Journal of
Marine Science and Engineering

Article

Optimizing Berth Allocation in Maritime Transportation with Quay Crane Setup Times Using Reinforcement Learning

Yonggai Dai [1], Zongchen Li [2,*] and Boyu Wang [3]

1 Wuhan Hangke Logistics Company Limited, CCCC Second Harbor Engineering Company Ltd., Wuhan 430013, China
2 Mechanical Systems Engineering, EMPA-Swiss Federal Laboratories for Materials Science and Technology, 8600 Duebendorf, Switzerland
3 Shenzhen CaiGao Tech, Shenzhen 518067, China
* Correspondence: zongchen.li@empa.ch

Abstract: Maritime transportation plays a critical role in global trade as it accounts for over 80% of all merchandise movement. Given the growing volume of maritime freight, it is vital to have an efficient system for handling ships and cargos at ports. The current first-come-first-serve method is insufficient in maintaining operational efficiency, especially under complicated conditions such as parallel scheduling with different cargo setups. In addition, in the face of rising demand, data-driven strategies are necessary. To tackle this issue, this paper proposes a mixed-integer model for allocating quay cranes, terminals, and berths. It considers not only cargo types, but also the time required for a quay crane setup. The proposed model features a greedy-insert-based offline algorithm that optimizes berth allocation when vessel information is available. In situations where vessel information is uncertain, the model utilizes an online optimization strategy based on a reinforcement-learning algorithm that is capable of learning from feedback and of adapting quickly in real time. The results of the numerical experiments demonstrate that both the offline and online algorithms can significantly enhance cargo handling efficiency and overall harbor operation. Furthermore, they have the potential to be extended to other complex settings.

Keywords: berth allocation; online optimization; reinforcement learning; quay crane setup

check for updates

Citation: Dai, Y.; Li, Z.; Wang, B. Optimizing Berth Allocation in Maritime Transportation with Quay Crane Setup Times Using Reinforcement Learning. *J. Mar. Sci. Eng.* **2023**, *11*, 1025. https://doi.org/10.3390/jmse11051025

Academic Editors: Mateusz Zając and Tomislav Rožić

Received: 17 April 2023
Revised: 8 May 2023
Accepted: 9 May 2023
Published: 11 May 2023

1. Introduction

The harbor plays a pivotal role in the operation of modern society since it connects maritime transportation and land transportation, which makes it a center for industrial activities and logistics. During the past few centuries, due to the rapid development of logistics, harbor management is undergoing emerging challenges such as sped-up transportation and an increasing volume of cargo. Specifically, the boom of the differentiated commodity economy highlights the interaction between the harbor and its relevant logistics. For the sake of the aforementioned challenges, a high-efficiency operation system is in great demand by harbor managers.

The berth allocation problem (BAP) is a key obstacle in the development of harbor management. As seen in Figure 1, when a ship arrives at a harbor, berth space must be assigned, and a time schedule established. Cargos are typically delivered between the ship and port using quay cranes (QCs). When considering the BAP, ship owners seek to minimize duration at berth as it increases costs with no revenue being generated. Additionally, a multipurpose terminal is capable of handling various types of cargo, including general cargo, bulk cargo, and containers, while using a multipurpose gantry crane as its primary handling equipment. Different types of cargo require different lifting devices for loading and unloading. For general cargo, such as steel products, a hook is commonly used as the lifting device. For bulk cargo, such as minerals, a grab is typically used. For container handling, a container spreader is used to lift and move containers. This flexible handling approach allows multipurpose terminals to meet the diverse cargo handling needs of different industries. However, harbor managers must minimize berth time to accommodate more ships and to increase freight volume. Factors such as the limited capacity of QCs and the need to change the setup for different types of cargo (e.g., bulk cargo, containers, and roll-on roll-off cargo) can impact loading efficiency. Furthermore, the vessel schedule is often unpredictable due to weather and political uncertainties. Online optimization is necessary to balance the various constraints in order to optimize the process.

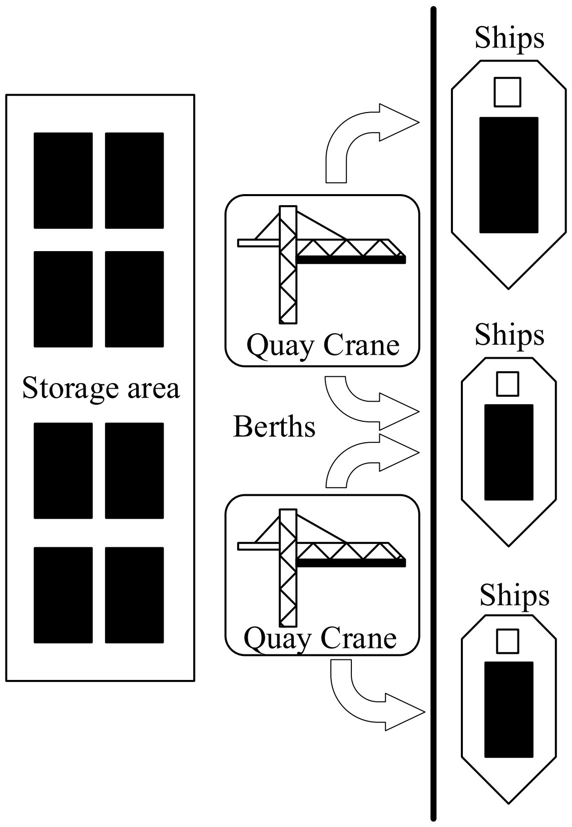

Figure 1. The layout of berths in a harbor.

In recent years, both academics and the industry have devoted considerable effort to developing innovative statistical BAP solutions based on operations research and game theory in response to the increasing freight volume. Before proposing solutions, it is necessary to derive a mathematical model for BAP and to quantify the multiple constraints and objectives. The current literature contains a number of state-of-the-art models, which focus on the uncertainty posed by the arrival time and priority of different ships, but overlook the unique characteristics of each harbor, such as their layout and equipment. To meet this need, a comprehensive, universal BAP model is required to provide optimal solutions. Furthermore, most existing models are continuous in nature, but a discrete model is more suitable for practical applications.

