Hindawi
Discrete Dynamics in Nature and Society
Volume 2023, Article ID 5566244, 13 pages
https://doi.org/10.1155/2023/5566244

Research Article

An Integrated Approach to Scheduling B-CAVs in Container Terminals considering Battery Management

Shuo Wang,[1] Jiliang Luo,[2] Weimin Wu,[1] Dahai Zhang,[3] and Tao Zhang[1]

[1]*State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China*
[2]*College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China*
[3]*College of Ocean Engineering, Zhejiang University, Hangzhou 310027, China*

Correspondence should be addressed to Weimin Wu; wmwu@iipc.zju.edu.cn

Received 8 June 2023; Revised 7 July 2023; Accepted 20 July 2023; Published 25 August 2023

Academic Editor: Chun Wei

With the aim of promoting environmental sustainability and enhancing transport efficiency, battery-powered connected and automated vehicles (B-CAVs) are employed to replace diesel-powered ones in horizontal transport systems (HTSs) of container terminals. The operational efficiency of an HTS can be increased by the cooperation of B-CAVs. However, it is time-consuming to charge them. Therefore, their battery management becomes a critical issue of a transport schedule. To run container terminals more economically and efficiently, this work proposes an integrated scheduling approach to B-CAVs tasks' dispatch and route planning, where battery management is taken into account. An integer programming model is constructed with the goal of minimizing the total travel distance. Then, a sustainable charging policy is designed to ensure the consistent transport capacity of an HTS. Furthermore, a congestion-free path plan-based improved genetic algorithm is presented to obtain a near-optimal plan for dispatching B-CAVs to perform transporting and charging operations. A series of experiments are carried out to verify the effectiveness and efficiency of our approach.

1. Introduction

Container terminals play a vital role in the world's freight transportation, and a horizontal transport system (HTS) is a core component of container terminals, significantly impacting the efficiency of freight transportation. With the increment of container vessels' loading capacity, the operational efficiency of a terminal is required to be higher. However, due to improper task allocations and path planning, the HTS in traditional container terminals often operates inefficiently, resulting in a backlog of vessels and containers [1]. To address this issue, many automated container terminals (ACTs) are built by employing automatic guided vehicles (AGVs) [2, 3]. However, the yard layout needs to be changed for adopting AGVs, and the cost of this kind of HTS is huge.

In the near future, based on automatic driving, battery, and information technologies, battery-powered connected and automated vehicles (B-CAVs) will be deployed [4–7], as shown in Figure 1. Cooperative strategies, which are realized by vehicle-to-vehicle and vehicle-to-infrastructure communications, make transportation operations more efficient, flexible, and secure. Furthermore, compared with AGVs in ACTs, container terminals equipped with B-CAVs require less structural modification and cost to improve the operations of terminals.

As shown in Figure 2, a typical container terminal [8] consists of the quayside and yardside. The HTS is responsible for transporting containers between the quayside and yardside using B-CAVs. Quay cranes (QCs) load and unload containers between B-CAVs and vessels at the quayside. At the yardside, rubber-tired gantry cranes or other yard cranes (YCs) load and unload containers to or from B-CAVs. In reality, to minimize the berth time of a vessel, QCs and YCs should load and unload containers continuously once the vessel arrives. Consequently, a sound scheduling policy of

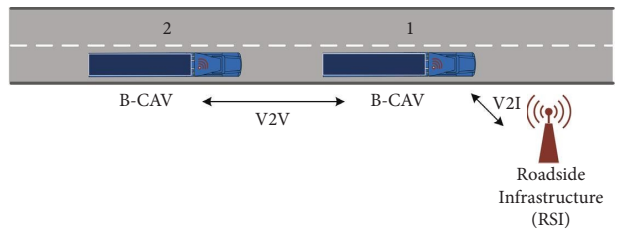

Figure 1: A battery-powered connected and automated vehicle system.

