Automated Guided Vehicle (AGV) Scheduling in Automated Container Terminals (ACTs) Focusing on Battery Swapping and Speed Control

Xurui Yang 1, Hongtao Hu 2,*, Chen Cheng 2 and Yuren Wang 2

1 Institute of Logistics Science & Engineering, Shanghai Maritime University, Shanghai 201306, China; yang_xurui@163.com
2 School of Logistics Engineering, Shanghai Maritime University, Shanghai 201306, China; c.c.owen@foxmail.com (C.C.); wyr8193@163.com (Y.W.)
* Correspondence: hthu@shmtu.edu.cn

Abstract: Because they are environmentally friendly and safe, automated guided vehicles (AGVs) are increasingly used in newly constructed automated container terminals. However, their scheduling strategy is constrained by their limited battery capacity. When batteries reach their charging threshold, the AGVs need to be returned to battery-swapping stations. Moreover, the movement speed of AGVs has a significant impact on their energy consumption and operating times. Therefore, in this paper, a speed control strategy that considers the traffic environment of the terminal is proposed from the perspective of energy conservation and emission reduction. In addition, the charging capacity of the terminal is discretized to model its limited handling capacity to avoid congestion in the battery-swapping stations. To minimize the costs of delays and carbon emissions of AGV operations, a mixed integer programming model is established. It optimizes the efficiency and carbon emissions of the operations by assigning and prioritizing container transportation and AGV battery-swapping tasks. An improved genetic algorithm-based approach is designed where a better initial solution is obtained through a greedy strategy, while simulated annealing is adopted for population selection to prevent the algorithm from falling into local optima. Furthermore, an adaptive adjustment strategy for crossover and mutation probabilities is adopted to improve the algorithm’s convergence. Finally, a series of numerical experiments is conducted to verify the efficiency of the proposed method. The experimental results indicate that considering the variability of AGV speed more accurately characterize their energy consumption, and increasing the number of AGVs and enhancing the battery-swapping capacity can effectively reduce the costs of delays and carbon emissions.

Keywords: automated container terminal; automated guided vehicles; charging; speed control; energy consumption

1. Introduction

As critical nodes in the global supply chain, container terminals play a significant role in international trade. As technology advances and operational equipment is upgraded, ports are gradually transitioning toward constructing automated or semiautomated container terminals (ACTs). Horizontal transportation equipment used for the delivery of containers between the seaside and the landside is a key component in the operation of container terminals. Among these, automated guided vehicles (AGVs) constitute the transportation equipment with the largest market share in the global container port business.

In recent years, to achieve high throughput and storage density in ACTs, yards equipped with unilateral cantilever rail-mounted gantry cranes combine the advantages
of conventional and general automated container yards and have been adopted in newly constructed automated terminals, such as the Tuas Port in Singapore and the Haifa Port in Israel. In this new yard layout, AGVs can enter the interior of blocks arranged perpendicularly to the shoreline for interaction with the yard cranes (YCs). This approach has resulted in the development of long-range AGVs operating in more complex environments, requiring more maneuvering and under more complex scheduling requirements.

Figure 1 shows a typical layout of a new ACT. The terminal is divided into three parts: the seaside operation area, the horizontal transportation area, and the yard blocks. In addition, a battery-swapping station providing battery recovery for AGVs is arranged within the horizontal transportation area. In addition, the AGV-drivable area is divided into three parts: the apron area, the horizontal transportation area, and the block area. Among them, AGVs in the apron area need to handle containers in cooperation with quay cranes (QCs), while in the block area of the yard, AGVs need to interact with YCs to handle containers. Because of the lack of space allowing wide movement, as in the horizontal transportation area, interaction with YCs is more challenging. The roads inside the yard block are narrow and crowded, the traffic environment is more complex, and the speed control requirements are high.

Figure 1. Automated container terminal (ACT) layout.

In the conventional container terminal truck scheduling, vehicles are often controlled by drivers. With the advancement of Industry 4.0, the terminal’s transportation system has been updated to automation and digital intelligent technology, posing additional operational challenges. Unoccupied vehicles have to rely on control systems for scheduling. In addition to completing container transportation tasks, it is also necessary to control the vehicle’s driving speed and monitor the vehicle’s electricity level. Because of the limited capacity of their battery packs, AGVs often need to return to the battery stations for battery swapping during the yard operations. However, the handling capacity of the battery-swapping stations is limited, and too many AGVs entering at the same time may cause congestion. In addition, the traffic environment in different areas of the terminal can also have an impact on the movement speed of AGVs, thereby affecting their energy consumption and, by extension, the CO₂ emissions associated with replenishing the batteries.
Unlike the conventional AGV scheduling [1–5], when studying the AGV scheduling problem in a new type of automated container terminal, it is necessary to pay attention to its battery consumption, ensuring that the depleted AGVs can be restored in a timely manner while completing their task, thereby ensuring the smooth completion of container transportation. At the same time, it is necessary to consider the charging capacity of the terminal to avoid congestion of AGVs with depleted batteries at the battery-swapping station and consider factors that affect the energy consumption of AGVs, such as their driving speed, often influenced by the traffic environment in different areas of the terminal. Therefore, this paper schedules AGV operations to determine the task and sequence of AGV operations while also exploring the impact of AGV speed variability on carbon emissions and task completion.

With the construction of smart terminals, intelligent equipment and technologies have also been widely applied, and low-carbon environmental protection is receiving increasing attention. Therefore, this paper studies the scheduling problem of AGVs in new terminals from the perspective of energy conservation and emission reduction, ensuring that AGVs complete their tasks while reducing carbon emissions. It also provides an academic reference for the operation of smart terminals and the intelligent upgrading of traditional ones.

This paper studies the AGV scheduling problem in a new type of automated container terminal, taking into account the battery swapping and speed control of AGVs and optimizing their task assignment and operation sequence while avoiding congestion at the battery-swapping station, thereby minimizing the operation delay and carbon emission costs of AGVs. The contributions of this paper mainly lie in the consideration of the following:

1. A speed control strategy considering the traffic environment of new ACTs is proposed, and a mixed integer programming (MIP) model is established with the goal of minimizing CO₂ emissions and delays due to AGV operations. The new model takes into account the battery recovery of AGVs and the impact of traffic conditions on speed in different areas of new ACTs;

2. Because of the complexity of the proposed model, an improved genetic algorithm (GA)-based solution is developed. A greedy strategy is used to generate a better initial solution. A simulated annealing (SA)-based population selection approach and an adaptive crossover and mutation probability calculation prevent the algorithm from falling into local optima while increasing the rate of convergence, thus improving the solution efficiency.

In the rest of this paper, Section 2 offers a brief review of the AGV scheduling problem. This problem in new ACTs is defined in Section 3, and a MIP model is established. The proposed approach is introduced in Section 4. The results of simulation experiments are presented in Section 5. Finally, the study’s conclusion is discussed, and future directions for this field are put forward in Section 6.

2. Literature Review

In this section, the relevant literature on the AGV scheduling problem is reviewed, and the following aspects are considered: AGV scheduling in ACTs, AGV scheduling considering battery recovery, and AGV scheduling considering speed control.

Scholars have conducted extensive research on the AGV scheduling problem in ACTs. Sun et al. (2022) provided a comprehensive review of AGV-based transportation systems of ACTs and summarized the literature presented in the past 25 years from different research perspectives [6]. Grunow et al. (2005) studied the multi-load AGV dispatching problem. On this foundation [1], Grunow et al. (2006) further considered the uncertain handling time of cranes and release times of orders [7]. Yang et al. (2018) established a bi-level programming model to study AGV scheduling in ACTs and proposed an improved GA to solve the proposed model [2]. Zhong et al. (2020) integrated the AGV
scheduling problem and AGV conflict-free path-planning problem into a hybrid genetic algorithm-particle swarm optimization designed to solve the proposed mixed integer programming model [3]. Wang et al. (2022) investigated the AGV scheduling problem considering bidirectional paths in ACTs and designed a branch-and-bound algorithm to generate conflict-free routes and minimize the total completion time [4]. Duan et al. (2023) studied the AGV scheduling problem considering the interaction with QC, and a mixed integrated programming model was established with the goal of minimizing the completion time of ships and the idle time of AGVs [5]. Yang et al. (2023) studied the AGV scheduling problem under the sea rail intermodal transportation model and established a mixed integer programming model to minimize equipment energy consumption [8]. Hu et al. (2023) investigated AGV path planning in conflict situations, and the multiagent deep deterministic policy gradient (MADDPG) method was proposed to obtain the shortest path [9].

There is also relevant research on yard vehicle assignment and scheduling. Dotoli et al. (2014) developed a Petri nets-based framework for intermodal freight transport terminals that can evaluate the operational performance of transportation systems and the terminals’ efficiency [10]. Subsequently, Cavone et al. (2016a) proposed two further control policies to optimize the terminal performance [11], and Cavone et al. (2016b) used timed Petri nets to solve the strategic planning and the capacity/frequency of the transportation equipment [12]. Iris et al. (2018) integrated the management of loading operations, planning, and scheduling of the transport vehicles and established a mixed integer programming model to minimize tardiness and vessel service time [13].

