

Article

Automatic Guided Vehicle Scheduling in Automated Container Terminals Based on a Hybrid Mode of Battery Swapping and Charging

Shichang Xiao ^{1,2}, Jinshan Huang ¹, Hongtao Hu ^{1,2,*} and Yuxin Gu ¹

¹ Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China; scxiao@shmtu.edu.cn (S.X.); 202130210375@stu.shmtu.edu.cn (J.H.); 17601241625@163.com (Y.G.)

² Engineering Research Center of Container Supply Chain Technology, Ministry of Education, Shanghai 201306, China

* Correspondence: hthu@shmtu.edu.cn

Abstract: Automatic guided vehicles (AGVs) in the horizontal area play a crucial role in determining the operational efficiency of automated container terminals (ACTs). To improve the operational efficiency of an ACT, it is essential to decrease the impact of battery capacity limitations on AGV scheduling. To address this problem, this paper introduces battery swapping and opportunity charging modes into the AGV system and proposes a new AGV scheduling problem considering the hybrid mode. Firstly, this study describes the AGV scheduling problem of the automated container terminals considering both loading and unloading tasks under the hybrid mode of battery swapping and charging. Thereafter, a mixed-integer programming model is established to minimize the sum of energy costs and delay costs. Secondly, an effective adaptive large neighborhood search algorithm is proposed to solve the problem, in which the initial solution construction, destroy operators, and repair operators are designed according to the hybrid mode. Finally, numerical experiments are conducted to analyze the effectiveness of the model and the optimization performance of the algorithm. The results demonstrate that the hybrid mode of battery swapping and charging can effectively reduce the number of battery swapping times and scheduling costs compared to the existing mode.

Keywords: automated container terminals; AGV scheduling; battery swapping and charging; energy cost; adaptive large neighborhood search



Citation: 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. *J. Mar. Sci. Eng.* **2024**, *12*, 305. <https://doi.org/10.3390/jmse12020305>

Academic Editor: Mihalis Golias

Received: 4 January 2024

Revised: 6 February 2024

Accepted: 7 February 2024

Published: 9 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the advancement of carbon neutrality strategies and the emergence of large-scale container ships, automated container terminals (ACTs) are currently undergoing a crucial period of energy conservation, carbon reduction, and intelligent transformation [1,2]. To enhance the operational efficiency of terminals, battery-driven AGVs are widely employed in ACTs. These AGVs form a horizontal transport system that connects the seaside and landside of the ACTs, and the scheduling performance directly affects the operational efficiency of the ACTs as well as their energy consumption. However, due to the limited capacity of AGV batteries, the AGVs need to leave the operation area of the ACTs for power replenishment. The time length of the power replenishment and the location of the charging and battery swapping stations can result in the sequence of subsequent loading and unloading operations and operation time being uncertain, thus affecting the synergy of loading and unloading equipment and the terminal's operational efficiency.

Currently, AGV power replenishment methods mainly include battery swapping [3] and battery charging [4]. For instance, the Shanghai Yangshan Phase IV Automated Terminal has installed a battery swapping station at the front of the terminal, which can be used for the battery swapping of AGVs. When the power level of AGVs drops below the threshold, the power-deficient AGVs are dispatched to the battery swapping stations

to replace their batteries with new ones [5]. Xiamen Yuanhai Terminal and Qingdao Port Qianwan Terminal use the battery charging mode, where AGVs use the waiting time to recharge during operation [6]. Battery swapping makes AGV scheduling more flexible; however, it will affect the continuity of AGV operations due to the battery swapping station being far away from the operation area. Battery charging can replenish power in the AGV operation area but requires frequent consideration of charging time, making the constraints of the AGV scheduling problem more complex. Therefore, optimizing the scheduling of AGVs while considering power replenishment has become an urgent research problem that needs to be addressed.

In the existing ACT-related studies, the scheduling of AGVs usually considers the integration with quay crane (QC) operations and yard crane (YC) operations to improve the synergy between different loading and unloading equipment, and the energy consumption or efficiency is usually taken as the optimization objective. Fan et al. [7] established a two-phase model for the joint scheduling problem of dual-trolley QCs and AGVs to minimize the energy consumption of the QCs and AGVs. Zhao et al. [8] established a mixed-integer programming model for the energy consumption problem in the joint scheduling of automated quay cranes (AQCs) and AGVs, taking into account the capacity limitation of the automated QC transfer platform to minimize the total energy consumption. Xu et al. [9] established a multi-objective scheduling model to minimize the total completion time and carbon emissions of the equipment joint scheduling problem of U-shaped ACTs. Xu et al. [10] investigated the unloading process of U-shaped ACTs. They established an integrated scheduling model of the QCs, AGVs, and double-cantilever rail cranes to minimize the processing time and a conflict-free path planning model to minimize the transportation time of the AGVs. Zhong et al. [11] constructed a mixed-integer programming model to optimize the joint scheduling of AGVs and YCs, minimizing the total energy consumption for given loading and unloading tasks. They considered conflict-free path planning for AGVs and the capacity constraints of the AGV mate, and designed a novel bi-level genetic algorithm for solving this model. Xing et al. [12] investigated the integrated scheduling problem of QCs, YCs, and a speed-optimized AGV-integrated scheduling problem, and constructed a mixed-integer programming model for minimizing the total completion time of QCs and YCs and the energy consumption of AGVs. Yue et al. [13] established a two-phase mathematical model for dual-trolley QCs and AGVs by considering constraints such as vessel stability and aiming at the total energy consumption during loading and discharging operations. Duan et al. [14] developed a two-phase mathematical model for the integrated scheduling problem for the QCs and AGVs to minimize the makespan and the unloaded time of AGVs. The related research results are valuable for the improvement of the overall operation efficiency and energy consumption optimization of terminals, but there is still a lack of specific depictions of the operation flow of AGV subsystems and related constraints. Meanwhile, the influence of battery swapping or battery charging constraints on the scheduling of AGVs is seldom considered.

In recent years, research in the field of AGVs has made significant progress concerning AGV battery capacity and charging mode. Shi and Liang [15] have studied the waiting time for AGVs at battery swapping stations and constructed a mixed-integer optimization model for AGV scheduling that incorporates the battery-swapping process. Ding and Chen [16] have developed a mixed-integer programming model for dynamic job scheduling of multi-loading AGVs by adopting a rolling time-domain optimization strategy based on a cycle-event hybrid drive, taking into account AGV loading capacity and battery endurance capacity as constraints. Zhou et al. [17] considered the battery endurance capacity and the difference in power consumption between the heavy and empty conditions of AGVs. The priority of the tasks was set based on the task distance and urgency, then an end condition for AGV charging was set to reduce the waiting time of AGVs. Thereafter, a scheduling model aimed at minimizing the total task completion time was established. Zhao et al. [18] considered the impact of AGV battery swapping on its operational efficiency in ACT scenarios, and proposed a two-layer scheduling model to minimize the makespan

and total battery swapping time. Bian et al. [19] proposed an event-driven approach for dispatching electric AGVs and established a distribution model with event-driven considerations for battery capacity. Xiang and Liu. [20] investigated the effects of batching swapping and plug-in charging strategies on terminal efficiency and proposed an optimal task allocation strategy for AGVs that accounts for battery management. Based on the principle of a blocking-type hybrid flow shop problem, Sun et al. [21] established a multi-resource collaborative scheduling optimization model which considered AGV pooling efficiency and charging effects. This model aimed to minimize the makespan of QCs and transportation energy consumption of AGVs. Wang et al. [22] considered the impact of wind power and logistics scheduling on terminal AGV charging strategies, and proposed a joint economic optimization method for AGV logistics scheduling and orderly charging. A joint optimization model to minimize the cost of purchasing electricity at the terminal was established. Li et al. [23] constructed a two-stage stochastic programming model for the joint scheduling problem of battery swapping and task operation with random tasks and employed a double-threshold constraint for battery swapping decision-making to enhance AGV utilization. Ma et al. [24] investigated a multi-AGV task scheduling problem with consideration of battery capacity. To minimize the completion times of AGVs, they created a mixed-integer programming model, and proposed a charging strategy for AGVs in split congestion road networks. Nevertheless, the aforementioned studies primarily concentrate on either the charging modes or the swapping modes, with less attention paid to the AGV scheduling problem in the hybrid mode of battery swapping and charging.

