Journal of *Marine Science and Engineering*

MDPI

Article

A Machine Learning-Based Approach for Multi-AGV Dispatching at Automated Container Terminals

Yinping Gao [1,*], Chun-Hsien Chen [2] and Daofang Chang [3]

[1] School of Management, Shanghai University, 99 Shangda Road, Shanghai 200444, China
[2] School of Mechanical & Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
[3] Logistics Engineering College, Shanghai Maritime University, 1550 Haigang Avenue, Shanghai 201306, China
* Correspondence: gaoyp@shu.edu.cn

Abstract: The dispatching of automated guided vehicles (AGVs) is essential for efficient horizontal transportation at automated container terminals. Effective planning of AGV transportation can reduce equipment energy consumption and shorten task completion time. Multiple AGVs transport containers between storage blocks and vessels, which can be regarded as the supply sides and demand points of containers. To meet the requirements of shipment in terms of timely and high-efficient delivery, multiple AGVs should be dispatched to deliver containers, which includes assigning tasks and selecting paths. A contract net protocol (CNP) is employed for task assignment in a multiagent system, while machine learning provides a logical alternative, such as Q-learning (QL), for complex path planning. In this study, mathematical models for multi-AGV dispatching are established, and a QL-CNP algorithm is proposed to tackle the multi-AGV dispatching problem (MADP). The distribution of traffic load is balanced for multiple AGVs performing tasks in the road network. The proposed model is validated using a Gurobi solver with a small experiment. Then, QL-CNP is used to conduct experiments with different sizes. The other algorithms, including Dijkstra, GA, and PSO, are also compared with the QL-CNP algorithm. The experimental results demonstrate the superiority of the proposed QL-CNP when addressing the MADP.

Keywords: AGV dispatching; distribution balance; energy consumption; Q-learning; multiagent

Citation: Gao, Y.; Chen, C.-H.; Chang, D. A Machine Learning-Based Approach for Multi-AGV Dispatching at Automated Container Terminals. *J. Mar. Sci. Eng.* **2023**, *11*, 1407. https://doi.org/10.3390/jmse11071407

Academic Editor: Kamal Djidjeli

Received: 18 May 2023
Revised: 7 July 2023
Accepted: 7 July 2023
Published: 13 July 2023

1. Introduction

The operational efficiency of container terminals has always been a popular research issue, especially in the context of the widespread utilization of automation technology. The berth time is the key indicator to evaluate the efficiency, and containers to be shipped are required to be loaded as soon as possible. The automated guided vehicle (AGV) is the horizontal transportation equipment at an automated container terminal (ACT), which is responsible for delivering containers to the quay crane (QC). The berth time of ships can be reduced by optimizing the transportation time of container tasks. If AGVs are well organized to transport container tasks, the energy consumption of AGVs can be better utilized. When assigning tasks to AGVs, the shortest path is primarily considered. Soylu et al. [1] attempted to find the shortest path for the single AGV when conducting the pickup and delivery tasks. Yang et al. [2] also regarded the AGV route planning to be the shortest path problem, in which the speed was considered constant. Another essential factor in previous AGV path research was collision. There is a conflict between two or more AGVs when one's planned path clashes with another's. Accordingly, some strategies were proposed to deal with the collision or a path was even preplanned for AGVs to prevent it. Nishi et al. [3] restricted two or more AGVs from passing the node at the same time when optimizing the production scheduling and routing. The sequence and time window of processing all jobs were determined under a hierarchical control architecture to generate the

collision-free trajectories of AGVs, as proposed by Xin et al. [4]. Moreover, Zhong et al. [5] analyzed the impact of AGV strategy on path planning and constructed the mathematical model to prevent conflict and deadlock for multi-AGV scheduling.

Using more AGVs can promote task completion progress to some extent. If more AGVs are assigned to transport container tasks, the makespan may decrease while equipment consumption will increase. However, when the number of AGVs reaches an extreme value, the operation efficiency would decrease. Yue et al. [6] optimized the configuration of QCs and AGVs to minimize the makespan of all tasks, and then analyzed the inefficient time of different AGVs. Although utilizing more AGVs can help to reduce the makespan, it may result in the inefficiency of AGVs and lengthen the delay time of QCs. In addition, an increase in the number of AGVs brings a high possibility of congestion and conflict. The relationship among the number of vehicles, the speed of vehicles, and the container throughput were investigated by Roy et al. [7], who claimed that the effective vehicle speed steeply decreased and congestion occurred when a certain number of vehicles was exceeded on the travel path at a container terminal. They also suggested that reasonably allocating the traffic flow on paths can be regarded as a feasible solution to congestion. Salman and Alaswad [8] applied the Markov chain to traffic assignment and used vehicle density as the optimization object to alleviate congestion. The flow-paths were described as a square topology of AGV traffic network, and the control algorithm based on the chains of elementary reservations was proposed to prevent collision and deadlock by Małopolski [9]. Therefore, AGVs should be well assigned to optimize the equipment configuration and consumption.

This study proposes the dispatch of multiple AGVs for container transportation by balancing the traffic flow between the storage yard and QC. The storage yard is regarded as the supply side due to the storage area for containers to be shipped, while the QC serves as the demand point to receive containers and then load containers on the vessel. AGVs, as the carriers of containers, have various transportation routes between the supply side and demand point that can form a traffic network. The first step is to assign an AGV to a QC considering the completion of all tasks with the least time and shortest distance. It should be noted that the same route in the traffic network may be chosen by multi-AGV many times. The occurrence of multiple AGVs on a path not only adds to the transportation time of AGVs, but also increases the probability of congestion and conflicts. The next step is to allocate the distribution of AGVs in the traffic network. The balance of transportation load between the supply side and demand point is an adaptable approach to effectively address multi-AGV dispatching from the origin of congestion and conflict. If the traffic load remains balanced, the possibility of congestion will be decreased. In a nutshell, the multi-AGV dispatching problem (MADP) can be seen as multiagent task allocation, in which an AGV is taken as an agent for receiving and executing tasks. Multiple agents can communicate, coordinate, and collaborate with each other in a multiagent system. Moreover, the collaboration between agents can be established through a negotiation protocol, such as the well-known contract network protocol (CNP). Agents can share resources and cooperate to achieve global goals on the basis of a negotiation protocol in the network. Li et al. [10] studied the scheduling problem of manufacturing resources, in which a multiagent system was developed to address the problem using the CNP. The system allowed internal negotiation among subsystem agents and external sharing among intermediary agents, and then introduced an optimization algorithm to independently schedule jobs for agents. Framiñán Torres et al. [11] also compared the performance of different scheduling rules using the CNP in the job shop. The decentralized rules were demonstrated to have a high potential in job shop scheduling, and the CNP could be integrated with machine learning methods in a multiagent system to optimize complex decisions.

In this study, all tasks of transporting containers are required to be allocated to multiple AGVs in an effective and balanced way, and the CNP can be applied to perform task allocation in a multi-AGV system. AGVs are considered bidders to select container tasks

and transportation paths by bidding according to the CNP. However, numerous path nodes in the traffic network add difficulty and complexity for multiple AGVs to make decisions. Additionally, a machine learning method is used to improve the CNP, in which Q-learning can help multiple agents to choose better actions during the bid evaluation. Q-learning is employed to update the reward of selected paths at different states for multiple AGVs, which is beneficial to obtain the action that can get the most value.

The main efforts of this study are summarized below. Firstly, the relationship among the storage yard, QC, and AGV is regarded as a supply chain. Correspondingly, the AGV is taken as the carrier, and the container blocks and QCs are respectively taken as the supply side and demand point according to supply chain theory. The transportation routes of multiple AGVs constitute a traffic network between the supply side and demand point. Then, the MADP is decomposed into assigning AGVs to QCs and allocating AGV path distribution, thus formulating a two-stage mathematical model. The delay time of all tasks is optimized in the first stage when assigning multi-AGV transportation. The shortest path is prioritized in order to transport the containers to be shipped as soon as possible. To avoid congestion and collision, the transportation load of various routes is balanced between the supply side and demand point. The congestion rate is introduced to describe the condition of traffic load in the transportation network of multiple AGVs, which refers to the rate of the waiting time and transportation time. In addition, a solution algorithm called QL-CNP is designed which combines QL and CNP, as numerous path nodes in the traffic network increase the complexity of model solving. The CNP has a potential to address task allocation in a multiagent system through negotiation protocols among agents. Moreover, the CNP is improved through its combination with QL, in which the bidder of choosing paths is evaluated for multiple AGVs. Experiments of different sizes are also conducted to verify the established models and the QL-CNP algorithm, demonstrating the effectiveness of the proposed approach.

The remainder of this paper is organized as follows: Section 2 reviews the literature related to AGV dispatching and machine learning methods for AGV routing; a problem description and mathematical models are given in Section 3; Section 4 illustrates the designed solutions; experimental results are provided in Section 5; conclusions of this study and recommendations for further research are presented in Section 6.

2. Literature Review

As this study focuses on the MADP at an ACT and applies a machine learning method to optimize the task allocation and path planning of multiple AGVs, the relevant research on scheduling problems and machine learning methods for optimization are discussed below.

2.1. Scheduling Problem Research

AGV is commonly used for horizontal transportation at container terminals, workshops, and other factories. Task allocation and path planning of AGVs were studied to improve efficiency. Singh et al. [12] considered the battery constraints when scheduling heterogeneous AGVs. Accordingly, a metaheuristic-based adaptive large neighborhood search algorithm was proposed to assign transportation requests and process sequences, as well as other task information, with the minimum travel and tardiness costs. Yu et al. [13] focused on multi-AGV online scheduling and path planning in automated sorting warehouses and introduced the grid blocking degree to the path planning algorithm. Through the improved algorithm, six priority rules were designed and further improved to develop conflict-free paths for multiple AGVs. A traffic control system was developed with emergency traffic control schemes to prevent collisions and deadlocks among AGVs, which was simulated in the automated container terminal to evaluate its performance by Li et al. [14], whose results showed that each AGV could transport without colliding with any other vehicle. To plan the conflict-free path for AGVs, Zhong et al. [15] designed a priority-based speed control strategy. The strategy was also combined with the Dijkstra depth-first search algorithm to solve the model whose objective was the shortest distances of AGVs between QCs and

yard cranes (YCs). Li et al. [16] studied the conflict-free path problem of AGVs used in large warehouses, which applied a branch-and-bound algorithm to minimize energy consumption when completing all transportation tasks. Xin et al. [17] established a time–space network to address the AGV path planning where the multiple vehicle pickup and delivery (MVPD) problem was described as a multicommodity network flow problem. A receding horizon planning (RHP) methodology was proposed to obtain the paths with the minimal total arrival times of all AGVs in the network. Furthermore, Xin et al. [18] considered energy consumption and conflicts in the path planning of AGVs, where a mixed-integer nonlinear programming and a hybrid metaheuristic algorithm were developed to solve the problem.