The traditional first-in-first-serve (FIFS) scheduling strategy for ships at berth in harbors is often inefficient, particularly under extreme conditions. To improve operational efficiency, computer-aided optimization algorithms have been introduced to harbor management. Various algorithms, including genetic algorithms, particle swarm optimization, neural network algorithms, and reinforcement learning algorithms, each offer unique advantages for different application scenarios.

This paper aims to comprehensively summarize and categorize the various specs of a harbor in order to address the aforementioned problem. In order to address the challenges posed by the rising demand and scalability of various scenarios, a data-driven model, which considers the capacity of loading, type of cargo, and the setup switching time, for a

QC ship system is constructed. To minimize the total waiting time of ships in scenarios where all vessel information is known beforehand, an offline solution with a greedy-based algorithm is proposed. A greedy insert algorithm is used to achieve this goal. Compared to other algorithms, such as the genetic algorithm, the greedy insert algorithm can reduce the computation load and can increase the calculation speed. As real-world scenarios often involve unknown information, such as the arrival time and the capacity of ships, the online optimization of berth allocation is necessary. Therefore, a reinforcement learning algorithm is proposed for online optimization. This machine learning algorithm is advantageous in terms of convergence and scalability, making it suitable for the rapid optimization of berth scheduling and allocation. A numerical experimental study is conducted using a typical case from a harbor to validate the functionality of the proposed model and solutions. The results of the comparison between the calculation results and the baseline reveal that the proposed reinforcement learning algorithm performs excellently and can significantly improve the operating efficiency of the harbor.

The other sections of the paper are organized as per the following below. Section 2 summarizes the previous efforts in developing BAP models and solutions. Section 3 defines the BAP problem with regard to quay crane setup times. Section 4 presents the offline case with a heuristic algorithm, while Section 5 demonstrates the reinforcement learning formulation for the online case. Section 6 presents and discusses the numerical results. Section 7 concludes the paper and incorporates insight into this research.

2. Literature Review

Maximizing the operating efficiency of the harbor is the top priority target for harbor management. Prior to optimizing the operation of the harbor, the BAP and quay crane scheduling problem (QCSP) should be addressed [1]. This type of research has attracted the attention of the industry since the end of the last century. Several studies have proposed solutions for the BAP and QCSP, including the use of novel set partitioning models and variable reduction techniques [2]. Other studies have focused on the integration opportunities and environmental issues that are related to cargo shipping through optimization, showing that an accurate speed discretization can result in better economic and environmental results [3].

Lai et al. [4] improved the conventional berth allocation strategy by changing the allocation standard. According to the average waiting time of ships, the operating time of the berth allocation system and the average utilization of berths, three different berth allocation strategies are proposed. They also observed that the BAP cannot be addressed with a constant standard. Kim et al. [5] have contributed to making this topic attractive to academia in their publication, which presents a detailed model of QC scheduling. This model considers some factors such as the container groups, non-crossing restrictions, and safety distances, which make the model more accurate in practical applications. Iris et al. [6] have integrated berth allocation and energy management approaches in ports. Bierwirth et al. [7] have summarized the existing research in optimizing berth allocation and the QCSP.

Prior to addressing the BAP, an accurate model that can quantify the impact of various variables on the waiting time should be constructed. The major challenge that had a large impact on the model is the uncertainty of vessel information, including the ship's arrival time and its priority of service. Therefore, as categorized by Rodrigues et al. [8], several types of models are proposed in this field, including the stochastic model [9], the robust optimization model [10], the fuzzy model [11], and deterministic models [12].

Nowadays, due to rapidly increasing transportation demands and complicated setups in practical scenarios, data-driven methods are being introduced to solve the BAP in order to minimize the impact caused by the uncertainty or intermittence of ships' schedules. For instance, the method proposed by Xiang et al. [13] considers the uncertainty of the vessel arrival time, as well as the operating time of the QCs. Based on this model, a berth allocation model that can minimize the operating cost is proposed. Agra et al. [14]

comprehensively consider the BAP and QCSP and propose a relative position formulation equation. Tengecha et al. [15] utilize a constraint programming approach to address the QCSP.

Based on the algorithms utilized, there are several types of methodologies. The basic method is based on the approximation algorithms introduced by Lee et al. [16] and Liu et al. [17]. The mixed-integer linear programming (MILP) standard solver [18–23] is an extensively applied method. Since the BAP is a multi-objective optimization problem, some Pareto algorithms, such as the genetic algorithm [24–32] and unidirectional-search-based algorithm [33,34], are also appropriate.

Among all the aforementioned algorithms, reinforcement learning is appropriate for the optimal control of nonlinear systems [35,36], traffic control [37], manufacturing automation [38], and enterprise management [39]. It was significantly improved by Mnih et al. [40] by applying a deep Q-network (DQN) structure. Through its learning to make and correcting mistakes, it has superiority in processing very complex problems that cannot be solved by conventional computational techniques, and it can achieve long-term good results. However, the application of reinforcement learning to solving the BAP has not yet been reported. This paper will discuss the application of reinforcement learning in solving the online berth allocation and QC scheduling problems.

3. Problem Definition

Prior to conducting optimization, it is necessary to understand the operation of a harbor via defining the problem and all of the considered variables. In other words, the final objective should be constructed. The conventional berth allocation method follows the FIFS principle, while the operation efficiency is not considered. The optimized method should aim at minimizing the total waiting time of the ships in the berths.

Herein, we assume a set of vessels V that are served at a set of berths B. Each vessel ships with a single type of cargo are understood from the designated cargo set C, and each ship should be allocated to one berth exactly. The service time of a vessel $v \in V$ carrying cargo with type $c \in C$ depends on its cargo weight and the corresponding quay crane resource; thus, this can be simplified as t_v^b. The arrival time t_v^a is deterministic since it is usually given by the ship owners. As mentioned in Section 1, there are various types of cargo, including liquid cargo, dry bulk cargo, etc. It usually takes some time for QCs to re-setup if the cargo types change. This setup time is denoted by a matrix T_s. For example, $T_s(1,2)$ refer to the consuming time for a quay crane that changes its setup from processing a type 1 cargo into a type 2 cargo. It should be noted that T_s can be different in different harbors, and $T_s(\mathrm{i},\mathrm{j})$ could be a large positive number if the QCs are not able to switch between cargo i and cargo j. The final objective of this setting is to minimize the total waiting time. Based on the previous analysis, the objective function can be derived as given in Equation (1).

$$Minimize \sum\nolimits_{i=1}^{|V|} (t_i^s - t_i^a), \tag{1}$$

where t_i^s is the start service time of vessel i. The definitions of each variable utilized in the following sections can be found in Table 1.