With the construction of green ports, electric vehicle scheduling has gradually attracted the attention of several scholars. Iris and Lam (2019) and Zhou et al. (2022) provided a comprehensive review of energy efficiency and application of AGV in green ports, respectively [14,15]. Schmidt et al. (2015) evaluated the cost efficiency of electric AGVs in ACTs from a revenue perspective and proposed strategies to further enhance their profitability [16]. Ma et al. (2020) studied the planning of charging station facilities and operational strategies of electric AGVs in ACTs and proposed two layout designs and two charging strategies for deploying charging stations and controlling electric AGV charging [17].

In addition, scholars have conducted in-depth research on the issue of battery power in AGV scheduling. Xiang et al. (2021) established an ACT-nested semi-open queuing network model, which evaluates the ACT performance under different plug-in charging and battery-swapping strategies [18]. Singh et al. (2022) studied AGV scheduling in ACTs considering battery charging to minimize the tardiness and travel costs of AGVs and developed an adaptive large-neighborhood search algorithm to solve the proposed MIP model [19]. Sun et al. (2023) investigated AGV scheduling focusing on battery charging and established a collaborative scheduling optimization model that trades off terminal operational efficiency and energy savings [20].

Other scholars have studied the energy recovery of electric vehicles in other application scenarios. Keskin et al. (2019) studied electric vehicle routing with time windows, taking into account the queuing time of vehicles at charging stations [21]. Hosseini et al. (2020) proposed a novel, fully distributed control strategy for large-scale electric vehicles, minimizing total charging costs and battery degradation while achieving optimal charging [22]. Dang et al. (2021) studied the scheduling problem of heterogeneous AGVs in factories, considering limited battery capacities and partial charging at critical battery thresholds [23]. Jamshidi et al. (2021) studied the scheduling problem of electric taxis and developed a charging algorithm that considers movement distance, charging facility capacity, and charging time [24]. In order to explore the relationship between AGV scheduling, charging threshold, and power consumption, Li et al. (2022) studied AGV scheduling in a manufacturing factory [25]. Lam et al. (2022) studied electric vehicle routing considering time windows and charging constraints, but in this case, the energy recovery was modeled as a piecewise linear function of charging time [26]. Boccia et al. (2023) studied AGV
scheduling with battery constraints in manufacturing workshops and determined the optimal scheduling of transportation and charging operations for AGV fleets [27].

Some scholars’ focus on AGV speed control in AGV scheduling is also worth noting. Nishi et al. (2014) studied the optimization of collision-free paths for AGVs with acceleration and deceleration. By discretizing the transportation area into several regions, a continuous time model was established to represent the dynamic changes in the vehicle’s state [28]. Adamo et al. (2018) determined the AGV path and speed on each arc to prevent conflicts, focusing on time windows and minimizing the total energy consumption [29]. Chen et al. (2021) studied automatic truck scheduling and established a mixed integer second-order cone programming model based on fuel consumption functions related to movement speed and fleet formation to minimize the total operating costs [30]. Considering the impact of speed on AGV travel time and energy consumption, Xing et al. (2023) studied AGV scheduling in ACTs and established an MIP model to minimize the total operation time of cranes and the energy consumption of the AGVs [31].

The literature related to automated vehicle scheduling is classified in Table 1 on the basis of the following six main consideration parameters: (1) charging demand, (2) limited charging capacity, (3) speed, (4) types of constraints, (5) types of models, and (6) solution approach. Overall, only a few papers have considered a speed factor in vehicle scheduling problems. Additionally, only a few works have considered both AGV speed control and energy consumption in ACT settings. Moreover, these studies did not simultaneously consider the charging demand of AGVs. In the AGV scheduling of urban roads and factory workshops, only a few scholars have considered the charging demand while addressing the limited charging capacity.

Table 1. Literature on automated vehicle scheduling.

<table>
<thead>
<tr>
<th>Literature</th>
<th>Charging Demand</th>
<th>Limited Charging Capacity</th>
<th>Speed</th>
<th>Constraints</th>
<th>Model Type</th>
<th>Solution Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adamo et al. (2018) [29]</td>
<td></td>
<td></td>
<td></td>
<td>Time windows</td>
<td>MIP</td>
<td>Heuristic</td>
</tr>
<tr>
<td>Boccia et al. (2023)</td>
<td></td>
<td></td>
<td></td>
<td>Battery capacity; job sequence</td>
<td>MIP</td>
<td>Heuristic</td>
</tr>
<tr>
<td>Chen et al. (2021) [30]</td>
<td></td>
<td></td>
<td></td>
<td>Fuel consumption; time windows</td>
<td>MISOCPP</td>
<td>Heuristic</td>
</tr>
<tr>
<td>Dang et al. (2021) [23]</td>
<td>√</td>
<td></td>
<td></td>
<td>AGV capacity; time windows</td>
<td>MIP</td>
<td>ALNS</td>
</tr>
<tr>
<td>Duan et al. (2021) [5]</td>
<td></td>
<td></td>
<td></td>
<td>Job sequence; time windows</td>
<td>MIP</td>
<td>MOPSO</td>
</tr>
<tr>
<td>Jamshidi et al. (2023)</td>
<td></td>
<td></td>
<td></td>
<td>Battery capacity</td>
<td>MILP</td>
<td>CG</td>
</tr>
<tr>
<td>Keskin et al. (2019)</td>
<td></td>
<td></td>
<td></td>
<td>Job sequence; time windows; battery capacity</td>
<td>MIP</td>
<td>ALNS</td>
</tr>
<tr>
<td>Lam et al. (2022) [26]</td>
<td></td>
<td></td>
<td></td>
<td>Number of routes; time windows</td>
<td>MIP</td>
<td>BCP</td>
</tr>
<tr>
<td>Li et al. (2023) [25]</td>
<td></td>
<td></td>
<td></td>
<td>Job sequence; time windows; battery capacity</td>
<td>Stochastic programming model</td>
<td>S-ACO</td>
</tr>
<tr>
<td>Ma et al. (2020) [17]</td>
<td></td>
<td></td>
<td></td>
<td>Time windows</td>
<td>Simulation model</td>
<td>CPLEX</td>
</tr>
<tr>
<td>Nishi et al. (2014) [28]</td>
<td></td>
<td></td>
<td></td>
<td>Vehicle acceleration and deceleration</td>
<td>MIP</td>
<td>CG</td>
</tr>
<tr>
<td>Schmidt et al. (2015)</td>
<td></td>
<td></td>
<td></td>
<td>Charging time window</td>
<td>-</td>
<td>Simulation</td>
</tr>
</tbody>
</table>
Singh et al. (2022) [19] √ Charging sequence; battery capacity; time windows MIP ALNS
Sun et al. (2023) [20] √ Battery capacity; job sequence; time Windows MIP SA-GA
Wang et al. (2022) [4] Job sequence; time windows; path conflict MIP BB
Xiang et al. (2021) [18] √ Number of AGV; battery capacity queueing model MIP Approximation
Xing et al. (2023) [31] √ Job sequence; time windows MIP Heuristic
Yang et al. (2023) [8] Time windows; number of AGV MIP SCGA
Zhong et al. (2020) [3] Job sequence; time windows MIP HGA-PSO
Cavone et al. (2016b) [12] Time windows - Timed PN
Hosseini et al. (2020) [22] √ Time windows; battery capacity CQP DWA
Grunow et al. (2006) [7] Time windows - PN


Therefore, from the perspective of green port construction, in this paper, AGV scheduling is studied in the context of container transportation operations in ACTs, taking into account the charging demands of AGVs and the limited charging capacity of the terminal. The traffic environment in different areas is also analyzed to determine the optimal AGV speed and the CO2 emissions incurred during AGV transportation.

3. Problem Description

3.1. Automated Guided Vehicle Scheduling in Automated Container Terminals

3.1.1. Decision Problem

In this paper, the container transportation process is regarded as a container task, and the handling capacity of the charging station is discretized into several battery-swapping tasks. A container-loading task requires the AGV to reach the starting position of the task and transport the container to its destination. A battery-swapping task requires the AGV to travel to the battery-swapping station and remain there for the period of time required by the swapping robot to load and unload a battery pack.

The number of unloaded AGV transitions needs to be decreased effectively by optimizing AGV scheduling and altering the order of AGV operations. Simultaneously, the optimization of AGV speed can reduce AGV energy consumption. When the cumulative energy consumption of the AGV reaches a charging threshold $E_{\text{max}}$, the AGV needs to return to the battery-swapping station to prevent its battery from depleting completely and
stalling, which would compromise the overall terminal’s operation. Furthermore, because of the limited handling capacity of the battery-swapping station, the AGV charging sequence should also be considered to avoid congestion in the battery-swapping station. As a result, the research focus of this paper is on the scheduling of AGV container tasks and battery-swapping tasks, the adjustment of operational sequences and movement speeds, and the replacement of depleted AGV batteries to ensure that all container transportation tasks are completed within a specific allocated time period while minimizing the CO₂ emissions generated by AGV charging.

3.1.2. Impact of Speed Variability

AGVs are commonly considered pollution-free vehicles because they consume electrical energy. However, in many countries, electricity is generated through fuel combustion, which results in the release of a large amount of greenhouse gases. Therefore, drawing inspiration from Zhang et al.’s research on carbon emissions from electric vehicles (2018) [32], in this paper, the carbon dioxide (CO₂) emissions caused by the operation of AGVs are calculated through the AGVs’ energy consumption.