There were studies which integrated AGV scheduling with other resources at container terminals. Luo and Wu [19] integrated AGV dispatching and container storage allocation, and then applied the genetic algorithm (GA) to solve it. However, they assumed that there was no traffic congestion for AGVs on the path. Yang et al. [20] studied the cooperative scheduling of an automated rail-mounted gantry (ARMG) and AGV in the sea-rail operation environment and designed a self-adaptive chaotic genetic algorithm to solve the problem. A bilevel programming model of scheduling AGVs, QCs, and automated stacking cranes (ASCs) was established considering AGV conflicts by Ji et al. [21]. Then, an approach based on the conflict resolution strategy and the bilevel adaptive genetic algorithm was verified in both small-size and large-size experiments. GA was typically used as an alternative algorithm in optimization problems.

2.2. Machine Learning Methods for Optimization

As traditional algorithms are not available for tackling combinatorial optimization problems that require much computational time when solving larger-scale cases, deep reinforcement learning has been investigated and reviewed on how to address optimization problems [22]. Gumuskaya et al. [23] combined machine learning and operation research to determine barge plan under uncertain scenarios. A decision tree was used to predict the arrival of containers, and the sample average approximation was proposed to make barge call decisions based on the prediction. Peng et al. [24] studied ship energy consumption using machine learning methods, and analyzed the important characteristics of reducing ship energy consumption. The experimental results showed that improving the efficiency of facilities could effectively reduce the berthing energy consumption of ships. Gao et al. [25] applied the Q-learning algorithm to optimize the operating strategies of ASCs using digital twin, which was compared with GA and other solution algorithms. The solutions with a digital twin-based Q-learning approach achieved optimal energy consumption. For the scheduling problem, Kintsakis et al. [26] proposed a sequence-to-sequence neural network architecture and used reinforcement learning algorithm to train the model. In this architecture, each alternative scheduling decision was entered into the policy network for evaluating the reward function; then, an optimal decision with the lowest makespan was selected for a workflow management system. Kim et al. [27] proposed a reinforcement learning approach to solve the production scheduling of precast concrete because the metaheuristic approach needed more computational time on large-size problems. The proposed reinforcement learning approach had fast computation time and good performance in solving various sizes of problems. In the real-time charging scheduling of AGVs, Lin et al. [28] developed a Markov decision model to fulfill uncertain requests. Then, a feature-based reinforcement learning approach was improved to obtain charging strategies that optimized utilization time and charging cost of AGVs. Cals et al. [29] solved the batching problem by deep reinforcement learning (DRL) where the learning agents studied and compared different combinations of picking periods and picking methods to minimize the number of tardy orders.

Machine learning was also applied to address the vehicle routing problem (VRP). An online routing strategy based on DRL was proposed to solve VRP, which could develop routes with minimal time [30]. Ahamed et al. [31] solved the crowd shipping problem

using deep Q learning (DQN), which referred to allocating transportation requests between different locations with a limited time window and the lowest cost. All routing sequences allocated to the requests were presented for selection based on heuristic rules, which improved the training efficiency of DQN. For the multi-vehicle routing problem with soft time windows (MVRPSTW) problems, Zhang et al. [32] proposed a novel reinforcement learning architecture, called the multiagent attention model (MAAM), which utilized a DRL strategy to find feasible routes for vehicles. Jelen et al. [33] developed a multiagent system to plan paths for electric vehicle fleets using machine methods, which included charging station agents and depot agents. Then, a multiagent simulation was performed to verify routing of electric vehicle fleet. To adapt to the changing environment in an ACT, Choe et al. [34] designed an online preference learning algorithm to dispatch AGVs. The preference function was used to evaluate each dispatch decision, and the old samples were replaced to achieve online learning. To plan the anti-conflict path of AGVs, Hu et al. [35] proposed the multiagent deep deterministic policy gradient (MADDPG) method which introduced reinforcement learning.

2.3. Research on Contract Net Protocol

The contract net protocol (CNP) is a task-sharing protocol in multiagent systems [36]. It has been applied to resource allocation and scheduling of multiple agents in different domains. In a diagnostic services scheduling system, Gao et al. [37] designed CNP and simulated annealing-based metaheuristics to negotiate patient selection and preference scheduling, which effectively improve the scheduling performance. Jégou et al. [38] solved the reactive hoist scheduling problem using the auction mechanisms of CNP. Cardon and Vacher [39] combined CNP and GA to address job-shop scheduling problems where each agent has its own rules and negotiates with each other for a better solution. Liang and Kang [40] improved the CNP to realize task optimal allocation for modeling an agent-oriented unmanned underwater vehicle swarm system. To handle the load balance of unmanned aerial vehicle swarm, two improved CNP assignment algorithms were proposed for the one-to-many and many-to-one modes [41]. It can be found that CNP was applied for task allocation and scheduling as it could realize the reaction and negotiation among agents through bidding and rating. Therefore, CNP was exploited to allocate multiple AGVs for balancing the transportation load between supply sides and demand points.

To summarize, Table 1 compares the problems and solutions of AGV optimization. It can be seen from Table 1 that heuristic algorithms were mostly used to solve AGV scheduling problems, and most studies considered path planning with conflicts. In this study, the path distribution of AGVs is taken into consideration, and machine learning and CNP are combined as a novel approach.

Table 1. Comparison of AGV optimization in problems and solutions.

Problems	Solutions	Authors
AGV scheduling considering battery to minimize travel and tardiness costs	A metaheuristic-based adaptive large neighborhood search algorithm	Singh et al. [12]
Multi-AGV online scheduling in warehouses	Improved path planning algorithm with grid blocking degree	Yu et al. [13]
AGV conflict-free path planning to minimize distances	Dijkstra depth-first search algorithm	Zhong et al. [5]
AGV conflict-free path planning to minimize energy consumption	A branch and bound algorithm	Li et al. [16]
AGV path planning in MVPD problem	A receding horizon planning methodology	Xin et al. [17]
AGV dispatching and container storage allocation	GA	Luo & Wu [19]
Integrated scheduling considering AGV conflicts	An approach based on the conflict resolution strategy and bilevel adaptive GA	Ji et al. [21]
AGV online dispatching	An online preference learning algorithm	Choe et al. [34]
AGV anti-conflict path at ACT	MADDPG combining reinforcement learning	Hu et al. [35]

3. Problem Description and Model Formulation

3.1. Problem Description

Containers are stacked in the scattered container blocks of the storage yard before being loaded on the vessels. Each vessel has a few QCs to load these containers. The layout of the automated container terminal is provided in Figure 1. The number of the arranged QCs, which may be two, three, or even more, depends on the number of containers to be shipped. AGV is responsible for transporting containers from the stored container blocks to the designated QC. There are various routes between container blocks and QCs which are selected by AGVs. To ensure the vessel departs as soon as possible, each AGV prefers the shortest path to complete the transportation. However, two or more AGVs may travel on the same path and meet at the intersection, causing congestion and collision. Hence, AGVs need to be well dispatched to QCs when assigning the containers to be loaded on the vessel.

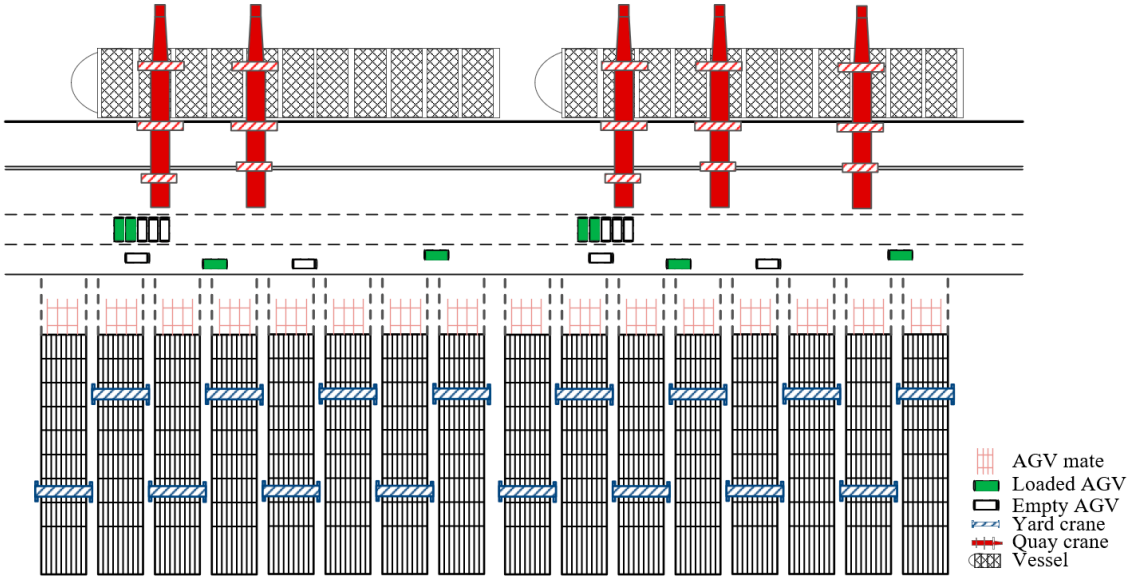

Figure 1. The layout of the automated container terminal.