Table 1. Notations utilized for the BAP.

Notation	Meaning
V	Set of vessels
B	Set of berths
C	Set of cargo types
t_v^b	Service time of vessel v in berth b
t_v^s	Start service time of vessel v
t_v^a	Arrival time of vessel v
T_s	Setup changing time between cargo types

No matter whether all the information is known ahead of time or not, the final target of the optimization can be summarized in the form of Equation (1).

4. Offline Case

The offline case occurs when all information such as the arrival time, cargo type, and QC setup time is revealed. With the known information, the global optimization can be conducted. We first consider an offline setting where the vessel's information $\left\{v \in V \middle| \left\langle C_v, t_v^a,\ t_v^b \right\rangle\right\}$ is known ahead of time, and the optimal allocation is searched to minimize the total waiting time. In this section, we first formulate our setting as an integer programming problem and then propose a heuristic algorithm to solve it.

4.1. Optimization Formulation

Based on Equation (1), the problem can be formulated as follows:

$$min \sum_{i=1}^{|V|} (t_i^s - t_i^a) \tag{2}$$

$$s.t.\ \sum_{i=1}^{|V|} x_{0i}^b \le 1,\ b \in B \tag{3}$$

$$\sum_{i=0}^{|V|} \sum_{b=1}^{|B|} x_{ik}^b = 1,\ k \in V \tag{4}$$

$$\sum_{k=1}^{|V|+1} x_{ik}^b = y_i^b,\ i \in V,\ b \in B \tag{5}$$

$$\sum_{k=0}^{|V|} x_{ki}^b = y_i^b,\ i \in V,\ b \in B \tag{6}$$

$$\sum_{b=1}^{|B|} y_i^b = 1,\ i \in V \tag{7}$$

$$\sum_{b=1}^{|B|} \sum_{k=0}^{|V|} (x_{ki}^b t_k^s + T_{ki} x_{ki}^b + x_{ki}^b t_k^b) = t_i^s,\ i \in V \tag{8}$$

$$t_0^s, t_0^b, T_{0k} = 0 \tag{9}$$

$$x_{ki}^b = \begin{cases} 1 \text{ if vessel i immediately follows vessel k on berth b,} \\ 0 \text{ otherwise,} \end{cases} \tag{10}$$

$$y_i^b = \begin{cases} 1 \text{ if vessel i assigned on berth b,} \\ 0 \text{ otherwise,} \end{cases} \tag{11}$$

T_{ki} denotes the setup switching time from vessel k to vessel i. Constraint (3) ensures that for each berth selected to allocate vessels, only one real vessel follows the dummy vessel 0. Constraint (4) ensures that if a vessel is allocated into a specific berth, it will be served after one vessel, and a vessel must be served only at one position of a berth. Constraints (5)–(6) determines that if two vessels are served sequentially, then they are allocated in the same berth. Constraint (7) states that each vessel is scheduled to exactly one berth. Constraints (8)–(9) restrict the vessel sequence of the starting times by the starting time of the previous vessel, the service time of the previous vessel, and the time required for setup switching. Finally, Constraints (10)–(11) define the domain of the decision variables.

4.2. Greedy Insert Algorithm

To solve this problem, we propose a greedy insert algorithm to enhance the allocation performance step-by-step based on an initial solution. A solution is a vector of a numerical sequence representing the vessels, and the $|B| - 1$ zeros represent the separation of the different berths. Figure 1 shows an example of the solution representation for a three berths, seven vessels case. Thus, the total waiting time can be calculated based on the vessel information, berth information, and the allocation.

The main idea of the proposed greedy insert algorithm is first sorting the vessels according to their arrival time, then allocating them into berths following the FIFS (first-in-first-serve) rule in which the incoming vessel will be allocated into the berth with the earliest idle time. After an initial solution is obtained, based on the FIFS rule as shown in Figure 2, a vessel is randomly selected at each step. Herein, a greedy strategy is adopted by reinserting this vessel into the allocation, and the best position is picked where the total waiting time reduces the most. Finally, if the total waiting time increases or if the computation time exceeds a specific threshold, the iteration stops. The solution that can be calculated with the aforementioned algorithm indicates the allocation of the vessels at berth and the serve time for each ship. Even though the solution is generally not the optimal value, the results are still acceptable considering the complexity of the calculation. Algorithm 1 illustrates the process of the proposed greedy insert algorithm, where M is the preset iteration threshold, f(.) represents the total waiting time calculation, and $\lceil x_{k-1}, I \rceil_p$ represents the I^{th} non-zero entry of the vector x_{k-1} shift to position p.

Algorithm 1. Greedy Insert

```
1: Input: $\left\{\left\langle V_n, C_n, t^a_{V_n}, t^s_{V_n}, t^b_{V_n} \right\rangle\right\}, T_s$
2: Initialize: $x_0 = FIFS\left(\left\{\left\langle V_n, C_n, t^a_{V_n}, t^s_{V_n}, t^b_{V_n} \right\rangle\right\}, T_s\right)$
3: **while** $T_k < T_{k-1}$ or $k \leq M$:
4:         Randomly select $V_k$ with index $I_k$
5:         $I^* = min_p f\left(\lceil x_{k-1}, I_k \rceil_p\right)$
6:         $x_k :\leftarrow \lceil x_{k-1}, I^* \rceil_p$
7: **end while**
8: **return x**
```