To enhance the flexibility of electrical vehicle scheduling, the field of electrical vehicles has delved into the problem of scheduling for hybrid mode battery swapping and battery recharging. Mao et al. [25] proposed a vehicle scheduling optimization model that considered partial recharging and battery swapping strategies, and designed experiments to compare the results under multiple recharging option strategies. The findings revealed that the combination of these two strategies can effectively reduce costs. Raeesi et al. [26] proposed an ECV path problem with a time window, recharging, and battery swapping vehicles to optimize the number of vehicles and travel distances. Ferro et al. [27] considered the selection of different types of charging stations, and developed a mixed-integer programming model, which was used to minimize the traveling distance cost and energy costs. Kumar et al. [28] studied the electric vehicle path problem with time windows, modeling the problem with battery swapping, partial charging, charging flexibility, and energy prices for different charging levels. The results related to electric vehicle scheduling verify the effectiveness of the hybrid mode of battery swapping and charging and provide an important reference for the optimization of AGV scheduling in ACTs under the battery swapping and charging mode.

From the above literature, it is found that some progress has been made in AGV scheduling in the field of ACTs; however, studies on AGV scheduling in the hybrid mode of battery swapping and charging has not been conducted. The research gap can be summarized as follows:

- (1) Although the advantage of hybrid mode of simultaneous swapping and charging has been verified, the application scenario of the electrical vehicle scheduling problem is obviously different from that of AGV scheduling under the scenario of ACTs;
- (2) Due to the complex operation process and frequently assigned scheduling tasks, there are higher requirements for the optimization performance and optimization efficiency of AGV scheduling.

Therefore, to ensure the efficiency of ACTs, it is vital to study the AGV scheduling of ACTs considering the hybrid mode of battery swapping and charging. This study selects the Shanghai Yangshan Port Phase IV as an example of ACTs that aim to decrease the impact of the power replenishment process on the AGV scheduling, thus enhancing operational efficiency and reducing the energy consumption of the terminal.

The contributions of this paper are summarized as follows:

- (1) An opportunity charging mode is introduced for ACTs, combining the layout of ACTs with the unique characteristics of battery-swapping AGVs;
- (2) To minimize the energy consumption and delay costs of the AGVs, a hybrid mode of battery swapping and charging based on the AGV scheduling problem is proposed, and then a mathematical programming model is established;
- (3) An adaptive large-neighborhood search algorithm is proposed to provide effective decision support for AGV scheduling in the hybrid mode.

The remaining parts of the paper are as follows. We provide the problem analysis and description in Section 2. Section 3 establishes the mathematical model of the focused problem. Section 4 illustrates the provided algorithm to solve the mathematical model, followed by the numerical experiments in Section 5 and the conclusions in Section 6.

2. Problem Description

2.1. Problem Analysis

This study investigates the AGV scheduling problem in the context of the hybrid mode of battery swapping and charging, which is based on the layout of the ACT of Shanghai Yangshan Port Phase IV. The terminal includes three main areas, which are the seaside operation area, the yard operation area, and the horizontal transportation area. The layout is depicted in Figure 1.

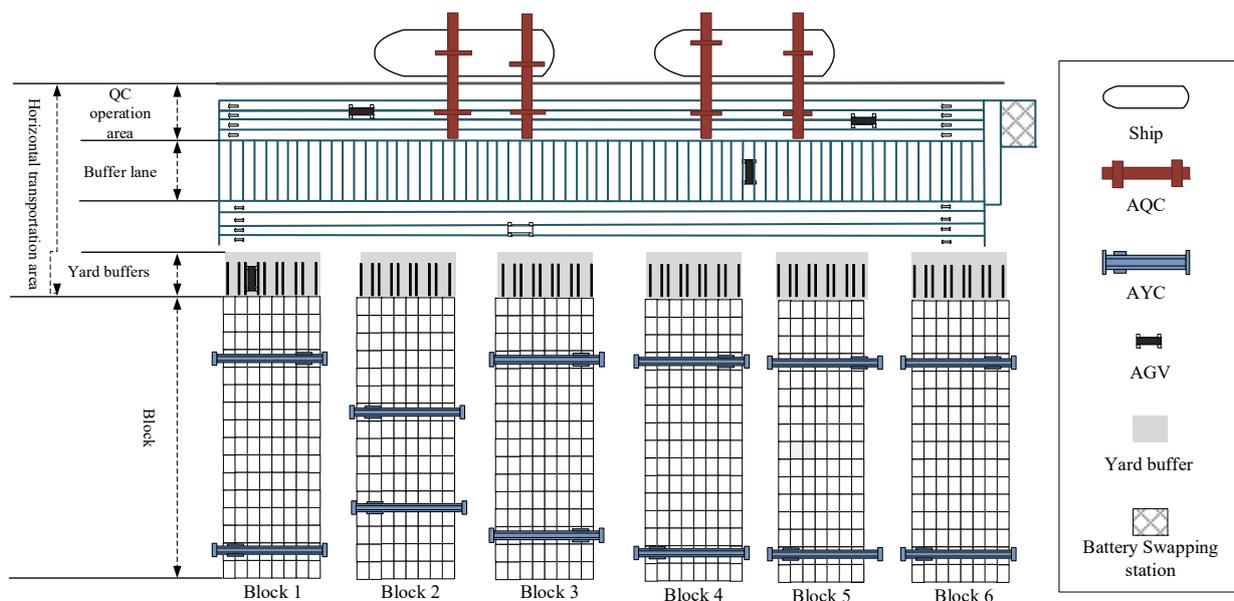


Figure 1. The schematic layout of Shanghai Yangshan Port Phase IV.

The problem scenario targeted by this study has the following typical features: the blocks of the ACT are perpendicular to the shoreline; there is a buffer in front of each block; there are five AGV mates in each of the buffers on which AGVs or AYCs can temporarily place containers; charging facilities are configured for each of these AGV mates to allow AGVs to stay and charge; the area between the AQCs and the buffers and its intermediate area is the traveling area of the AGVs; a battery swapping station is set up on the right side of the ACT; AGVs carry out charging in the buffers of the yard when the power level of their batteries is higher than the swapping threshold; and AGVs go to the battery swapping station to replace a battery when the power level is lower than a predefined threshold.

2.2. Energy Consumption Evaluation

The energy consumption formula proposed by Bektaş et al. [29] was adopted to evaluate the AGV energy consumption. To calculate the mechanical power on each path (i, j) of the AGV, the average speed is set to be s_{ij} , and the total load as $M = w + L_{ij}$, where

L_{ij} is the load carried by the vehicle on the path and w is the mass of the vehicle itself. The length (i, j) of the path is denoted as d_{ij} and the mechanical energy ME_{ij} required on the path is calculated by Equation (1),

$$ME_{ij} \approx P_t(d_{ij}/s_{ij}) = \alpha_{ij}(w + L_{ij})d_{ij} + \beta s_{ij}^2 d_{ij} \tag{1}$$

where α_{ij} is the arc-specific factor, β is the vehicle-specific factor, and the unit of mechanical energy is kWh.

The electrical energy consumption of the AGV is calculated by converting mechanical energy into electrical energy of the battery as in Equation (2),

$$E_{ij} = eff_d \cdot eff_m \cdot ME_{ij} = eff_d \cdot eff_m \cdot [\alpha_{ij}(w + L_{ij})d_{ij} + \beta s_{ij}^2 d_{ij}] \tag{2}$$

where eff_m denotes the motor efficiency and eff_d denotes the battery discharge efficiency. The calculation of the loading energy consumption w_i is related to the loading distance between the start and end positions of task i , the loading speed, and the amount of loading, which is calculated by Equation (3).

$$w_i = eff_d \cdot eff_m \cdot [\alpha_i(w + L_i)D_i + \beta s_i^2 D_i] \tag{3}$$

The calculation of loading task energy consumption is based on Equation (3), while the no-load energy consumption is determined by setting the load L_{ij} to zero and updating the distance D_i travelled by i to the distance d_{ij} travelled from the end position of task i to the start point of task j .

2.3. AGV Scheduling Based on the Hybrid Mode

The tasks performed by AGVs were classified as unloading tasks, loading tasks, and battery swapping tasks. The location of the tasks from the start point to the end point was known and path conflicts were not considered. The operation time of the loading task is the sum of the time when the AQC picks up the container from the AGV and the time when the AGV picks up the container from the AGV mate. The operation time of the unloading task is the sum of the time the AQC delivers the container to the AGV and the time the AGV delivers the container to the AGV mate.