From the perspective of supply chain, the storage yard supplies the containers that are required by the vessel, while AGVs are chosen to be the carriers. The total demand is equal to the supply as all containers to be shipped need to be transported from their stored blocks. Each QC assigned to the vessel is known as one demand point, and each block with stacked containers is considered as a supply side. However, the number of containers handled by each QC is not equal, which means that the number of demand containers is different among all demand points, as well as all supply sides. Containers that will be transported to a certain demand point are stored at different supply sides, and an AGV is dispatched to different demand points at various timepoints. The path network between demand points and supply sides is shown in Figure 2. To avoid congestion and collision, this study aims to balance the allocation of AGVs in each path. The reason for congestion is that many AGVs are densely distributed on the same route. Meanwhile, conflicts arise when two AGVs need to pass through the same intersection, which requires assigning AGVs to the appropriate routes.

This study focuses on the AGV assignment for transporting the containers to be shipped under the context of balancing the traffic load between supply sides and demand points. An AGV transporting a container from the supply side to the demand point is considered as one task, and containers to be shipped are all bound to be delivered. Firstly, AGVs are dispatched to a certain QC to complete all tasks with the least delay time. Then, the appropriate path is assigned for AGVs to minimize the congestion rate of the road

network, which means to balance the traffic load distribution in various routes. The mathematical models are formulated below.

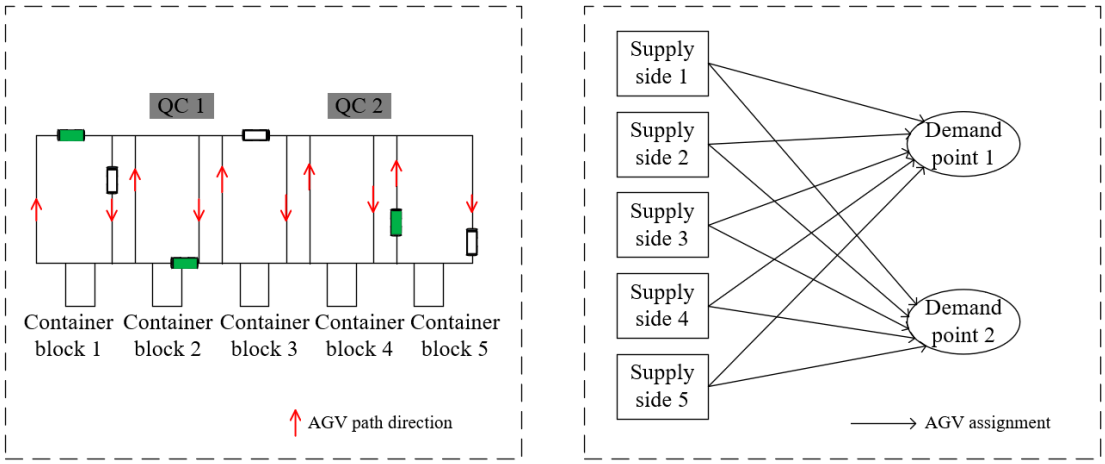

Figure 2. The path network between demand points and supply sides.

3.2. Model Formulation

In this study, multiple AGVs are responsible for transporting containers from the stored container blocks to QCs serving one vessel, and each AGV returns to the next target block for the next task assignment. Therefore, AGV has two travels including the loaded travel to the QC of the current task and the unloaded travel from the current QC to the container block of the next task. Some assumptions are given below.

(1) There are virtual locations which are the starting points of AGVs. This means that AGVs staye at the starting locations before being assigned. When AGVs complete their tasks, they return to the starting locations.

(2) The task information of containers is known. Containers to be shipped are delivered to the storage yard, and the detailed container blocks are also allocated for storage.

(3) AGV has enough battery to transport containers continuously. The new technology of charging allows the AGV to replace the whole lithium battery, while the configuration capacity of the lithium battery meets the demand for the continuous operation of ACT.

(4) To balance the workload of all AGVs, each AGV has its own transportation capacity. Then, tasks are arranged fairly to AGVs, instead of one AGV taking on all tasks. The transportation capacity also refers to the travel time limit of AGV.

At the first stage, the mathematical model is established describing how to dispatch multiple AGVs to transport tasks. The symbols involved in the first model are described in Table 2.

$$f = \min \sum_{i \in N} (et_i + lt_i). \tag{1}$$

$$\sum_{k \in M} x_{ik} = 1, \forall i \in N. \tag{2}$$

$$\sum_{j \in N} y_{ijk} = 1, \forall i \in N, k \in M. \tag{3}$$

$$\sum_{j \in N} y_{ijk} = x_{ik}, \forall i \in N, k \in M. \tag{4}$$

$$\sum_{i \in N} y_{q_i b_j k} = x_{jk}, \forall j \in N, k \in M. \tag{5}$$

$$\sum_{b_i \in L} y_{o_k b_i k} = z_k, \forall k \in M. \tag{6}$$

$$\sum_{q_j \in L} y_{q_j o_k k} = z_k, \forall k \in M. \tag{7}$$

$$\sum_{i \in N} x_{ik} \cdot t_i + \sum_{i \in N} \sum_{j \in N} y_{ijk} \cdot t_{q_i b_j} \leq C_k \cdot z_k, \forall k \in M. \tag{8}$$

$$st_i + t_i + t_{q_i b_j} \leq st_j + G(3 - y_{ijk} - x_{ik} - x_{jk}), \forall i, j \in N, k \in M. \tag{9}$$

$$st_j + t_j + t_{q_j b_i} \leq st_i + G(2 - x_{ik} - x_{jk} + y_{ijk}), \forall i, j \in N, k \in M. \tag{10}$$

$$st_i \geq ET_i - et_i, \forall i \in N. \tag{11}$$

$$st_i \leq lt_i + LT_i, \forall i \in N. \tag{12}$$

Table 2. Symbol definitions of the first model.

Sets and indices	N	Set of container tasks, $i, j \in N = \{1, 2, \ldots, n\}$
	B	Set of container blocks, $b_i \in B = \{b_1, b_2, \ldots, b_n\}$
	Q	Set of QCs, $q_i \in Q = \{q_1, q_2, \ldots, q_n\}$
	M	Set of AGVs, $k \in M = \{1, 2, \ldots, m\}$
	O	Set of virtual locations that AGVs start, $o \in O = \{o_1, o_2, \ldots, o_k,\}$
	L	Set of locations that AGVs pass, $l \in L = \{o_1, o_2, \ldots, o_k, b_1, b_2, \ldots, b_n, q_1, q_2, \ldots, q_n\}$
Parameters	t_{lp}	The transportation time between location l and location p, $l, p \in L$
	t_i	Transportation time of task i
	ET_i	Earliest time of task i in the time window
	LT_i	Latest time of task i in the time window
	C_k	Transportation capacity of AGV k
	G	The giant number
Decision variables	x_{ik}	If task i is assigned to AGV k, $x_{ik} = 1$; otherwise, $x_{ik} = 0$
	y_{ijk}	If AGV k completes task i and then is dispatched to transport task j, $y_{ijk} = 1$; otherwise, $y_{ijk} = 0$
	z_k	If AGV k is dispatched to the task, $z_k = 1$; otherwise, $z_k = 0$;
	st_i	Start time of task i
	et_i	Time difference of the earliest time of task i
	lt_i	Time difference of the latest time of task i

The objective function is to minimize the delay time of all tasks in Equation (1). Constraints (2) and (3) guarantee that one task can only be transported by one AGV. Constraint (4) represents that, when the AGV is dispatched to a task, it must transport from the block to QC of the task. Constraint (5) ensures that the AGV travels from the QC of the current task to the block of the next task. Constraints (6) and (7) imply that the AGV travels from the start location and returns to the start location. It is noted that the variable y_{ijk} means that the AGV transfers from task i to task j, which also refers to the change in AGV status. Hence, the variables used in Constraints (5)–(7) represent the state transitions of AGVs and their tasks. Constraint (8) means that the AGV works within its own transport

capacity. Constraints (9) and (10) restrict the sequence of two adjacent tasks which are assigned to the same AGV. Only when task *i,j* are assigned to AGV and task *i* is transported before task *j* can Constraint (9) stand. The stand condition of Constraint (10) is the opposite of that of Constraint (9), referring to task *j* being transported before task *i*. Constraints (11) and (12) define the time difference of earliest time and latest time.

According to the solution of the first stage, multiple AGVs are dispatched to transport their own tasks. The AGV drives between the supply side and the demand point as the storage location and destination of each task are provided. Then, the driving path of multiple AGVs is further planned to balance the traffic load of the multi-AGV network at the second stage. When two AGVs have the need to pass through the same path, a collision will. As a result, one AGV needs to wait for the other AGV to pass. Then, the congestion rate is used to describe the condition of traffic load during the transportation of multiple AGVs. A low congestion rate shows that the path planned for the AGV is appropriate for the task, and that the traffic load is balanced in the road network. The second model with the goal of minimizing the congestion rate is formulated below. The sets of tasks and AGVs are the same as the first model. The other parameters and variables are given in Table 3.

$$f = \min \frac{t}{T}. \tag{13}$$

$$t = \sum_{p \in P} \sum_{i \in N} \sum_{k,k\prime \in M} x_{pik} \cdot y_{pkk\prime} \cdot (t_1 + t_2). \tag{14}$$

$$T = \frac{\sum_{p \in P} \sum_{i \in N} \sum_{k \in M} x_{pik} \cdot w_p}{v}. \tag{15}$$

$$\sum_{p \in P} x_{pik} = 1, \forall i \in N, k \in M. \tag{16}$$

$$\sum_{p \in P} y_{pkk\prime} = 1, \forall k, k\prime \in M. \tag{17}$$

$$\sum_{k \in K} y_{pkk\prime} \leq 1, \forall p \in P, k\prime \in M. \tag{18}$$

$$\sum_{k\prime \in K} y_{pkk\prime} \leq 1, \forall p \in P, k \in M. \tag{19}$$

$$x_{pik} = \{0, 1\}, p \in P, i \in N, k \in M. \tag{20}$$

$$y_{pkk\prime} = \{0, 1\}, p \in P, k, k\prime \in M. \tag{21}$$

Table 3. The meanings of parameters and variables in the second model.

