systems

MDPI

Article

Simulation and Optimization of Automated Guided Vehicle Charging Strategy for U-Shaped Automated Container Terminal Based on Improved Proximal Policy Optimization

Yongsheng Yang *, Jianyi Liang and Junkai Feng

Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China
* Correspondence: yangys_smu@126.com

Abstract: As the decarbonization strategies of automated container terminals (ACTs) continue to advance, electrically powered Battery-Automated Guided Vehicles (B-AGVs) are being widely adopted in ACTs. The U-shaped ACT, as a novel layout, faces higher AGV energy consumption due to its deep yard characteristics. A key issue is how to adopt charging strategies suited to varying conditions to reduce the operational capacity loss caused by charging. This paper proposes a simulation-based optimization method for AGV charging strategies in U-shaped ACTs based on an improved Proximal Policy Optimization (PPO) algorithm. Firstly, Gated Recurrent Unit (GRU) structures are incorporated into the PPO to capture temporal correlations in state information. To effectively limit policy update magnitudes in the PPO, we improve the clipping function. Secondly, a simulation model is established by mimicking the operational process of the U-shaped ACTs. Lastly, iterative training of the proposed method is conducted based on the simulation model. The experimental results indicate that the proposed method converges faster than standard PPO and Deep Q-network (DQN). When comparing the proposed method-based charging threshold with a fixed charging threshold strategy across six different scenarios with varying charging rates, the proposed charging strategy demonstrates better adaptability to terminal condition variations in two-thirds of the scenarios.

Keywords: u-shaped automated container terminal; AGV; charging strategy; deep reinforcement learning; plant simulation

Citation: Yang, Y.; Liang, J.; Feng, J. Simulation and Optimization of Automated Guided Vehicle Charging Strategy for U-Shaped Automated Container Terminal Based on Improved Proximal Policy Optimization. *Systems* **2024**, *12*, 472. https://doi.org/10.3390/systems12110472

Academic Editors: Shuqi Xue, Yun Wang and Xiaomeng Shi

Received: 14 October 2024
Revised: 2 November 2024
Accepted: 4 November 2024
Published: 5 November 2024

1. Introduction

ACTs allow terminal operators to densify their operations, maximizing the utilization of terminal space [1]. As land resources become increasingly scarce, ACTs are expected to become a prevailing trend in port development. Based on research on traditional automated container terminal handling systems and integrating the unique characteristics of the Qinzhou Terminal in Beibu Gulf Port, Shanghai Zhenhua Heavy Industries Co., Ltd. has proposed a novel U-shaped terminal yard layout [2], which has garnered significant attention from the academic community.

The layout of the ACT system, including the berth line layout that determines the location of the quay crane, the layout of the yard involving the location of the yard crane, and the block and path of the AGV, has a significant impact on the performance of ACTs [3]. AGV paths vary under different yard layouts. With the advancement of carbon neutrality strategies and the emergence of large container vessels, ACTs are at a pivotal stage of energy-saving, decarbonization, and intelligent transformation [4]. In this context, automated terminals have widely adopted electrically powered AGVs. However, due to the limitation of battery capacity, AGVs must proceed to charging areas for power replenishment when their battery levels drop below a certain threshold. The different charging methods and the layout of power supply equipment influence the duration of AGV power

replenishment. Uncertainty in charging time can result in unpredictable loading and unloading times, affecting the synchronization of equipment and the overall operational efficiency of the terminal.

The charging methods for electrically powered AGVs primarily include battery swapping strategies [5] and plug-in charging strategies [6]. Shanghai Yangshan Phase IV terminal implements a battery swapping strategy, where AGVs with insufficient power travel to battery swapping stations at the terminal's end to replace their batteries. This strategy allows for more flexible AGV scheduling, but the distance between the battery swapping station and the operational area affects the continuity of AGV operations [7]. The Guangzhou Nansha Phase IV ACTs and the Qingdao Qianwan Terminal adopt the plug-in charging strategy, where AGVs can use idle time to charge at charging piles. However, the plug-in charging strategy requires consideration of battery charging time, making the coordination of AGV charging and equipment scheduling more complex. Beibu Gulf Port ACT has adopted the plug-in charging strategy along with a decentralized charge station layout among the currently constructed U-shaped automated container terminals.

As shown in Figure 1 [8], the layout of U-shaped ACTs features loading and unloading places distributed perpendicularly along the shoreline. These points are evenly spread across both sides of the yard, and a double-cantilever rail crane (DCRC) is utilized to enable multi-point loading and unloading. AGVs and external container trucks (ECTs) can access the interior of the yard directly, interacting with loading and unloading places, which enhances operational efficiency. AGVs enter and exit the yard via two single-lane roads between paired yard blocks, while ECTs exit the yard through the U-shaped roadway [9]. This traffic flow separation ensures physical isolation between AGVs and ECTs, ensuring production safety at the automated terminal. However, the U-shaped layout introduces new challenges: First, as AGVs need to travel deeper into the yard, the increased distance required to complete loading and unloading tasks leads to higher energy consumption, resulting in more frequent AGV charging. Second, uncertain events often occur in actual terminal production environments, causing frequent changes in terminal working conditions. How to adopt appropriate charging strategies for varying working conditions is an urgent issue that needs to be addressed. Third, the increased number of loading and unloading places in the U-shaped layout complicates mathematical modeling, requiring a solution that reflects the actual terminal environment and is easy to solve.

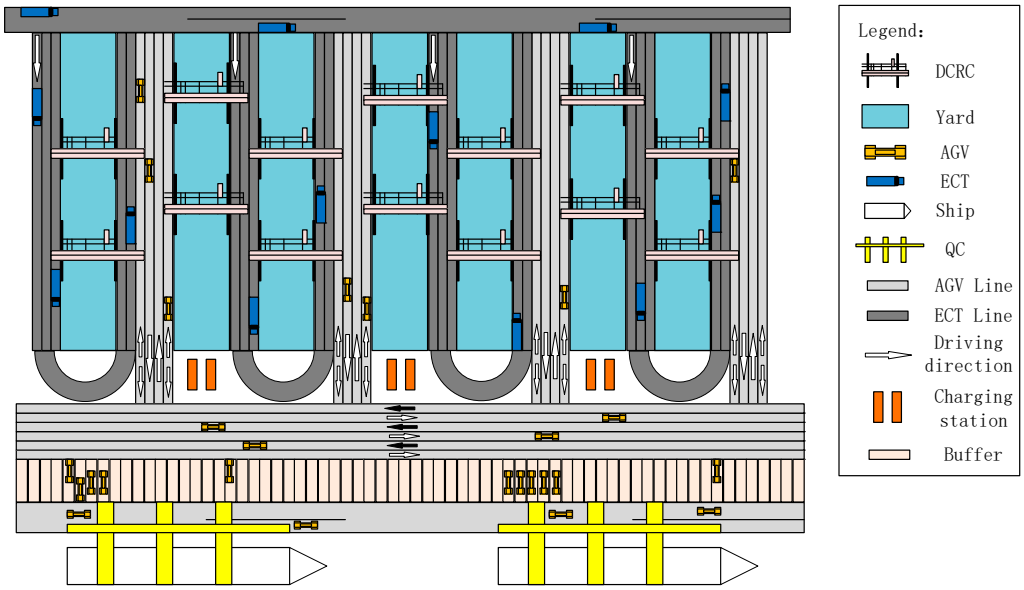

Figure 1. Layout of U-shaped ACTs.

Therefore, in response to the need for modeling that closely aligns with actual U-shaped ACT production operations and with a focus on AGV charging strategies under

varying working conditions in the U-shaped layout, this paper employs simulation to establish a model for U-shaped ACTs. Simulation modeling as the algorithm environment, a novel dynamic adjustment method for AGV charging thresholds based on deep reinforcement learning, is proposed for the first time.

The main contributions of this paper are as follows:

1. Current research on charging strategies for automated container terminals predominantly employs fixed charging thresholds without considering working conditions to optimize AGV charging strategies. The dynamic charging threshold strategy for AGVs is a pressing issue in U-shaped ACTs and an essential academic topic in terminal simulation research. For the first time, this paper introduces a simulation-based optimization method for AGV dynamic charging thresholds in U-shaped ACTs using an improved PPO algorithm. By simulating U-shaped ACTs and iteratively training the improved PPO, our charging strategies apply appropriate charging thresholds based on varying working conditions in automated terminals.
2. To meet the simulation optimization requirements for AGV charging in U-shaped ACTs, we improved the PPO algorithm's neural network and clipping function to enhance convergence speed and reduce training time in large-scale simulation scenarios. By utilizing the ability of the Gated Recurrent Unit (GRU) to leverage historical state information, a GRU structure was incorporated into the PPO deep neural network to extract temporal correlations from input state information. The GRU enables the algorithm to learn patterns in terminal working condition variations from historical data and apply them to decision-making. To address the clipping function in the PPO algorithm that fails to effectively constrain the ratio of new to old policies' probability updates within the specified range, we improved the clipping function by utilizing a hyperbolic tangent function. This adjustment ensures that the ratio of policy updates is constrained within the designated limits while facilitating smoother and more natural transitions between the old and new policies.

The remainder of this paper is structured as follows: Section 2 presents the literature review. Section 3 describes the simulation system. Section 4 introduces the improved PPO algorithm and its training process. Section 5 conducts comparative experiments. Conclusions are given in Section 6.

2. Literature Review

In this section, we review the existing research related to this study, which can be divided into the following four categories: charging strategy optimization based on mathematical models, charging strategy optimization based on simulation methods, charging strategy optimization based on deep reinforcement learning, and simulation-based charging strategy optimization using deep reinforcement learning in U-shaped ACTs.