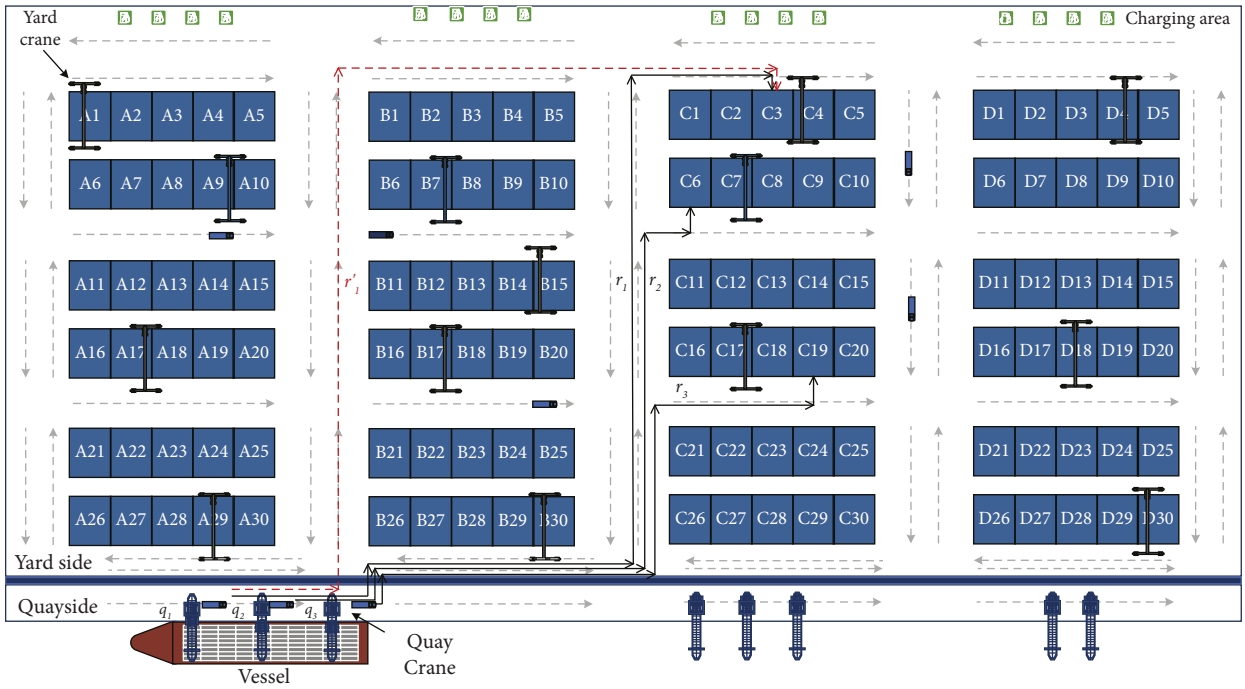

Figure 2: The layout of a typical container terminal.

HTS is significant to ensure the efficient loading and unloading of a container terminal.

However, unlike diesel-powered vehicles, a B-CAV needs a long time to fully charge its battery after the battery depletes. Hence, we must consider the battery management of B-CAVs when designing a transport plan for an HTS to prevent a shortage of B-CAVs. Furthermore, considering that the number of loading and unloading containers among vessels varies, the berth time could be different for different vessels. For example, three vessels $v1$, $v2$, and $v3$ arrive at the terminal in sequence, and their estimated berthing plan is present in Figure 3. Thus, the power demand of HTS varies when serving different vessels. An integrated scheduling strategy should not only consider container transport tasks for the current vessel but also reserve sufficient power for the upcoming vessels. To reduce transportation distance while ensuring uninterrupted and reliable container throughput, this work focuses on designing an integrated B-CAV scheduling approach, where the battery management of B-CAVs is taken into account to enhance the transportation efficiency of the HTS.

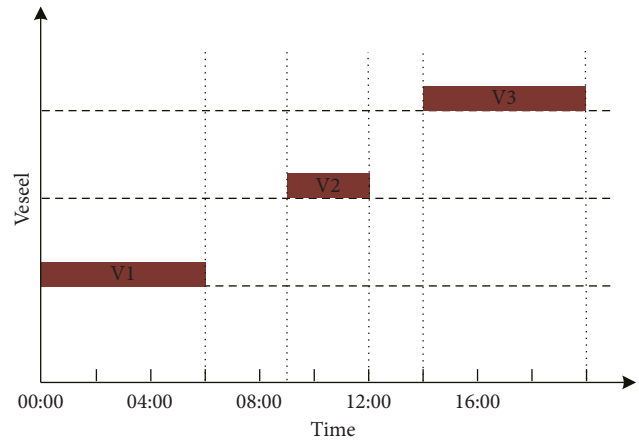

Figure 3: Example of vessels berth time plan.