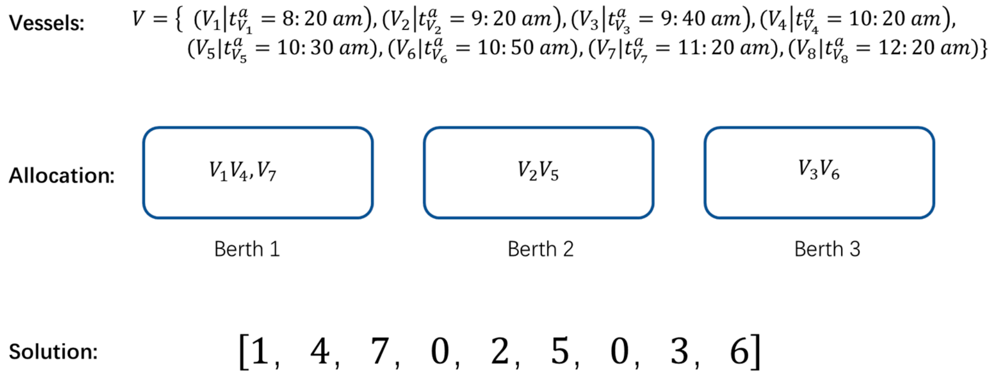
Vessels:
$V = \{ (V_1|t^a_{V_1} = 8{:}20\ am), (V_2|t^a_{V_2} = 9{:}20\ am), (V_3|t^a_{V_3} = 9{:}40\ am), (V_4|t^a_{V_4} = 10{:}20\ am), (V_5|t^a_{V_5} = 10{:}30\ am), (V_6|t^a_{V_6} = 10{:}50\ am), (V_7|t^a_{V_7} = 11{:}20\ am), (V_8|t^a_{V_8} = 12{:}20\ am)\}$
Allocation:
$V_1 V_4, V_7$
$V_2 V_5$
$V_3 V_6$
Berth 1
Berth 2
Berth 3
Solution:
[1, 4, 7, 0, 2, 5, 0, 3, 6]

Figure 2. An example of the allocation representation.

A typical example to illustrate the greedy insert algorithm is the given vessels set V from Table 2 being allocated into three berths with an identical service time; the setup changing times are assumed to be {A to B: 15 min, A to C: 20 min, B to A: 15 min, B to C: 45 min, C to A: 20 min, and C to B: 30 min}. Then, the solution is initialized (as shown in Figure 2), where vessels (V1, V4, V7), (V2, V5), and (V3, V6) are assigned to Berth 1, 2, and 3, respectively. Numerically, the solution is in the form of a vector [1,4,7,0,2,5,0,3,6]. Note that each berth is separated by the number zero. The total waiting time for this solution can be calculated as 195 min. On Iteration 1, suppose V_4 is randomly selected and can be possibly relocated into any berth with any order. The corresponding total waiting time can be calculated and the shortest total waiting time will be selected, which is 185 min, which corresponds to a solution [1,7,0,2,4,5,0,3,6]. Then, move to the next iteration. In this way, the berth allocation and QC scheduling can be gradually improved.

Table 2. An example of a vessel set.

Vessel	Arrival Time	Cargo Type	Service Time (min)
V_1	8:20 a.m.	A	140
V_2	9:20 a.m.	A	90
V_3	9:40 a.m.	B	100
V_4	10:20 a.m.	C	65
V_5	10:30 a.m.	C	70
V_6	10:50 a.m.	B	90
V_7	11:20 a.m.	A	120

5. Online Case

The offline case is more straight forward since all information is known. However, in real-world applications, some information is usually unknown, or it may change due to various unpredicted reasons. For instance, the weather forecast usually has considerable errors and it can significantly affect the arrival time of the vessel ships. Therefore, online cases are more critical for practical applications.

In this case, we now consider an online setup where the vessels arrive sequentially, and the information for arrival times, types of cargo, and handling time are unavailable until the vessel arrives at the port. Thus, once a new vessel arrives, the port will schedule the vessel to one of the berths to minimize the overall waiting time. The new variables that will be used for online case optimization are listed in Table 3.

Table 3. State space for online optimization cases.

Notation	Meaning	Dimension
S_p	In-processing berth status	$N_b \times N_c$
S_h	Processing history	$N_b \times N_c$
S_a	Service time of the arrival vessel	$N_b \times 1$
S_t	Type of cargo	1

In Table 3, N_b and N_c denote the number of berths and cargo type, respectively. The berth status S_p and the processing history S_h are all in the form of matrix with the dimension $N_b \times N_c$.

5.1. MDP Formulation

The online BAP considered in this paper is formulated as a Markov decision process (MDP) problem. The MDP is a discrete-time stochastic process used to model decisions, and it can be represented mathematically in a tuple (S, A, P, R, γ), where S is the state space, A is the action space, P is the transition probability, R is the reward after an action is taken, and γ is the discount factor.

States: We define the states by observing the current status of the berths and vessels. The proposed state consists of two matrices and a vector as follows:

- S_p: The most important status is to describe the current working situation of the quay cranes of the corresponding berths. We define the matrix S_p to represent the remaining working time of each berth. In S_p, each row means a berth, and each column means a type of cargo; thus, the entries are the remaining time of a specific type of cargo processed in a specific berth;
- S_h: To capture the history of the berths and the quay cranes, we define a matrix S_h, which has a similar structure to S_p but which represents the processed working time for each type of cargo at each berth;
- S_a: We also define a vector to represent the service time of the arrival vessels that are being allocated into different berths;
- S_t: A scalar represents the type of cargo that the arrival vessel carries.

Actions: the actions are defined by simply allocating the arrival vessel to one of the berths, and the action set A is defined as follows:

$$A = \{a|a = 1, \dots N_b\} \tag{12}$$

An action $a_n = k$ indicates that a vessel arrives at time slot n, and it will be allocated into the k^{th} berth. It should be noted that the port only makes an action at one time slot when there a vessel arrives at that time slot.

Rewards: The reward defined in this paper is the waiting time of the vessel until it starts to be served. It can be expressed as is given in (13).

$$r_n = -t_r^b \tag{13}$$

State transitions: After executing a_n, the state transitions from S_n into S_{n+1}. Intuitively, the in-processing berth status and the last action will be updated accordingly in terms of the allocation.

Example: Figure 3 describes an example of vessel allocation and depicts how the MDP model evolves. V_1, V_2, and V_3 are the three vessels that are parking at different berths and being handled by the quay cranes. When V_4 arrives at Time slot 2, it is allocated into the third berth by taking action $a_2 = 3$. Then, the state transitions from S_2 into S_3. Specifically for S_p, the current working status is updated by switching from a steel cargo into a container, and the remaining processing time is updated by considering the addition of V_4 into the queue of Berth 3. Similarly, for S_h, the history will be updated accordingly. In addition, S_a and S_t return to 0.