2.1. Charging Strategy Optimization Based on Mathematical Models

The charging strategy optimization method based on mathematical models establishes a mathematical model for the optimization objective and solves the model using intelligent optimization methods. Dang et al. [10] and Singh et al. [11] addressed scenarios where each transportation request requires different AGV capacities, allowing AGVs to charge partially under a critical battery threshold. Their studies prioritize charging requests and transportation assignments and develop a mixed-integer linear programming (MILP) model. They proposed a hybrid adaptive extensive neighborhood search and integrated a local search method to solve for feasible scheduling solutions. Practical calculations demonstrate that this approach can reduce costs by 20–50%. Jun et al. [12] developed a mathematical model for the pickup and delivery problem in a manufacturing environment, considering partial and complete charging strategies. The objective was to minimize the total delay in transportation requests. To solve the model, they developed a memetic algorithm that combines genetic algorithms with local optimization techniques. Simulation experiments showed that the proposed algorithm outperforms other algorithms in terms of

average total delay. Yang et al. [13] discretized the capacity of battery swapping stations at terminals and modeled their limited processing capabilities. A mixed-integer programming (MIP) model was established to minimize the delay costs associated with AGV operations and carbon emission costs. An improved genetic algorithm was employed to solve the model. By optimizing the allocation and prioritization of container transportation and AGV battery swapping tasks, the efficiency of terminal operations and carbon emissions were significantly improved. Song et al. [14] studied the flexible scheduling problem of AGVs with battery constraints. Their study considered the varying power consumption of AGVs under empty and loaded conditions and the nonlinear characteristics of battery charging. An improved charging strategy, incorporating two charging thresholds, was proposed for the flexible scheduling model to minimize the total operational time required to complete transportation tasks. A novel meta-heuristic algorithm based on an adaptive extensive neighborhood search was employed to solve the model. Real-case calculations demonstrated the effectiveness of the proposed charging strategy. Mousavi et al. [15] developed a multi-objective AGV scheduling model to minimize the makespan and the number of AGVs, considering battery charging constraints. They optimized the model using a hybrid fuzzy genetic algorithm and particle swarm optimization. The model was evaluated and validated through simulations conducted with Flexsim software. Li et al. [16] addressed the joint scheduling problem of battery swapping and task operations with random tasks by constructing a two-stage stochastic programming model. They proposed a simulation-based ant colony optimization algorithm, where dual thresholds constrained the battery swapping strategy. The results demonstrated that scheduling schemes that account for random tasks in advance exhibit greater robustness and stability. Abderrahim et al. [17] addressed the scheduling problem of manufacturing facilities in AGV-operated job shops, aiming to minimize the makespan while considering AGV charging. They proposed a meta-heuristic algorithm based on a General Variable Neighborhood Search (GVNS) to solve the model. However, the mathematical model-based charging strategy optimization methods mentioned above fail to fully capture the dynamic complexity of actual terminal environments. Additionally, the multi-point loading and unloading configuration in U-shaped ACTs further increases the difficulty of solving these models.

2.2. Charging Strategy Optimization Based on Simulation Methods

The ACT system is a complex dynamic system with numerous discrete events and simulation methods widely used in terminal optimization. Chen et al. [18] developed a simulation model based on actual production environment data to optimize the design and operational settings of the AGV system. This model determined the required number of AGVs, charging systems, positioning, scheduling, and routing rules. They conducted factorial experiments using simulation models with different charging systems and constructed a metamodel. An optimization method based on the global metamodel was applied to optimize AGV utilization and throughput responses. Kabir et al. [19] developed a simulation model to study the impact of charging less than total capacity on manufacturing system productivity, considering the nonlinear characteristics of battery charging curves. The results indicated that this method could significantly enhance the productivity of the manufacturing system. Ma et al. [20] investigated the impact of charging pile facility planning and battery-powered AGV operation strategies on terminal system performance. They developed a simulation model consisting of a ship generator, a scheduler, and a traffic network. Their study examined the effects of two charging pile layouts and two charging strategies on system performance. The experimental results showed that a decentralized charging pile layout and a progressive charging strategy outperformed the alternatives. Kabir QS et al. [21] studied how the routing of AGVs to charge stations affects the productivity of manufacturing facilities using four heuristic algorithms and a simulation model. The results showed that optimal productivity can be achieved when the routing heuristic algorithm attempts to jointly minimize the total travel distance and the waiting time at battery stations. Han et al. [22] addressed the issue of traditional AGV scheduling systems

incurring significant additional time due to charging needs by proposing a dynamic AGV scheduling method based on digital twins. This method includes four essential functions: a technical support system, a scheduling model, scheduling optimization, and scheduling simulation. The experimental results showed that this method, compared to traditional dynamic AGV scheduling approaches, reduces the makespan by 10.7% and decreases energy consumption by 1.32%. Park et al. [23] addressed practical issues such as buffer space constraints and battery charging in AGV scheduling decisions, proposing a simulation-based multi-AGV scheduling program. They introduced job selection rules, AGV selection rules, and charging pile selection rules for AGV scheduling in actual workshops. Flexsim simulation demonstrated that job selection rules had a more significant impact on average waiting time compared to the other rules. However, the above simulation-based charging strategy optimization methods did not consider the impact of fluctuating working conditions at terminals, which may lead to deviations from real-world operations when studying the proposed charging strategies.

2.3. Charging Strategy Optimization Based on Deep Reinforcement Learning

To overcome the dynamic complexity of actual ACT environments and the vast amount of state information generated, deep reinforcement learning (DRL) methods have been recently introduced [24]. Drungilas et al. [25] addressed the problem of energy consumption optimization for battery-powered AGVs by establishing a model that includes the AGV transportation process from the quay crane to the yard. They proposed an AGV speed control algorithm based on DRL. The experimental results, compared with actual measurements, showed that the proposed method reduced energy consumption by 4.6%. Gao et al. [26], focusing on the impact of a dynamically complex environment in ACTs on AGV operational efficiency, proposed a digital twin-based decision support method to improve AGV scheduling efficiency. They used a mathematical programming model and a Q-learning algorithm to generate AGV scheduling plans for battery charging. They mapped physical space operations to virtual space to validate the solution's effectiveness. The results indicated that this method outperformed genetic algorithms and particle swarm optimization. Zhang et al. [27] tackled the dynamic scheduling problem of AGV battery swapping strategies in logistics systems, modeling the bi-objective joint optimization of AGV scheduling and battery swapping management as a Markov Decision Process (MDP). They developed a novel dueling double deep Q-network algorithm to maximize the long-term reward of minimizing material handling delays and energy consumption. Gong et al. [28] proposed a novel multi-agent deep deterministic policy gradient (MADDPG)-based scheduling algorithm called MDAS to solve the multi-AGV hybrid scheduling problem by reducing AGV energy consumption and total turnaround time in ACTs. Simulation experiments demonstrated that this method effectively reduced AGV energy consumption compared to baseline methods.

2.4. Simulation-Based Charging Strategy Optimization Using Deep Reinforcement Learning in U-Shaped ACTs

Based on a simulation model that can account for ACTs' complex working condition variations, state information related to environmental changes is provided to deep reinforcement learning, which outputs action values suited to the current ACTs' working condition. Building on this concept, many scholars have integrated simulation models with deep reinforcement learning to study AGV systems. Zhang et al. [29] addressed the AGV scheduling optimization problem in logistics systems considering spatiotemporal and kinematic constraints in AGV path planning. They proposed a digital twin-enhanced deep reinforcement learning optimization framework using an improved competitive double deep Q-network algorithm with count-based exploration. This algorithm interacted with a high-fidelity digital twin model that integrated static path planning agents using A* and dynamic collision avoidance agents to learn better scheduling strategies. The experimental results showed that this method achieved shorter delays and lower energy consump-

tion. Zheng et al. [30], considering the complexity and uncertainty of terminal operations, aimed to improve terminal operational efficiency by establishing an ACT simulation model through PlantSimulation software. They developed an adaptive learning algorithm based on a deep Q-network (DQN) to generate optimal scheduling strategies. The proposed algorithm was trained using data obtained through interactions with the simulation environment. Simulation experiments demonstrated that this approach outperformed heuristic algorithms in terms of effectiveness and efficiency. Hu et al. [31] proposed a new algorithm, Artificial Potential Field (APF)-D3QNPER, to overcome the limitations of traditional AGV path planning algorithms which rely on high-precision maps and lack generalization and obstacle avoidance capabilities in unknown environments. The APF action output method was combined with the double deep Q-network algorithm, and improvements were made to the experience replay and state feature extraction network. Comparisons with traditional path planning algorithms on the Gazebo simulation platform showed that the proposed method exhibited superior generalization ability and performance.

Based on the review of the above literature, the following bottlenecks in optimizing AGV charging strategies for U-shaped ACTs are summarized:

1. Due to the multi-point loading and unloading operations and the dynamic complexity inherent in U-shaped ACTs, mathematical modeling approaches face difficulties in obtaining solutions and fully accounting for the dynamic complexities of actual ACTs' operational environments.
2. Current research on AGV charging strategies based on simulation modeling has not adequately considered the impact of terminal working condition variations.
3. No researchers have employed deep reinforcement learning methods for optimizing AGV charging strategies in U-shaped ACTs.

To address the abovementioned issues, we establish a U-shaped ACT simulation model that accounts for terminal working condition variations by simulating different working condition changes through varying ship arrival intervals. The simulation model serves as the environment, providing state information to the improved PPO algorithm. After iterative training, the model outputs charging thresholds adapted to the current working conditions. Finally, the effectiveness of the proposed method is validated through comparative experiments, offering a feasible approach for optimizing AGV charging strategies in U-shaped ACTs.

3. Simulation System Description

In this section, we introduce the simulation model of the U-shaped ACTs, explaining its various modules to provide a foundation for the subsequent algorithm design.