The main contributions of this study are that we designed a new integer programming model for the HTS scheduling problem in container terminals, which includes the assignment of transportation tasks, route planning, and

particular attention to both the current and the next vessel's power demand. Then, a sustainable charging policy is designed to address battery management. Finally, a novel improved genetic algorithm (IGA) is proposed to incorporate the proposed charging policy and space-time routing method to generate the scheduling plan. The chromosome of the IGA is divided into two parts in terms of the remaining battery, and experiments verify that relying on the given crossover and mutation rules can effectively find better results.

The remainder of the paper is organized as follows. In Section 2, relevant works are reviewed. In Section 3, we presented an integrated programming model formalizing a B-CAV scheduling problem, and in Section 4, an improved genetic algorithm that combines a sustainable charging policy with a space-time route approach is proposed to generate the B-CAVs schedule. Subsequently, in Section 5, the case study is presented, describing our simulation for verifying the proposed method. Finally, in Section 6, the paper ends with concluding remarks.

2. Literature Review

The scheduling of transport vehicles is crucial for enhancing the operational efficiency of a container terminal. In the context of this study, we classify terminals according to their utilized vehicles, thereby categorizing them into three types: terminals with trucks, ACTs with AGVs, and improved traditional terminals with B-CAVs. Numerous strategies for the transport tasks assignment and vehicle route planning relevant to these terminal environments have been suggested.

In the context of scheduling truck in traditional terminals, Bish [9] divided trucks into distinct sets, employing a heuristic method to dispatch trucks for container transportation. Nishimura et al. [10] implemented a dynamic truck assignment method to establish a schedule with the shortest travel distance. They propose an efficient schedule principle and a heuristic algorithm to determine a near-optimal solution. Adelman [11] constructed an internal pricing mechanism to manage service trucks to generate higher daily shipping profits in a closed network. Lee et al. [12] leveraged a hybrid insertion algorithm to minimize total yard truck travel time. Hop et al. [13] deployed an adaptive particle swarm optimizer for searching a desired truck schedule plan by automatically adjusting its parameters. However, these studies primarily concentrate on the transport tasks assignment problem but fail to consider the impact of vehicle routing plans on transportation efficiency. Consequently, their schedules may not effectively circumvent the problem of efficiency degradation due to congestion.

In terms of AGVs utilized in the container terminal, Nguyen and Kim [14] discussed how to dispatch AGVs by harnessing the location and time information of impending delivery tasks. A mixed-integer programming model is provided to assign optimal delivery tasks to AGVs and employ a heuristic algorithm to reduce computational time. Ho and Liao [15] suggested a dynamic zone strategy which depends on zone partition design and dynamic zone regulation to avert AGV collisions and to maintain load equilibrium among AGVs across varying zones. Nishi et al. [16] designed a bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing of AGV systems. In detail, the upper level is used to assign tasks, and the lower level is to plan a path. Wu and Zhou [17] proposed a deadlock and blockage prevention method based on the framework of colored resource-oriented Petri net and the deadlock-free operation condition to obtain the shortest routes of bidirectional AGVs. Miyamoto and Inoue [18] presented a mixed-integer programming model aiming to minimize both transportation distance and delay time for AGVs. Fazlollahtabar and Saidi-Mehrabad [19] implemented the minimum cost flow model in conjunction with the enhanced network simplex algorithm for conflict-free and minimal delay time AGV route planning. Qin et al. [20] tackled the distinct hybrid flow terminal scheduling problem by integrating mixed-integer programming and constraint programming techniques. Chen et al. [21] proposed a multicommodity network flow model with two sets of flow balance constraints for cranes and AGV scheduling, and two side constraints are introduced to deal with interdevice constraints. The alternating direction algorithm of the multipliers method is adopted to decompose the problem into a set of crane-specific and vehicle-specific subtasks. The cost-effective solutions can be obtained by iteration. Zhong et al. [22] constructed a mixed-integer programming model focusing on path optimization, integrated scheduling, and conflict avoidance with the aim to minimize AGV delay time. Subsequently, a hybrid methodology combining genetic algorithm and particle swarm optimization is utilized to obtain the results. Ma et al. [23] established a novel mathematical model to describe multiload AGVs operating in ACTs and proposed an improved shuffled frog leaping algorithm. Xing et al. [24] devised the concept of glued nodes in roadmaps to analyze possible occupancy conflicts among different plans of the AGV path. Skaf et al. [25] utilized a heuristic method to derive a near-optimal solution for QC and yard truck scheduling problems. These methods show enormous potential in improving the operation efficiency of ACTs. However, battery management is not considered in their scenario.