The loading and unloading vessel plan provides a time window $[e_i, l_i]$ for each loading and unloading container task. The notation e_i denotes the earliest available start time for the task i , and l_i denotes the latest available start time for the task i . Since the scheduling priority of the AQCs and AYC is higher than that of the AGVs, when the AGV executes task i later than the latest available start time of the task, it will cause delays in the scheduling of the AQCs or AYC as well as delays in the subsequent AGV tasks that may result in delay costs. Discretizing the swapping tasks at the battery swapping station, the duration of each swapping task is known, and each swapping task can only serve one AGV that is short of power at the same time.

To determine the timing of charging, four types of articulation of AGV loading and unloading tasks are analyzed: an AGV performs a loading task after performing an unloading task; an AGV performs a loading task after performing a loading task; the AGV performs an unloading task after performing an unloading task; and the AGV performs an unloading task after performing a loading task.

As shown in Figure 2, tasks 1 and 3 are unloading tasks, and tasks 2 and 4 are loading tasks. The time windows and paths of all the tasks are known. After executing unloading task 1, AGV A immediately goes to the starting position of task 2, and assuming that AGV A arrives at the starting position of task 2 in 220 s, AGV A can make use of the waiting time of 30 s before executing task 2 to carry out charging operations. AGV B travels to the start position of task 3 immediately after executing task 4. Since the task is not conducted on the buffer side, it is not possible to perform power replenishment. Assuming that AGV B

arrives at the start position of task 3 in 290 s, the delay time for AGV B to execute task 3 is 10 s.

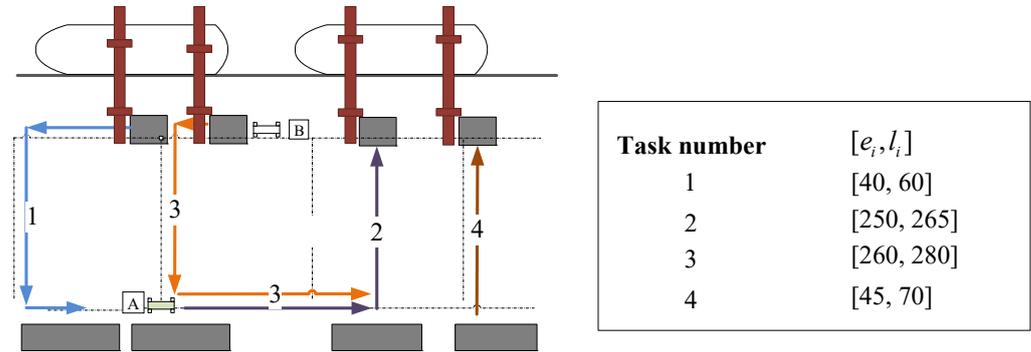


Figure 2. Schematic diagram of the loading and unloading tasks.

From the above analysis, it was found that optimizing the AGV task allocation, sequence of operations, and timing of charging and swapping can effectively reduce the possibilities of no-loading and delays of the AGVs. The objective is to schedule the sequence of loading and unloading tasks of AGVs and coordinate the recharging and swapping process so that the AGVs can complete the specified container delivery tasks in a given time window and minimize the energy consumption and the delay cost of the AGVs.

3. Mathematical Modelling

3.1. Assumptions

The AGV scheduling problem for an ACT in the hybrid mode of battery swapping and charging satisfies the following assumptions:

- (1) Each AGV has the same specifications and the same size;
- (2) Each AGV transports only one container at a time during the execution of a task;
- (3) Each AGV travels at a common speed between any two nodes;
- (4) During the scheduling plan period, the time window and the start and end position of each container task are known;
- (5) In this scheduling problem, the conflicts of AGVs on the road are not considered;
- (6) The batteries in the battery swapping station are sufficient in number and fully charged;
- (7) The charging facility is set at the AGV mate in the buffer of the yard;
- (8) The charging mode of the AGVs is linear charging, and the amount of charging is proportional to the charging time.

To accurately model the AGV scheduling problem under the hybrid mode of battery swapping and charging, it is necessary to define the studied problem using the above assumptions. For example, the type of AGVs and their speed were set at the same value. One standard container can be transported by one AGV at a time. Although many uncertainties exist in real-world ACTs, studying a mathematical model is still an important means to propose an effective scheduling solution to reduce the impact of AGV power supplementation on the efficiency of automated container terminals.

3.2. Notations

The notation variables and corresponding definitions are shown in Table 1.

The decision variables are shown in Table 2.

Table 1. Definition of notation variables.

Notation	Description
I_{load}	The set of loading tasks
I_{unload}	The set of unloading tasks
I_r	The set of unloading and loading tasks
I_0	The dummy starting task
I'_0	The dummy ending task
I_t	The set of battery-swapping tasks
I	The set of all tasks
i, j	The index for task, i, j are positive integers
K	The set of AGVs
k	The index for AGVs; k is a positive integer
e_i	The earliest starting time of unloading and loading task i
l_i	The latest starting time of unloading and loading task i
y	The time for a AQC to deliver to or pick up containers from an AGV
q	The time for an AGV to deliver to or pick up containers from an AGV mate
t^{charge}	The time for an AGV to swap a battery at the battery swapping station
h_i	The execution time for task i
z_{ij}	The no-load energy consumption for travelling from the end position of task i to the start position of task j
w_i	The loading energy consumption of task i
C	The total power of the AGV
c	The charging rate
E_{max}	The AGV battery swapping threshold
b_k	The initial power of AGV k
v_1	The no-load speed of an AGV
v_2	The load speed of an AGV
d_{ij}	The no-load distance travelled by an AGV from the end position of task i to the start position of task j
D_i	The load distance for an AGV performing task i
c_e	The energy cost of one unit
c_t	The penalty cost per time unit

Table 2. Definitions of decision variables.

Decision Variables	Description
x_{ijk}	0–1 variable, which is 1 when AGV k executes task j immediately after task i , and 0 otherwise
y_{ik}	0–1 variable, which is 1 when AGV k performs task i and 0 otherwise
g_{ik}	The remaining power of AGV k after the execution of task i
s_{ik}	The start time of AGV k performing task i
u_{ik}	The wait time of AGV k performing task i
τ_{ik}	The delay time of AGV k performing task i

3.3. The Mathematical Model

In the AGV scheduling problem of ACTs with the hybrid mode of battery swapping and charging, two main optimization objectives are considered and transformed as cost functions. The first objective is to minimize the electrical energy consumption, including both stages of the loading and unloading transportation of AGVs. The second objective is to minimize the penalty cost if the task cannot be started within the given time window. Therefore, the mathematical model of the AGV scheduling considering the hybrid mode is shown as follows.

Objective function:

$$\min f = \{c_e(\sum_{i \in I} \sum_{j \in I} \sum_{k \in K} z_{ij} x_{ijk} + \sum_{i \in I} \sum_{k \in K} w_i y_{ik}) + c_t \sum_{i \in I} \sum_{k \in K} \tau_{ik}\} \quad (4)$$

Subject to:

$$\sum_{k \in K} y_{ik} = 1, \forall i \in I_r \quad (5)$$

$$\sum_{k \in K} y_{ik} \leq 1, \forall i \in I_t \tag{6}$$

$$y_{ik} = \sum_{j \in I} x_{jik}, \forall i \in I_r \cup I_t, \forall k \in K \tag{7}$$

$$x_{iik} = 0, \forall i \in I, \forall k \in K \tag{8}$$

$$\sum_{j \in I} x_{ijk} = 1, \forall i \in I_0, \forall k \in K \tag{9}$$

$$\sum_{i \in I} x_{ijk} = 1, \forall j \in I'_0, \forall k \in K \tag{10}$$

$$\sum_{i \in I} x_{ijk} - \sum_{h \in I} x_{jhk} = 0, \forall j \in I_r \cup I_t, \forall k \in K \tag{11}$$

$$e_i \leq s_{ik}, \forall i \in I, \forall k \in K \tag{12}$$

$$s_{ik} + h_i + D_i/v_2 + d_{ij}/v_1 - M(1 - x_{ijk}) \leq s_{jk}, \forall i, j \in I, i \neq j, \forall k \in K \tag{13}$$

$$u_{ik} = (s_{jk} - s_{ik} - h_i - D_i/v_2 - d_{ij}/v_1)x_{ijk}, \forall i, j \in I, i \neq j, \forall k \in K \tag{14}$$