Parameters	D	Set of path Nodes, $m, n \in D$
	P	Set of paths formed by two nodes, $p \in P$
	a	The acceleration of AGV
	t_1, t_2	The deceleration time and the acceleration time of AGV
	d_s	The safe distance of AGV
	w_p	The distance of the path p formed by two nodes
Variables	x_{pik}	AGV k transporting the task i chooses the path p
	$y_{pkk\prime}$	AGV k and k' meets at the path p
	t	The waiting time of AGV during the transportation
	T	The total transportation time of AGV passing paths

Equation (13) describes the congestion rate, i.e., the rate of the waiting time and transportation time. Equation (14) defines the waiting time of AGV as meeting the conflicts when transporting tasks. Equation (15) refers to the total time of AGV passing all paths for the transportation tasks. Equation (16) limits each AGV to only choosing one path to transport one task at one time. Equation (17) represents that two AGVs meet on the same path. Equations (18) and (19) restrict that at most one AGV will meet with another AGV on the same path due to the one-way traffic rule in the road network. Equations (20) and (21) are the value ranges of the variables.

4. Approach Design

The concept of a multiagent system is introduced to dispatch AGVs for transportation tasks at ACT. CNP is an essential approach to perform task allocation that is achieved by bidding [42]. The bidding flow of CNP is provided in Figure 3a. In a multiagent system, there is one manager agent who sends requirements and evaluates bids according to the CNP. The remaining agents are bidders who have the freedom to submit a tender and to accept the bid. In this study, classic CNP is combined with QL to assign path requirements for multiple AGVs, called QL-CNP, as shown in Figure 3b. When AGVs receive the task requirements, they submit the proposals to obtain the corresponding task. However, each AGV has many path choices to execute the transportation task, which adds the complexity of evaluating bids. The manager agent needs to consider the total performance of all bidders, including matching the bidder with the task and the path for bidders. The state, environment, and choice of agent who sends the task proposal are all the elements for the manager to assess the bid. Only when all the conditions are considered comprehensively can the manager agent decide who is the winning bidder. Therefore, QL is applied to evaluate actions of agents so as to choose the best action for the bidder. The descriptions of agent, state, action, and reward function in the proposed QL-CNP are shown below.

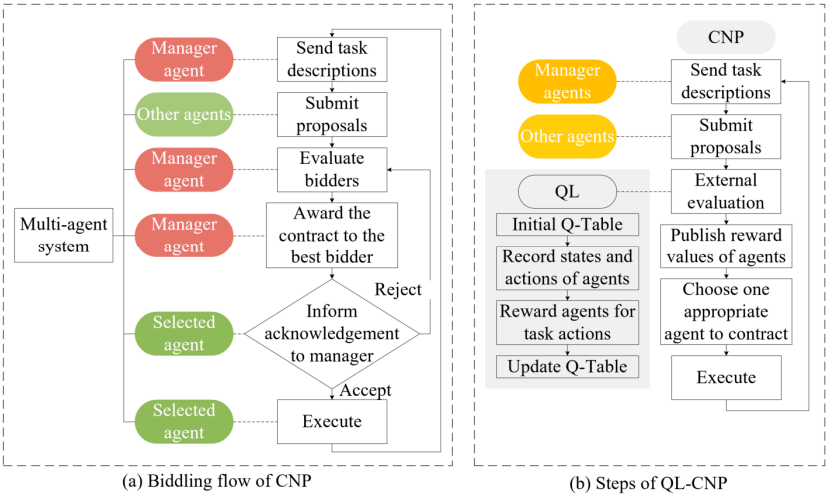

Figure 3. The algorithm steps QL-CNP.

4.1. Agent

Each AGV in the road network is considered an agent. In the path network (see Figure 4), all AGVs are the same in driving speed and transportation preference. At the ACT, AGVs are responsible for transporting containers from the container block to the QC. One task involves one container stacked in the container block being transported by an AGV to the QC. When one container has been transported to the QC by an AGV, the AGV is released and free to drive toward the container block for the next task. There are many tasks in the multi-AGV system, and the AGV can choose any task for execution. Edges composed of different nodes form the path network of AGV transportation. However, the traffic load has to be considered in the path network for transporting quickly and safely. The state of AGVs directly affects transportation efficiency. If one AGV is surrounded by other AGVs on the edges, it would restrict driving, and conflicts may occur. Therefore, the AGV needs to choose one edge which is not relatively busy in the path network for the transportation task in advance.

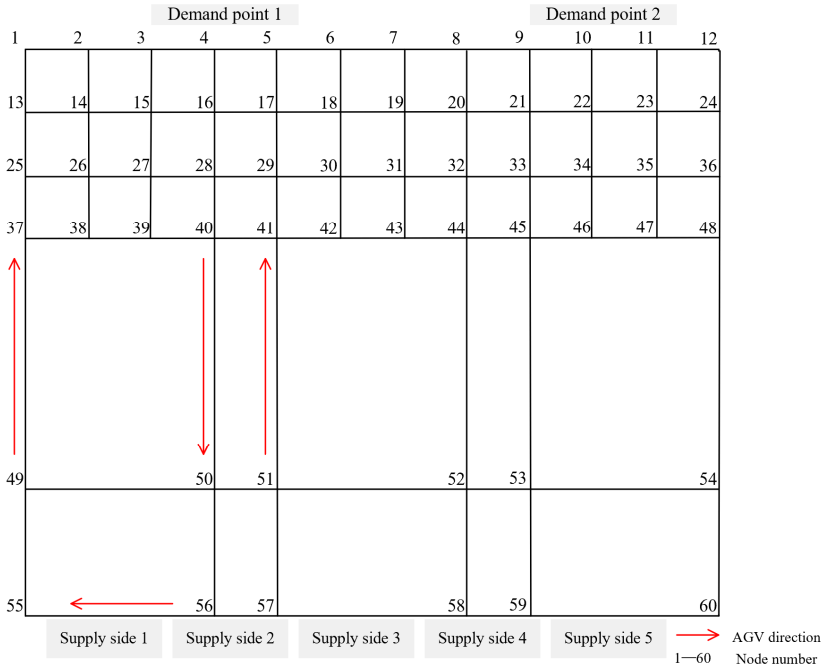

Figure 4. The road network of AGVs.

4.2. State

When AGV chooses one path to transport the current task, the traffic load of the selected path will be added. The environment of AGV transforms from being in the starting point to being on edge, which can be defined as

$$s_k = \left\{ s_k^n, p_k^n, t_k^p \right\}, \tag{22}$$

where s_k^n denotes that agent k performs the task n, and p_k^n is the path chosen by the agent. The time of agent in the path is shown as t_k^p.

Once the task starts, AGV needs to pass through various paths to reach the destinated QC. The process of the task can be expressed as

$$s_n = \left\{ m, p_1, p_2, \ldots, p_m, C_{p_m} \right\}, \tag{23}$$

where the state of the task includes the number of paths passing m, the path chosen by agent p_m, and the remaining capacity of the path C_{p_m}.

4.3. Action

Before performing any task, the AGV needs to choose which path to maximize efficiency. Starting from the supply point where the task is located, the AGV should consider each node to go to the next step until reaching the destinated demand point. Then, the edge to be chosen by the task is expressed as

$$a_{kn}^p = \{0, 1\}, \tag{24}$$

$$p_m = p_j^i, \text{ nodes } i, j = \{1, 2, \ldots, 86\} \text{ formulates the path } p_m, \tag{25}$$

where $a_{kn}^p = 1$ means that the agent chooses the edge formulated with node i and j to transport the task.

When the agent takes an action, the ε-greedy strategy is used. The parameter ε is set to 0.9, referring to [43], which means to choose the random action with a probability of 0.1 and the optimal action with a probability of 0.9.

4.4. Reward Function

Reward refers to the feedback given by the environment and the action that the agent performs. The reward setting is for agents to choose better actions. A Q-table is composed of states, actions, and corresponding Q-values, which is used to select the action that can get the most rewards on the basis of the Q-value. It is also the main idea of Q-learning.

To evaluate the action of agent, the reward function is defined as

$$r_{nk} = \alpha \cdot \sum_m C_{P_m} \cdot a_{kn}^p - \beta \cdot m_n, \tag{26}$$

where C_{P_m} is the remaining capacity of the path, and m_n is the number of paths that are passed by the agent. The parameters α and β are the reward coefficients for the capacity and number of paths. The path also refers to the edge that is traversed by the AGV. Two nodes formulate an edge which is the alternative path. When more edges are chosen by an agent, the agent can get fewer rewards. The more edges the AGV passes, the farther it will travel. Less reward means that the agent is not recommended to perform such an action. In addition, greater rewards can be received by the agent if there is more remaining capacity for AGVs. The distribution of traffic load in the path network can be controlled by encouraging the remaining capacity.

4.5. Algorithm Implementation

The proposed QL-CNP algorithm combines QL and CNP to assign tasks to multiple AGVs and plan paths for task transportation. On the basis of the CNP, the states of agents can be observed and known by the manager in a multiagent system. The manager is responsible for allocating tasks to agents according to their states and considering whether they can execute tasks or not. The main idea of QL-CNP is to enter the state of agents into QL and estimate the maximum reward that can be obtained for each action taken in the state. QL is used to help agents to choose better actions in various states, and then the states, actions, and corresponding rewards are stored in the Q-table. The algorithm of QL-CNP is described below.

(1) Initialize the Q-table recording the state, action, reward, and agent;
(2) Observe the environment of multiagent system, including states and agents;
(3) Release the transportation tasks by the manger agent;
(4) Submit the requirement proposal by agents;
(5) Choose an action for the agent;
(6) Calculate the reward of the state and action;
(7) Update the Q-table;
(8) Announce the results of the bid by the manager agent;
(9) Contract the protocol between the manager and bidder agent;
(10) Cycle in turn until all task assignments end.

The QL-CNP algorithm can be implemented by pseudo-procedures which contain three main parts: establishing the environment, updating the Q-table, and finding the optimal action. The reward function for choosing action at a state is expressed as R(s, a), and the Q-value is the expected value of accumulated reward received. On the basis of Bellman's equation, the Q-value can be described by Equation (27), where α and γ respectively represent the learning rate and discount factor. The pseudocode is shown in Table 4.

$$Q^{new}(s,a) \leftarrow Q(s,a) + \alpha \cdot (R(s,a) + \gamma \cdot \max Q(s',a') - Q(s,a)). \quad (27)$$

Table 4. The pseudocode of QL-CNP algorithm.