More recently, battery-powered AGVs started to see usage in ATCs, leading to research into corresponding battery management policies. Ma et al. [26] employed a simulation-based approach to demonstrate that the decentralized charging stations layout and the progressive recharging policy lead to excellent performance. Xiang and Liu [27] established a nested semiopen queuing network model to estimate the performance with respect to resource allocation and layout design. These studies focus on optimizing resource allocation and layout design of the terminal considering battery management of battery-powered AGVs. However, these studies do not incorporate task dispatching and AGV path planning into their optimization framework. Contrastingly, Li et al. [28] constructed a two-stage stochastic programming model to address the joint scheduling problem of battery swapping and task operation under the

influence of random tasks and proposed a simulation-based ant colony optimization algorithm as a solution model. In this work, we introduce an integrated scheduling approach that focuses on scheduling of HTS, particularly considering both current and future battery requirements, and employ a novel genetic algorithm for a faster search of the efficient scheduling plan.

3. Problem Description

The application background of this work is in an improved traditional container terminal. Terminal equipment includes QCs, YCs, and B-CAVs, among which B-CAVs are used to transport containers between QCs and YCs. The container terminal scheduling module dispatches new tasks and plans a reasonable path for a B-CAV. Then, the B-CAV self-drives from the origin place to the destination by following the given path. The next task is assigned to the B-CAV in time after it completes the current task. The layout of a typical container terminal [8, 27] is shown in Figure 2. The yard layout is the parallel layout with a single edge, in which vehicles travel along only one side of a block since two adjacent blocks in a row of the layout are grouped together. In addition, charging stations are distributed along the yard's edge.

In this work, a container terminal traffic network is represented by a directed graph $G = (N, E)$, where $G = (N, E)$ is a finite set of nodes that represent traffic intersections, and $n \in N$; $E \subseteq N \times N$ is a set of directed edges that represent road within the network, and $e \in E$. For a road, storage capacity is used to qualify its space. The traffic network is prone to congestion when the number of B-CAVs exceeds its storage capacity. Specifically, the storage capacity of the road can be formulated as follows:

$$c_e = \frac{l_e}{(l_v + bf_1)}, \tag{1}$$

where c_e is the storage capacity of road e, l_e is the length of e, l_v is the length of a B-CAV, and bf_1 represents the necessary buffer length between two B-CAVs for maintaining safe driving conditions. The new routing plan of a B-CAV should consider the occupied status of a road by other B-CAVs to avoid potential traffic congestion. To depict the traffic state in a network, i.e., the congestion level of roads, this work presents a busy factor concept and combines it with time window. The busy factor of road e during the kth time window is given by

$$p_{ek} = \frac{n_{ek}}{c_e}, \tag{2}$$

where n_{ek} denotes the quantity of B-CAVs traversing through road e during the kth time window, and $p_{ek} > 1$ implies that the number of B-CAVs passing through road e in kth time window is greater than the capacity of road e. In this study, we make the assumption that the position of a working B-CAV at a given time t can be established according to its route plan and speed, and the value of n_{ek} on road l during the kth time window can also be determined. Subsequently, the traffic condition on each road is evaluated by examining its busy factor. The busy factors for all routes during all time windows P are recorded and updated over time to provide an accurate representation of the traffic situation.