3.1. U-Shaped ACT Simulation Model

The simulation model of the U-shaped ACTs was developed using Siemens Tecnomatix Plant Simulation 15.0, as shown in Figure 2. The modeling process leverages the built-in components of the simulation software and the integrated SimTalk language to simulate various operations in the terminal. The multiple operations include unloading containers from the ship to the AGV via quay cranes, AGVs waiting in the buffer zone for loading and unloading, transporting containers to the designated yard locations, and moving containers in the yard via a yard crane. The simulation process can be divided into four main modules, the quay crane module, yard module, AGV module, and charging module, each with corresponding function files.

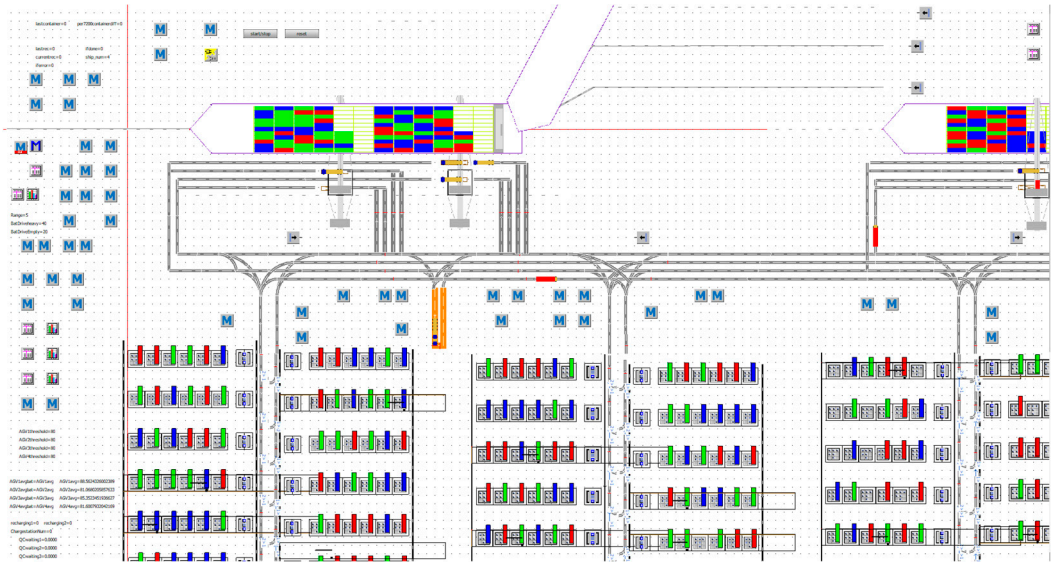

Figure 2. Schematic diagram of U-shaped ACT simulation model.

3.2. Quay Crane Module

The quay crane module includes both the ship and the quay crane unloading modules. The shipping module has a generator that sets the ship arrival intervals and function files that generate container entities once the ship reaches a designated position. The quay crane unloading module contains function files related to the generation of quay cranes and the control of quay crane unloading movements. During the initialization of the simulation, quay cranes and unloading places are generated at specified track positions. When a ship arrives, function files control the movement of the quay crane gantry, trolley, and hook to designated positions, completing the container handling and unloading processes.

3.3. Yard Module

The yard module includes the AGV buffer module and the yard crane module. When an AGV reaches the designated position in the yard, it waits at the unloading place under the yard crane for container handling. The AGV buffer module transfers the AGV to buffer control after it arrives at the yard unloading position, waits for the yard crane, and ensures that the AGV leaves buffer control after unloading is completed. The yard crane module controls the movement of the yard crane to the specified buffer position, grabbing containers from the AGV and moving the yard crane to the designated bay for container placement.

3.4. AGV Module

This module includes the AGV class and AGV roadway modules. The AGV class is responsible for generating different AGV fleets and contains functions for tracking the status of each AGV. When an AGV is generated, it is assigned a fixed loading/unloading location. If the number of AGVs under a quay crane is less than two, the AGV prioritizes picking up a container from that quay crane; otherwise, it heads to its designated loading/unloading place. During the simulation, AGV status can be accessed in real time, and the collected information is summarized in an Excel sheet, as shown in Figure 3. The AGV roadway module includes the paths that AGVs follow between the quay cranes and the yard. This path can be divided into the quay-side AGV waiting buffer area, the horizontal transport area between the quay cranes and the yard, and the AGV paths inside the U-shaped yard between different yard blocks. When an AGV has no charging task and has completed unloading in the yard, and the quay crane unloading place is busy, the AGV moves to the waiting buffer area. After collecting a container from the quay crane, the AGV leaves the quay crane unloading place and enters the horizontal transport area, which connects

all yard blocks and quay cranes. The AGV selects the path based on the principle of the shortest route. Upon entering the designated unloading place in the yard via the left-side travel lane, the AGV completes the unloading task and then turns around to exit the yard via the right-side travel lane.

string	objAGV	Remain	ChargeCount	Chargewait	distance	ChargePortion	chargedistance	chargecritic
1	*.ApplicationObjects.QuayCrane.UserObjects.AGV1:1	74.73	5	3:25:59.0071	63547.73	0.11	967.89	44
2	*.ApplicationObjects.QuayCrane.UserObjects.AGV1:2	84.40	4	2:21:42.4482	59040.55	0.11	1757.86	37
3	*.ApplicationObjects.QuayCrane.UserObjects.AGV1:3	75.60	5	3:50:34.4197	68643.97	0.12	2323.78	44
4	*.ApplicationObjects.QuayCrane.UserObjects.AGV1:4	80.24	4	2:59:57.6972	66327.85	0.11	1545.90	32
5	*.model.framework.AGV1avg	314.98						
6	*.ApplicationObjects.QuayCrane.UserObjects.AGV2:1	71.37	4	2:17:41.8079	58266.25	0.09	950.22	37
7	*.ApplicationObjects.QuayCrane.UserObjects.AGV2:2	83.91	5	3:04:00.1274	56846.31	0.11	3142.76	31
8	*.ApplicationObjects.QuayCrane.UserObjects.AGV2:3	73.71	4	3:28:27.5994	55969.22	0.10	1219.99	29
9	*.ApplicationObjects.QuayCrane.UserObjects.AGV2:4	81.91	4	3:03:44.0608	54581.09	0.10	4944.28	28
10	*.model.framework.AGV2avg	310.89						
11	*.ApplicationObjects.QuayCrane.UserObjects.AGV3:1	74.96	6	4:24:18.8301	70693.25	0.12	4056.39	54
12	*.ApplicationObjects.QuayCrane.UserObjects.AGV3:2	78.60	7	3:02:55.3358	74061.39	0.14	2861.40	43
13	*.ApplicationObjects.QuayCrane.UserObjects.AGV3:3	76.22	6	3:17:35.2267	69055.59	0.12	2200.08	48
14	*.ApplicationObjects.QuayCrane.UserObjects.AGV3:4	81.72	5	3:31:57.8877	65147.83	0.11	2288.48	35
15	*.model.framework.AGV3avg	311.49						
16	*.ApplicationObjects.QuayCrane.UserObjects.AGV4:1	88.19	5	2:46:09.2017	61318.83	0.11	2282.38	35
17	*.ApplicationObjects.QuayCrane.UserObjects.AGV4:2	80.69	4	2:20:39.5311	61042.09	0.10	1547.49	31
18	*.ApplicationObjects.QuayCrane.UserObjects.AGV4:3	90.37	5	3:57:02.0377	64440.06	0.13	2807.03	45
19	*.ApplicationObjects.QuayCrane.UserObjects.AGV4:4	79.77	5	2:56:51.6712	61798.40	0.10	2204.34	32
20	*.model.framework.AGV4avg	339.02						
21	*.model.framework.Agv1Chargewait			12:38:13.5722				
22	*.model.framework.Agv2Chargewait			11:53:53.5954				
23	*.model.framework.Agv3Chargewait			14:16:47.2803				
24	*.model.framework.Agv4Chargewait			12:00:42.4417				
25	*.model.framework.ChargeCount		78					
26	*.model.framework.totalCW			2:02:49:36.8896				
27	*.model.framework.totaldistance				1010314.00			
28	*.model.framework.totalchargedistance						37100.26	
29	*.model.framework.chargecritc							605

Figure 3. Status information summary.

3.5. Charge Module

The charging module includes the charging pile and AGV charging strategy modules. In Figure 2, the orange lanes represent the charging piles. AGVs can enter the charging piles via the horizontal transport area for recharging. The charging pile module outputs the number of AGV charging events and the AGV charging waiting time. The charging strategy used by the AGVs in the simulation is shown in Figure 4. This strategy is embedded into each AGV's charging control through the charging strategy module, allowing for real-time modification of the charging threshold by accessing global variables, which lays the foundation for algorithm integration.

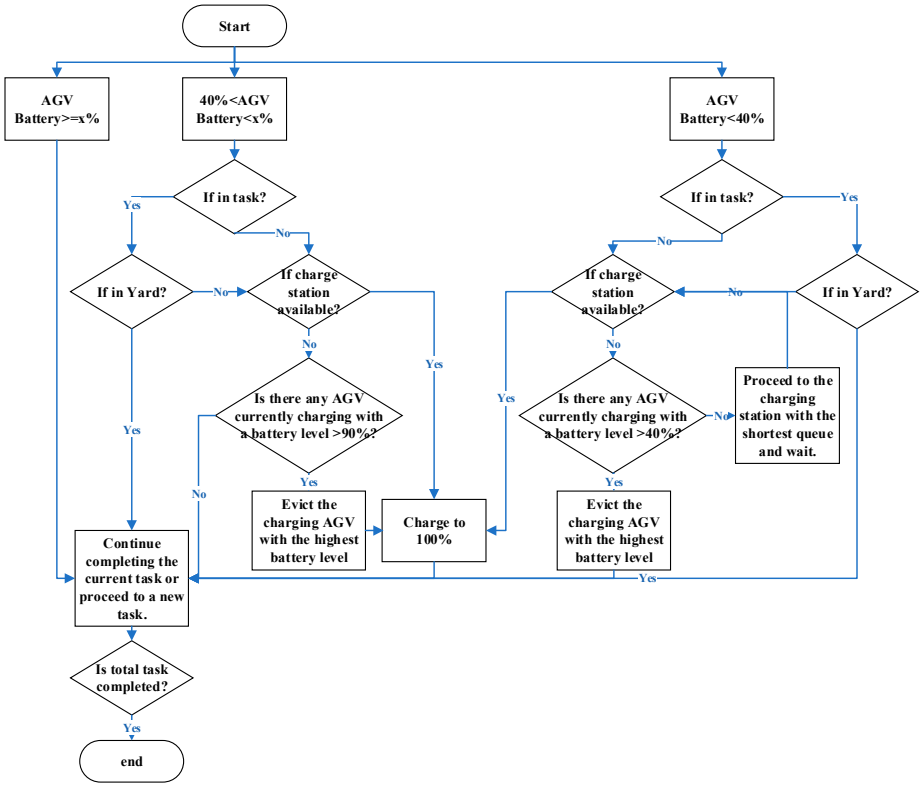

Figure 4. Charging strategy flowchart.