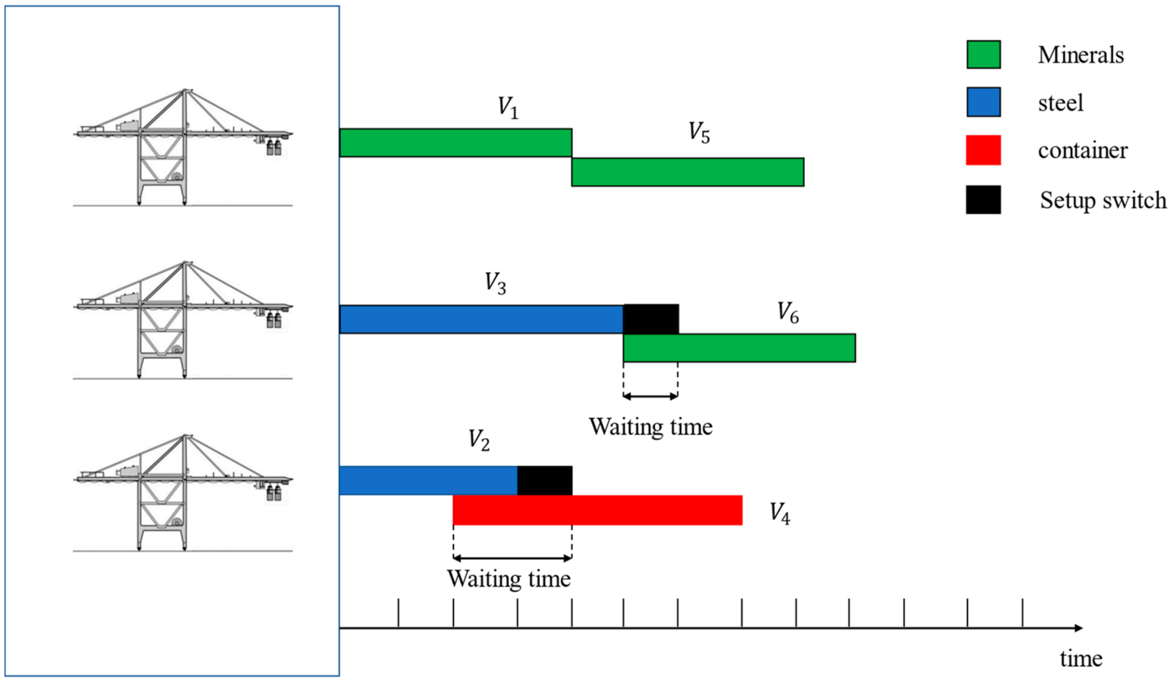

Figure 3. An example of vessel allocation.

5.2. Dueling-DQN-Based Algorithm

A dueling deep Q-network (DDQN) was trained to learn the state-value function. DDQN is a variant of DQN; it takes a state s as an input and outputs Q-values $Q_\theta(s, a)$ for all candidate actions $a \in A$ with parameters θ. However, DQN performs inefficiently in practice, and the DDQN structure was proposed to overcome this issue by decoupling the value and advantage in DQN. Figure 4 shows the proposed fully connected network architecture where the input of DDQN is a matrix constructed by concatenating the state matrices S_p, S_h, and S_a. Following the fully connected hidden layer, two sequences are constructed to estimate the value and advantage functions separately. Finally, the value function and the advantage function are aggregated into Q-values.

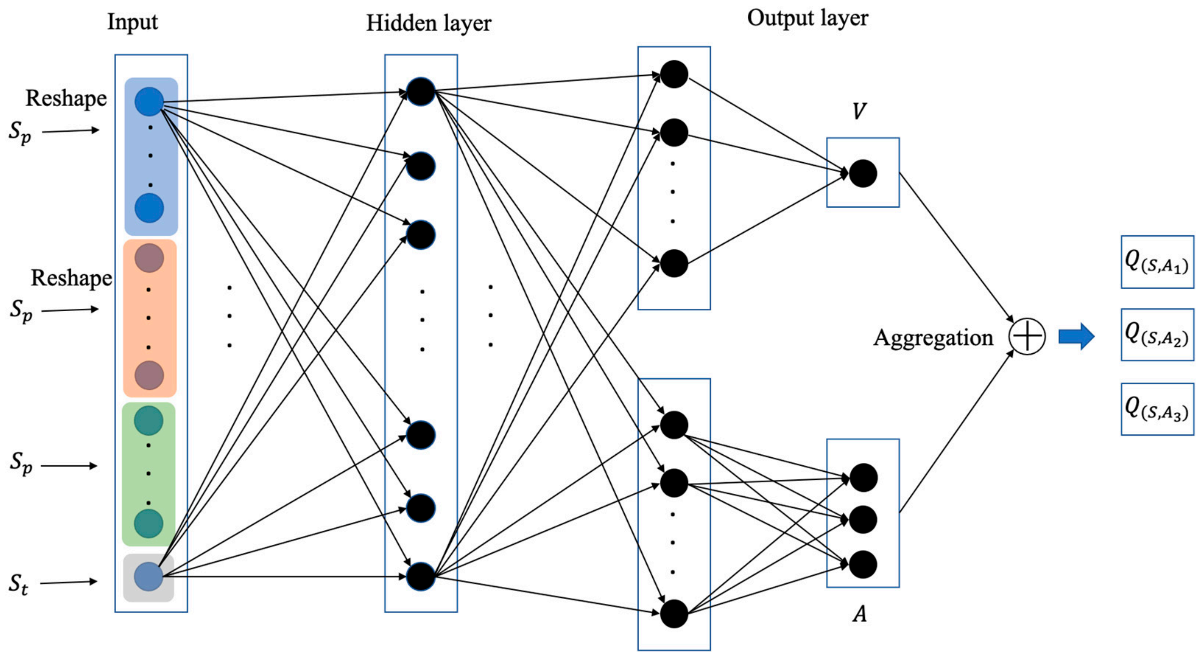

Figure 4. Architecture of the DDQN.

The training of DDQN is similar to regular DQN. First, the action is determined by an ε-greedy policy to balance the exploration and exploitation. Then, the performance will be stored as tuples in a replay memory. The small batch of tuples from this memory are selected randomly, and the value function $\hat{V}(s)$ and advantage function $\hat{A}(s,a)$ are calculated and aggregated into the Q-value. Finally, a gradient descent step is performed, and parameters are updated accordingly. The training process is defined in Algorithm 2.