Initialize the environment for agents transporting Create a new m × n-dimensional environment matrix Construct a reward matrix Set parameters of QL
For episode = 1 to M
Obtain the state of tasks and agents Select all possible actions of the state Calculate Q-value of all possible actions Select one action and take it as the next state Find the best Q-value which is the best state for the next action Update the Q-table
If to the end Break; End if
End for

The AGV attempts to find the optimal path for its task transportation in the road network, as shown in Figure 5. The starting point and destination point (S and D in Figure 5) are known by the AGV, and the bar in the road matrix indicates that the way cannot be passed. Various paths from S to D are tried to evaluate the total reward. Finally, an optimal path is provided for AGV to execute the task.

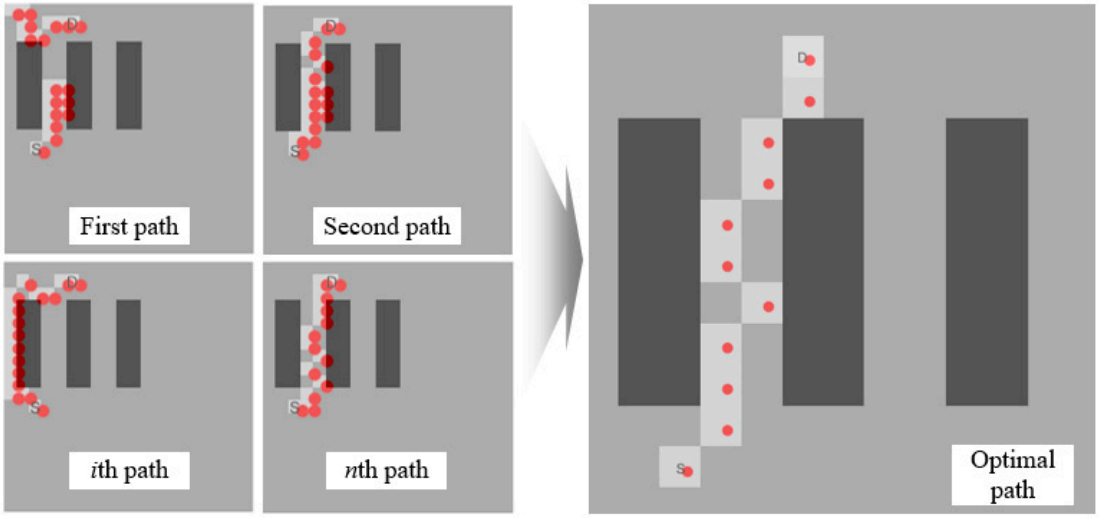

Figure 5. The process of finding paths by QL-CNP.

5. Case Study

5.1. Experiment Design

The tasks of containers loading to one vessel are taken as an example to verify the established model and proposed algorithm. There are five container blocks and two QCs considered in the example where the proportion of equipment is set with 1:2.5. Task information in this case is incomplete but realistic due to factors such as storage location and destination. Some data samples of the tasks are provided in Appendix A.

The QL algorithm is trained before being used to address the MADP, including determining the hyperparameters. The update of Q-value is related to the reward function, learning rate, and discount factor according to the above Bellman's equation. In the training experiment, a 5 × 5 grid is chosen from the road networks of AGVs to be the environment. The agent needs to choose an optimal path from the start location to the goal location. The comparison experiments of different learning rates and discount factors are conducted with 1000 episodes, as shown in Figure 6.

The learning rate ranges from 0 to 1, where $\alpha = 0.5$ means a moderate value for the learning. According to Bellman's equation, a larger learning rate denotes a lower effect of retaining the previous training. Then, two small learning rates are compared with the moderate learning rates, as shown in Figure 6a. When the learning rate is 0.1, its convergence performance is better than that of the other two learning rates. In addition, the trend of the learning rate (0.1) gradually becomes stable, while the other two show obvious fluctuation. Similarly, the discount factor is in the range of $0 \leq \gamma < 1$, where a larger value means more attention to the training. However, the discount factor of 0.8 has the best performance among the three results presented in Figure 6b, mainly because its fluctuation is minimal during the later episodes. Therefore, the learning rate and discount factor are respectively set to 0.1 and 0.8 in this case study.

In addition, the path from the start to the goal in a 5 × 5 grid is provided in Figure 7. During the process, the agent has four action selections: up, down, left, and right. The agent can get the reward when choosing an action at one state. The Q-table recording Q-values is an m × n table, where there are m states and n actions. Some Q-values when finding the optimal path in the grid are provided in Table 5.

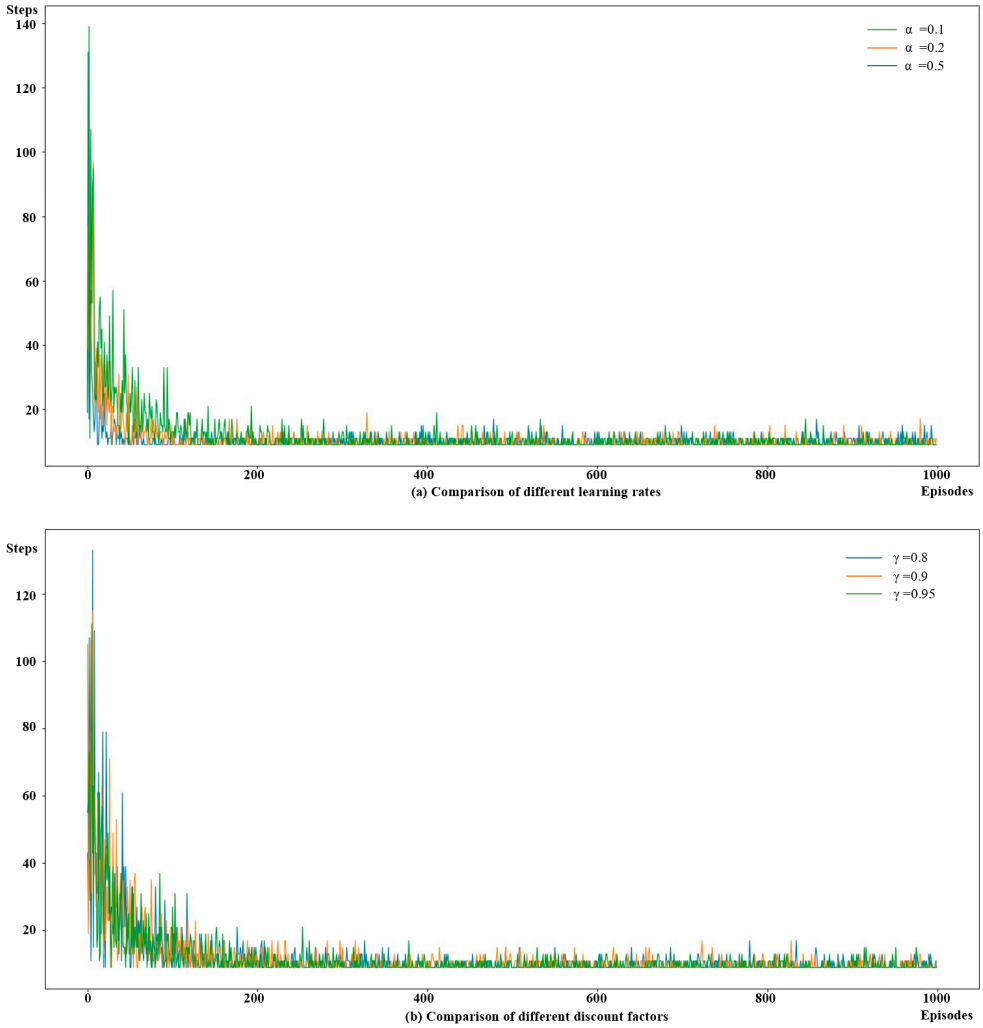

Figure 6. The training results of different parameter values.

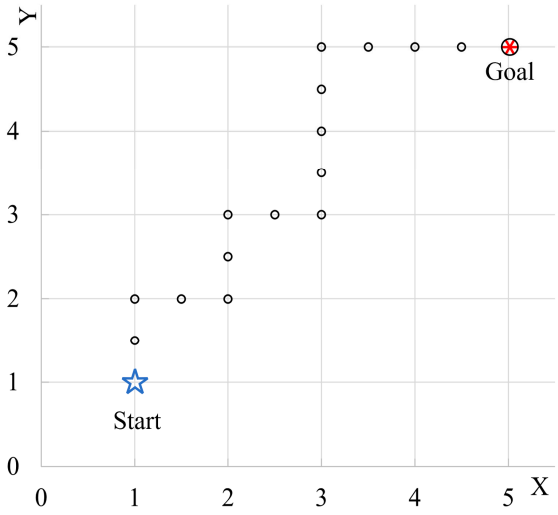

Figure 7. The path-finding process from the start to the goal location.

Table 5. Part Q-values of the state and action.

State \ Action	1	2	3	4
1	0.00	0.00	0.00	0.00
2	0.86	0.35	0.46	0.86
3	0.96	0.28	0.14	0.90
4	0.61	0.40	0.10	0.79
5	0.81	0.08	0.03	0.94

The mathematical models and QL-CNP algorithm are proposed to solve the MADP in this study. The approach is implemented in Python with an Intel Core i7-11800H @ 2.30 GHz system. The corresponding results are analyzed below.

5.2. Experimental Results

5.2.1. Small-Scale Case

A small-scale case is provided in this section to validate the established mathematical model. There were 10 tasks and three AGVs in the small-scale experiment. The Gurobi solver was used to solve the mathematical models, and the solution results are given in Table 6. Three AGVs were dispatched to transport a total of 10 tasks, and the transportation paths were planned with the balanced traffic load distribution. The passing nodes for each task were also provided according to the road network of AGVs. As for AGV 1, it completed the transportation of tasks 2, 4, and 8. The starting locations of the three tasks were respectively nodes 59, 55, and 57, while their goal location was node 4. The three tasks were transported to QC 1.