The charging strategy can generally be divided into three sections: when the battery level is below 40%, between 40% and the charging threshold x (determined by the DRL agent), and above the charging threshold x. When the battery level exceeds the charging threshold x, the AGV continues its current task or proceeds to a new one. When the battery level drops below 40%, the AGV first checks for available charging piles. Suppose there is an available station, the AGV charges until the battery reaches 100%. If no station is available, it forces the AGV with the highest battery level (above 40%) to stop charging, or if this is not possible, it goes to the charging pile with the shortest queue.

4. Simulation Optimization Method for Charging Strategy Based on Improved PPO

This section provides a detailed introduction to the simulation optimization method for the charging strategy based on the improved PPO algorithm, including the definition of the action and state spaces, improvements in the policy and value network architectures, enhancements in the clipping function, the design of the reward function, and the algorithm's training process.

4.1. Action Space and State Space

Defining the action and state space for DRL methods is fundamental to solving the problem. In DRL, the action space refers to the set of all actions that the algorithm can execute. In this study, the actions correspond to the charging thresholds in the charging strategy. The state space represents a series of states related to the charging strategy within the simulation. The state and action spaces can be classified as discrete or continuous. This study employs a continuous state space, while discrete action values are output.

At time step k, the action space is ak = [Thk], where Thk represents the charging threshold output by the algorithm at time step k. The action space is set to {60, 63, 66, 70, 73, 76, 80, 83, 86}. The state space is defined as Sk = [Bk, Ck, Pk-1, Cspk-1, Fcsk-1, Cck, Qwk, Cwk, Cdk, if_done], where the following holds:

- Bk is the average AGV battery level at time step k;
- Ck is the increase in the number of charging AGVs at time step k;
- Pk-1 is the number of ships in the berthing queue at time step k-1;
- Cspk-1 is the number of AGVs queuing at the charging piles at time step k-1;
- Fcsk-1 is the number of available charging piles at time step k-1;
- Cck is the AGV charging score at time step k;
- Qwk is the increase in quay crane waiting time at time step k;
- Cwk is the increase in AGV charging waiting time at time step k;
- Cdk is the proportion of AGV charging mileage at time step k;
- if_done is a signal indicating whether the simulation has reset;
- All input state variables are normalized to ensure their range is between 0 and 1.

4.2. Policy Network and Value Network Architecture

The state space and action space were defined in the previous section. However, they may become exceedingly large, particularly as the number of training iterations increases, making storing a separate value for every state or state–action pair impractical. To address this, we use deep neural networks to estimate the value function or policy function efficiently.

Various factors influence ship arrival times in ACTs. However, overall, ship arrival times exhibit a time-dependency characteristic. Therefore, the GRU network structure is considered, as it can leverage historical data to learn the patterns and dependencies of ship arrival times, thereby improving prediction accuracy. The GRU network structure is shown in Figure 5a. The input of the network consists of the state information output from the simulation, and the hidden layer is composed of a forward GRU, which can account for historical information and extract more useful information from the observed data. The structure of a GRU is shown in Figure 5b. In the GRU structure, the update gate and reset gate manage the flow of information within the network. By learning when to

retain or discard information, the GRU helps effectively capture long-term dependencies when processing sequential data. The output of the GRU can be obtained using the following equations:

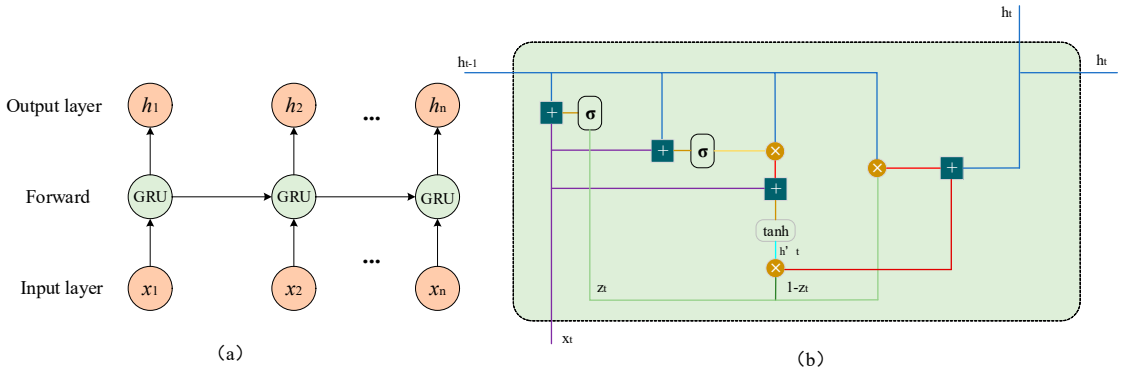

Figure 5. (**a**) GRU structure. (**b**) GRU.

The expression for the update gate z_t at the current time step is as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + \Delta b_z) \tag{1}$$

where W_z is the weight matrix for the update gate, h_{t-1} is the hidden state from the previous time step, x_t is the input at the current time step, and Δb_z is the bias parameter for the update gate.

The expression for the reset gate rt at the current time step is given by the following:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + \Delta b_r) \tag{2}$$

where W_r is the weight matrix for the reset gate, and Δb_r is the bias parameter for the reset gate.

Based on the output of the reset gate, the candidate hidden state $\widetilde{h}_t$ at the current time step is calculated as follows:

$$\widetilde{h}_t = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + \Delta b_h) \tag{3}$$

where W_h is the weight matrix, and Δb_h is the bias parameter.

Finally, the hidden state h_t at the current time step t can be obtained as follows:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t \tag{4}$$

The deep neural network is shown in Figure 6. In the simulation model, the AGV module automatically retrieves relevant state information S, which is input into the policy and value networks. The input layer of both networks consists of a GRU structure, which processes the state information and outputs the final hidden state h_t to the hidden layer. The policy and value networks have only one hidden layer, with a dimension of 128*64. The output of the policy network represents the charging threshold for the AGV charging strategy, while the output of the value network estimates the advantage of the current state. The parameters of the policy and value networks are denoted by θ and ζ, respectively. During training, these parameters are continuously optimized through the objective function. The objective of the iterative training is to ensure that the charging thresholds generated by the policy network align with the current working conditions of the ACTs, thereby preventing inefficient charging behaviors.

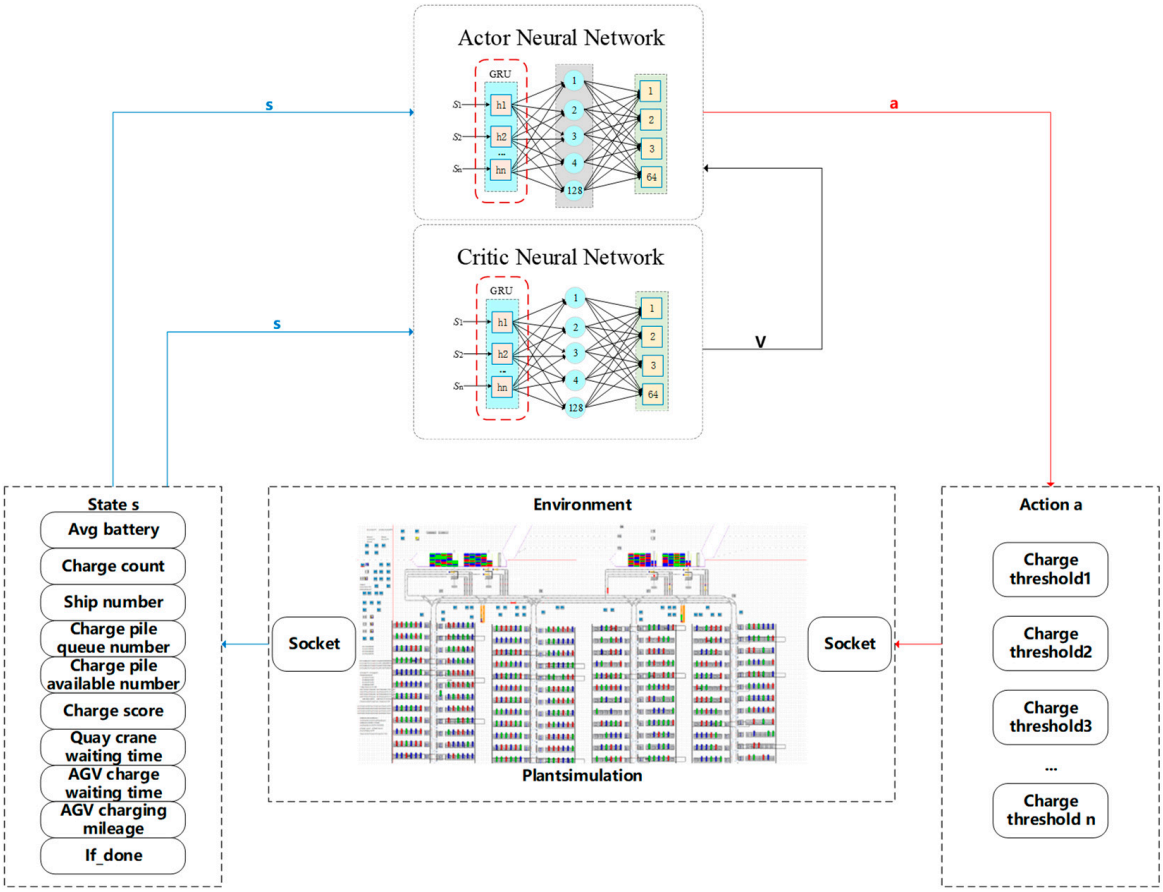

Figure 6. Simulation optimization method for AGV dynamic charging threshold strategy based on DRL.