Algorithm 2. DDQN Training

```
1:  Input: $\left\{\left\langle V_n, C_n, t^a_{V_n}, t^s_{V_n}, t^b_{V_n}\right\rangle\right\}, T_s$
2:  Initialize: Initialize the weight $\omega$ of network $Q(s,a)$
3:  **for** $e = 1, 2, \ldots, E$ **do**
4:          Initialize states
5:      **for** $t = 1, 2, \ldots, T$ **do**
6:          select action a based on $\varepsilon$-greedy
7:          Update state $S$
8:          Store transition in replay memory $D$
9:          Sample random minibatch of transitions from $D$
10:         Calculate value function $\hat{V}(s)$ and advantage function $\hat{A}(s,a)$
11:         Aggregate Q value
12:         Perform a gradient descent step
13:     **end for**
14: **end for**
15: **return w**
```

6. Numerical Results

6.1. Offline Case

To validate the functionality of the proposed offline algorithm, a numerical case study was conducted. A total of 3 berths, 5 types of cargo, and 80 vessels, as well as 10 cases were tested; the results are listed in Table 4. The original information of the ships is based on the practical data of a harbor in China. The computer utilized for the numerical study was an Apple MacBook with a 2.2 GHz CPU and 16 GB RAM, and the programming environment was Python 3.8. The cargo type was generated with discrete uniform distribution, and the cargo-setup switching time is the real data from the harbor.

Table 4. Numerical results for the offline cases.

Case No.	Optimal	FIFS	Greedy Insert
Computation time	225 s	0.03 s	1.1 s
1	18,233 min	21,823 min	20,435 min
2	19,338 min	27,949 min	22,391 min
3	17,391 min	26,596 min	18,935 min
4	19,476 min	31,450 min	28,234 min
5	16,758 min	29,273 min	26,743 min
6	17,698 min	28,229 min	23,210 min
7	18,290 min	27,652 min	21,431 min
8	24,210 min	35,183 min	28,947 min
9	19,261 min	26,832 min	23,998 min
10	18,992 min	27,079 min	21,284 min

Three different groups of results are compared. The optimal results are calculated via a mathematics-based solver by defining the optimization formulation in Equation (2) with high computational complexity. The average computation time for optimal results is 225 s. The average computation time of the greedy insert and the FIFS are 1.1 s and 0.03 s, respectively.

To better compare the performance of the greedy insert and the FIFS, the average waiting time of the FIFS and greedy insert are referred to the optimal results, which are listed in Table 4 and plotted in Figure 5. As shown in Figure 5, the greedy insert shows a superiority in terms of waiting time over the FIFS.

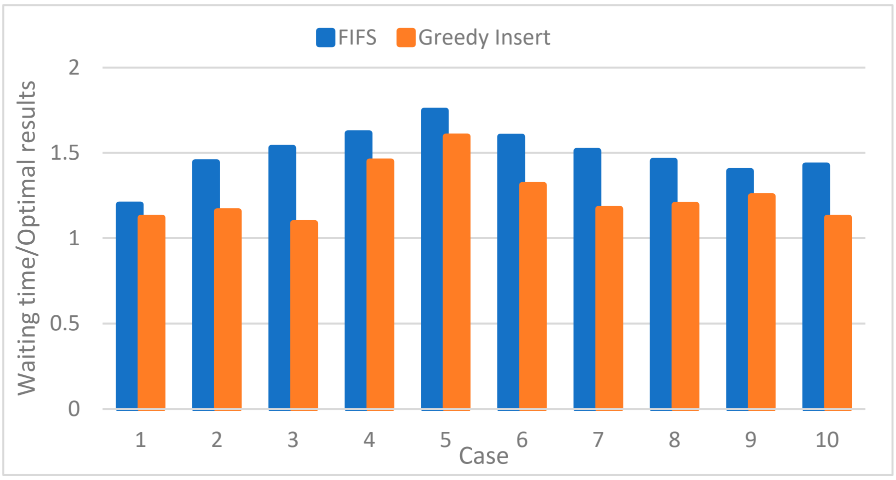

Figure 5. Waiting time for the FIFS and greedy insert over the optimal results.

6.2. Online Case

To validate the functionality of the online optimization, a numerical study was conducted with the known information, such as the number of berths, types of cargo, QC capacity, etc., while the vessel arrival time, amount of cargo, etc., were unknown. The computer utilized was the same one that was utilized with the offline case validation. The original data input is from the real harbor daily operation. Nine groups of experimental studies were conducted with the FIFS and DDQN, respectively, and the results are plotted in Figure 6. To investigate the performance of the DDQN algorithm, the amounts of berths and cargo types are different in each case. It should be noted that, in Figure 6, the vessel amount is selected to be 50, 80, and 120. The relationship between the vessel amount and the total waiting time is straight, as is shown in Figure 6.

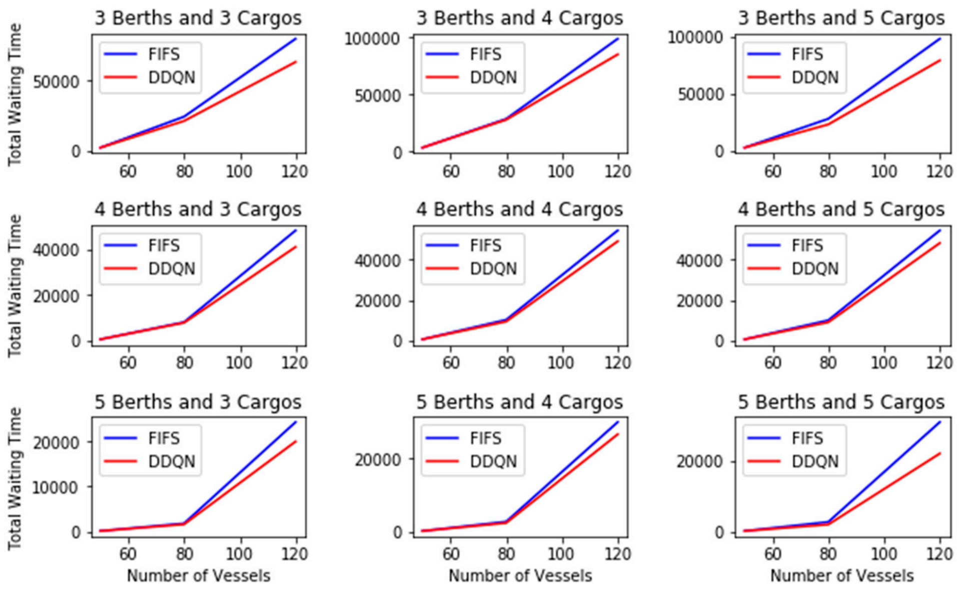

Figure 6. Comparison of the FIFS and DDQN for different online cases.