In order to calculate the total CO₂ emissions from AGV operation, it is necessary to calculate the energy consumed as an AGV moves between any two points in the terminal. Assuming, as in Equation (1), that \( i \) and \( j \) are the starting and ending points that define an arc, the energy required by an AGV moving along arc \((i, j)\) is \( \eta_{ij}, e_{\text{ff}} \) is the motor efficiency, \( e_{\text{ff}} \) is the electrical energy efficiency, \( \alpha \) is an arc-specific coefficient, \( \beta \) is a vehicle-specific coefficient, \( g \) is the weight of the AGV, \( \text{Load}_{ij} \) is the weight of the container transported by the AGV, \( d_{ij} \) is the length of arc \((i, j)\), and the velocity of AGV along arc \((i, j)\) is \( v_{ij} \), then

\[
\eta_{ij} = e_{\text{ff}} \cdot \text{Load}_{ij} \cdot \eta_{ij} \cdot e_{\text{ff}} \cdot \left[ \alpha(g + \text{Load}_{ij})d_{ij} + \beta v_{ij}^2d_{ij} \right]
\]  

(1)

The energy consumption of an AGV on the arc \((i, j)\) is related to its load, its speed, and the arc’s length. Given the container task, the energy consumption of AGV is only related to its speed. Therefore, the speed of the AGV affects not only the task start and completion times but also the energy consumption of the AGV, thereby affecting its CO₂ emissions \( CE_{ij} \) during operation. This is shown in Equation (2)

\[
CE_{ij} = e_{\text{ff}} \cdot r \cdot \eta_{ij} = e_{\text{ff}} \cdot r \cdot e_{\text{ff}} \cdot e_{\text{ff}} \cdot \left[ \alpha(g + \text{Load}_{ij})d_{ij} + \beta v_{ij}^2d_{ij} \right]
\]  

(2)

where \( e_{\text{ff}} \) is the charging efficiency, and \( r \) is the rate of electric energy conversion into emissions based on the country’s energy mix. Considering the different conditions in each ACT area, dividing the route of AGV into smaller segments allows for a more accurate representation of the variability of the AGV’s speed on a given route. At the same time, traffic speed can be defined for each segment from on the traffic conditions of each area. To further illustrate the speed changes in AGVs in different areas, Figure 2 shows the route of an AGV as it performs a container-unloading task in a layout such as the one shown in Figure 1. The container-unloading task requires the AGV to pick up the container from the QC, then move through the horizontal transportation area, enter the corresponding yard block, and finally stop next to the designated bay to deliver the container to the YC. Because the allocated storage location of the container has already been determined at a higher decision-making level, the distance traveled by the AGV to complete the task is known.
Figure 2. AGV speed variability.

Assuming that the distance traveled by the AGV to complete the unloading operation Task 1 is 270 m, if the traffic environment is not taken into account, it is assumed that the movement speed of the AGV is 5 m/s. Considering that AGVs operate in different areas, the transportation route can be divided into three segments of 20 m, 100 m, and 150 m lengths, as shown in Figure 2. According to the traffic rules and safety regulations of the terminal, combined with the actual production environment, we have limited the maximum speed of AGV driving in each area to control their speed. We assumed that information on the traffic environment in different areas could be obtained and that the AGV movement speeds in the three segments are specified as 5 m/s, 6 m/s, and 4 m/s, respectively. If an AGV completes the container task moving with a speed of 5 m/s, assuming that the traffic speed is constant along the entire route, the CO2 emitted by the energy consumed to perform Task 1 is 4.73 g. However, if the traffic environments of the AGVs in different areas of the terminal are considered, and the AGVs travel at the specified speeds on each segment, the CO2 emissions generated are 4.56 g. Compared with the constant speed movement, a variable AGV movement speed will result in CO2 emissions reduced by 3.73%. Therefore, considering the speeds in different transportation areas will make the calculation of AGV CO2 emissions more accurate.

3.2. Assumptions

Based on the preceding discussion, the following assumptions are stated;

(1) Each AGV has the same battery capacity;
(2) An AGV only transports a single container at a time;
(3) The initial energy consumption of AGVs obeys a uniform distribution;
(4) The start and end times of each task can be calculated from the known container-loading and -unloading sequences of automated QCs and YCs and are available prior to investigating the AGV scheduling;
(5) The acceleration and deceleration of AGVs are instantaneous.

3.3. Mathematical Model

The mathematical model can be established following the objections and constraints expressed by Equations (3)–(27):

$$
\begin{align*}
\min \ & \sum_{i \in A} \sum_{k \in K} f_i \cdot f_k \cdot \bar{d}_{ik} \cdot \delta_{i}^{\alpha} + \sum_{i \in I} \sum_{k \in K} \sum_{j \in J} f_i \cdot f_j \cdot d_{ijk} \cdot \delta_{j}^{\alpha} + \sum_{i \in I} f_i \cdot \mu_i \\
\text{s.t.} \ & \sum_{k \in K} \alpha_{ik} = 1 \quad \forall i \in A \\
\ & \sum_{k \in K} \alpha_{ik} \leq 1 \quad \forall i \in T \\
\ & \sum_{i \in I} \beta_{0ik} = 1 \quad \forall k \in K \\
\ & \sum_{i \in I} \beta_{0ik} = \sum_{i \in I} \beta_{0ik} \quad \forall k \in K \\
\ & \sum_{j \in J} \beta_{ijk} = \alpha_{ik} \quad \forall i \in I, k \in K
\end{align*}
$$
\[
\sum_{j \in J^i} \beta_{ijk} = \alpha_{ik} \quad \forall i \in I, k \in K
\]

(9)

\[
a_i \cdot \alpha_{ik} \leq \rho_{ik} \leq b_i \cdot \alpha_{ik} + \mu_i \quad \forall i \in A, k \in K
\]

(10)

\[
a_i \cdot \alpha_{ik} \leq \rho_{ik} \leq b_i \cdot \alpha_{ik} \quad \forall i \in T, k \in K
\]

(11)

\[
\rho_{ik} - \rho_{ik} + \sum_{l \in L_j} d_{ij} / \delta_{ijkl} + \sum_{j \in J} d_{ij} / \delta_{ijkl} \leq M \cdot (1-\beta_{ijk}) \quad \forall i \in A^o, j \in I, i \neq j, l \in L_j, k \in K
\]

(12)

\[
\rho_{ik} - \rho_{ik} + c_i + \sum_{l \in L_j} d_{ij} / \delta_{ijkl} \leq M \cdot (1-\beta_{ijk}) \quad \forall i \in T, j \in A^o, l \in L_j, k \in K
\]

(13)

\[
\eta_{ik} = (\eta_{ik} + q \cdot \sum_{l \in L_j} d_{ij} \cdot \delta_{ijkl}^2 + q \cdot \sum_{j \in J} d_{ij} \cdot \delta_{ijkl}^2) \leq M \cdot (1-\beta_{ijk}) \quad \forall i \in I^o, j \in A, i \neq j, l \in L_j, k \in K
\]

(14)

\[
\eta_{ik} = (\eta_{ik} + \sum_{l \in L_j} d_{ij} \cdot \delta_{ijkl}^2 + \sum_{j \in J} d_{ij} \cdot \delta_{ijkl}^2) \geq M \cdot (\beta_{ijk} - 1) \quad \forall i \in I^o, j \in A, i \neq j, l \in L_j, k \in K
\]

(15)

\[
\eta_{ik} = 0 \quad \forall i \in T, k \in K
\]

(16)

\[
E_{max} - (\eta_{ik} + q \cdot \sum_{l \in L_j} d_{ij} \cdot \delta_{ijkl}^2) \geq M \cdot (\beta_{ijk} - 1) \quad \forall i \in I^o, j \in T, k \in K
\]

(17)

\[
E_{max} - (\eta_{ik} + q \cdot \sum_{l \in L_j} d_{ij} \cdot \delta_{ijkl}^2) \geq M \cdot (\beta_{ijk} - 1) \quad \forall i \in A, j \in T, k \in K
\]

(18)

\[
\delta_{ijkl} \leq u_{ijkl} \quad \forall i \in I^o, j \in I^o, i \neq j, l \in L_j, k \in K
\]

(19)

\[
\overline{\delta}_{ijkl} \leq u_{ijkl} \quad \forall i \in I, l \in L_j, k \in K
\]

(20)

\[
\alpha_{ik} \in \{0, 1\} \quad \forall i \in I^o, k \in K
\]

(21)

\[
\beta_{ijk} \in \{0, 1\} \quad \forall i \in I^o, j \in I^o, k \in K
\]

(22)

\[
\eta_{ik} \geq 0 \quad \forall i \in I^o, k \in K
\]

(23)

\[
\rho_{ik} \geq 0 \quad \forall i \in I^o, j \in I^o, k \in K
\]

(24)

\[
\mu_i \geq 0 \quad \forall i \in A
\]

(25)

\[
v^{ib} \leq \delta_{ijkl} \leq v^{ib} \quad \forall i \in I^o, j \in I^o, i \neq j, l \in L_j, k \in K
\]

(26)

\[
v^{ib} \leq \overline{\delta}_{ijkl} \leq v^{ib} \quad \forall i \in I^o, l \in L_j, k \in K
\]