4.3. Clipping Function Improvement

In the PPO algorithm, the clipping function limits the magnitude of updates between the new and old policies. The objective function for the policy network is L^{CLIP}.

$$L^{CLIP}(\theta) = \hat{E}_t\left[min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right] \tag{5}$$

In Equation (5), the clipping function uses fixed upper and lower bounds. This approach has certain drawbacks, such as failing to constrain the likelihood ratio between the new and old policies within the specified range and the sudden flattening of the likelihood ratio curve, which can result in reduced optimization efficiency. To address the above issues, we improve the clipping function by applying a hyperbolic tangent function to limit the ratio that exceeds the bounds, preventing excessive policy update magnitudes. The improved clipping function is shown in Equation (6):

$$clip(r_t(\theta), \varepsilon, \alpha) = \begin{cases} -2\alpha tanh(r_t(\theta)) + 1 - \varepsilon, & r_t(\theta) < 1-\varepsilon \\ -2\alpha tanh(r_t(\theta)) + 1 + \varepsilon, & r_t(\theta) \geq 1+\varepsilon \\ r_t(\theta), & otherwise \end{cases} \tag{6}$$

where α is a hyperparameter that determines the degree of clipping in the function, and as α increases, the upper and lower bounds increase.

4.4. Reward Function

After constructing the policy and value networks in the previous section, designing an appropriate reward function to train the network and optimize its parameters is essential.

The reward function guides the agent and offers timely feedback during interactions with the environment, helping the agent quickly identify the optimal strategy.

In this study, to ensure the agent can adopt appropriate charging thresholds under varying working conditions, the total reward function is designed based on the number of ships in the queue as a criterion for distinguishing different working conditions. The overall reward function is defined as follows:

$$R_t = R_{bat} + R_{th} + R_c + R_n + R_{cc} + R_{Qw} \tag{7}$$

$$R_{bat} = \frac{(B_{t-1} + B_t)}{2} \times W_{bat} \tag{8}$$

In Equation (8), R_{bat} represents the reward value for the average battery level of the AGV fleet, which is used to assess the overall battery level of the AGVs at the terminal. W_{bat} is the weight coefficient for the battery-level reward.

$$R_{th} = H_{th} + W_{cc} \times 2 \times abs(p_{t-1} - Csp_{t-1}),\ n+1 \quad a \le p_{t-1} \le b\ and\ c \le a_{t-1} \le d \tag{9}$$

In Equation (9), R_{th} represents the reward value for the charging threshold, which assesses whether the output action is appropriate for the current working condition. This reward is calculated based on the number of ships in the queue at the terminal at the time the action was generated, using the ship queue length p_{t-1} as the indicator for the current terminal working condition. A fixed positive reward value H_{th} is given when the output charging threshold falls within the target range under different working conditions. W_{cc} is the weight coefficient for the charging rationality score, which is measured by the difference between the ship queue length and the queue length at the charge piles. The variables *a* and *b* represent the minimum and maximum ship queue lengths for each working condition. The terminal working conditions are divided into three categories based on different working conditions: 0–2, 3–6, and 7–8 ships in the queue. The charging thresholds c and d define each working condition's upper and lower bounds of the charging threshold range. The parameter n tracks the number of times the algorithm's output action falls within the corresponding charging threshold range.

$$R_c = -W_c \times c_t \tag{10}$$

In Equation (10), R_c represents the penalty for the number of charging events, which is used to assess how well the charging threshold matches the current working condition. N is the average number of AGV charging events under different working conditions, and W_c is the weight coefficient for the penalty on charging frequency.

$$R_n = W_n \times n \tag{11}$$

In Equation (11), R_n represents the count reward, which tracks the number of times the output action aligns with the current working condition. W_n is the weight coefficient for the count reward.

$$R_{cc} = W_{cd} \times C_{dt} + W_{cw} \times C_{wt} \tag{12}$$

In Equation (12), R_{cc} represents the reward value for charging-related evaluation metrics, which is used to assess the effectiveness of the current charging threshold under the given working condition. W_{cc} is the weight coefficient for the charging score, W_{cd} is the weight coefficient for the charging distance reward, and W_{cw} is the weight coefficient for the charging waiting time reward.

$$R_{Qw} = \begin{cases} -H_{Qw} & Q_{wt} > 0 \\ 0 & else \end{cases} \tag{13}$$

Equation (13) represents the penalty for quay crane waiting time in the simulation. To avoid situations where quay cranes are waiting for AGVs, a fixed penalty value of H is applied whenever such waiting occurs.

4.5. Policy Training

After completing the network architecture setup and reward function design, this section focuses on designing an efficient training strategy to optimize the network parameters.

PPO, a popular policy gradient method in deep reinforcement learning, is chosen for its training stability, flexibility, and broad applicability. Therefore, PPO is used to train both the policy and value networks. The PPO training framework and algorithm flowchart are shown in Figure 7 and Algorithm 1, respectively.

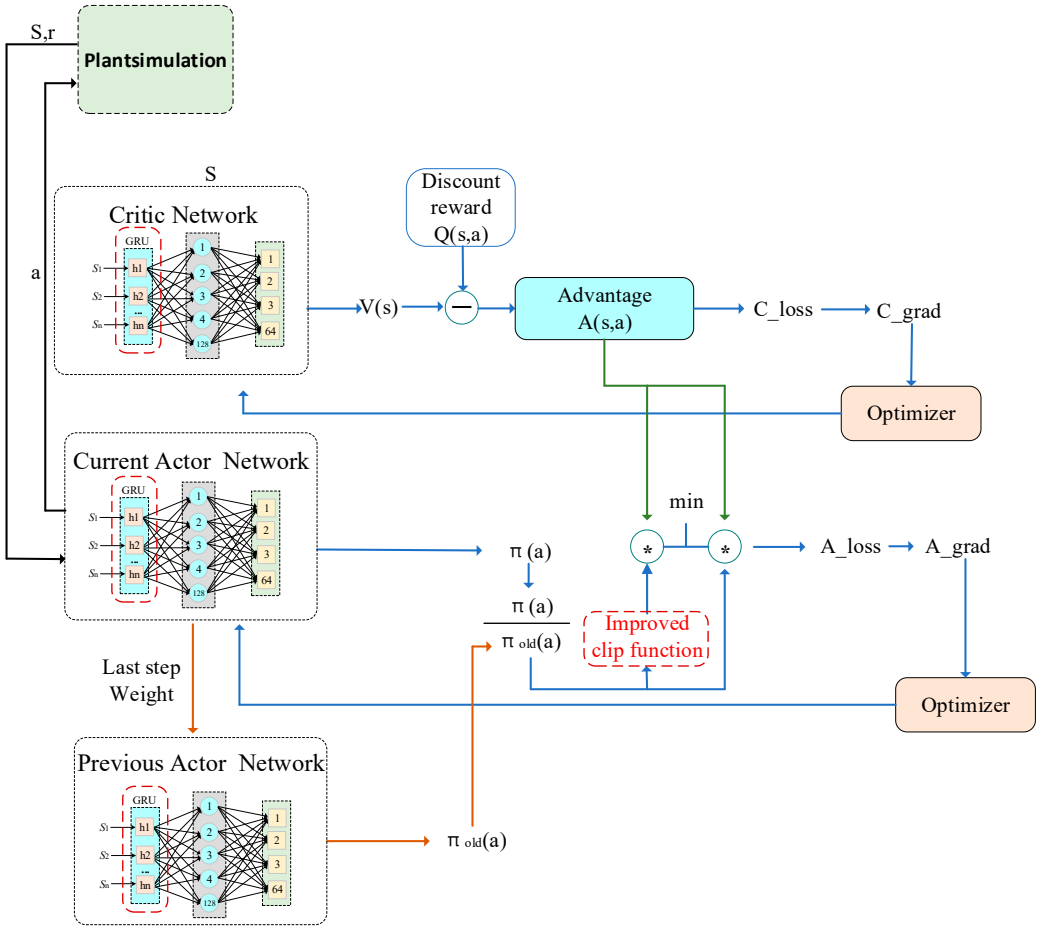

Figure 7. Policy training process flowchart.

First, the agent retrieves the initial state value from the simulation environment and generates action values based on the new policy network to interact with the simulation environment. After the AGV module in the simulation collects data, the state changes S in the simulation environment during this period are fed back to the value network and the new policy network. The state values are input into the value network to obtain the state function $V(s)$, and the advantage function $A(s)$ is calculated using the following equation:

$$Q^{\pi}(s,a) = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s, a_t = a\right] \tag{14}$$

$$A(s,a) = Q(s,a) - V(s) \tag{15}$$

$$L^V(\zeta) = \left(V_\zeta(s_t) - V_t^{target}\right)^2 \tag{16}$$

In Equation (14), $Q^\pi(s, a)$ represents the long-term return, and γ is the discount factor for the long-term return. Equation (15) represents the calculation process of the advantage function, and $V(s)$ denotes the estimated value of the current state. Equation (16) defines the objective function for the value network, which is minimized using backpropagation and the Adam optimizer to adjust the parameters ζ in the value network.