Table 6. The solution results of the small-scale case.

AGV Number	Task Sequence	Passed Nodes
AGV 1	Task 2–Task 4–Task 8	[59, 58, 57, 56, 55, 49, 37, 25, 13, 14, 2, 3, 4]–[55, 49, 37, 38, 26, 14, 2, 3, 4]–[57, 56, 55, 49, 37, 38, 26, 14, 2, 3, 4]
AGV 2	Task 3–Task 5–Task 6–Task 10	[57, 56, 53, 45, 33, 21, 22, 10]–[59, 58, 57, 51, 41, 42, 30, 18, 6, 7, 8, 9, 10]–[57, 56, 55, 49, 37, 25, 26, 14, 2, 3, 4]–[59, 58, 57, 56, 55, 49, 37, 38, 39, 27, 15, 3, 4]
AGV 3	Task 1–Task 7–Task 9	[60, 59, 58, 57, 51, 41, 29, 30, 18, 6, 7, 8, 9, 10]–[57, 56, 55, 49, 37, 25, 13, 1, 2, 3, 4]–[55, 49, 37, 25, 26, 14, 2, 3, 4]

The task routes of three AGVs are shown in Figure 8. The AGV drove to the container blocks for the assigned tasks at different time. The minimized delay was 1981.6 s at the first-stage model, and the congestion rate was 4.86%, calculated as 29.5 s of waiting time and 606.6 s of transport time.

5.2.2. Large-Scale Case

Large-scale experiments were conducted with the proposed QL-CNP algorithm, and the results of 10 examples are shown in Table 7. Although the number of tasks increased, the number of total alternative paths did not increase. The road network of AGVs did not change with the scale of tasks. Hence, the search space of path solutions did not become larger in the larger-scale experiments. In addition, the CNP can help agents in assigning tasks according to the requirement of agents. On the basis of the respective states, agents can propose their own action selections, which can improve the feasibility of learning to some extent. This was also the goal of combining QL and CNP in this study.

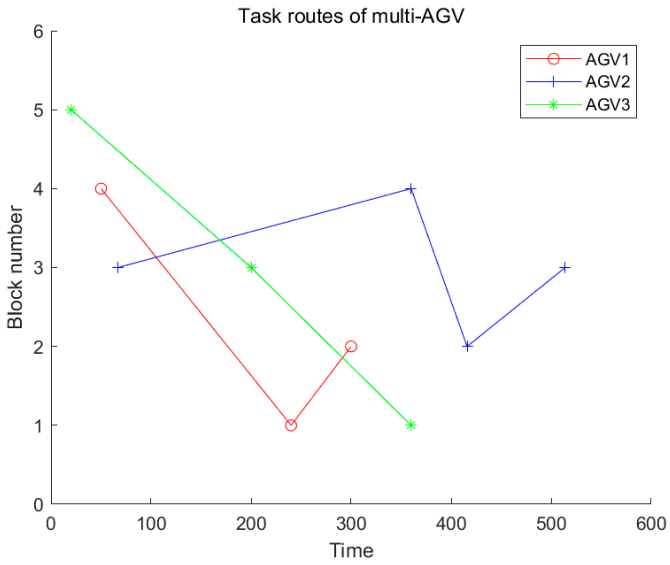

Figure 8. The task routes of multiple AGVs.

Table 7. Experimental results of different scale examples.

Experiment ID	Number of Tasks	Number of AGVs	Delay Time (s)	Congestion Rate
1	20	3	1754.83	3.14%
2	30	4	3339.53	3.06%
3	40	4	4222.47	2.74%
4	50	5	4811.17	2.50%
5	60	5	5750.67	2.44%
6	80	6	8068.1	2.24%
7	100	7	8773.6	1.93%
8	120	8	10,346.24	1.82%
9	150	9	12,860.38	1.75%
10	200	10	12,994.32	1.63%

The delay of completing tasks within the time window increased while the congestion rate decreased as the example scale became larger. There was no proportional relationship between the increase in delay time and the increase in task number. It can be found that the growth rate of delay time became smaller when adding the number of tasks and AGVs at the same time. The main reason is that the increased AGVs helped to take on the workload, which could avoid delay. However, increasing the number of AGVs was incapable of slowing the growth rate of the delay time when the number of tasks reached a certain scale. For example, the gap in delay time among the later large-scale experiments became larger and larger even though more AGVs joined the tasks. This is because the equipment configuration could not meet the task requirements when the number was saturated. Thus, it was necessary to make an appropriate assignment between supply sides and demand points.

In addition, the congestion rate of different experiments decreased gradually as the scale of examples increased, mainly due to the reduced waiting time. The transporting time of AGVs increased when the number of tasks increased. Meanwhile, multiple AGVs were allocated to various paths for their respective tasks. The reasonable paths formulated by different passing nodes led to a lower waiting time of AGVs with fewer conflicts of transportation. Even if the number of tasks and the time of transportation increased, the

waiting time decreased, thereby reducing the congestion rate. This also indicates that the distribution of traffic load was balanced because few waits occurred. Therefore, the proposed QL-CNP algorithm was effective in finding the optimal paths of multiple AGVs in the above cases.

5.3. Comparison of Different Approaches

Different approaches are compared in this section to verify the effectiveness of the QL-CNP algorithm. The proposed QL-CNP algorithm was used to solve the MADP in this study. Furthermore, the general contract net algorithm (CNA) and Dijkstra algorithm were also compared to conduct performance analysis of different approaches. Dijkstra is a classical shortest-path algorithm that is usually used to find optimal routes [44]. Three different scales of tasks were chosen to perform the experiments. The congestion rate and solution time of the three approaches are given in Table 8.

Table 8. Experimental results of different scales with three approaches.

	QL-CNP		CNA		Dijkstra	
Task Scale	**Congestion Rate**	**Time**	**Congestion Rate**	**Time**	**Congestion Rate**	**Time**
50	2.50%	444	3.99%	4738	4.02%	3049
100	1.93%	529	3.06%	7680	3.12%	4738
150	1.75%	1020	NA	NA	2.87%	4918
Average	2.06%	664.33	3.525%	6209	3.34%	4235

When the scale of tasks expanded, it was difficult for CNA to obtain the target value while QL-CNP and Dijkstra could still obtain the solutions. This is mainly because optimal paths among many nodes could not be found for a large number of tasks based on the CNA. In addition, QL-CNP and Dijkstra struggled to find the solution when the number of tasks increased. On the one hand, it took much more time to assign tasks to multiple AGVs within the given time window. The increased number of tasks led to longer calculation time. On the other hand, dispatching multiple AGVs to the optimal path also spent substantial energy for balanced traffic load distribution. This shows that the improved CNA is desirable as the QL-CNP algorithm obtained the solution within a reasonable time.

The congestion rate solved by CNA was lower than that of Dijkstra, while the calculation time of CNA was much larger. The CNA needs to search all possible nodes in the road network that multiple AGVs would pass for the task, with the manager evaluating all possible selections. Thus, there is more time for CNA to complete the whole process. The Dijkstra algorithm is more inclined to find the shortest path for transportation tasks, which may cause more AGVs to choose the same routes. As a result, the balance of traffic load distribution was ignored to some degree, and the congestion rate of its solutions was higher than for QL-CNP and CNA. Comparing the congestion rate and calculating time of the three approaches, the proposed QL-CNP algorithm could guarantee the quality of solutions within a limited time.

Moreover, other methods were introduced to test the capability of the proposed approach in this study. GA and PSO have received much attention in finding the optimal solutions to path planning problems in the literature [45,46]. These algorithms were also used to make a performance comparison. In line with previous studies, the crossover probability, mutation probability, and maximum generation were respectively set to 0.9, 0.1, and 500 in the algorithms. Each algorithm was run independently 10 times, and the average value was taken for quantitative analysis. In this experiment, there were 60 tasks and five AGVs. The QL-CNP algorithm, GA, and PSO were evaluated, and the experimental results are provided below.

The task distributions of AGVs solved using the three methods are described in Figure 9. Among the three task allocation schemes, the performance of QL-CNP was better than that of GA and PSO. Tasks were intensively distributed to multiple AGVs in the QL-CNP algorithm as the gaps in the Gantt chart of task distribution were relatively small (Figure 9a). On the contrary, the gaps in the Gantt chart of task distribution were larger in Figure 9b,c. The assignment of tasks obtained from GA and PSO was relatively scattered. In addition, the time required for all tasks in QL-CNP was lower than for GA and PSO. The last task 58 was finished at time 2395 in QL-CNP, as shown in Figure 9a. The last task was finished by AGV 3 at time 2638 in GA, while the last task 5 was finished by AGV 2 at the time 2695 in PSO, as shown in Figure 9b,c. The container tasks being assigned to AGVs in a balanced way enabled all AGVs to transport evenly, avoiding the heavy workload of some certain AGVs. Each AGV is introduced to complete container transportation, whereby no equipment remains idle. Moreover, each AGV has a relatively balanced task distribution, which produces similar energy consumption. The excessive use of AGVs is detrimental to equipment life and maintenance.