(27)

Objective (3) aims to minimize the CO2 emission cost and penalty costs caused by operational delays. The CO2 emission costs include those incurred during complete tasks and those incurred during no-load operations. Constraints (4) and (5) state that each container and battery-swapping task, respectively, can be completed at most once.
Constraints (6) and (7) state that all AGVs must depart and ultimately return to dummy nodes. Constraints (8) and (9) guarantee that each node has only one immediately preceding and one immediately following node. Constraints (10) and (11) limit the start time of the container and battery-swapping task nodes within specified ranges. Constraint (12) limits the start time of the ensuing tasks after the AGV has completed a container task; that is, the start time of an ensuing task must be later than the start time of the preceding container task plus its operational time and the no-load traveling time between the two task nodes. Constraint (13) restricts the start time of an ensuing container task after the AGV has completed a battery-swapping task, which means that the start time of the ensuing task must be later than the start time of the preceding swapping task plus the duration of stay at the battery-swapping station and the no-load traveling time between the two task nodes. Constraints (14) and (15) pertain to the cumulative energy consumption associated with completing each container task. Constraint (16) ensures that the cumulative energy consumption of the AGV after completing a battery-swapping task is reset to zero. Constraints (17) and (18) ensure that the AGV has sufficient power to travel to the battery-swapping station. Constraints (19) and (22) restrict the speed of each segment to the maximum speed allowed by the traffic environment. Constraints (21)–(27) are the value ranges of the decision variables.

3.4. Model Linearization

The objective function of the model and the constraints applied include speed decision variables related to nonlinear terms. Therefore, linearization of the model is required. In this paper, the same strategy used by Nasri et al. (2018) is adopted [33]; thus, the speed variable is discretized, i.e., the velocity is represented using a set of discrete values \( R = \{1, ..., r, ... \} \). At the same time, two binary decision variables \( \lambda_{ijlk}^r \) and \( \lambda_{ilk}^r \) are introduced to designate the speed selection in different segments. The exact linearization process is shown below:

Sets

- \( R \): the set of speed \( v^1 = v^a \) and \( v^r = v^{ab} \)

Decision variables

- \( \lambda_{ijlk}^r \): binary variable \( \lambda_{ijlk}^r = 1 \), if AGV \( k \) moves at speed level \( r \) on segment \( l \) of the arc \( (i, j) \); otherwise \( \lambda_{ijlk}^r = 0 \); 

- \( \lambda_{ilk}^r \): binary variable \( \lambda_{ilk}^r = 1 \), if AGV \( k \) moves at level \( r \) on segment \( l \) of the node \( i \); otherwise \( \lambda_{ilk}^r = 0 \).

The objective function is to minimize:

\[
\min \sum_{i \in A} \sum_{l \in L} \sum_{k \in K} \left( f \cdot f \cdot \tilde{d}_{ij} \cdot \sum_{r \in R} \lambda_{ijlk}^r \cdot v^2 \right) + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \left( f \cdot f \cdot d_{ij} \cdot \sum_{r \in R} \lambda_{ilk}^r \cdot v^2 \right) + \sum_{i \in A} \mu_i \quad (28)
\]

Subject to:

(4)–(11), (16), (21)–(25)

\[
\rho_{ij} - \rho_{jk} + \sum_{l \in L} \tilde{d}_{ij} \cdot \sum_{r \in R} \lambda_{ijlk}^r / v^r + \sum_{l \in L} d_{ij} \cdot \sum_{r \in R} \lambda_{ilk}^r / v^r \leq M \cdot (1 - \beta_{ij})
\]

\( \forall i \in A, j \in I^0, i \neq j, k \in K \) \hspace{1cm} (29)

\[
\rho_{ij} - \rho_{jk} + c_j + \sum_{l \in L} d_{ij} \cdot \sum_{r \in R} \lambda_{ilk}^r / v^r \leq M \cdot (1 - \beta_{ij})
\]

\( \forall i \in T, j \in A^0, k \in K \) \hspace{1cm} (30)
\[ \eta_{jk} - (\eta_i + q \cdot \sum_{k \in I_R} d_{jk} \cdot \sum_{r \in R} \tau_{jk} \cdot v_r^2 + q \cdot \sum_{k \in I_R} d_{il} \cdot \sum_{r \in R} \lambda_{ilk} \cdot v_r^2) \leq M \cdot (1 - \beta_{jk}) \] (31)

\[ \forall i \in I^0, j \in A, i \neq j, k \in K \]

\[ \eta_{jk} - (\eta_i + q \cdot \sum_{k \in I_R} d_{jk} \cdot \sum_{r \in R} \tau_{jk} \cdot v_r^2 + q \cdot \sum_{k \in I_R} d_{il} \cdot \sum_{r \in R} \lambda_{ilk} \cdot v_r^2) \geq M \cdot (\beta_{jk} - 1) \] (32)

\[ \forall i \in I^0, j \in A, i \neq j, k \in K \]

\[ E_{\max} - (\eta_i + q \cdot \sum_{k \in I_R} d_{jl} \cdot \sum_{r \in R} \lambda_{ljk} \cdot v_r^2) \geq M \cdot (\beta_{jk} - 1) \] (33)

\[ \forall i \in I^0, j \in T, i \neq j, k \in K \]

\[ E_{\max} - (\eta_i + q \cdot \sum_{k \in I_R} d_{il} \cdot \sum_{r \in R} \lambda_{ilk} \cdot v_r^2) \geq M \cdot (\beta_{il} - 1) \] (34)

\[ \forall i \in A, j \in T, i \neq j, k \in K \]

\[ \sum_{r \in R} \lambda_{ijk} \cdot v_r^2 \leq u_{ij} \quad \forall i \in I^0, j \in I^0, i \neq j, l \in L, k \in K \] (35)

\[ \sum_{r \in R} \lambda_{rjk} \cdot v_r \leq \pi_i \quad \forall i \in A, l \in L, k \in K \] (36)

\[ \sum_{r \in R} \lambda_{ilk} = \beta_{ik} \quad \forall i \in I^0, j \in I^0, i \neq j, l \in L, k \in K \] (37)

\[ \sum_{r \in R} \lambda_{ilk} = \alpha_{il} \quad \forall i \in A, l \in L, k \in K \] (38)

\[ \lambda_{ijk} \in \{0, 1\} \quad \forall i \in I^0, j \in I^0, i \neq j, l \in L, k \in K, r \in R \] (39)

\[ \tau_{ilk} \in \{0, 1\} \quad \forall i \in A, l \in L, k \in K, r \in R \] (40)

Constraints (29)–(36) play the same role as Constraints (12)–(15) and Constraints (17)–(20). Constraints (37) and (38) ensure that each AGV must move at only one speed on each segment. Constraints (39) and (40) are the value ranges of the decision variables.

4. Solution Approach

Owing to the complexity of the developed model, the solution space of the problem expands as the scale of the numerical experiments increases, and it is difficult to obtain the results within a reasonable time through the direct application of commercial solvers such as GUROBI. The main factors affecting the complexity of the proposed model come from the integer decision variables of task assignment and task sequence. The integer encoding rules of GA are very suitable for solving such decision variables [2,8,20]. Therefore, in this paper, a greedy and simulated annealing strategy-based adaptive genetic algorithm (GSAAGA) is proposed to improve the efficiency of the model solution. The randomness of the algorithm search is used to improve the global search capability and find the optimal solution among the combinations of task assignments that guarantee minimum battery power consumption.

4.1. Coding and Decoding
The MIP model proposed in this paper consists of eight decision variables, among which task assignment $\alpha_{ik}$ and operation sequence $\beta_{ijk}$ are the main decision variables in the AGV scheduling problem. Due to the soft time window constraints of container operation tasks and the potential impact of delays in ACT transportation operations on the scheduling of equipment downstream, the delay costs account for a relatively high proportion of the objective function. Therefore, after determining the tasks assigned to the AGVs, the task sequence can be obtained by sorting the task time windows in ascending order.

The decision variables involved in the speed control of each road segment are not determined using algorithms due to their fine granularity, high similarity in decision content, and large decision space. Thus, the maximum speed is used in the algorithm for energy and cost calculations, and the GUROBI solver was invoked to obtain the speed of each road segment after determining the key decision variables.

### 4.1.1. Solution Coding

A solution is encoded as a vector $z = \{z_1, z_2, \ldots\}$, representing the assignment of container tasks, and its length is the total number of container tasks. The number at each position indicates which AGV performs the corresponding container task, and the number takes on a value ranging from zero to $|K| - 1$. The encoding method indicates which AGV performs which container transportation task. Figure 3 demonstrates a solution coding of the solution, in which $z_2 = 1$, $z_3 = 1$, and $z_6 = 1$ indicate that Tasks 2, 3, and 6 are all performed by AGV 1.

![Figure 3. Solution coding.](image)

The AGV battery-swapping operation tasks add additional waiting time and no-load travels, increasing delay and CO2 emission costs. Battery-swapping tasks need to be performed only when an AGV reaches the swapping threshold; that is, battery-swapping tasks depend on the operational tasks assigned to the AGV. Therefore, battery-swapping tasks are not explicitly assigned to AGVs during coding.