In the PPO strategy, the policy network is divided into a new policy network and an old policy network. The new policy network interacts with the environment and generates action values, while the old policy network retains the parameter weights from the previous step of the new policy network. The state values are input into the new policy network to generate the policy function $\pi(a \mid s)$, and at the same time, the old policy network generates the old policy function $\pi_{old}(a \mid s)$. The PPO strategy updates the policy network using the following objective function, with L^{CLIP} as the final objective function for the policy network:

$$L^{CLIP}(\theta, \varepsilon, \alpha) = \hat{E}_t\left[min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), \varepsilon, \alpha)\hat{A}_t\right)\right] \tag{17}$$

Algorithm 1 can be summarized as the following pseudocode:

Algorithm 1: Policy Training Algorithm

Input: discount factor γ; actor learning rate lr_a, critic learning rate lr_c; actor update steps A_UPDATE_STEPS; critic update steps C_UPDATE_STEPS; batch size BATCH; max episode EP_MAX; episode length EP_LEN; initial $\pi_{\theta 0}$, $V_{\zeta 0}$;
Output: π_θ, V_ζ

Initialize: $\pi_\theta = \pi_{\theta 0}$, $V_\zeta = V_{\zeta 0}$
For episode i $\rightarrow$ 1 to EP_MAX **do**
    Reset simulation environment;
For timestep j $\rightarrow$ 1 to EP_LEN **do**
Collect data $\{S_{ij}, a_{ij}, r_{ij}\}$;
Compute advantage;
end for
if j %BATCH == 0 or j == EP_LEN $-$ 1 **then**
for k $\rightarrow$ 1 to A_UPDATE_STEPS **do**
Optimize L^{CLIP} (θ) w.r.t. θ with lr_a;
$\theta_{old} \rightarrow \theta$;
end for
    For k $\rightarrow$ 1 to C_UPDATE_STEPS **do**
        Optimize L^V (ζ) w.r.t. ζ with lr_c;
        $\zeta_{old} \rightarrow \zeta$;
end for
end if
end for

5. Experiments and Results

As shown in Figure 8, a U-shaped ACT is simulated in the software, including two berths, four quay cranes, eight container yards, with each yard equipped with three yard cranes, sixteen AGVs, and two charge piles(The containers in the image have different colors and numbers, but their size and weight are the same.). Based on the characteristics of the U-shaped ACT, AGVs enter the yard through the left-side passageway between two yards and proceed to the designated unloading place. After completing the unloading task, they turn around and exit the yard via the right-side passageway. The ship arrival intervals are controlled through a list, simulating different working conditions (working conditions) within the ACTs. Each ship carries 144 containers, and the target positions of the containers are generated based on the principle that 80% are placed in the nearest yard and 20% in

other yards. Once all containers have been unloaded, the ship automatically departs from the berth.

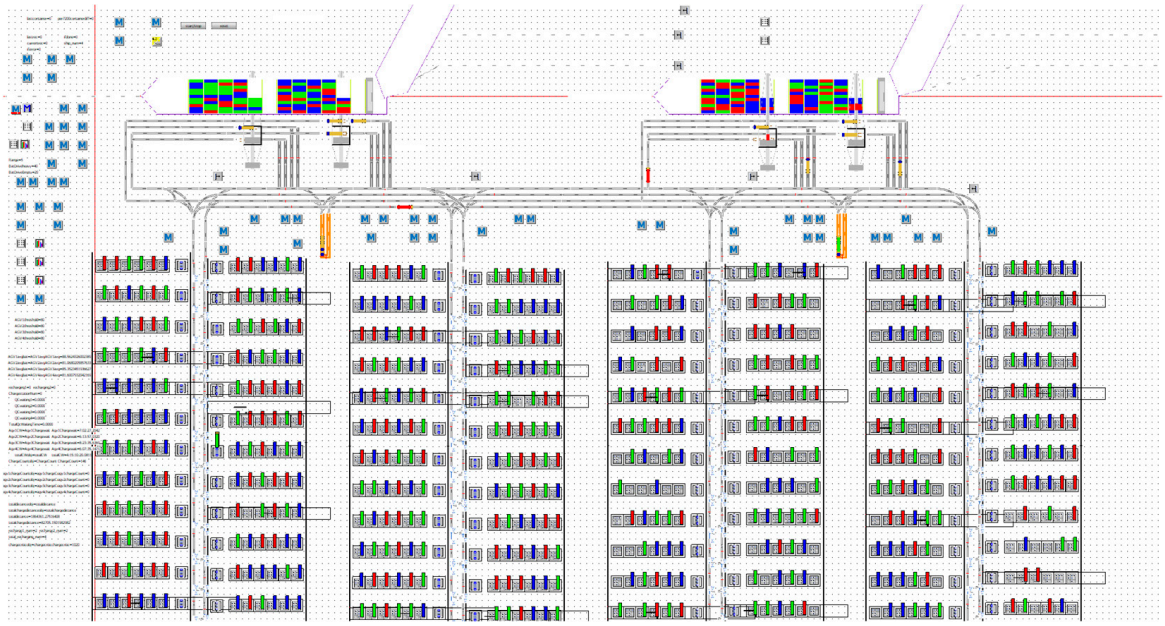

Figure 8. Schematic diagram of U-shaped automated container terminal simulation model.

To better align with the actual operational processes of the U-shaped ACTs, the following assumptions are made:

1. The containers used in the simulation are standardized 20ft TEUs;
2. The problem of turning over the box is not considered;
3. Each container information is randomly generated, with 80% placed in the nearest yard and 20% in other yards;
4. All containers are import containers;
5. Each AGV can transport only one container at a time;
6. All AGV roadways in the simulation are one-way;
7. The quay crane unloads containers from the ship in a top-to-bottom, right-to-left order;
8. Two AGVs can wait under the quay crane for loading;
9. After an AGV reaches the designated unloading place in the yard, it is transferred to a buffer, and only after the yard crane has picked up the container can the AGV leave the buffer and exit the yard;
10. The simulation sets the movement speed of the quay crane, yard crane, and hook and the height of the gantry to simulate the time for the hook to ascend and descend;
11. If the AGV does not need charging after completing the unloading task, it will move to the buffer zone to wait for the next loading;
12. All AGVs navigate using the shortest path principle;
13. AGVs are divided into loaded and unloaded states, with varying energy consumption depending on the load;
14. In the simulation, the AGV speed is set to 6.5 m/s. Regardless of whether loaded or unloaded, the AGV can operate for 11 h with 60% battery. The energy consumption per ten minutes is 1.4%, while AGV waiting consumes no energy. The specific parameters for the ACT are shown in Table 1.

Table 1. Simulation parameters for U-shaped ACTs.

Parameters	Value
Quay crane loading/unloading speed	Triangle (90, 144, 180)
Yard crane gantry movement speed	1 m/s
Yard crane hook movement speed	1 m/s
AGV speed	6.5 m/s
AGV unloaded mileage power consumption	0.6%/km
AGV loaded mileage power consumption	1.2%/km
AGV 60% battery operational time	11 h

5.1. Comparison of Different Algorithms

Before the training of different algorithms, the data-related components in the simulation are generated uniformly as follows: The arrival time intervals for all vessels are controlled using the same scheduling table. The initial energy level of each AGV is generated based on a normal distribution, ranging between 85 and 95, during the simulation initialization. The storage location for each container is determined according to the principle of placing 80% in the nearest storage yard, with the specific bay locations generated randomly. The unloading time for each quay crane placing containers onto AGV is randomly generated following a triangular distribution with parameters (90, 144, 180). Based on the aforementioned data generation method for the simulation, the overall simulation process is as follows: After the AGVs are generated, they automatically proceed to the quay crane loading and unloading points until there are two AGVs under each quay crane. When a vessel arrives at the berth, the quay crane transfers containers from the ship onto the AGV. The AGV then transports the containers to their designated locations based on randomly generated storage positions. After completing the transport, the AGV returns to the idle buffer area to wait or proceed to recharge.

During algorithm training, the simulation environment first sends the initial environment information to the algorithm, generating action values and sending feedback to the simulation. After receiving the action values from the algorithm, the simulation runs for 2 h before sending the relevant state information back to the algorithm. The algorithm evaluates the action values based on the feedback and iteratively trains the model. If the simulation run exceeds 20 days or congestion occurs in the yard, the simulation resets automatically.

DQN, PPO, and the proposed improved PPO are compared during training, with the hyperparameters for each algorithm listed in Table 2. The training process is conducted on a computer with an i5-12600KF CPU @ 4.90 GHz and with 32 G RAM. The reward curves for model training are shown in Figure 9. Owing to the introduction of the GRU structure, which enables the improved PPO to utilize historical information more effectively, the proposed method demonstrates faster convergence compared to PPO and DQN.

Table 2. Hyperparameters.