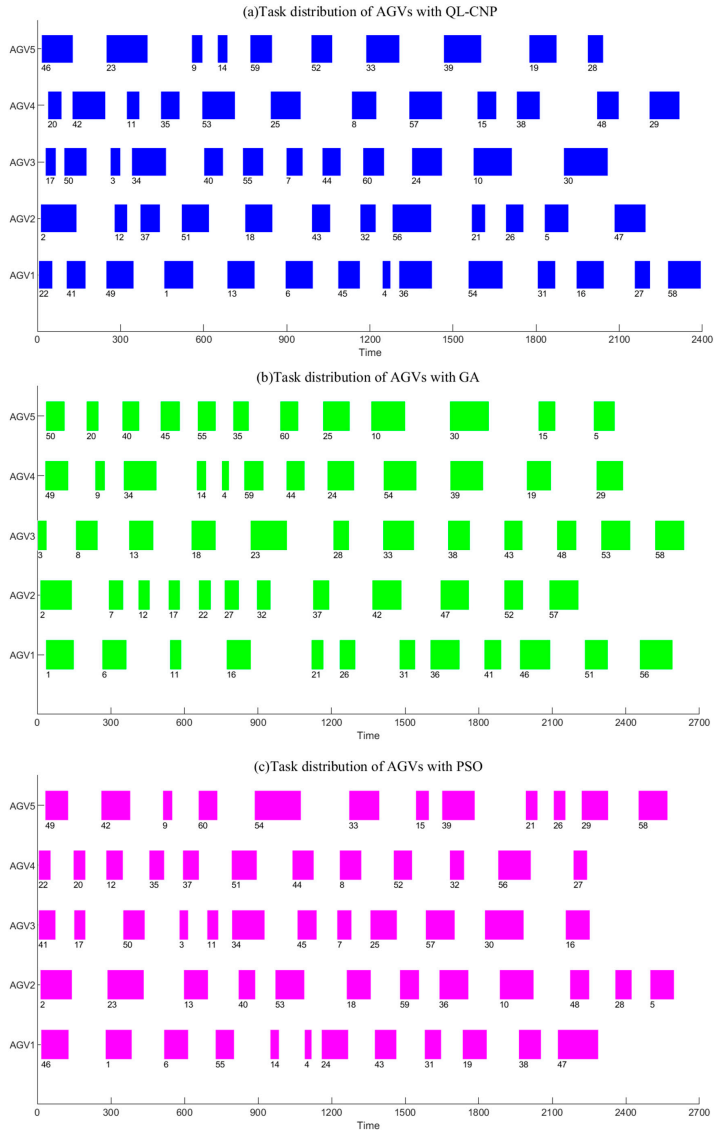

Figure 9. The task distribution of multiple AGVs using three methods.

The paths of AGVs in different solutions are drawn in Figure 10. There were five AGVs assigned to the tasks, and the path of each AGV transporting task was solved using QL-

CNP, GA, and PSO. The path nodes in the road network selected by the AGV formulated the final transportation paths. As shown in Figure 10, the five blue paths of AGVs obtained from QL-CNP were better than the green and red paths respectively obtained from GA and PSO. There was just one cross node among the AGV paths with QL-CNP, while five cross nodes and three cross nodes appeared on the green and red paths. This indicates that the AGV path of QL-CNP was more appropriate than those of GA and PSO due to the fewer conflicts. Moreover, the performance of the AGV path solved by PSO was better than that solved by GA. The AGV paths of PSO were similar to the paths of QL-CNP as some parts of the blue and red paths overlapped. The conflicts in the paths of PSO were also fewer than for GA.

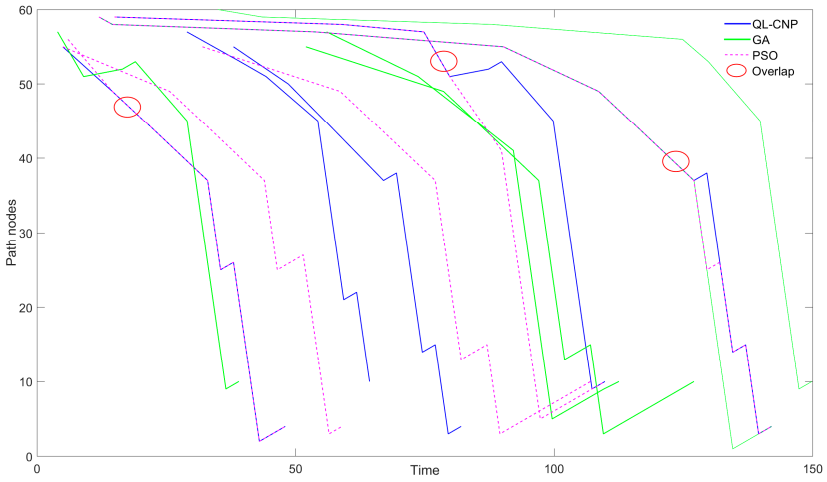

Figure 10. The AGV paths of solutions obtained using three methods.

The delay time of all tasks and congestion rate of AGVs were calculated to compare the performance of the three methods, as shown in Figure 11. The delay time of the solution obtained from QL-CNP was 5750.67, while the delay time was 6728.90 for GA and 6398.50 for PSO. Additionally, the congestion rate solved using QL-CNP was the lowest (~2.44%), followed by the congestion rate that solved using PSO. The solution from GA had the highest congestion rate of 4.01%. Hence, the solution performance of QL-CNP was superior to that of both GA and PSO.

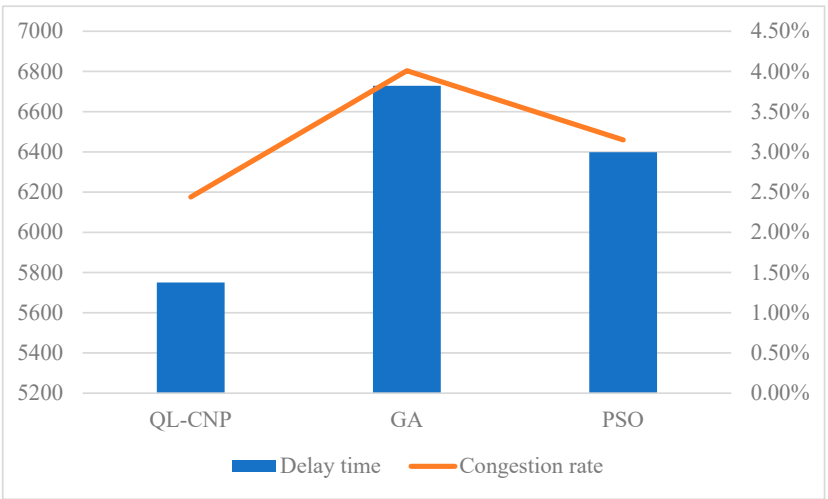

Figure 11. The delay time and congestion rate comparison of the three solutions.

6. Conclusions

This study focused on the MADP at the ACT which was divided into the task assignment and path plan for multiple AGVs. To meet the requirements of shipment, i.e., loading containers to vessels by QCs, the QL-CNP approach was proposed, involving the combination of QL and CNP to solve task allocation and path planning. Using the machine learning-based approach can reward actions executed by agents according to a certain policy by observing the state of the environment. After the environment receives the agent's action, the state will be updated and the reward feedback will be given to the agent. Then, the path plan with the maximum cumulative benefit is obtained for multiple AGVs to execute container transportation. A reasonable arrangement of AGVs contributes to the normal use of energy consumption and equipment maintenance. The process of delivering containers was described as a supply chain where the supply side was the container block storing containers and the demand point was the QC receiving containers. All available paths for AGVs formulated a traffic road network between supply sides and demand points. The traffic load distribution was introduced to estimate the transportation condition of multiple AGVs. A two-stage mathematical model was established to arrange the task sequence and select the driving path for multiple AGVs. All tasks were assigned to multiple AGVs with the goal of minimizing the lateness of transporting containers, and then the passing paths were optimized to minimize the congestion rate. The mathematical models of the MADP were solved using the Gurobi solver in small-size cases to verify their feasibility. Meanwhile, experiments of different scales were conducted using the QL-CNP algorithm. The comparison of QL-CNP, can, and Dijkstra algorithms indicated that the proposed QL-CNP was desirable to find solutions for MADP. Additionally, the experimental results of QL-CNP, GA, and PSO demonstrated the effectiveness of the proposed approach.

The numerical results obtained using QL-CNP were superior to those obtained using the two other heuristic algorithms evaluated in this study, showing the lowest delay time and congestion rate in the solutions of task paths. The CNP demonstrated the potential of addressing task allocation in a multiagent system. Combined with a machine learning method, the MADP proposed in this study, being decomposed into task allocation and path planning, was tackled to obtain an optimal solution. When dispatching multiple AGVs to execute container transportation, a reasonable assignment of containers and AGVs can promote better task completion, and the results can provide decision-making support for port operation. In addition to analyzing the problem itself, a combination of different approaches can be tried and applied to complex decision problems.

However, there were some limitations in this study. A traditional training method was used in QL, e.g., the ε-greedy strategy, when choosing the action. An intelligent algorithm can be considered to evaluate the action selections. For example, QL and PSO were combined to search the better state–action values and improve the convergence rate. In addition, the double-cycle process can be further researched. The AGV transports the containers being loaded to vessels and containers being unloaded from vessels, which increases the route complexity. Deep reinforcement learning can be considered to solve the integrated dispatching of AGVs and other associated equipment during transportation. Moreover, some factors should be studied in the MADP, such as the path conflicts and the battery capacity of AGVs. The realistic constraints can be considered when planning AGV paths to simulate the practical operational environment. Digital twin is also an alternative approach to help decision making involving the scheduling of multiple AGVs in port.

Author Contributions: Conceptualization, Y.G.; methodology, C.-H.C.; software, Y.G.; validation, D.C. and Y.G.; formal analysis, C.-H.C.; investigation, C.-H.C. and D.C.; resources, Y.G. and C.-H.C.; data curation, Y.G.; writing—original draft preparation, Y.G. and C.-H.C.; writing—review and editing, D.C.; visualization, Y.G. and C.-H.C.; supervision, D.C. 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: Data is unavailable due to privacy or ethical restrictions.

Acknowledgments: The authors are grateful to the editors and anonymous referees for their valuable comments and detailed suggestions, which helped to improve the quality of this paper.

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

Appendix A

Some data samples of tasks are provided in Table A1, where the time is converted to numeric format.

Table A1. Some data samples of tasks.