### 4.1.2. Decoding

The container tasks assigned to each AGV can be determined through the above chromosome coding, and the operational sequence of AGVs can be sorted based on the time window of each task. In addition, it is also necessary to know precisely which battery-swapping tasks are required and where these tasks have been planned in the task sequence. In this paper, simulations are used to determine the task sequence of each AGV and calculate the cost of that task sequence while decoding. The process of decoding is shown in Algorithm 1.

The decoded solution of the chromosomes in Figure 3 is shown in Figure 4. It can be seen that this decoding method can not only obtain the sequence in which AGVs complete the container tasks but also determine when they need to swap their batteries, thereby determining which battery-swapping tasks the AGVs will need to perform.
Algorithm 1. Solution decoding process.

**Algorithm** Decoding of AGV scheduling scheme based on time window and simulation

**Input**
- load segment \( \overline{d}_{ij} \)
- chromosome \( c \)
- load segment \( \overline{d}_{ij} \)
- no-load segment \( d_{ij} \)
- earliest starting time \( a_i \)
- initial cumulative energy consumption \( \eta_{0k} \)
- energy consumption coefficient \( q \)
- cost coefficient \( f_e \)
- \( f_p \)

**Output**
- decision variables \( \alpha_{ik}, \beta_{ijk}, \text{fitness} \)

**Foreach** (container task \( i, j \in I \))
- Calculate the process time of task \( i: time_i = \sum \overline{d}_{ii} / \overline{w}_i \), no-load travel time of arc \((i,j): time_y = \sum d_{ij} / u_{ij} \), energy consumption of task \( i: bat_i = q \cdot \text{time}_i \), no-load energy consumption of arc \((i,j): bat_y = q \cdot \text{time}_y \).

Create AGV task sequence list: \( \text{tasks} = [\text{List}_1, \ldots, \text{List}_k] \), and swapping state list of each AGV \( \text{changed}_k \).

**Foreach** (container task \( i \in A \))
- let \( v \leftarrow c[i] \), \( last \leftarrow \text{AGV} v \) last completed task

**Foreach** (container task \( i \in \text{tasks}_k \), AGV \( k \in K \))
- If \( \eta_{last} + \text{bat}_{last} + \text{bat}_i + \text{bat}_r \leq E_{\text{max}} \)
  - delay time of task \( i \): \( \text{delay}_i \leftarrow \text{Max} (\text{End}_{i} - \text{time}_{last} + \text{time}_{last} - c_r, 0) \)
  - Add \( \text{fitness} \) to \( \text{tasks}_k \), update the cumulative energy consumption of AGV \( v \), \( \text{fitness} + \) = emission cost of last task to task \( i \) + delay cost of task \( i \) + emission cost of task \( i \)
- Else If \( \text{changed}_k = \text{false} \)
  - task, add task \( t \) which satisfy \( \text{End}_{i} - \text{time}_{last} + \text{time}_{last} - c_r \geq 0 \) (If there is no swapping task that meets the conditions, it is considered that the chromosome is infeasible and the decoding algorithm is exited). Update the cumulative energy consumption of AGV \( v \) to 0, swapping state list: \( \text{changed}_k \leftarrow \text{true} \), the delay time of task \( i \):
    - \( \text{delay}_i \leftarrow \text{Max} (\text{End}_{i} - \text{time}_{last} + \text{time}_{last} + c_r + \text{time}_{last} - c_r, 0) \)
  - Add \( \text{fitness} \) to \( \text{tasks}_k \), update the cumulative energy consumption of AGV \( v \), \( \text{fitness} + \) = delay cost of task \( i \) + emission cost of task \( i \) + emission cost of last task to BSS+ emission cost of BSS to task \( i \)
- Else
  - The AGV requires a battery swapping to continue the task, but the battery has already been swapped, which means this chromosome is infeasible and the algorithm is exited.
- Let \( a_{ik} \leftarrow 1, \beta_{last} \leftarrow 1 \)

**Output** \( \alpha_{ik}, \beta_{ijk}, \text{fitness} \)
4.2. Standard Genetic Algorithm

This section describes the basic steps of a standard GA, including initial solution generation, crossover and mutation operation, and fitness calculation.

4.2.1. Initial Solution Generation

The algorithm randomly generates \( n \) chromosome encoding vectors constituting the initial chromosome population. For each chromosome, the availability of battery-swapping tasks of each AGV operation is determined. If the chromosome is determined to be infeasible, it is generated again randomly.

4.2.2. Fitness Function

Since the solution objective of the proposed model is to minimize CO\(_2\) emission and delay costs, the inverse of the objective function is chosen as the fitness function in the GA. The calculation is shown in Equation (41), where \( f(x) \) represents the objective function value of the model as

\[
\text{fitness} = \frac{1}{f(x)} \tag{41}
\]

4.2.3. Crossover Operator

Chromosomes are encoded using multiple gene points, and a single-point crossover strategy is adopted. A portion of the genes of two parent chromosomes is exchanged to create two offspring chromosomes. The crossover operation is performed to ensure the diversity of transportation task plans for AGVs, thus expanding the search range of task sequences in the solution space. The crossover operation is shown in Algorithm 2.
Algorithm 2. Crossover operation.

**Algorithm** Crossover operator

| Input | Parent chromosomes $f_i$ and $f_2$, $f_i(i)$ is the value in the $i$ th gene on the chromosome, $i \in \{1,\ldots,|N|\}$ |
|-------|--------------------------------------------------|
| Output | Offspring chromosomes $c_1$ and $c_2$, $c_i(i)$ is the value in the $i$ th gene on the chromosome, $i \in \{1,\ldots,|N|\}$ |

Step 1 Randomly generates two numbers: $s_1$ and $s_2$, without loss of generality, let $s_1 \leq s_2$.

Step 2 Set $s_1$ as the starting point of the crossover operator, $s_2$ as the ending point of the crossover operator, and obtain the gene length for the crossover based on $s_2 - s_1$.

Step 3 According to the starting point $s_1$ of the crossover operator, exchange the genes with length $s_2 - s_1$ for the parent chromosomes $f_1$ and $f_2$ to obtain the offspring chromosomes $c_1$ and $c_2$, let $c_1(i) \leftarrow f_2(i)$, $i \in \{1,\ldots,|N|\}$, $c_2(i) \leftarrow f_1(i)$, $i \in \{1,\ldots,|N|\}$

Step 4 Obtain offspring chromosomes

---

4.2.4. Mutation Operator

The mutation operator randomly changes some genes in the chromosome, resulting in a diverse population of chromosomes in the algorithm. This avoids falling into local optima in the solution space and allows the algorithm to optimize with good searchability. In this paper, a two-point mutation strategy is thus carried out according to the process shown in Algorithm 3, which results in new offspring chromosomes.

Algorithm 3. Mutation operation.

**Algorithm** Mutation operator

| Input | Parent chromosomes $f$, $f(i)$ is the value in the $i$ th gene on the chromosome, $i \in \{1,\ldots,|N|\}$ |
|-------|--------------------------------------------------|
| Output | Offspring chromosomes $c$, $c(i)$ is the value in the $i$ th gene on the chromosome, $i \in \{1,\ldots,|N|\}$ |

Step 1 Randomly select two genes as mutation points on parent chromosome $f$

Step 2 Swap the positions of genes at two mutation points and copy them to the corresponding positions of the offspring chromosome $c$

Step 3 Copy all gene on the parent chromosome $f$ except for mutation points to the corresponding positions on the offspring chromosome $c$, ensuring that the gene sequence is consistent with that on the $f$.
4.2.5. Termination Condition

The GA termination conditions are: (1) when the optimal fitness value has not changed for 20 consecutive iterations or (2) when the maximum number of iterations has been reached.

The flowchart of the standard GA is shown in Figure 5.

![Standard GA flowchart](image)

**Figure 5.** Standard GA flowchart.

4.3. Improved Genetic Algorithm

In order to improve the solution efficiency of GA, in this paper, some improvements are introduced to the algorithm. These are the generation of initial solutions using a greedy strategy, the selection of populations based on SA, and adaptive crossover and mutation probability. The flowchart of the improvement methodology in the proposed solution approach is shown in Figure 6.
4.3.1. Greedy Strategy-Based Initial Solution Generation

Because of the significant impact of randomly generated initial solutions on the results of the GA, in some cases, the algorithm converges after several tens of iterations, while in others, the optimal solution was not reached even when the maximum number of iterations had been performed. Therefore, it was necessary to generate initial solutions with high fitness to accelerate convergence. In this paper, an initial solution generation method based on a greedy strategy is introduced and used to generate half of the initial population.

The difference between the chromosomes generated using the greedy strategy lies in the different order of traversing AGVs when assigning container tasks and the different first tasks assigned to them. There are significant changes in subsequent task sequences due to the different initial energy levels of the AGV. When assigning container tasks, priority should be given to AGVs that can minimize the added costs of operations, which include the CO₂ emission and delay costs generated during the no-load transitions between the previous task and the ensuing container task. On the basis of the relationship between the remaining energy of the AGV and the charging threshold, a battery-swapping task should be inserted to ensure the feasibility of AGV operations. The flowchart of the algorithm is shown in Figure 7.
Sort container tasks and battery swapping tasks by time window

Randomly generate AGV order

Input AGV and container information

All container tasks been assigned

Calculate the cumulative energy consumption of each AGV by adding the currently assigned task

The remaining battery capacity can reach the swapping station

Calculate the increased cost of the AGV completing current task

Calculate the increased cost of the AGV completing current task before completing current task

Selection the task plan with the lowest increased cost

According to the AGV task sequence, change the gene

Output chromosome

End

Figure 7. Flowchart of greedy strategy-based initial solution generation.