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & \ddots & & p_{2k} \\ \vdots & & & \vdots \\ p_{e1} & \cdots & & p_{ek} \end{bmatrix}. \tag{3}$$

Based on the abovementioned concepts, we formulate the mathematical model of the scheduling problem as follows. The formulated scheduling problem is subjected to several constraints and takes the minimum total transportation distance as the optimization objective. In the proposed mathematical model, the following symbols are adopted:

I: set of B-CAVs, $(1, 2, \ldots, i) \in I$

J: set of tasks, $(1, 2, \ldots, j) \in J$

Q: set of QCs, $(1, 2, \ldots, q) \in Q$

Y: set of YCs, $(1, 2, \ldots, y) \in Y$

K: set of time windows, $(1, 2, \ldots, k) \in K$

θ_j: if task j is a loading task $\theta_j = 1$, if task j is a unloading task $\theta_j = -1$

x_{ij}: 1 if task j is assigned to B-CAV i, otherwise 0

d_{ij}: path length elapsed by B-CAV i when delivering task j

z_i: maximum load container number of B-CAV i

t_{ij}: travel time of B-CAV i delivering task j

t_{qj}: start time of QC q operating task j

t'_{qj}: end time of QC q operating task j

t_{yj}: start time of YC y operating task j

t'_{yj}: end time of YC y operating task j

i^b: remaining power of B-CAV i

b_1: first remaining power threshold of B-CAV

b_2: second remaining power threshold of B-CAV with $b_2 > b_1$

b_w: warning remaining power threshold of B-CAV

r_i: route plan from o_i to d_i

wa_i: 1 if B-CAV i is in the work area, otherwise 0

ws_i: 1 if B-CAV i is in the work state, otherwise 0

Applying the abovementioned notation, the multi-B-CAVs scheduling problem in container terminals can be formulated as the following integer programming model.

Objective function:

$$\min \sum_{i \in I} \sum_{j \in J} x_{ij} \cdot d_{ij}, \tag{4}$$

subject to

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J, \tag{5}$$

$$z_i \leq 1, \quad \forall i \in I, \tag{6}$$

$$t_{q(j+1)} \geq t'_{qj}, \quad \forall 1, 2 \cdots j-1 \in J, \forall q \in Q, \tag{7}$$

$$t_{y(j+1)} \geq t'_{yj}, \quad \forall 1, 2 \cdots j-1 \in J, \forall y \in Y, \tag{8}$$

$$t_{yj} + t_{ij} \leq t_{qj}, \theta_j = 1, \quad \forall j \in J, \forall i \in I, \forall y \in Y, \forall q \in Q, \tag{9}$$

$$t_{qj} + t_{ij} \leq t_{yj}, \theta_j = -1, \quad \forall j \in J, \forall i \in I, \forall y \in Y, \forall q \in Q, \tag{10}$$

$$b_w \leq i^b, \quad \forall i \in I, \tag{11}$$

$$p_{ek} \leq 1, \quad \forall e \in E, \forall k \in K, \tag{12}$$

$$wa_i + ws_i \neq 1, \quad \forall i \in I. \tag{13}$$

Equation (4) serves as the objective function that aims to minimize the HTS cumulative transportation distance during the loading and unloading of one vessel's containers. Constraint (5) ensures that each task can only be served by exactly one B-CAV. Constraint (6) illustrates that the number of containers carried by the B-CAV never exceeds its capacity, i.e., one B-CAV can transport up to one container at a time. Constraint (7) represents that QC starts to handle container $j+1$ only if it finishes handling the former container j. Constraint (8) dictates that the YC can only start to operate on container $j+1$ after it has finished handling container j. Constraint (9) implies that the B-CAV should deliver the loading container j to the transfer point before the QC starts to handle the container. Constraint (10) implies that YC starts to handle container transportation task j after the B-CAV i has completed the delivery of this task. Constraint (11) imposes that the remaining energy of each B-CAV i must over the threshold of warning remaining power. Constraint (12) means that the busy factor of road should be less than 1. Constraint (13) confirms that a B-CAV in the work area is in the work state.

