data Uncertainties. *Measurement* **2022**, *190*, 110686.
20. Sezer, A.; Altan, A. Detection of solder paste defects with an optimization-based deep learning model using image processing techniques. *Solder. Surf. Mt. Technol.* **2021**, *33*, 291–298. [[CrossRef](#)]
21. Tran, M.Q.; Elsis, M.; Liu, M.K.; Vu, V.Q.; Mahmoud, K.; Darwish, M.M.; Lehtonen, M. Reliable Deep Learning and IoT-Based Monitoring System for Secure Computer Numerical Control Machines Against Cyber-Attacks with Experimental Verification. *IEEE Access* **2022**, *10*, 23186–23197. [[CrossRef](#)]
22. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An introduction to deep reinforcement learning. *arXiv* **2018**, arXiv:1811.12560.

23. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Hassabis, D. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)] [[PubMed](#)]
24. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
25. Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
26. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
27. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
28. Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [[CrossRef](#)]