4.3.2. Simulated Annealing-Based Population Selection

Because of the search process randomness of GA, it is impossible to guarantee that the crossover and mutation operations during each iteration will lead the population to search in a more optimal direction. Therefore, in this paper, the GA is improved by introducing the temperature acceptance criteria of SA. After the offspring population has been obtained, a potentially poor population may be used as a new parent for iteration based on the current annealing temperature through the difference between the offspring’s average fitness and that of the parent population. The process of the algorithm is as follows:

1. After generating the initial population using GA, the current temperature is set as $T_{\text{cur}} = T_0$, where $T_0$ is the preset starting temperature;
2. The fitness of each population is calculated, the average fitness of the current population is recorded as $\text{avg}(\text{fitness}_{\text{cur}})$ and the average fitness of the offspring population is recorded as $\text{avg}(\text{fitness}_{\text{new}})$;
3. Based on the average fitness of two population generations, the Metropolis acceptance criterion (as shown in Equation (42)) is used to determine whether the original population will be used as the current population for selecting parents in the next iteration:

$$\text{Acceptance} = \begin{cases} 1, & \text{avg fitness}_{\text{cur}} < \text{avg fitness}_{\text{new}} \\ \exp((\text{avg fitness}_{\text{cur}} - \text{avg fitness}_{\text{new}})/T_{\text{cur}}), & \text{avg fitness}_{\text{cur}} \geq \text{avg fitness}_{\text{new}} \end{cases} \quad (42)$$

When the average fitness of the offspring is higher than that of the current population, the offspring is directly accepted as the new current population. At high
annealing temperatures, it is easier to accept new populations with larger fitness differences, while at low annealing temperatures, only new populations with smaller fitness differences will be accepted;

4. The temperature current after one iteration is updated as \( T_{cur} = T_{cur} \cdot rate_{cool} \), where \( rate_{cool} \) represents the cooling coefficient.

4.3.3. Adaptive Crossover and Mutation Probability Design

The mutation and crossover probabilities \( P_m \) and \( P_c \) will directly affect the GA solution results. Therefore, in this section, an adaptive strategy is introduced to ensure the convergence of the algorithm, which automatically changes \( P_m \) and \( P_c \) depending on the fitness value.

The maximum and average fitness values of the population are represented as \( g_{\text{max}} \) and \( g_{\text{avg}} \), respectively; the larger fitness value of two chromosomes that need to cross is represented as \( g_m \), and the fitness value of the chromosome that needs to mutate is represented as \( g_v \). The constant is represented as \( n (n \in [0,1]) \), and the four constants are represented as \( n_1, n_2, n_3, n_4 \), respectively. \( P_m \) and \( P_c \) will be adjusted according to Equations (43) and (44)

\[
P_m = \begin{cases} 
  n_1 \frac{(g_{\text{max}} - g_x)}{g_{\text{max}} - g_a}, & g_x \geq g_a \\
  n_2, & g_x < g_a 
\end{cases} \quad (43)
\]

\[
P_c = \begin{cases} 
  n_1 \frac{(g_{\text{max}} - g_x)}{g_{\text{max}} - g_m}, & g_m \geq g_a \\
  n_4, & g_m < g_a 
\end{cases} \quad (44)
\]

Although traditional adaptive genetic strategies can obtain and retain relatively optimal solutions effectively and can find global optimal solutions by changing fitness values around local optimal solutions, the results of these methods in the early stages of evolution may not be ideal. In the early stages, the probability of obtaining a relatively optimal solution is minimal, resulting in the optimal solution remaining approximately unchanged. As a result, when entering the later stages, the evolution results are prone to falling into local optima, ultimately slowing down the search speed of the GA and even causing the search to halt.

An improved adaptive strategy is proposed by introducing three weighted parameter probability values \( P_{(c)\text{max}} \), \( P_{(c)\text{mid}} \), \( P_{(c)\text{min}} \). These place certain constraints between different fitness values and continuously adjust the crossover and mutation probabilities according to the evolution of the population. In the early evolutionary stages, by adjusting the parameter operations, the probability values of crossover and mutation are increased, thereby improving the global search ability, avoiding premature convergence to local optima, and ensuring individual diversity in the chromosome population. In the later stages, the fitness value of individual chromosomes is relatively high. At this time, if the search strategy of the algorithm is relatively random, this could lead to a convergence speed that is too slow. In order to improve the speed of convergence, the crossover and mutation probability can be reduced by adjusting the weighted parameter values. The improved adaptive crossover and mutation probabilities are adjusted as shown in Equations (45) and (46)
The maximum and minimum values of the crossover and mutation probabilities are represented as $P_{c\text{ max}}$ and $P_{c\text{ min}}$, and $P_{v\text{ max}}$ and $P_{v\text{ min}}$, respectively, while $P_{v(c)\text{ mid}}$ is the median of $P_{v(c)\text{ min}}$ and $P_{v(c)\text{ max}}$.

4.3.4. Proposed Solution Method

The flowchart of the proposed solution approach is shown in Figure 8. Half of the chromosomes in the initial population are generated using the greedy strategy, and the crossover and mutation probabilities are adjusted adaptively on the basis of the fitness of the parental chromosomes. In addition, the SA is integrated into the GA to enhance the latter’s global search ability.
Figure 8. Flowchart of the proposed solution approach.

5. Numerical Experiments

This section presents a series of experiments to verify the effectiveness and efficiency of the proposed MIP model and solution method. The proposed mathematical model was implemented and solved using the popular commercial solver GUROBI. The proposed solution method was implemented in C#, and the model was solved using GUROBI 10.0. All experiments were conducted on a computer with a 3.80 GHz CPU and 64 GB RAM.

5.1. Parameter Settings

The planning horizon in this paper is one hour, and the QC and YC operation time and sequence of each container are assumed to have been determined at a higher level of planning, which is beyond the scope of this paper. The processing time of the QC and the YC follow a uniform distribution of U (40, 200) s, and the initial cumulative power consumption of the AGVs follows a uniform distribution of U (0, 100) kw/h.

The length of the shoreline was assumed to be 2350 m, and the width of the horizontal transportation area was set to 120 m. The battery-swapping station was assumed to be equipped with three swapping robots, and the AGV stay time at the battery-swapping station was set to 5 min. The maximum speed of the AGVs was set to 6 m/s, which led to a discrete set of velocities \( R = \{1, 2, 3, 4, 5, 6\} \). Considering the environmental traffic limitations in different areas of the yard, the maximum speed in different segments was set as \( u_{ij} = \{5, 6, 5\} \) when the AGVs were in a no-load state, and the maximum speeds in different segments were set as \( v_{ij} = \{3, 4, 3\} \) when the AGVs were loaded. The settings of the other parameters are shown in Table 2.

Table 2. Parameter settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{eff}_d )</td>
<td>1.11</td>
<td>( \text{eff}_m )</td>
<td>1.25</td>
</tr>
<tr>
<td>( \text{eff}_p )</td>
<td>1.25</td>
<td>( r )</td>
<td>0.69 kg/kWh</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0.098</td>
<td>( g )</td>
<td>2.5 t</td>
</tr>
<tr>
<td>( \beta )</td>
<td>5</td>
<td>( f_e )</td>
<td>30 CNY/ton</td>
</tr>
<tr>
<td>( L_i )</td>
<td>25t</td>
<td>( f_p )</td>
<td>7.2 CNY/h</td>
</tr>
</tbody>
</table>

The feasibility and effectiveness of the model and algorithm were verified in small, medium, and large-scale cases. Detailed information on all cases is shown in Table 3.

Table 3. Cases setting.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>No. of Tasks</th>
<th>No. of AGVs</th>
<th>No. of QCs</th>
<th>No. of Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S2</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S3</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S4</td>
<td>12</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S5</td>
<td>16</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S6</td>
<td>16</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S7</td>
<td>20</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>M1</td>
<td>60</td>
<td>6</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>M2</td>
<td>60</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>M3</td>
<td>80</td>
<td>12</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>M4</td>
<td>100</td>
<td>15</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>M5</td>
<td>100</td>
<td>16</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>
5.2. Numerical Experiment Results

5.2.1. Analysis of Results for Different Scale Cases

This section presents the results of numerical experiments to evaluate the effectiveness of the proposed model and solution approach. The results of the standard GA, a GA based on SA (SAGA), an adaptive genetic algorithm based on SA (SAAGA), a greedy algorithm (Greedy), and the proposed GSAAGA were compared with the results of the GUROBI solver. Because of the randomness of the heuristic algorithms (HAs), each case was run 10 times, and the average values are reported as the results. The numerical experimental results of all cases are shown in Table 4.