$$\tau_{ik} \geq s_{ik} - l_i, \forall i \in I, \forall k \in K \tag{15}$$

$$g_{ik} = b_k, \forall i \in I_0, \forall k \in K \tag{16}$$

$$g_{ik} = C, \forall i \in I_t, \forall k \in K \tag{17}$$

$$g_{jk} - qc + w_j \leq g_{ik} - z_{ij} + M(1 - x_{ijk}), \forall i \in I, j \in I_{unload}, \forall k \in K \tag{18}$$

$$g_{jk} - qc + w_j \geq g_{ik} - z_{ij} + M(x_{ijk} - 1), \forall i \in I, j \in I_{unload}, \forall k \in K \tag{19}$$

$$g_{jk} - qc + w_j \leq g_{ik} - z_{ij} + cu_{jk} + M(1 - x_{ijk}), \forall i \in I, j \in I_{load}, \forall k \in K \tag{20}$$

$$g_{jk} - qc + w_j \geq g_{ik} - z_{ij} + cu_{jk} + M(x_{ijk} - 1), \forall i \in I, j \in I_{load}, \forall k \in K \tag{21}$$

$$E_{max} - (g_{ik} - z_{ij}) + M(1 - x_{ijk}) \geq 0, \forall i \in I, \forall j \in I_t, \forall k \in K \tag{22}$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in I, \forall k \in K \tag{23}$$

$$y_{ik} \in \{0, 1\}, \forall i \in I, \forall k \in K \tag{24}$$

$$0 \leq g_{ik} \leq C, \forall i \in I, \forall k \in K \tag{25}$$

$$s_{ik} \geq 0, \forall i \in I, \forall k \in K \tag{26}$$

$$u_{ik} \geq 0, \forall i \in I, \forall k \in K \tag{27}$$

$$\tau_{ik} \geq 0, \forall i \in I, \forall k \in K \tag{28}$$

where Equation (4) is the objective function that aims to minimize the sum of the energy consumption cost and delay cost of the AGVs. The energy consumption cost consists of loading energy consumption and no-load energy consumption. The delay cost is the cost generated when the task start time exceeds its latest available start time. Constraint (5) ensures that each loading and unloading task can only be executed once by one AGV. Constraint (6) ensures that each battery-swapping task can be performed at most once by one AGV. Constraint (7) guarantees that for any loading/unloading or battery swapping task, there is at most one immediately preceding task. Constraint (8) guarantees that tasks cannot be self-connected. Constraints (9) and (10) ensure that each AGV starts from the execution of a dummy starting task and ends with the execution of a dummy ending task. Constraint (11) is an arc balance constraint that ensures that the AGV should satisfy arc balance during the execution of loading, unloading, and battery-swapping tasks. Constraint (12) ensures that the start time of task i execution by AGV k is greater than or equal to the earliest time window of task i . Constraint (13) restricts the feasibility of the sequential processing time for AGV k to execute tasks i, j . If AGV k executes task j immediately after task i , that is, $x_{ijm} = 1$, the constraint (13) becomes $s_{ik} + h_i + D_i/v_2 + d_{ij}/v_1 \leq s_{jk}$. Constraint (14) calculates the waiting time before AGV k executes task j . If AGV k executes task j immediately after task i , that is, $x_{ijm} = 1$, the constraint (14) becomes $u_{jk} = s_{jk} - s_{ik} - h_i - D_i/v_2 - d_{ij}/v_1$. Constraint (15) defines the delay time for AGV k to execute task i . Constraints (16) and

(17) denote the remaining power constraints after AGV k executes the dummy starting task and the battery swapping task, respectively. Constraints (18) and (19) denote the remaining power constraints before the AGV k performs the unloading task j . If AGV k executes unloading task j immediately after task i , that is, $x_{ijm} = 1$, constraints (18) and (19) become $g_{jk} - qc + w_j = g_{ik} - z_{ij}$. Constraints (20) and (21) represent the remaining power constraints before the AGV k performs the loading task j , when the immediately following task j is a loading task and the AGV k can use the waiting time for opportunity charging. When $x_{ijm} = 1$, the constraint (20) becomes $g_{jk} - qc + w_j = g_{ik} - z_{ij} + cu_{jk}$. Constraint (22) indicates that the AGV k needs to have enough power to travel to the battery swapping station for the battery swapping task. That is, the AGV k battery swapping threshold must be greater than or equal to the remaining power of the AGV k when completing task i minus the no-load energy consumption of AGV k from the end position of task i to the start position of task j . Constraints (23) to (28) represent the range of values of the decision variables.

4. An Adaptive Large-Neighborhood Search Algorithm

The AGV scheduling problem modeled in this paper is a mixed-integer programming model with a large number of decision variables and constraints, which is difficult to solve in a reasonable time using commercial solvers, so an adaptive large-neighborhood search algorithm is proposed.

4.1. Framework of the Proposed Algorithm

The adaptive large-neighborhood search (ALNS) is a heuristic algorithm proposed by Ropke and Pisinger [30], which has been applied in the fields of vehicle path optimization and ACT scheduling [31–33]. This study adopts the architecture of the ALNS algorithm shown in Figure 3.

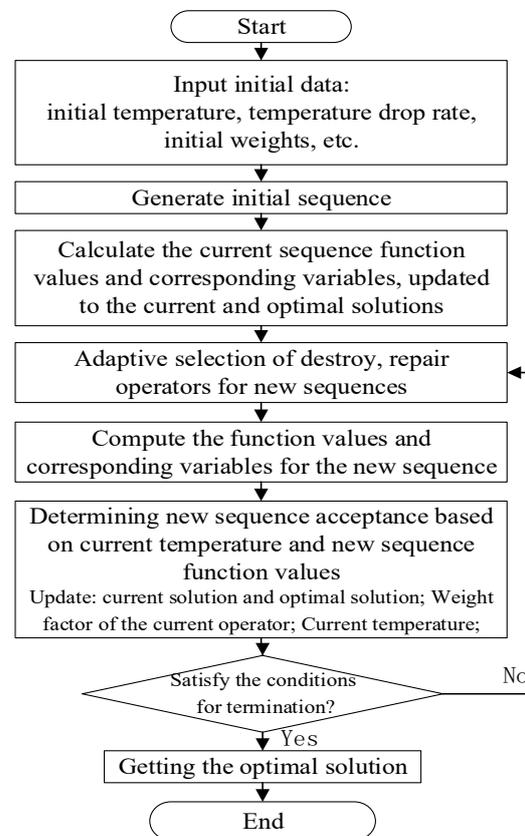


Figure 3. Flowchart of ALNS algorithm.

The framework of ALNS consists of five parts: the construction of the initial solution, the selection of the destroy and repair operators for destroying and repairing the solutions, the updating of the weights of the operators, the adaptive selection of the operators, and the acceptance criterion of the solutions. To solve the proposed mathematical model effectively, the coding process is developed by considering all the constraints of the AGV scheduling problem in the focused ACT, by which the effectiveness of the scheduling sequence is ensured. Moreover, the initialization process considers the time windows of the container tasks as well as the battery-swapping process and the opportunity charging process, by which the accuracy of the scheduling decision can be ensured. The destroy and repair approach is designed according to the characteristics of the studied model. In addition, the acceptance criterion of the solutions is the simulated annealing algorithm that permits solutions that are slightly worse than the current solutions. Then, the algorithm can jump out of local optima. The ALNS stops either when the number of non-improvement temperature decrements exceeds a specified number or after having run the maximum number of allowed iterations.

4.2. Coding for the AGV Scheduling

The solution encoding and decoding approach is described through the attributes and sequence of tasks. The sequence of tasks consists of three parts, i.e., the dummy tasks, the loading and unloading tasks, and the battery-swapping tasks. The task sequence chromosome is encoded using natural numbers, each chromosome sequence represents a set of loading and unloading tasks, each natural number in the chromosome represents the index of a loading and unloading task, and the length of the chromosome is equal to the number of all loading and unloading tasks. In this study, the loading and unloading types of the loading and unloading tasks are known, and the numbers of the operational AQC and block of each loading and unloading task are known. Taking eight loading and unloading tasks, two operational AQC, and four operational blocks as an example, the initial task sequence is a chromosome whose length is the number of loading and unloading tasks, and the encoding is shown in Figure 4.

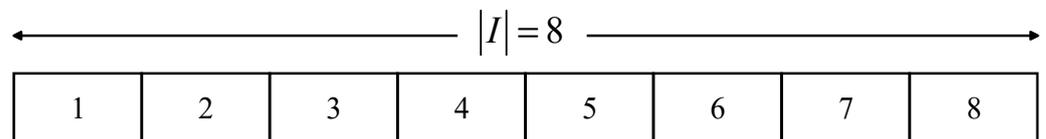


Figure 4. Schematic diagram of coding.