7. Discussion and Conclusions

7.1. Discussion

The numerical results of the offline cases, including the amount of cargo, type of cargo, and setup switch, are shown in Table 3. The greedy insert approach yields wait times that are nearly close to optimal but are much shorter than the FIFS in all cases. The greedy insert approach also requires slightly more computation time than the FIFS, but it still only takes 1.1 s to finish the calculation, making it an acceptable tradeoff between performance and efficiency. Therefore, it is suitable for the practical operation of berth allocation and QC scheduling for a harbor.

For the offline case optimization, since all information is known prior to calculation, there are various algorithms reported in the existing references. Based on the survey given by Bierwirth et al. [7], the genetic algorithm can give a better result while the computation time is correspondingly longer. In the following research, it is interesting to conduct a comparative study between the state-of-the-art optimization algorithms.

The results of the online numerical case study are displayed in Figure 6. It can be seen that reinforcement learning has a more significant advantage as the number of vessels increases. Both the type and the number of cargo and berths significantly affect the optimization outcomes. Comparing the group with five berths and five types of cargo to the group with three berths and three types of cargo, it is clear that reinforcement learning can reduce the waiting times more effectively in larger, more complex harbors. The DDQN drastically reduces the total waiting time for the 5 berths, 5 types of cargo, and 120 vessels setup, requiring only 60% of the total waiting time that is required by the FIFS. It should be noted that the outcome of the computation may vary between rounds, and the performance can be improved further with appropriate data training.

A major advantage of the DDQN is its high efficiency due to the off-policy feature. It can increase the convergence speed over the DQN. In the following research, more machine learning algorithms such as proximal policy optimization can be employed for the BAP solution and can be compared with the DDQN.

Future work can explore the potential for developing and applying different reinforcement-learning-based models to optimize harbor operations. In addition, different setups can be explored, such as considering multiple terminals, diverse cargo types, and varying environmental factors. The effectiveness of alternative optimization algorithms and decision-making frameworks can also be examined. Additionally, the implementation of real-time data feeds and advanced analytics techniques, such as predictive modeling and anomaly detection, could further enhance the efficiency and reliability of harbor operations.

7.2. Conclusions

This study provides a comprehensive analysis of the BAP and QCSP. To address these issues, a Markov decision process model was constructed based on a real harbor layout. This model considers the loading capacity of quays, the types of cargo, and the setup time of switching. Two cases were discussed: an offline case, where all vessel information is known, and an online case, where the uncertainty of vessel arrival time is taken into account. The offline optimization utilized the greedy insert algorithm, which drastically reduced waiting time while keeping computation time to a minimum when compared to the conventional FIFO method. Online optimization was accomplished through the deep double Q-network (DDQN) reinforcement learning algorithm, which is capable of learning from feedback and quickly adapting in real-time; in addition, it further reduced the waiting time when compared to the FIFO approach. The optimization algorithms proposed in this study have demonstrated the capacity to substantially enhance efficiency and flexibility in harbor operations while accommodating uncertainties. Additionally, the feasibility of extending these algorithms to more intricate scenarios has been explored through a data-driven approach.

Author Contributions: Conceptualization, Y.D., B.W. and Z.L.; methodology, Y.D.; software, Y.D. and B.W.; validation, Y.D., Z.L. and B.W.; formal analysis, Y.D.; investigation, Y.D. and Z.L.; resources, Y.D.; data curation, Y.D., B.W. and Z.L.; writing—original draft preparation, Y.D. and Z.L.; writing—review and editing, Y.D., Z.L. and B.W.; visualization, Y.D.; supervision, Y.D., Z.L. and B.W.; project administration, Y.D., Z.L. and B.W.; funding acquisition, Y.D., Z.L. and B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All of the data of this study is contained within the article.

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

References

1. Mnasri, S.; Alrashidi, M. A comprehensive modeling of the discrete and dynamic problem of berth allocation in maritime terminals. *Electronics* **2021**, *10*, 2684. [CrossRef]
2. 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]
3. 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]
4. Lai, K.K.; Shih, K. A study of container berth allocation. *J. Adv. Transp.* **1992**, *26*, 45–60. [CrossRef]
5. Kim, K.H.; Park, Y.M. A crane scheduling method for port container terminals. *Eur. J. Oper. Res.* **2004**, *156*, 752–768. [CrossRef]
6. Iris, Ç.; Lam, J.S.L. Optimal energy management and operations planning in seaports with smart grid while harnessing renewable energy under uncertainty. *Omega* **2021**, *103*, 102445. [CrossRef]
7. Bierwirth, C.; Meisel, F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **2015**, *244*, 675–689. [CrossRef]
8. Rodrigues, F.; Agra, A. Berth allocation and quay crane assignment/scheduling problem under uncertainty: A survey. *Eur. J. Oper. Res.* **2022**, *303*, 501–524. [CrossRef]
9. Iris, Ç.; Lam, J.S.L. Recoverable robustness in weekly berth and quay crane planning. *Transp. Res. Part B Methodol.* **2019**, *122*, 365–389. [CrossRef]
10. Xiang, X.; Liu, C. An expanded robust optimisation approach for the berth allocation problem considering uncertain operation time. *Omega* **2021**, *103*, 102444. [CrossRef]