Task ID	Container Block	QC	Earliest Time	Latest Time
15151050	5	2	0	2100
15150726	4	1	0	720
15151322	3	2	60	3600
15152214	1	1	60	8400
15155329	4	2	120	26520
15151520	2	1	180	4560
15151632	3	1	240	4740
15153108	2	1	240	13020
15151248	1	1	240	3240
15153185	3	1	300	13560

References

1. Soylu, M.; Özdemirel, N.E.; Kayaligil, S. A self-organizing neural network approach for the single AGV routing problem. *Eur. J. Oper. Res.* **2000**, *121*, 124–137. [CrossRef]
2. Yang, Y.; Zhong, M.; Dessouky, Y.; Postolache, O. An integrated scheduling method for AGV routing in automated container terminals. *Comput. Ind. Eng.* **2018**, *126*, 482–493. [CrossRef]
3. Nishi, T.; Hiranaka, Y.; Grossmann, I.E. A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Comput. Oper. Res.* **2011**, *38*, 876–888. [CrossRef]
4. Xin, J.; Negenborn, R.R.; Corman, F.; Lodewijks, G. Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance. *Transp. Res. Part C Emerg. Technol.* **2015**, *60*, 377–396. [CrossRef]
5. Zhong, M.; Yang, Y.; Dessouky, Y.; Postolache, O. Multi-AGV scheduling for conflict-free path planning in automated container terminals. *Comput. Ind. Eng.* **2020**, *142*, 106371. [CrossRef]
6. Yue, L.; Fan, H.; Ma, M. Optimizing configuration and scheduling of double 40 ft dual-trolley quay cranes and AGVs for improving container terminal services. *J. Clean. Prod.* **2021**, *292*, 126019. [CrossRef]
7. Roy, D.; Gupta, A.; De Koster, R.B. A non-linear traffic flow-based queuing model to estimate container terminal throughput with AGVs. *Int. J. Prod. Res.* **2016**, *54*, 472–493. [CrossRef]
8. Salman, S.; Alaswad, S. Alleviating road network congestion: Traffic pattern optimization using markov chain traffic assignment. *Comput. Oper. Res.* **2018**, *99*, 191–205. [CrossRef]
9. Małopolski, W. A sustainable and conflict-free operation of AGVs in a square topology. *Comput. Ind. Eng.* **2018**, *126*, 472–481. [CrossRef]
10. Li, K.; Zhou, T.; Liu, B.H.; Li, H. A multi-agent system for sharing distributed manufacturing resources. *Expert Syst. Appl.* **2018**, *99*, 32–43. [CrossRef]
11. Framiñán Torres, J.M.; Pérez González, P.; Fernández-Viagas Escudero, V.; González, V. Assessing the potential of decentralised scheduling: An experimental study for the job shop case. In 10th IFAC Conference on Manufacturing Modelling, Management and Control, MIM 2022. *IFAC PapersOnLine* **2022**, *55*, 2617–2622.
12. Singh, N.; Dang, Q.V.; Akcay, A.; Adan, I.; Martagan, T. A matheuristic for AGV scheduling with battery constraints. *Eur. J. Oper. Res.* **2022**, *298*, 855–873. [CrossRef]
13. Yu, N.N.; Li, T.K.; Wang, B.L.; Yuan, S.P.; Wang, Y. Reliability oriented multi-AGVs online scheduling and path planning problem of automated sorting warehouse system. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1043*, 22035. [CrossRef]

14. Li, Q.; Pogromsky, A.; Adriaansen, T.; Udding, J.T. A control of collision and deadlock avoidance for automated guided vehicles with a fault-tolerance capability. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 64. [CrossRef]
15. Zhong, M.; Yang, Y.; Sun, S.; Zhou, Y.; Postolache, O.; Ge, Y.E. Priority-based speed control strategy for automated guided vehicle path planning in automated container terminals. *Trans. Inst. Meas. Control* **2020**, *42*, 3079–3090. [CrossRef]
16. Li, C.; Zhang, L.; Zhang, L. A route and speed optimization model to find conflict-free routes for automated guided vehicles in large warehouses based on quick response code technology. *Adv. Eng. Inform.* **2022**, *52*, 101604. [CrossRef]
17. Xin, J.; Wei, L.; Wang, D.; Xuan, H. Receding horizon path planning of automated guided vehicles using a time-space network model. *Optim. Control. Appl. Methods* **2020**, *41*, 1889–1903. [CrossRef]
18. Xin, J.; Wei, L.; D'Ariano, A.; Zhang, F.; Negenborn, R. Flexible time-space network formulation and hybrid metaheuristic for conflict-free and energy-efficient path planning of automated guided vehicles. *J. Clean. Prod.* **2023**, *398*, 136472. [CrossRef]
19. Luo, J.; Wu, Y. Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *79*, 49–64. [CrossRef]
20. Yang, Y.; He, S.; Sun, S. Research on the cooperative scheduling of ARMGs and AGVs in a sea-rail automated container terminal under the rail-in-port model. *J. Mar. Sci. Eng.* **2023**, *11*, 557. [CrossRef]
21. Ji, S.; Luan, D.; Chen, Z.; Guo, D. Integrated scheduling in automated container terminals considering AGV conflict-free routing. *Transp. Lett.* **2021**, *13*, 501–513.
22. Wang, Q.; Tang, C. Deep reinforcement learning for transportation network combinatorial optimization: A survey. *Knowl. Based Syst.* **2021**, *233*, 107526. [CrossRef]
23. Gumuskaya, V.; van Jaarsveld, W.; Dijkman, R.; Grefen, P.; Veenstra, A. Integrating stochastic programs and decision trees in capacitated barge planning with uncertain container arrivals. *Transp. Res. Part C Emerg. Technol.* **2021**, *132*, 103383. [CrossRef]
24. Peng, Y.; Liu, H.; Li, X.; Huang, J.; Wang, W. Machine learning method for energy consumption prediction of ships in port considering green ports. *J. Clean. Prod.* **2020**, *264*, 121564. [CrossRef]
25. Gao, Y.; Chang, D.; Chen, C.H. A digital twin-based approach for optimizing operation energy consumption at automated container terminals. *J. Clean. Prod.* **2023**, *385*, 135782. [CrossRef]
26. Kintsakis, A.M.; Psomopoulos, F.E.; Mitkas, P.A. Reinforcement learning based scheduling in a workflow management system. *Eng. Appl. Artif. Intell.* **2019**, *81*, 94–106. [CrossRef]
27. Kim, T.; Kim, Y.W.; Lee, D.; Kim, M. Reinforcement learning approach to scheduling of precast concrete production. *J. Clean. Prod.* **2022**, *336*, 130419. [CrossRef]
28. Lin, C.C.; Chen, K.Y.; Hsieh, L.T. Real-time charging scheduling of automated guided vehicles in cyber-physical smart factories using feature-based reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 4016–4026. [CrossRef]
29. Cals, B.; Zhang, Y.; Dijkman, R.; van Dorst, C. Solving the online batching problem using deep reinforcement learning. *Comput. Ind. Eng.* **2021**, *156*, 107221. [CrossRef]
30. James, J.Q.; Yu, W.; Gu, J. Online vehicle routing with neural combinatorial optimization and deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3806–3817.
31. Ahamed, T.; Zou, B.; Farazi, N.P.; Tulabandhula, T. Deep reinforcement learning for Crowdsourced urban delivery. *Transp. Res. Part B Methodol.* **2021**, *152*, 227–257. [CrossRef]
32. Zhang, K.; He, F.; Zhang, Z.; Lin, X.; Li, M. Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *Transp. Res. Part C Emerg. Technol.* **2020**, *121*, 102861. [CrossRef]
33. Jelen, G.; Babic, J.; Podobnik, V. A multi-agent system for context-aware electric vehicle fleet routing: A step towards more sustainable urban operations. *J. Clean. Prod.* **2022**, *374*, 134047. [CrossRef]
34. Choe, R.; Kim, J.; Ryu, K.R. Online preference learning for adaptive dispatching of AGVs in an automated container terminal. *Appl. Soft Comput.* **2016**, *38*, 647–660. [CrossRef]
35. Hu, H.; Yang, X.; Xiao, S.; Wang, F. Anti-conflict agv path planning in automated container terminals based on multi-agent reinforcement learning. *Int. J. Prod. Res.* **2023**, *61*, 65–80. [CrossRef]
36. Smith, R.G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.* **1980**, *29*, 1104–1113. [CrossRef]
37. Gao, J.; Wong, T.; Wang, C. Coordinating patient preferences through automated negotiation: A multiagent systems model for diagnostic services scheduling. *Adv. Eng. Inform.* **2019**, *42*, 100934. [CrossRef]
38. Jégou, D.; Kim, D.W.; Baptiste, P.; Lee, K.H. A contract net based intelligent agent system for solving the reactive hoist scheduling problem. *Expert Syst. Appl.* **2006**, *30*, 156–167. [CrossRef]
39. Cardon, A.; Vacher, J.P. Multi-objective genetic agents based on a contract-net system for job-shop scheduling problems. *IFAC Proc.* **2000**, *33*, 951–956. [CrossRef]
40. Liang, H.; Kang, F. A novel task optimal allocation approach based on contract net protocol for agent-oriented UUV swarm system modeling. *Optik* **2016**, *127*, 3928–3933. [CrossRef]
41. Zhen, Z.; Wen, L.; Wang, B.; Hu, Z.; Zhang, D. Improved contract network protocol algorithm based cooperative target allocation of heterogeneous UAV swarm. *Aerosp. Sci. Technol.* **2021**, *119*, 107054. [CrossRef]
42. González-Briones, A.; De La Prieta, F.; Mohamad, M.S.; Omatu, S.; Corchado, J.M. Multi-agent systems applications in energy optimization problems: A State-of-the-Art Review. *Energies* **2018**, *11*, 1928. [CrossRef]

43. Chen, H.; Ji, Y.; Niu, L. Reinforcement learning path planning algorithm based on obstacle area expansion strategy. *Intell. Serv. Robot.* **2020**, *13*, 289–297. [CrossRef]
44. Qiu, L.; Hsu, W.J.; Huang, S.Y.; Wang, H. Scheduling and routing algorithms for AGVs: A survey. *Int. J. Prod. Res.* **2002**, *40*, 745–760. [CrossRef]
45. Soltani, A.R.; Tawfik, H.; Goulermas, J.Y.; Fernando, T. Path planning in construction sites: Performance evaluation of the Dijkstra, A∗, and GA search algorithms. *Adv. Eng. Inform.* **2002**, *16*, 291–303. [CrossRef]
46. Sun, P.; Shan, R. Predictive control with velocity observer for cushion robot based on PSO for path planning. *J. Syst. Sci. Complex.* **2020**, *33*, 988–1011. [CrossRef]