Parameters	Value
Clipping Parameter	0.2
Discount Factor	0.85
Learning Rate of Actor	0.0001
Learning Rate of Critic	0.0003
Maximum Episodes	3500
Maximum Episode Length	5

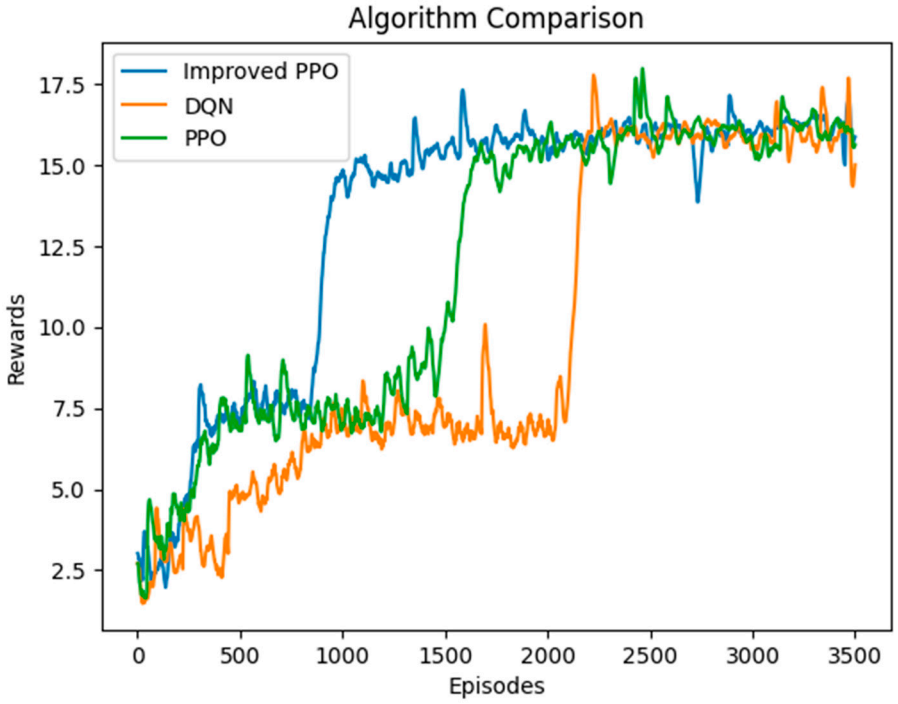

Figure 9. Reward curves for three methods.

5.2. Comparison with Charging Strategies Using Fixed Charging Thresholds

When comparing our proposed charging strategy with the fixed charging threshold strategy, we will embed the trained algorithm model into our charging strategy. Regarding the interaction time between the algorithm and the simulation, tests indicate that it takes 1 s for the algorithm to receive state information from the simulation and output new action values. Since our proposed charging strategy sends state information to the algorithm every 2 h, we can conclude that the computation time of the algorithm has a minimal impact on the real-time performance of the charging strategy.

The same ship arrival list and initial AGV battery distribution are used when compared with a fixed charging threshold strategy. For the fixed charging threshold strategy, thresholds are set at 60% and 80%. The metrics shown in Table 3 are used to evaluate these two charging strategies. In the table, "Max" represents the maximum value of each metric. Under this charging strategy, the weight assigned to these metrics is smaller since the set charging threshold more significantly influences the charging frequency and charging distance. The charging score measures the rationality of AGV charging behavior. For instance, in urgent working conditions at the ACTs, AGVs should reduce their charging frequency to minimize the loss of operational capacity. A higher charging score indicates more rational AGV charging behavior. Charging and quay crane waiting times are two metrics that directly reflect the AGV charging strategy's efficiency and the ACTs' overall operational efficiency. Longer charging waiting times will directly impact quay crane operations, and since quay cranes are the core equipment in ACTs, minimizing the situation where quay cranes wait for AGVs is critical. Therefore, these two metrics are given higher weights in the evaluation.

Table 3. Charging strategy evaluation metrics.

	Charging Frequency	**Charging Wait Time**	**Charging Distance**	**Charging Score**	**Quay Crane Wait Time**
Normalization	$\frac{Max-x}{Max}$	$\frac{Max-x}{Max}$	$\frac{Max-x}{Max}$	$\frac{x}{Max}$	$\frac{Max-x}{Max}$
Weight	0.1	0.3	0.1	0.2	0.3

Since AGV charging is mainly affected by the charging speed of the charge piles, different charging rates are set for evaluating the charging strategies. Table 4 shows the five charging rates used in the evaluation. Tables 5–10 display the average values of various metrics under different charging rates. When the charging rate falls below 0.0105%/s, AGVs become concentrated at the charge piles, leading to a severe shortage of available AGVs and making it difficult to complete tasks. Therefore, 0.0105%/s is set as the minimum charging rate.

Table 4. Charging rates.

Charging Rate	Rate 1	Rate 2	Rate 3	Rate 4	Rate 5	Rate 6
Rate (%/s)	0.0105	0.0111	0.0125	0.0138	0.0152	0.0166

Table 5. Charging strategy evaluation metrics (rate 1).

Metric	Charging Frequency	Total Charging Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time
Algorithm	984	58:08:10:48	624,551	3.66	2:09
80%	1575	69:06:31:16	1,162,095	3.2	9:01
60%	781	61:09:16:22	561,925	3.9	17:00

Table 6. Charging strategy evaluation metrics (rate 2).

Metric	Charging Frequency	Total Charging Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time
Algorithm	1018	55:13:00:53	619,893	3.7	1:13
80%	1856	63:09:59:23	1,036,996	3.3	2:45
60%	789	47:21:04:30	375,163	3.12	13:45

Table 7. Charging strategy evaluation metrics (rate 3).

Metric	Charging Frequency	Total Charging Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time
Algorithm	1142	39:08:46:41	562,470	2.85	0
80%	1838	35:18:58:10	791,452	2.15	0
60%	783	31:06:29:45	338,660	2.47	0:11

Table 8. Charging strategy evaluation metrics (rate 4).

Metric	Charging Frequency	Total Charging Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time
Algorithm	1152	32:15:27:47	544,256	2.75	0
80%	1726	29:03:34:32	697,727	2.42	0
60%	775	27:15:55:16	350,432	2.45	0

Table 9. Charging strategy evaluation metrics (rate 5).

Metric	Charging Frequency	Total Charging Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time
Algorithm	1180	28:01:15:58	531,395	2.53	0
80%	1654	26:02:12:17	670,936	2.49	0
60%	770	24:02:13:41	322,979	2.66	0

Table 10. Charging strategy evaluation metrics (rate 6).

Metric	Charging Frequency	Total Charging Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time
Algorithm	1170	24:19:49:48	495,517	2.75	0
80%	1603	24:09:58:16	624,276	2.76	0
60%	758	22:00:37:02	320,904	2.87	0

After collecting the corresponding data for different charging rates, the charging strategy evaluation metrics are standardized, and the final scores are calculated. The evaluation scores for each charging strategy under various charging rates are shown in Tables 11–16. The summary of charging strategy evaluation scores under different charging rates is shown in Table 17.

Table 11. Charging strategy evaluation scores (rate 1).

Metric	Charging Frequency	Total Charge Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time	Weighted Score
Algorithm	0.451	0.124	0.403	1	0.912	0.5333
80%	0	0	0	0.892	0.800	0.3962
60%	0.574	0.245	0.638	0.843	0	0.3830

Table 12. Charging strategy evaluation scores (rate 2).

Metric	Charging Frequency	Total Charge Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time	Weighted Score
Algorithm	0.376	0.153	0.462	0.938	0.874	0.6583
80%	0	0	0	0.821	0.470	0.4984
60%	0.504	0.113	0.516	1	0	0.3849

Table 13. Charging strategy evaluation scores (rate 3).

Metric	Charging Frequency	Total Charge Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time	Weighted Score
Algorithm	0.378	0	0.289	1	1	0.6567
80%	0	0.091	0	0.754	1	0.5223
60%	0.574	0.204	0.572	0.867	0	0.4321

Table 14. Charging strategy evaluation scores (rate 4).

Metric	Charging Frequency	Total Charge Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time	Weighted Score
Algorithm	0.332	0	0.22	1	1	0.6520
80%	0	0.107	0	0.88	1	0.5271
60%	0.55	0.151	0.498	0.891	1	0.6739

Table 15. Charging strategy evaluation scores (rate 5).

Metric	Charging Frequency	Total Charge Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time	Weighted Score
Algorithm	0.286	0	0.208	0.951	1	0.5544
80%	0	0.069	0	0.936	1	0.5553
60%	0.534	0.141	0.519	1	1	0.7124

Table 16. Charging strategy evaluation scores (rate 6).

Metric	Charging Frequency	Total Charge Wait Time	Charging Distance	Charging Score	Quay Crane Wait Time	Weighted Score
Algorithm	0.27	0	0.206	0.958	1	0.5636
80%	0	0.015	0	0.961	1	0.6183
60%	0.527	0.112	0.486	1	1	0.7485

Table 17. Summary of evaluation scores.

	Rate 1	Rate 2	Rate 3	Rate 4	Rate 5	Rate 6
Algorithm	0.5333	0.6583	0.6567	0.6520	0.5544	0.5636
80%	0.3962	0.4984	0.5223	0.5271	0.5553	0.6183
60%	0.3830	0.3849	0.4321	0.6739	0.7124	0.7485

As shown in Figure 10, the proposed charging strategy outperforms the fixed charging threshold strategy at charging rates 1, 2, and 3. At charging rate 4, the proposed method performs similarly to the strategy with a fixed 60% threshold, while at charging rates 5 and 6, the fixed 60% threshold strategy delivers the best results.

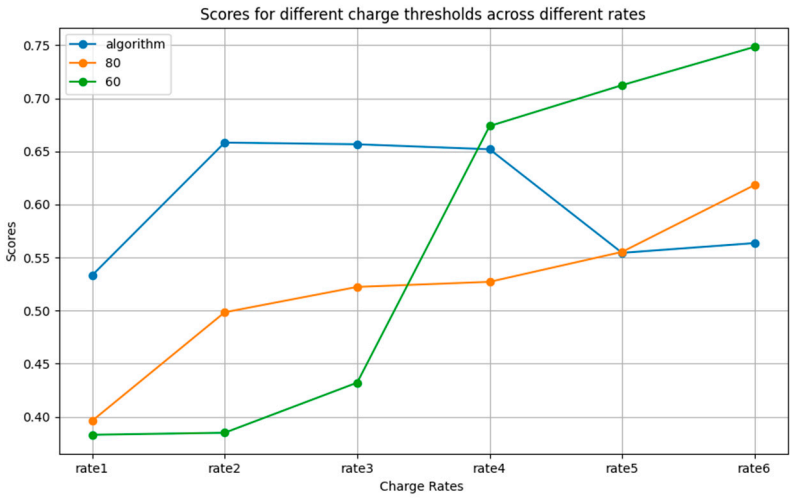

Figure 10. Summary of charging strategy evaluation scores.