The tasks are assigned to AGVs and executed sequentially based on the attributes of the tasks, time window constraints, and energy consumption. A task sequence represents the traveling route of an AGV k , and the total number of sequences is the same as the total number of AGVs, which is K . All AGVs start from the dummy starting task, execute the tasks sequentially according to the time window constraints and power constraints in the order of the tasks within the route, and return to the dummy ending task. The AGV calculates whether its remaining power is below a threshold or cannot complete the next task after each task, inserting a battery-swapping task if it is, or continuing to execute the next task if the power is sufficient. It should be noted that the dummy starting task and dummy ending task are not involved in the destruction and repair step since they are the dummy tasks, and can be executed repeatedly.

After assigning AGVs to each loading and unloading task, the sequence of the tasks to be accomplished by each AGV is shown in Figure 5 below:

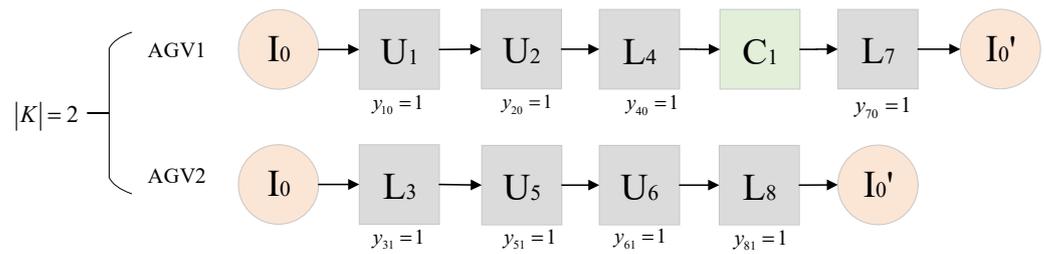


Figure 5. Decoding schematic.

After the decoding process, the tasks are assigned to the AGVs, and the task sequences of different AGVs are generated. C1 denotes the inserted battery swapping task. The task route of AGV1 is $I_0 \rightarrow U_1 \rightarrow U_2 \rightarrow L_4 \rightarrow C_1 \rightarrow L_7 \rightarrow I_0'$, and the task route of AGV2 is $I_0 \rightarrow L_3 \rightarrow U_5 \rightarrow U_6 \rightarrow L_8 \rightarrow I_0'$.

4.3. Initial Solutions

The initial solution construction process considers time windows and power constraints to ensure the feasibility of the initial solution in the following steps:

- (1) Let the increment of the objective function value of the sequence of tasks inserted by task i into AGV k be Δf_{ik} ; different insertion positions lead to different increments of the objective function value, define $c_i = \min\{\Delta f_{ik}\}, \forall i \in I, \forall k \in K$ and take the minimum value of c_i corresponding to AGV k as the best choice for task i .
- (2) Iterate all tasks i in the set of loading and unloading tasks and insert them into the AGV task sequence.
- (3) Update the remaining power g_{ik} of the AGV after task i is inserted into AGV k . The update of the power g_{ik} is divided into the power update after battery swapping and the power update after charging, where the power update of charging is divided into fixed charging and opportunity charging. Fixed charging is the charging of AGV k that utilizes the interaction time between the buffer in the yard and the AGV mate, assuming that the last task is j and the current task is i . The remaining power is calculated by Equation (29):

$$g_{ik} = g_{jk} + qc - z_{ji} \tag{29}$$

where Equation (29) indicates that the remaining power g_{ik} of the AGV k after executing task i is the remaining power of the AGV k after executing the last task j minus the no-load energy consumption of the AGV k traveling from the ending position of task j to the starting position of task i , plus the supplemental power of the fixed charging time.

Opportunity charging needs to determine the loading and unloading container attribute of task i . If task i is a loading task L , the power is increased according to the early arrival time; if not, the power remains unchanged and the power after opportunity charging is calculated by Equation (30):

$$g_{ik} = g_{jk} + cq - z_{ji} + cu_{ik} \tag{30}$$

where Equation (30) represents the remaining power g_{ik} of the AGV k after executing task i , which equals to the remaining power of the AGV k after executing the task j minus the no-load energy consumption of the AGV k traveling from the ending position of task j to the starting position of task i , plus the supplemental power for the fixed charging time and the supplemental power utilizing the waiting time before executing task i .

The power update after battery swapping needs to determine whether the remaining power of AGV k is sufficient to travel to the battery-swapping station, as shown in inequality (31):

$$g_{jk} - z_{jh} \leq E_{\max} \tag{31}$$

where the inequality (31) indicates whether the remaining power of AGV k after executing the last task j minus the no-load energy consumption of AGV k traveling to the battery-swapping station is less than or equal to the swapping threshold, and if so, if it is necessary to go to the battery-swapping station to swap the battery.

- (4) Update the start time and end time of task i based on the time window of the task and the position of AGV k . Define s_{ik} as the start time of executing task i , o_{ik} as the end time of executing task i , and t_{ij} as the no-load time from the ending position of task i traveling to the starting position of task j .

When AGV k executes task i after task j , the expression for the start time s_{ik} of task i is updated as

$$s_{ik} = \max \{ o_{jk} + t_{ji}, e_i \} \tag{32}$$

The end time o_{ik} of the execution of task i by AGV k is updated by Equation (33),

$$o_{ik} = s_{ik} + h_i + D_i/v_2 \tag{33}$$

- (5) Based on the updated power and time in steps (4) and (5), calculate the increment Δf_{ik} of the objective function value for each task i to be inserted into the AGV k , and keep iterating until all the tasks are assigned to the AGV.

4.4. Destroy Operators

After constructing an initial sequence of solutions, the algorithm removes a certain number of tasks from the current sequence by using a destroy operator and saves them in an unsorted list L of deleted tasks. The number of tasks p to be deleted by the destroy operators is determined by the size of the problem, setting the real number range of the number of deleted tasks p per iteration to $[0, q]$, where the upper limit $q = \rho n$, n denotes the total number of loading and unloading tasks, and ρ denotes the removal rate parameter. Four destroy operators were designed in this study:

- (1) The random destroy operator

The random destroy operator is a random selection of the number p of tasks to be removed from the task sequence and present in the list L of deleted tasks.

- (2) The worst target destroy operator

The destroying strategy of the worst target destroy operator is to compute the impact of each task i on the objective function; the higher the impact, the more it needs to be destroyed and reinserted, thus selecting p tasks with the highest impact for removal.

Define the insertion cost of each task i as $\text{cost}(i, s) = f(s) - f_{-i}(s)$, where $f(s)$ is the current objective function value and $f_{-i}(s)$ is the objective function value after removing task i . Sort $\text{cost}(i, s)$ in descending order, corresponding to each task i , to obtain a sequence of tasks. Select p tasks that have the largest impact on the objective function value to be removed from the sequence.

- (3) The worst time destroy operator The worst time destroy operator considers the time window constraints and removes the tasks that have a long waiting time before the AGV executes task i . This operator serves to minimize the time delay cost of the AGV. The main operation flowchart of this operator is as follows:

- (1) Calculate the waiting time $u_{ik} = s_{ik} - a_{ik}$ for each task, where s_{ik} is the time when AGV k starts to execute task i , and a_{ik} is the time when AGV k reaches the starting position of task i .
- (2) Sort the waiting times u_{ik} in descending order according to the magnitude of the values, which corresponds to the sequence of all the tasks i , and remove the first p tasks.

(4) The similarity destroy operator

The similarity destroy operator was proposed by Shaw [34], which serves to remove a set of tasks that have similar properties and can be easily exchanged. During the destruction process, pairs of tasks with high similarity need to be prioritized to obtain a better solution. Define the similarity value of two tasks i, j as $R(i, j)$, and calculate the similarity value between tasks i, j by Equation (34):

$$R(i, j) = \delta_1(d_{A(i),A(j)} + d_{B(i),B(j)}) + \delta_2(|e_i - e_j|) + \delta_3|L_i - L_j| \tag{34}$$

There are three main influencing factors for similarity judgment: similarity of task start and end positions, similarity of time, and similarity of load. $\delta_1, \delta_2, \delta_3$ represent the weight coefficients of the three influencing factors, respectively. $A(i), B(i)$ represent the start position and end position of task i , and L_i is the container load of task i .