Table 4. Results of numerical experiments.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Tmax</th>
<th>Large Scale Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>100</td>
<td>300 - 300</td>
</tr>
<tr>
<td>M2</td>
<td>100</td>
<td>500 - 500</td>
</tr>
<tr>
<td>M3</td>
<td>100</td>
<td>140 - 140</td>
</tr>
<tr>
<td>M4</td>
<td>100</td>
<td>140 - 140</td>
</tr>
<tr>
<td>M5</td>
<td>100</td>
<td>140 - 140</td>
</tr>
<tr>
<td>M6</td>
<td>100</td>
<td>140 - 140</td>
</tr>
<tr>
<td>M7</td>
<td>100</td>
<td>140 - 140</td>
</tr>
</tbody>
</table>

Because of the complexity of the problem, GUROBI could only solve the small-scale cases within the specified time. For those, GUROBI obtained the optimal solution in most cases, and the GAP (defined as GAP (%) = (OBJ\text{heuristics} – OBJ)/OBJ · 100%) between the HA-based solution methods and the optimal solution was 0, verifying the effectiveness of the HA-based solution methods. In addition, all HA-based solution methods obtained the
optimal solution within about 1 s on average, while GUROBI’s solution time was, on average, more than 30 times slower.

It is worth noting that for cases S3 and S5, GUROBI could not find the optimal solution within the specified time, while the GAP between the solution obtained using the HA-based method and the approximate solution was about 15% and 45%, respectively, further verifying the effectiveness of the HA-based solution methods. In addition, although the greedy algorithm-based methods could obtain a solution in a short time, the quality of the solution was not guaranteed because of the limited number of iterations allowed. In cases S5 and S6, the greedy algorithm-based methods obtained the worst solution.

5.2.2. Comparison of Algorithm Convergence

To further demonstrate the advantages of the solution approach proposed in this paper, the convergence of GA, SAGA, SAAGA, and GSAAGA in solving case L7 is shown in Figure 9. The population of the algorithm was set to 50, and the number of iterations was 500. The termination criterion was the absence of improvement in 20 generations. It can be seen from Figure 9 that the GA was easily trapped in local optima, and the population selection strategy based on SA helped the algorithm overcome these local optima effectively. The rate of convergence of SAAGA was faster than that of SAGA; that is, the adaptive crossover and mutation probabilities accelerated the rate of convergence of the algorithm. It is worth noting that half of the initial population in GAASGA adopts the greedy algorithm, which has a good initial solution from the beginning, allowing for a significant improvement in the quality of the solution.

![Figure 9. Comparison of algorithm convergence under different strategies.](image)

5.2.3. Scheduling Scheme

The scheduling results of the L7 large-scale case are shown in Figure 10. This case included 200 container tasks and 24 AGVs. The rectangles in the figure represent container and battery-swapping tasks for the AGVs. Solid rectangles represent inbound container tasks, dashed rectangles represent outbound container tasks, and red rectangles represent the battery-swapping tasks. The width of each rectangle represents the duration of the task, and the number in the rectangle represents the task number.
The scheduling scheme of the GA-based solution approach is shown in Figure 10a, while the scheduling scheme of GSAAGA is shown in Figure 10b. It can be seen that in the scheduling plan obtained using the GA-based solution approach, 9 AGVs had their batteries swapped, while in the case of GSAAGA, only 7 AGVs required battery swapping, which resulted in reduced cost of CO$_2$ emission generated through idle movement. It is worth noting that the GSAAGA-based solution approach resulted in the faster completion of container tasks and the effective reduction in the delay cost.
5.3. Sensitivity Analysis

This section presents a sensitivity analysis conducted using different numbers of AGVs and battery-swapping station capacities to analyze these parameters’ impact on scheduling results, CO₂ emission cost, and delay costs.

5.3.1. Sensitivity Analysis on the Number of Automated Guided Vehicles

In order to observe the impact of the number of AGVs on CO₂ emission cost and delay costs, a sensitivity analysis was conducted for ACT configurations with different numbers of AGVs. The number of container tasks was set to 20, 80, and 160, and the number of AGVs varied from 2 to 21. The specific settings and experimental results are shown in Table 5.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>No. of AGVs</th>
<th>No. of QCs</th>
<th>No. of Blocks</th>
<th>OBJ</th>
<th>Imp. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3.358</td>
<td>2770.085</td>
</tr>
<tr>
<td>A2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1.516</td>
<td>1195.726</td>
</tr>
<tr>
<td>A3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>0.881</td>
<td>652.991</td>
</tr>
<tr>
<td>A4</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>0.120</td>
<td>2.564</td>
</tr>
<tr>
<td>A5</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0.119</td>
<td>1.709</td>
</tr>
<tr>
<td>A6</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>0.118</td>
<td>0.855</td>
</tr>
<tr>
<td>A7</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>0.117</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>7</td>
<td>4</td>
<td>8</td>
<td>0.581</td>
<td>2.718</td>
</tr>
<tr>
<td>B2</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>0.573</td>
<td>1.213</td>
</tr>
<tr>
<td>B3</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>0.572</td>
<td>1.093</td>
</tr>
<tr>
<td>B4</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>0.571</td>
<td>0.872</td>
</tr>
<tr>
<td>B5</td>
<td>11</td>
<td>4</td>
<td>8</td>
<td>0.568</td>
<td>0.354</td>
</tr>
<tr>
<td>B6</td>
<td>12</td>
<td>4</td>
<td>8</td>
<td>0.567</td>
<td>0.246</td>
</tr>
<tr>
<td>B7</td>
<td>13</td>
<td>4</td>
<td>8</td>
<td>0.566</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>15</td>
<td>7</td>
<td>14</td>
<td>1.427</td>
<td>0.364</td>
</tr>
<tr>
<td>C2</td>
<td>16</td>
<td>7</td>
<td>14</td>
<td>1.425</td>
<td>0.257</td>
</tr>
<tr>
<td>C3</td>
<td>17</td>
<td>7</td>
<td>14</td>
<td>1.425</td>
<td>0.252</td>
</tr>
<tr>
<td>C4</td>
<td>18</td>
<td>7</td>
<td>14</td>
<td>1.425</td>
<td>0.216</td>
</tr>
<tr>
<td>C5</td>
<td>19</td>
<td>7</td>
<td>14</td>
<td>1.423</td>
<td>0.091</td>
</tr>
<tr>
<td>C6</td>
<td>20</td>
<td>7</td>
<td>14</td>
<td>1.422</td>
<td>0.059</td>
</tr>
<tr>
<td>C7</td>
<td>21</td>
<td>7</td>
<td>14</td>
<td>1.421</td>
<td></td>
</tr>
</tbody>
</table>

Note: Imp. (%) = (OBJ − min OBJ)/min OBJ · 100.

In order to demonstrate the impact of the number of AGVs on cost, for this experiment, Constraint (6) was relaxed, so not all AGVs were required to complete the tasks. Constraint (6) in the original model was replaced with Constraint (47)

\[
\sum_{n \in N} \beta_{nk} \leq 1 \quad \forall k \in K
\]  

The experimental results indicate that energy consumption decreased as the number of AGVs increased. This is because an increase in the number of AGVs will lower the average number of tasks performed by each AGV, reducing the demand that AGVs return to the battery-swapping stations for energy recovery, thereby reducing the overall CO₂ emission cost of the operation. In addition, when the number of container tasks is small, adding AGVs can bring more significant cost reductions. As the scale of container tasks expands, the cost improvement brought about through an increased investment in AGVs will gradually decrease.
5.3.2. Sensitivity Analysis on the Capacity of Battery-Swapping Stations

In this section, a sensitivity analysis is presented on the handling capacity of the battery-swapping stations, which was conducted by changing the number of swapping robots. The number of container tasks was set to 160, and three scenarios with different AGV numbers were considered, while the number of battery-swapping robots ranged from 1 to 5. The specific settings and experimental results are shown in Table 6.

Table 6. Sensitivity analysis results of the number of battery-swapping robots.

<table>
<thead>
<tr>
<th>AGVs</th>
<th>Robots</th>
<th>Operation Cost</th>
<th>No-Load Cost</th>
<th>Delay Cost</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.356</td>
<td>0.089</td>
<td>78.071</td>
<td>79.516</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.356</td>
<td>0.085</td>
<td>64.336</td>
<td>65.778</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.356</td>
<td>0.087</td>
<td>55.836</td>
<td>57.280</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.356</td>
<td>0.088</td>
<td>54.918</td>
<td>56.362</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.356</td>
<td>0.073</td>
<td>7.364</td>
<td>8.793</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.356</td>
<td>0.075</td>
<td>6.187</td>
<td>7.618</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.356</td>
<td>0.077</td>
<td>5.511</td>
<td>6.943</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.356</td>
<td>0.074</td>
<td>4.767</td>
<td>6.197</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.355</td>
<td>0.058</td>
<td>0.000</td>
<td>1.413</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.355</td>
<td>0.057</td>
<td>0.000</td>
<td>1.413</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.355</td>
<td>0.057</td>
<td>0.000</td>
<td>1.413</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.355</td>
<td>0.058</td>
<td>0.000</td>
<td>1.413</td>
</tr>
</tbody>
</table>

When the handling capacity of the battery-swapping station was low, some tasks could not be completed, resulting in the inability to honor the schedule. The experimental results indicate that when the handling capacity of the battery-swapping station is low, the total costs will gradually decrease as the number of battery-swapping robots increases because AGVs that require battery swapping have more options. In contrast, when the handling capacity of the battery-swapping stations is already sufficient, the overall cost reduction will decrease as the number of robots further increases. At that point, AGVs do not require additional swapping tasks. In addition, when the number of AGVs has already met the operational requirements of the container tasks, meaning there are no delay costs, a further increase in the number of robots will not result in a cost reduction. It is worth noting that increasing the number of robots leads to a significant improvement in delay costs, while its impact on CO2 emission is relatively small.