At slower charging rates (below charging rate 4), the proposed method achieves higher scores because it minimizes the impact of AGV charging tasks on quay crane operations. In contrast, at faster charging rates (greater than or equal to charging rate 4), the quicker completion of charging reduces the time AGVs spend charging, making lower fixed charging thresholds more effective as the charging speed increases.

6. Conclusions

This paper proposes a dynamic charging threshold method based on an improved Proximal Policy Optimization algorithm to optimize AGV charging strategies under varying working conditions in U-shaped automated container terminals. First, a U-shaped automated container terminal simulation model is established using the Tecnomatix PlantSimulation platform. To enhance the Proximal Policy Optimization's ability to utilize historical information and reduce the issue of excessive update magnitudes during the training process, an improved Proximal Policy Optimization algorithm is introduced. The improvements include leveraging Gated Recurrent Unit network structures to process historical information and refining the clipping function. This improved Proximal Policy Optimization algorithm is applied to AGV charging strategy simulations in U-shaped automated

container terminals. Then, the simulation model generates training data for the improved Proximal Policy Optimization, enabling the model to output corresponding charging thresholds under different working conditions. Finally, the improved Proximal Policy Optimization is compared against standard Proximal Policy Optimization and deep Q-network, demonstrating that the improved Proximal Policy Optimization achieves faster convergence in this scenario.

The charging strategy based on the improved Proximal Policy Optimization is compared with a fixed charging threshold strategy. The experimental results show that when charging pile rates are low, the proposed method minimizes the impact of AGVs on quay crane operations. However, as charging pile rates increase, the proposed method shows no significant advantage over using a lower fixed charging threshold.

In actual automated container terminal environments, in addition to the energy consumption issues of AGVs, there are many other factors that need to be considered, such as the long-distance movements required for the gantry cranes in U-shaped yards and the continuous operation of quay cranes. Therefore, in future research, energy optimization in automated container terminals considering multiple devices should be a major focus.

Author Contributions: Conceptualization, resources, review and editing, supervision, Y.Y.; methodology, software, validation, formal analysis, writing, J.L.; review and editing, validation, J.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

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

References

1. Knatz, G.; Notteboom, T.; Pallis, A.A. Container Terminal Automation: Revealing Distinctive Terminal Characteristics and Operating Parameters. *Marit. Econ. Logist.* **2022**, *24*, 537–565. [CrossRef]
2. Niu, Y.; Yu, F.; Yao, H.; Yang, Y. Multi-Equipment Coordinated Scheduling Strategy of U-Shaped Automated Container Terminal Considering Energy Consumption. *Comput. Ind. Eng.* **2022**, *174*, 108804. [CrossRef]
3. Xiang, X.; Liu, C.; Lee, L.H. Performance Estimation and Design Optimization of a Congested Automated Container Terminal. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 2437–2449. [CrossRef]
4. Chen, J.; Huang, T.; Xie, X.; Lee, P.T.-W.; Hua, C. Constructing Governance Framework of a Green and Smart Port. *JMSE* **2019**, *7*, 83. [CrossRef]
5. Yang, X.; Hu, H.; Jin, J. Battery-Powered Automated Guided Vehicles Scheduling Problem in Automated Container Terminals for Minimizing Energy Consumption. *Ocean Coast. Manag.* **2023**, *246*, 106873. [CrossRef]
6. Xiang, X.; Liu, C. Modeling and Analysis for an Automated Container Terminal Considering Battery Management. *Comput. Ind. Eng.* **2021**, *156*, 107258. [CrossRef]
7. Xiao, S.; Huang, J.; Hu, H.; Gu, Y. Automatic Guided Vehicle Scheduling in Automated Container Terminals Based on a Hybrid Mode of Battery Swapping and Charging. *JMSE* **2024**, *12*, 305. [CrossRef]
8. Yang, Y.; Sun, S.; Zhong, M.; Feng, J.; Wen, F.; Song, H. A Refined Collaborative Scheduling Method for Multi-Equipment at U-Shaped Automated Container Terminals Based on Rail Crane Process Optimization. *J. Mar. Sci. Eng.* **2023**, *11*, 605. [CrossRef]
9. Xu, B.; Jie, D.; Li, J.; Yang, Y.; Wen, F.; Song, H. Integrated Scheduling Optimization of U-Shaped Automated Container Terminal under Loading and Unloading Mode. *Comput. Ind. Eng.* **2021**, *162*, 107695. [CrossRef]
10. Dang, Q.-V.; Singh, N.; Adan, I.; Martagan, T.; Van De Sande, D. Scheduling Heterogeneous Multi-Load AGVs with Battery Constraints. *Comput. Oper. Res.* **2021**, *136*, 105517. [CrossRef]
11. 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]
12. Jun, S.; Lee, S.; Yih, Y. Pickup and Delivery Problem with Recharging for Material Handling Systems Utilising Autonomous Mobile Robots. *Eur. J. Oper. Res.* **2021**, *289*, 1153–1168. [CrossRef]
13. Yang, X.; Hu, H.; Cheng, C.; Wang, Y. Automated Guided Vehicle (AGV) Scheduling in Automated Container Terminals (ACTs) Focusing on Battery Swapping and Speed Control. *JMSE* **2023**, *11*, 1852. [CrossRef]
14. Song, X.; Chen, N.; Zhao, M.; Wu, Q.; Liao, Q.; Ye, J. Novel AGV Resilient Scheduling for Automated Container Terminals Considering Charging Strategy. *Ocean Coast. Manag.* **2024**, *250*, 107014. [CrossRef]
15. Mousavi, M.; Yap, H.J.; Musa, S.N. A Fuzzy Hybrid GA-PSO Algorithm for Multi-Objective AGV Scheduling in FMS. *Int. J. Simul. Model.* **2017**, *16*, 58–71. [CrossRef]

16. Li, L.; Li, Y.; Liu, R.; Zhou, Y.; Pan, E. A Two-Stage Stochastic Programming for AGV Scheduling with Random Tasks and Battery Swapping in Automated Container Terminals. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *174*, 103110. [CrossRef]
17. Abderrahim, M.; Bekrar, A.; Trentesaux, D.; Aissani, N.; Bouamrane, K. Manufacturing 4.0 Operations Scheduling with AGV Battery Management Constraints. *Energies* **2020**, *13*, 4948. [CrossRef]
18. Chen, J.C.; Chen, T.-L.; Teng, Y.-C. Meta-Model Based Simulation Optimization for Automated Guided Vehicle System under Different Charging Mechanisms. *Simul. Model. Pract. Theory* **2021**, *106*, 102208. [CrossRef]
19. Kabir, Q.S.; Suzuki, Y. Increasing Manufacturing Flexibility through Battery Management of Automated Guided Vehicles. *Comput. Ind. Eng.* **2018**, *117*, 225–236. [CrossRef]
20. Ma, N.; Zhou, C.; Stephen, A. Simulation Model and Performance Evaluation of Battery-Powered AGV Systems in Automated Container Terminals. *Simul. Model. Pract. Theory* **2021**, *106*, 102146. [CrossRef]
21. Kabir, Q.S.; Suzuki, Y. Comparative analysis of different routing heuristics for the battery management of automated guided vehicles. *Int. J. Prod. Res.* **2018**, *57*, 624–641. [CrossRef]
22. Han, W.; Xu, J.; Sun, Z.; Liu, B.; Zhang, K.; Zhang, Z.; Mei, X. Digital Twin-Based Automated Guided Vehicle Scheduling: A Solution for Its Charging Problems. *Appl. Sci.* **2022**, *12*, 3354. [CrossRef]
23. Park, J.-S.; Kim, J.-W. Multi-AGV Scheduling under Limited Buffer Capacity and Battery Charging Using Simulation Techniques. *Appl. Sci.* **2024**, *14*, 1197. [CrossRef]
24. Wan, Z.; Li, H.; He, H.; Prokhorov, D. Model-Free Real-Time EV Charging Scheduling Based on Deep Reinforcement Learning. *IEEE Trans. Smart Grid* **2019**, *10*, 5246–5257. [CrossRef]
25. Drungilas, D.; Kurmis, M.; Senulis, A.; Lukosius, Z.; Andziulis, A.; Januteniene, J.; Bogdevicius, M.; Jankunas, V.; Voznak, M. Deep Reinforcement Learning Based Optimization of Automated Guided Vehicle Time and Energy Consumption in a Container Terminal. *Alex. Eng. J.* **2023**, *67*, 397–407. [CrossRef]
26. Gao, Y.; Chang, D.; Chen, C.-H.; Sha, M. A Digital Twin-Based Decision Support Approach for AGV Scheduling. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107687. [CrossRef]
27. Zhang, L.; Yan, Y.; Hu, Y. Deep Reinforcement Learning for Dynamic Scheduling of Energy-Efficient Automated Guided Vehicles. *J. Intell. Manuf.* **2023**. [CrossRef]
28. Gong, L.; Huang, Z.; Xiang, X.; Liu, X. Real-time AGV scheduling optimisation method with deep reinforcement learning for energy-efficiency in the container terminal yard. *Int. J. Prod. Res.* **2024**, *62*, 7722–7742. [CrossRef]
29. Zhang, L.; Yang, C.; Yan, Y.; Cai, Z.; Hu, Y. Automated Guided Vehicle Dispatching and Routing Integration via Digital Twin with Deep Reinforcement Learning. *J. Manuf. Syst.* **2024**, *72*, 492–503. [CrossRef]
30. Zheng, X.; Liang, C.; Wang, Y.; Shi, J.; Lim, G. Multi-AGV Dynamic Scheduling in an Automated Container Terminal: A Deep Reinforcement Learning Approach. *Mathematics* **2022**, *10*, 4575. [CrossRef]
31. Hu, H.; Wang, Y.; Tong, W.; Zhao, J.; Gu, Y. Path Planning for Autonomous Vehicles in Unknown Dynamic Environment Based on Deep Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 10056. [CrossRef]