11. Gutierrez, F.; Lujan, E.; Asmat, R.; Vergara, E. Fuzziness in the berth allocation problem. In *Recent Advances in Computational Optimization: Results of the Workshop on Computational Optimization WCO 2017*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 149–174.
12. 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]
13. Xiang, X.; Liu, C.; Miao, L. A bi-objective robust model for berth allocation scheduling under uncertainty. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *106*, 294–319. [CrossRef]
14. Agra, A.; Oliveira, M. MIP approaches for the integrated berth allocation and quay crane assignment and scheduling problem. *Eur. J. Oper. Res.* **2018**, *264*, 138–148. [CrossRef]
15. Tengecha, N.A.; Zhang, X. An Efficient Algorithm for the Berth and Quay Crane Assignments Considering Operator Performance in Container Terminal Using Particle Swarm Model. *J. Mar. Sci. Eng.* **2022**, *10*, 1232. [CrossRef]
16. Lee, D.H.; Chen, J.H. An improved approach for quay crane scheduling with non-crossing constraints. *Eng. Optim.* **2010**, *42*, 1–15. [CrossRef]
17. Liu, M.; Zheng, F.; Li, J. Scheduling small number of quay cranes with non-interference constraint. *Optim. Lett.* **2015**, *9*, 403–412. [CrossRef]
18. de Andrade, J.L.M.; Menezes, G.C. A column generation-based heuristic to solve the integrated planning, scheduling, yard allocation and berth allocation problem in bulk ports. *J. Heuristics* **2023**, *29*, 39–76. [CrossRef]
19. Guo, L.; Zheng, J.; Du, H.; Du, J.; Zhu, Z. The berth assignment and allocation problem considering cooperative liner carriers. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *164*, 102793. [CrossRef]
20. Tang, M.; Ji, B.; Fang, X.; Yu, S.S. Discretization-strategy-based solution for berth allocation and quay crane assignment problem. *J. Mar. Sci. Eng.* **2022**, *10*, 495. [CrossRef]
21. Lv, X.; Jin, J.G.; Hu, H. Berth allocation recovery for container transshipment terminals. *Marit. Policy Manag.* **2020**, *47*, 558–574. [CrossRef]
22. Ji, B.; Tang, M.; Wu, Z.; Samson, S.Y.; Zhou, S.; Fang, X. Hybrid rolling-horizon optimization for berth allocation and quay crane assignment with unscheduled vessels. *Adv. Eng. Inform.* **2022**, *54*, 101733. [CrossRef]
23. Iris, Ç.; Lalla-Ruiz, E.; Lam, J.S.L.; Voß, S. Mathematical programming formulations for the strategic berth template problem. *Comput. Ind. Eng.* **2018**, *124*, 167–179. [CrossRef]
24. Zhao, S.; Zhao, X.; Farnell, C.; Mantooth, H.A.; Umuhoza, J.; Zhang, Y. A daily optimization method for a PV-battery microgrid considering the battery lifetime and time-of-use pricing. In Proceedings of the 2019 IEEE Applied Power Electronics Conference and Exposition (APEC), Anaheim, CA, USA, 17–21 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3243–3250.
25. Hu, X.; Ji, S.; Hua, H.; Zhou, B.; Hu, G. An Improved Genetic Algorithm for Berth Scheduling at Bulk Terminal. *Comput. Syst. Sci. Eng.* **2022**, *43*, 1285–1296. [CrossRef]
26. Jiang, X.; Zhong, M.; Shi, J.; Li, W.; Sui, Y.; Dou, Y. Overall Scheduling Model for Vessels Scheduling and Berth Allocation for Ports with Restricted Channels That Considers Carbon Emissions. *J. Mar. Sci. Eng.* **2022**, *10*, 1757. [CrossRef]
27. Yu, J.; Tang, G.; Song, X. Collaboration of vessel speed optimization with berth allocation and quay crane assignment considering vessel service differentiation. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *160*, 102651. [CrossRef]
28. Fatemi-Anaraki, S.; Tavakkoli-Moghaddam, R.; Abdolhamidi, D.; Vahedi-Nouri, B. Simultaneous waterway scheduling, berth allocation, and quay crane assignment: A novel matheuristic approach. *Int. J. Prod. Res.* **2021**, *59*, 7576–7593. [CrossRef]
29. Yu, F.; Shan, Q.; Xiao, Y.; Teng, F. Robust Low-Carbon Discrete Berth Allocation under Uncertainty. *Int. Trans. Electr. Energy Syst.* **2022**, *2022*, 5310004. [CrossRef]
30. Liu, B.; Li, Z.C.; Wang, Y. A two-stage stochastic programming model for seaport berth and channel planning with uncertainties in ship arrival and handling times. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *167*, 102919. [CrossRef]
31. Liu, B.; Li, Z.C.; Wang, Y.; Sheng, D. Short-term berth planning and ship scheduling for a busy seaport with channel restrictions. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *154*, 102467. [CrossRef]
32. Iris, Ç.; Pacino, D.; Ropke, S. Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *105*, 123–147. [CrossRef]
33. Meisel, F.; Bierwirth, C. A framework for integrated berth allocation and crane operations planning in seaport container terminals. *Transp. Sci.* **2013**, *47*, 131–147. [CrossRef]
34. Legato, P.; Trunfio, R. A local branching-based algorithm for the quay crane scheduling problem under unidirectional schedules. *4OR Q. J. Oper. Res.* **2014**, *12*, 123–156. [CrossRef]
35. Huang, Q.; Huang, R.; Hao, W.; Tan, J.; Fan, R.; Huang, Z. Adaptive power system emergency control using deep reinforcement learning. *IEEE Trans. Smart Grid* **2019**, *11*, 1171–1182. [CrossRef]
36. Xiao, X.; Waddell, C.; Hamilton, C.; Xiao, H. Quality Prediction and Control in Wire Arc Additive Manufacturing via Novel Machine Learning Framework. *Micromachines* **2022**, *13*, 137. [CrossRef]
37. Wan, Z.; Jiang, C.; Fahad, M.; Ni, Z.; Guo, Y.; He, H. Robot-assisted pedestrian regulation based on deep reinforcement learning. *IEEE Trans. Cybern.* **2018**, *50*, 1669–1682. [CrossRef]
38. Xiao, X.; Joshi, S. Process planning for five-axis support free additive manufacturing. *Addit. Manuf.* **2020**, *36*, 101569. [CrossRef]

39. Xiao, X.; Roh, B.M.; Hamilton, C. Porosity management and control in powder bed fusion process through process-quality interactions. *CIRP J. Manuf. Sci. Technol.* **2022**, *38*, 120–128. [CrossRef]
40. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]