5.3.3. Sensitivity Analysis on Speed Variability

To reflect the impact of speed on carbon emission costs, sensitivity analysis was conducted by changing the deriving speed of AGVs. Considering that the fixed driving speeds of AGV were 3, 4, 5, and 6, respectively, and the number of container tasks was set to 20, 80, and 160, the various costs were compared between fixed speed and variable speed strategies. The specific settings and experimental results are shown in Table 7.

It can be seen that compared with the fixed speed strategy, the variable speed has a greater impact on the calculation results. There is a 43.31% difference in the calculation results in small-scale cases, and this difference gradually increases as the scale of the cases increases. Therefore, considering the speed variability significantly impacts the calculation of carbon emissions and time for AGV operations.

In most cases, compared with a fixed speed, the total cost of variable speed was greater. However, the increase in total costs and the decrease in carbon emission costs were not necessarily related. Indeed, an increase in total costs did not necessarily mean an
increase in carbon emission costs. The total cost included carbon emission and delay costs, which were conflicting, with the latter playing a more important role in reducing the total cost. In addition, considering the traffic conditions of different driving environments in the actual operation of the terminal, the variable speed could more accurately depict the carbon emissions and B-AGV operation time.

### Table 7. Sensitivity analysis results of the different travel speeds of AGV.

<table>
<thead>
<tr>
<th>Speed (km/h)</th>
<th>OBJ (20)</th>
<th>Diff. (%)</th>
<th>OBJ (80)</th>
<th>Diff. (%)</th>
<th>OBJ (160)</th>
<th>Diff. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4.783</td>
<td>10.26</td>
<td>7.711</td>
<td>155.50</td>
<td>33.468</td>
<td>339.33</td>
</tr>
<tr>
<td>4</td>
<td>2.629</td>
<td>50.68</td>
<td>0.872</td>
<td>71.11</td>
<td>6.229</td>
<td>18.24</td>
</tr>
<tr>
<td>5</td>
<td>1.620</td>
<td>69.61</td>
<td>0.622</td>
<td>79.39</td>
<td>2.545</td>
<td>66.59</td>
</tr>
<tr>
<td>6</td>
<td>0.747</td>
<td>85.98</td>
<td>0.611</td>
<td>79.75</td>
<td>1.470</td>
<td>80.70</td>
</tr>
<tr>
<td>variable</td>
<td>5.330</td>
<td>0.00</td>
<td>3.018</td>
<td>0.00</td>
<td>7.618</td>
<td>0.00</td>
</tr>
<tr>
<td>Avg.</td>
<td>43.31</td>
<td>77.15</td>
<td>100.97</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: \(|\text{Diff.}\%) = \frac{|\text{OBJ (speed)} - \text{OBJ (variable)}|}{\text{OBJ (variable)}} \cdot 100.\)

### 5.4. Discussion

In terms of solving efficiency, comparing different solution methods shows that the greedy strategy has advantages in generating initial solutions but also has limitations as it is very unstable in solving results. In addition, the SA strategy and adaptive cross-mutation strategy can effectively address the drawbacks of GA being prone to local optima and slow convergence speed [20]. Therefore, compared with the standard GA-based method, the proposed method has significantly better solution and scheduling results. This means reducing the carbon emissions of AGVs while reducing task delays.

Moreover, as an important factor affecting the completion time and energy consumption of AGV tasks, speed has a significant impact on the actual operation of AGV. The experimental results also demonstrate the importance of variable speed considering the actual production environment of the terminal, which will more accurately depict the actual operation process of AGV.

In addition, to reduce the delay and carbon emission costs of AGVs in the actual operation of the terminal, terminal operators could increase the number of AGVs used while also increasing the difficulty of system scheduling. In addition, further improvements can be made to the infrastructure of the terminal, thereby enhancing its battery-swapping capacity.

### 6. Conclusions

In this paper, ACTs with limited charging capacity are analyzed using a model that considers a discrete handling capacity of battery-swapping stations. To reduce the CO$_2$ emission cost of horizontal transportation, an AGV scheduling plan based on speed control is proposed. In addition, an MIP model considering the charging demand and speed control is established to ensure that AGVs complete their tasks while the CO$_2$ emissions generated by transportation are minimized.

Because of the complexity of the problem, an improved GA-based solution approach is proposed to solve the established model effectively. The feasibility and effectiveness of the model and algorithm are verified through a series of numerical experiments. A greedy strategy is adopted to obtain a better initial solution for the algorithm in solving large-scale cases, while adaptive and SA strategies are used to improve the quality of the solution in small-scale and medium-scale cases.

At the same time, the impact of the number of AGVs and the handling capacity of battery-swapping stations on CO$_2$ emission cost and delay costs are analyzed. The experimental results indicate that although an increase in the number of AGVs can lead to reduced CO$_2$ emission cost and delay costs, this improvement will be less pronounced once
the number of AGVs exceeds a certain point. Therefore, a reasonable number of AGVs can reduce CO2 emission and delay costs effectively.

Moreover, the number of battery-swapping robots also has a significant impact on CO2 emission cost and delay costs. A shortage of battery-swapping robots can lead to the inability to complete tasks, while more swapping robots can help reduce CO2 emission costs and delay costs during AGV operations. Therefore, equipping a reasonable number of battery-swapping robots can ensure the smooth operation of the terminal.

With respect to the charging demand and speed control, the AGV scheduling model established in this paper applies not only to newly constructed automated terminals but also to traditional terminals undergoing automation upgrades. The research results not only expand the theoretical research on port resource scheduling but also provide terminal managers with references for decision making and the operation of green ports.

This paper only focuses on the short-term scheduling problem of AGV. In the future, scholars can further consider the conflict and congestion issues during AGV operation. In addition, with continuous equipment upgrading, different energy recovery methods have emerged at container terminals [15]. In addition to the widely used overall battery-swapping method, they also include plug-in charging methods. Researchers can further consider the AGV scheduling problem of combining different charging methods.

Author Contributions: Conceptualization, H.H.; Methodology, X.Y.; Software, C.C.; Formal analysis, C.C. and Y.W.; Investigation, X.Y.; Writing—original draft, X.Y.; Visualization, Y.W.; Funding acquisition, 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 [grant numbers 72271156, 71771143]; the Shanghai Shuguang Scholar Project, China [grant number 20SG46]; the Shanghai Frontiers Science Center of “Full penetration” far-reaching offshore Ocean energy and power, China; the Container Supply Chain Technology Engineering Research Center of the Ministry of Education, China.

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

Nomenclature

Indices

\( i, j \) node indices

\( k \) AGV index

\( l \) segment index

Sets

\( A \) the set of container task nodes; \( \{1, 2, 3, \ldots, |A|\} \in A \)

\( 0 \) dummy node

\( A^0 \) \( A^0 = A \cup \{0\} \)

\( T \) the set of available battery-swapping task nodes, \( \{|A|+1, |A|+2, |A|+3, \ldots, |A|+|T|\} \in T \)

\( I \) the set of all nodes, including container task nodes and battery-swapping nodes; \( \{1, 2, 3, \ldots, |A|, \ldots, |A|+|T|\} \in I \), \( I = A \cup T \)

\( I^0 \) \( I^0 = I \cup 0 \)

\( K \) the set of AGVs; \( \{1, 2, 3, \ldots, |K|\} \in K \)

\( L_y \) the set of segments on arc \((i, j)\)

\( L_i \) the set of segments in node \(i\)

Parameters

\( d_{yit} \) the distance of segment \(l\) in arc \((i, j)\), \( l \in L_y \)

\( \bar{d}_{yi} \) the distance of segment \(l\) in node \(i\), \( l \in L_i \)

\( u_{yit} \) the maximum travel speed in segment \(l\), \( l \in L_y \)
the maximum travel speed in segment \( l \), \( l \in L \)

the earliest start time of node \( i \)

the latest start time of node \( i \)

the processing time of battery-swapping node \( i \), \( i \in T \)

the battery-swapping threshold

the initial cumulative battery consumption of AGV \( k \)

the coefficient of energy consumption, \( q = \text{eff}_d \cdot \text{eff}_m \cdot \beta \)

the AGV CO2 emission cost

AGV CO2 emission coefficient, \( f_c = \text{eff}_d \cdot \text{eff}_m \cdot \beta \cdot \text{eff}_p \cdot r \)

the delay penalty

a large positive integer

\( \alpha_{ik} \) binary variable \( \alpha_{ik} = 1 \), if node \( i \) is assigned to AGV \( k \); otherwise \( \alpha_{ik} = 0 \)

\( \beta_{jk} \) binary variable \( \beta_{jk} = 1 \), if node \( i \) is consecutively processed by AGV \( k \) before node \( j \); otherwise \( \beta_{jk} = 0 \)

the starting time of AGV \( k \) to process node \( i \)

the delay time during the processing of node \( i \)

the travel speed of AGV \( k \) on segment \( l \), \( l \in L_y \)

the travel speed of AGV \( k \) on segment \( l \), \( l \in L_i \)

References


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.