The main operation process of this operator is as follows:

- (1) Select a task i in the task sequence randomly, and calculate the similarity between task i and all other tasks;
- (2) Sort the similarities in ascending order to obtain a new task sequence;
- (3) Define the random parameter $y \in [0, 1)$, and randomize the control parameter n , then the y^n task in the task sequence is removed;
- (4) Iterate sequentially until the p tasks with the largest similarity are selected to be removed from the sequence, and added to the list L of removed tasks.

4.5. Repair Operator

After the destruction of the solution using the destroy operator, it is necessary to reinsert these deleted tasks back into the sequence of tasks of the solution in a certain way. Therefore, two repair operators are designed to obtain a new feasible solution:

- (1) The greedy repair operator employs a greedy heuristic algorithm that, after inserting a task i into a path k , computes the increment Δf_{ik} of the increased target value, and if it is not possible to insert the path k , then it sets $\Delta f_{ik} = \infty$. Insert task i to the position that minimizes $c_i = \min\{\Delta f_{ik}\}$. Iterate this process until all tasks p are inserted into the path.
- (2) The regret value repair operator is an improvement of the greedy algorithm, the operator decides its insertion position according to the regret value generated after the task insertion, described as the difference between the cost of inserting the task into the optimal position and the sub-optimal position in the task sequence. The higher the regret value, the higher the difference between the insertion cost of the optimal and sub-optimal positions. The regret value is calculated by Equation (35):

$$c_i^* = \Delta f_{i2} - \Delta f_{i1} \tag{35}$$

where Δf_{i1} and Δf_{i2} denote the value added to the objective function after inserting task i into the optimal and sub-optimal positions, respectively. The regret values are sorted in a descending manner, and the p tasks with the highest regret values are selected in turn to be inserted into the optimal position.

4.6. Adaptive Selection Mechanism

The ALNS needs to select one of the destroy and repair operators in each iteration to realize adaptive adjustment. To dynamically adjust the selection probability of each operator, a weight factor scoring and adaptive probability selection mechanism is used. The weight factor scoring mechanism tracks the score of each heuristic operator to measure the performance of that operator.

Assuming that each operator has an initial score of 0 before an iteration, the operator increases the corresponding score after each iteration based on the following three scenarios: if the solution obtained in that iteration is a new global optimum, the operator's score

is increased by σ_1 ; if the solution obtained in that iteration is not better than the global optimum but better than the local optimum, the operator's score is increased by σ_2 ; and if the solution obtained in that iteration is worse than the solution obtained before the iteration, but still accepts this worse solution after passing the solution acceptance criterion, the iteration process will increase the operator's score to zero. If the algorithm still accepts this worse solution after passing the solution acceptance criterion, the score of the operator is increased by σ_3 during the iteration.

An operator scoring mechanism is used, where the recorded scores are used after each iteration to compute new weights. Define ω_{ij} as the weighting factor of operator i in the j th iteration. The initial weight of each operator is set to 1. At the end of each search, the adaptive weight ω_{ij} of each operator is updated based on the scores of each operator in this search, which is calculated by Equation (36):

$$\omega_{i,1+j} = \begin{cases} (1 - a)\omega_{ij} + a \frac{\pi_{ij}}{\theta_{ij}}, & \theta_{ij} > 0 \\ \omega_{ij}, & \theta_{ij} = 0 \end{cases} \tag{36}$$

where π_{ij} denotes the score of operator i in the j th iteration, θ_{ij} denotes the number of adaptive selections of operator i in the j th iteration, and $a \in [0, 1]$ is the speed factor of weight adjustment. The higher the score and the better the performance of the operator in each iteration, the larger the percentage of the operator after the weight update, and the easier it is to be selected by roulette in the next iteration with a higher probability.

4.7. Acceptance Criteria

The adaptive large-neighborhood search algorithm is allowed to accept worse solutions, the probability is calculated by Equation (37):

$$p = \begin{cases} e^{-(f(s^{new}) - f(s^{cur})) / T^{now}}, & f(s^{new}) < f(s^{cur}) \\ 1, & f(s^{new}) > f(s^{cur}) \end{cases} \tag{37}$$

When the new solution s^{new} is better than s^{cur} , the probability $p = 1$, i.e., the new solution s^{new} is accepted; when the new solution s^{new} is worse than s^{cur} , the probability $p = e^{-(f(s^{new}) - f(s^{cur})) / T^{now}}$ is used to decide whether to accept the new solution s^{new} or not, where T^{now} denotes the temperature of the simulated annealing acceptance criterion at the current iteration, and $T^{now} > 0$. An initial temperature $T_{start} = -\frac{\lambda}{\ln 0.5} f(s)$ needs to be generated before the start of the iteration such that the newly generated feasible solution in the initial condition is still accepted with a probability of 50% when it is worse than the initial solution with a value of λ . In each iteration, the annealing temperature T in the acceptance criterion decreases continuously at the rate of the equation $T = T \cdot v$, where v is the simulated annealing cooling rate and takes values in the range $0 < v < 1$.

5. Computational Experiments

To verify the effectiveness of the proposed AGV scheduling model and the ALNS algorithm, numerical experiments of two scales were designed for comparative analysis. Furthermore, the effectiveness of the model was analyzed by designing experiments related to the charging and battery-swapping factors. The numerical experiments were carried out on a Huawei MateBook14 laptop with an Intel Core i5 2.11 GHz processor and 16 GB of RAM. The model was validated using C# calling ILOG CPLEX 12.5, and the LNS algorithm, which does not include the operator adaptive selection mechanism, was used for comparative analysis. The algorithm parameters and the value settings are shown in Table 3.

Table 3. The parameter settings of the ALNS.

Parameters	Description	Value
σ_1	The algorithm iterates to obtain a globally optimal solution score	6
σ_2	The algorithm iterates to obtain the current optimal solution score	3
σ_3	The algorithm iterates to obtain an inferior solution score	1
δ_1	Destroy operator task distance similarity weighting factor	9
δ_2	Destroy operator task time similarity weighting factor	3
δ_3	Destroy operator task container weight similarity weighting factor	9
v	Cooling rate for simulated annealing acceptance criteria	20
λ	Initial temperature coefficients for simulated annealing acceptance criteria	0.05
a	Weight-adjusted speed factor	0.1

5.1. Instance Settings

Taking the Shanghai Yangshan Deepwater Port Phase IV ACT as an example, the AGV weighs 29 tons, of which the battery pack weighs 5.5 tons. The maximum speed of the AGV was 6 m/s (meters/seconds), and the working speed was set at 3 m/s. The maximum battery capacity of the AGVs was 300 kWh, and the threshold of the battery swapping was set at 40% of the remaining capacity of the battery. The average time to swap a battery at the battery swapping station is 300 s. The energy consumption evaluation of the AGV refers to the relevant studies by Feng and Figliozzi [35] and Murakami [36], and sets the motor efficiency at 1.25 and the battery discharge efficiency at 1.11. Referring to the study by Zhang et al. [37], the parameters in the energy consumption calculations were set to $\alpha = 5$, $\beta = 5$. The charging rate at the AGV mate in the front of the block is 0.08 kWh/s. The time for the container to be transported from the AQC transfer platform to the AGV was $U(20, 40)$ s, and the fixed time for the container to be unloaded from the AGV to the AGV mate was 25 s. The unit energy cost of the AGV is 0.8 CNY/kWh, and the cost of the unit delay time is 0.2 CNY/s. The cost of the AGV is 0.8 CNY/kWh.

The validity of the model and algorithm was verified by two groups of instances, large (L) and small (S), and the parameters of the instances are shown in Table 4.

Table 4. The parameters of the problem instances.

ID	Number of Tasks	Number of AGVs	Number of AQCs	Number of Blocks
S1	8	2	2	4
S2	10	2	2	4
S3	12	3	2	4
S4	14	3	5	10
S5	16	4	5	10
S6	18	4	5	10
S7	20	6	5	10
L1	40	8	5	10
L2	40	8	5	10
L3	60	10	10	20
L4	60	10	10	20
L5	80	10	10	20
L6	80	12	10	20
L7	100	14	10	30
L8	100	14	10	30

5.2. Computational Results

5.2.1. Experimental Results of Small-Scale Instance

Due to the NP-hard feature of the problem, the CPLEX solution time increases exponentially with the number of tasks and AGVs, which makes it difficult to obtain effective results in a reasonable time. Referring to the experimental scheme of Zhuang [28], T_{real} was set to be the running time of the CPLEX solution model, and the maximum solution time

T_{max} allowed by the corresponding algorithms was set according to the scale size of the arithmetic example.

In the small-scale instance, the CPLEX solution results were compared with the optimal solution of the proposed ALNS algorithm to verify the validity of the constructed model by using the proposed ALNS solution, and the results are shown in Table 5.

Table 5. Comparison of CPLEX and ALNS computation results.

ID	CPLEX		ALNS				GAP ₁
	$T_{real}(s)$	OBJ ₁	$T_{max}(s)$	Optimal Value	Average Value	σ^2_1	
S1	0.28	46.10	80	46.10	46.35	0.55	0.00%
S2	1.32	65.37	100	65.49	65.54	0.01	0.18%
S3	24.98	109.68	120	109.68	110.89	1.81	0.00%
S4	10.92	142.81	140	143.19	144.59	3.59	0.26%
S5	42.67	167.56	160	168.14	169.60	4.89	0.34%
S6	279.57	218.07	180	219.57	221.91	8.23	0.69%
S7	1483.99	269.56	200	274.03	276.98	11.03	1.66%
AVG	263.39	145.59	140	146.60	148.04	4.30	0.45%

$$GAP_1 = (ALNS' \text{ optimal value} - OBJ_1) / OBJ_1 \times 100\%.$$

From Table 5, in the instances where CPLEX can obtain the optimal solution, the average difference between the optimal value of ALNS and the optimal value of CPLEX is only 0.45%, and the difference in the average value is only 1.68%. The CPLEX solution further verifies the effectiveness of the model and also verifies the effectiveness of the proposed ALNS. In addition, comparing the computation times of CPLEX and ALNS, it was found that when the scale of the instance rose to 18 containers, the solution time of CPLEX increased dramatically, and when the scale of the instance reached 20 containers, it had already reached 1483.99 s, which is far more than the time for the ALNS algorithm to obtain an effective solution.

5.2.2. Experimental Results of Large-Scale Instance Experiments

The optimization of AGV scheduling in the horizontal transportation area of the terminal needs to consider the timeliness; however, during the solution test of CPLEX, when the scale of the instance reached 40 containers, even if the solution time was set to 7200 s, CPLEX still failed to obtain an acceptable solution.

Therefore, to further verify the effectiveness of the adaptive mechanism in the proposed ALNS algorithm, simulation experiments were carried out on eight groups of large-scale instances, which were compared with the LNS algorithm without an adaptive mechanism, and each group of instances was run 10 times and the optimal and average values were recorded, with the termination criterion that the optimal solution remained unchanged after 500 iterations, and the computational results are shown in Table 6.

From the data in Table 6, it can be seen that in the eight groups of experiments with larger scale of instances, the optimal value and the average value obtained using the proposed ALNS algorithm were better than the LNS algorithm, with the optimal value improving by 2.21%. The average value improved by 3.12%. In addition, the average variance of the ALNS algorithm has an obvious advantage over the LNS algorithm, and the stability of the solution performance is higher, which verifies the effectiveness of the proposed adaptive selection mechanism, which further verifies the feasibility of the ALNS for solving large-scale instances.

Table 6. Computational results of ALNS and LNS.

ID	ALNS			LNS			GAP ₁	GAP ₂
	Optimal Value	Average Value	σ^2_1	Optimal Value	Average Value	σ^2_2		
L1	475.90	484.80	60.89	489.95	504.44	78.61	−2.87%	−3.89%
L2	520.58	531.83	40.53	533.02	553.78	74.43	−2.33%	−3.96%
L3	1012.82	1044.15	471.46	1021.87	1072.65	815.67	−0.89%	−2.66%
L4	1034.62	1089.55	800.08	1050.01	1117.42	1408.05	−1.47%	−2.49%
L5	1357.36	1410.19	1529.14	1388.05	1453.56	2726.03	−2.21%	−2.98%
L6	1360.58	1415.58	1928.92	1399.53	1469.35	3686.35	−2.78%	−3.66%
L7	2787.04	2847.39	2372.12	2852.12	2946.48	3936.58	−2.28%	−3.36%
L8	2812.16	2899.27	2660.80	2883.67	2982.07	4018.73	−2.48%	−2.78%
AVG	1420.13	1465.35	1232.99	1452.28	1512.47	2093.06	−2.21%	−3.12%

GAP₁ = (ALNS’ optimal value-LNS’ optimal value)/LNS’ optimal value × 100%; GAP₂ = (ALNS’ average value – LNS’ average value)/LNS’ average value × 100%.

5.2.3. Analysis of the Effectiveness of the Hybrid Mode

In order to verify the effectiveness of the AGV scheduling method based on the charging-battery swapping hybrid mode, four sets of these large-scale instances were selected to analyze the differences between the charging–battery swapping hybrid mode proposed in this study and the existing overall battery swapping mode in the terminal. The experimental results are shown in Table 7 below.

Table 7. Comparison of charging and battery swapping modes.

ID	NC ₁	Obj ₁	NC ₂	Obj ₂	Dif ₁	Dif ₂
L1	1	475.90	2	518.86	−50.00%	−8.28%
L3	2	1012.82	4	1085.00	−50.00%	−6.65%
L5	4	1357.36	6	1491.33	−33.33%	−8.98%
L7	6	2787.04	11	2974.94	−45.45%	−6.32%
AVG	3.25	1408.28	5.75	1517.53	−43.48%	−7.20%

Dif₁ = (NC₁ – NC₂)/NC₂ × 100%; Dif₂ = (Obj₁ – Obj₂)/Obj₂ × 100%.

In Table 7, column 1 presents the number of the instance, columns 2 and 3 provide the number of AGV battery swaps and the optimal objective function value obtained in the hybrid mode of battery swapping and charging. Columns 4 and 5 show the number of AGV battery swaps and the optimal objective function value obtained in the overall battery swapping mode, and columns 6 and 7 provide the gap in the number of battery swaps and the gap in the objective function value, respectively. From the data in Table 7, it can be seen that the optimal time delay cost and no-load energy cost obtained from the AGV scheduling model in the hybrid mode are significantly reduced, and the number of AGV battery swaps and scheduling costs are better than that of the mode of battery swapping, with the average number of battery swaps reduced by 43.48% and the average objective function value optimized by 7.20%. It is further demonstrated that the power replenishment strategy based on the hybrid mode of battery swapping and charging can reduce the times of battery swapping in AGV scheduling and save the total cost.

6. Conclusions

This study focused on AGV scheduling in ACTs. The AGV scheduling problem in ACTs based on the hybrid mode of battery swapping and charging is investigated by considering the battery capacity limitation of AGVs.

Firstly, the AGV scheduling problem characteristics of the hybrid mode of battery swapping and charging are described, and the mixed-integer programming model of the problem is established to minimize the energy cost and time delay cost. Furthermore, based on the problem characteristics, an ALNS algorithm that is suitable for solving the problem is designed by considering the time window constraints and the battery swapping process.

Finally, the optimization performance of the proposed ALNS algorithm is verified through simulation experiments, along with the AGV scheduling costs under the hybrid mode of battery charging and swapping. Moreover, the single-battery swapping mode is compared to verify the validity and feasibility of the proposed AGV scheduling model based on the hybrid mode. From the simulation results, it is shown that the average number of battery swapping times is reduced by 43.48%, and the total cost is reduced by 7.2%, which provides an important theoretical basis and a case study reference for solving AGV scheduling problems under the power constraints of the ACTs.

As for the AGV scheduling problem in ACTs, the following two directions are provided for further study. First, the path planning of AGVs should be considered along with the scheduling process. The collision avoidance and collision of AGV paths were not considered in the assumptions. However, there are restrictions on intersection and lane capacity during the actual AGV transport process, and if the AGV does not have a good collision avoidance mechanism, scheduling accidents may occur. In addition, an adaptive large-neighborhood search algorithm with a better solution quality is needed. To improve the performance of the algorithm, the operator should be enhanced, and the algorithm should also have the ability to tune adaptive parameters.

Author Contributions: Conceptualization, H.H.; Data curation, J.H.; Formal analysis, H.H.; Investigation, S.X., J.H. and Y.G.; Methodology, S.X., J.H. and Y.G.; Software, J.H.; Supervision, H.H.; Validation, S.X., J.H. and Y.G.; Writing—original draft, S.X.; Writing—review and editing, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No. 72271156), and the Shanghai Municipal Commission of Science and Technology (No. 23692118400, 23692121400).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The authors confirm that the data supporting the findings of this study are available within the article.

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

References

1. Ministry of Transport; National Development and Reform Commission; Exchequer; Department of Natural Resources; Ministry of Ecology and Environment; Emergency Department; General Administration of Customs; General Administration of Market Regulation; National Railway Group. Guidelines on building world-class ports. *China Water Transp.* **2019**, *19*–22.
2. Chen, J.H.; Huang, T.C.; Xie, X.K.; Lee, P.T.W.; Hua, C.Y. Constructing Governance Framework of a Green and Smart Port. *J. Mar. Sci. Eng.* **2019**, *7*, 83. [[CrossRef](#)]
3. Kabir, Q.S.; Suzuki, Y. Comparative analysis of different routing heuristics for the battery management of automated guided vehicles. *Int. J. Prod. Res.* **2019**, *57*, 624–641. [[CrossRef](#)]
4. 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](#)]
5. Wu, S.P.; He, J.H.; Luo, X.J. Handling technology design for automated container terminal of Yangshan deepwater port phase IV project. *Port Waterw. Eng.* **2016**, *9*, 159–162+166.
6. Jin, Q.; Luo, X.J.; Chen, D.M. Layout pattern of AGV battery exchange station in ultra-large type automatic container terminal. *Port Waterw. Eng.* **2016**, *9*, 66–70.
7. Fan, H.m.; Guo, Z.F.; Yue, L.J.; Ma, M.Z. Joint Configuration and Scheduling Optimization of Dual-trolley Quay Crane and AGV for Container Terminal with Considering Energy Saving. *Acta Autom. Sin.* **2021**, *47*, 2412–2426.
8. Zhao, Q.R.; Ji, S.W.; Guo, D.; Du, X.M.; Wang, H.X. Research on Cooperative Scheduling of Automated Quayside Cranes and Automatic Guided Vehicles in Automated Container Terminal. *Math. Probl. Eng.* **2019**, *2019*, 6574582. [[CrossRef](#)]
9. Xu, B.W.; Jie, D.P.; Li, J.J.; Yang, Y.S. Green integrated scheduling of U-shaped automated terminals under uncertainty. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 4057–4068.
10. Xu, B.W.; Jie, D.P.; Li, J.J.; Zhou, Y.F.; Wang, H.L.; Fan, H.Y. A Hybrid Dynamic Method for Conflict-Free Integrated Schedule Optimization in U-Shaped Automated Container Terminals. *J. Mar. Sci. Eng.* **2022**, *10*, 1187. [[CrossRef](#)]
11. Zhong, Z.L.; Guo, Y.Y.; Zhang, J.H.; Yang, S.X. Energy-aware Integrated Scheduling for Container Terminals with Conflict-free AGVs. *J. Syst. Sci. Syst. Eng.* **2023**, *32*, 413–443. [[CrossRef](#)]

12. Xing, Z.; Liu, H.T.; Wang, T.S.; Chew, E.P.; Lee, L.H.; Tan, K.C. Integrated automated guided vehicle dispatching and equipment scheduling with speed optimization. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *169*, 102993. [[CrossRef](#)]
13. Yue, L.J.; Fan, H.M.; Zhai, C.X. Joint Configuration and Scheduling Optimization of a Dual-Trolley Quay Crane and Automatic Guided Vehicles with Consideration of Vessel Stability. *Sustainability* **2020**, *12*, 24. [[CrossRef](#)]
14. Duan, Y.T.; Ren, H.X.; Xu, F.Q.; Yang, X.; Meng, Y. Bi-Objective Integrated Scheduling of Quay Cranes and Automated Guided Vehicles. *J. Mar. Sci. Eng.* **2023**, *11*, 1492. [[CrossRef](#)]
15. Shi, N.L.; Liang, C.J. Optimization of AGV operation scheduling for automated terminal considering replacement processes. *Mod. Manuf. Eng.* **2019**, *2*, 6–11+139.
16. Ding, Y.; Chen, T. Charging Scheduling of Multi-Loading AGV Based on Rolling Time Domain Optimization Strategy. *Navig. China* **2020**, *43*, 80–85.
17. Zhou, X.F.; Chang, D.F.; Yu, F.; Gao, Y.P. Scheduling of AGV in container terminals considering charging and waiting time. *J. Shanghai Marit. Univ.* **2019**, *40*, 1–5+13.
18. Zhao, T.; Liang, C.J.; Hu, X.Y.; Wang, Y. Solution of AGV scheduling and battery exchange two layer model for automated container terminal. *J. Dalian Univ. Technol.* **2021**, *61*, 623–633.
19. Bian, Z.C.; Yang, Y.S.; Mi, W.J.; Mi, C. Dispatching Electric AGVs in Automated Container Terminals with Long Travelling Distance. *J. Coast. Res.* **2015**, *73*, 75–81. [[CrossRef](#)]
20. Xiang, X.; Liu, C.C. Modeling and analysis for an automated container terminal considering battery management. *Comput. Ind. Eng.* **2021**, *156*, 107258. [[CrossRef](#)]
21. Sun, B.F.; Zhai, G.S.; Li, S.; Pei, B. Multi-resource collaborative scheduling problem of automated terminal considering the AGV charging effect under COVID-19. *Ocean. Coast. Manag.* **2023**, *232*, 106422. [[CrossRef](#)] [[PubMed](#)]
22. Wang, X.; Wang, B.; Chen, Y.P.; Liu, H.; Ma, X.H. Joint Economic Optimization of AGV Logistics Scheduling and Orderly Charging in Low-Carbon Automated Terminal. *J. Shanghai Jiaotong Univ.* **2023**, 1–24. [[CrossRef](#)]
23. Li, L.M.; Li, Y.Q.; Liu, R.; Zhou, Y.M.; Pan, E.R. 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](#)]
24. Ma, N.L.; Hu, Z.H. Congestion-aware AGV charging strategy in automated container terminal. *Comput. Integr. Manuf. Syst.* **2022**, 1–22. Available online: <http://kns.cnki.net/kcms/detail/11.5946.TP.20220305.1541.010.html> (accessed on 1 February 2024).
25. Mao, H.T.; Shi, J.M.; Zhou, Y.Z.; Zhang, G.Q. The Electric Vehicle Routing Problem with Time Windows and Multiple Recharging Options. *IEEE Access* **2020**, *8*, 114864–114875. [[CrossRef](#)]
26. Raeesi, R.; Zografos, K.G. Coordinated routing of electric commercial vehicles with intra-route recharging and en-route battery swapping. *Eur. J. Oper. Res.* **2022**, *301*, 82–109. [[CrossRef](#)]
27. Ferro, G.; Paolucci, M.; Robba, M. Optimal Charging and Routing of Electric Vehicles with Power Constraints and Time-of-Use Energy Prices. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14436–14447. [[CrossRef](#)]
28. Kumar, A.; Aggarwal, A.; Rani, A.; Bedi, J.; Kumar, R. A mat-heuristics approach for electric vehicle route optimization under multiple recharging options and time-of-use energy prices. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7854. [[CrossRef](#)]
29. Bektas, T.; Laporte, G. The Pollution-Routing Problem. *Transp. Res. Part B Methodol.* **2011**, *45*, 1232–1250. [[CrossRef](#)]
30. Ropke, S.; Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **2006**, *40*, 455–472. [[CrossRef](#)]
31. Wen, M.; Linde, E.; Ropke, S.; Mirchandani, P.; Larsen, A. An adaptive large neighborhood search heuristic for the Electric Vehicle Scheduling Problem. *Comput. Oper. Res.* **2016**, *76*, 73–83. [[CrossRef](#)]
32. 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](#)]
33. Zhuang, Z.L.; Zhang, Z.L.; Teng, H.; Qin, W.; Fang, H.J. Optimization for integrated scheduling of intelligent handling equipment with bidirectional flows and limited buffers at automated container terminals. *Comput. Oper. Res.* **2022**, *145*, 105863. [[CrossRef](#)]
34. Shaw, P. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming—CP98*; Maher, M., Puget, J.F., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 417–431.
35. Feng, W.; Figliozzi, M. An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market. *Transp. Res. Part C Emerg. Technol.* **2013**, *26*, 135–145. [[CrossRef](#)]
36. Murakami, K. A new model and approach to electric and diesel-powered vehicle routing. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *107*, 23–37. [[CrossRef](#)]
37. Zhang, S.; Gajpal, Y.; Appadoo, S.S.; Abdulkader, M.M.S. Electric vehicle routing problem with recharging stations for minimizing energy consumption. *Int. J. Prod. Econ.* **2018**, *203*, 404–413. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.