In the proposed model, some B-CAVs' batteries deplete after a few hours of work and need to be dispatched to charging stations. The HTS needs to deploy other full-battery B-CAVs to replace them and execute tasks. Furthermore, as mentioned above, since the loading and unloading container quantities are "vessel-varying," the HTS should reserve enough energy for the next vessel. The energy required S_f of the next vessel is estimated by

$$S_f = J_f \cdot \overline{\beta}, \tag{14}$$

where J_f is the number of containers of the next vessel and $\overline{\beta}$ is the average energy consumption of a B-CAV for transporting a single container. The threshold of energy that B-CAVs should reserve is S_r, which can be computed as follows:

$$S_r = S_f \cdot (1-\gamma) - T_v \cdot \widehat{N}_c \cdot \alpha, \tag{15}$$

where γ is the energy recovery ratio of HTS during work time, and T_v is the idle time between the end berthing time of vessel v and the start berthing time of next vessel $v+1$, $\widehat{N}_c$ is the number of charging B-CAVs during T_v, and α is the energy recovery quantity per unit time of the B-CAV. Considering the constraint in the number of charging stations, only a subset of B-CAVs can be charged during T_v, therefore, let $\widehat{N}_c$ be the number of charging stations. S_a^{k+1} is the estimated total energy of HTS in $k+1$th time window which can be calculated as follows:

$$S_a^{k+1} = S_i^k + S_w^k + S_{wc}^k + \widehat{N}_c^k \cdot \alpha - \widehat{N}_w^k \cdot \overline{\beta}, \tag{16}$$

where S_i^k, S_w^k, and S_{wc}^k denote the battery state of charge (SOC) in period k of idle B-CAVs, recharging B-CAVs and working B-CAVs, respectively, $\widehat{N}_c^k$ is the number of recharging vehicles, and $\widehat{N}_w^k$ is the number of working vehicles. We set up the following constraint to determine the estimated total energy of the HTS in the next period which should be larger than the remaining energy threshold.

$$S_a^{k+1} \geq S_r, \quad \forall k \in K. \tag{17}$$

4. Integrated Scheduling Approach

As mentioned above, the objective of a container terminal is to minimize the overall transportation distance while ensuring uninterrupted loading and unloading operations for the serviced vessel. This section presents the details of the proposed integrated scheduling approach for B-CAVs within container terminals. The approach incorporates three key parts: a sustainable charging policy, a space-time routing method, and an improved GA. These algorithms are designed to collaboratively determine the task assignments and routing plans of the B-CAVs to make HTS more efficient and sustainable.

4.1. Sustainable Charging Policy. Maintaining sufficient available power for the HTS can prevent transport delays and throughput capacity decline at container terminals. Furthermore, advanced charging before the B-CAV SOC reaches the threshold [26] is beneficial in decreasing the traveling distance and the waiting time of B-CAV; thus, we develop an improved advanced recharge method called sustainable charging policy in this work. Before task assignment, we place all B-CAVs waiting for assignment in the idle set I_{id}. If a B-CAV is assigned tasks, it will be moved from I_{id} to the work set I_a. Meanwhile, the B-CAV in working status will be assigned a new transport or charging task when it completes the previous transport task. Moreover, if the SOC of a B-CAV falls below the threshold, it will be assigned a charging task. The B-CAV will drive into the parking area and convert to the idle state again after fully charging. B-CAVs in the set I_a and I_{id} can be further divided into four different preallocated subsets according to their next working type. The four different preallocated subsets

Input: I_{id}, I_a, b_1, and b_2;
Output: I_c, I_{cc}, I_w, and I_{cw};
(1) **foreach** i in I_a **do**
(2) **if** $i_b \le b_1$ **then**
(3) remove i from I_a, and put i in I_c;
(4) **if** $I_{id} \neq \varnothing$ **then**
(5) move one idle B-CAV from I_{id} to I_w;
(6) **while** $S_a^{k+1} > S_r$ **do**
(7) remove i with minimal energy from I_a, and put it in I_c;
(8) **if** $I_{id} \neq \varnothing$ **then**
(9) move one idle B-CAV from I_{id} to I_w;
(10) **foreach** i in I_a **do**
(11) **if** $i_b \le b_2$ **then**
(12) put i into I_{cc};
(13) **if** $I_{id} \neq \varnothing$ **then**
(14) move one idle B-CAV from I_{id} to I_{cw};
(15) **else**
(16) put i into I_w;
(17) **return** I_c, I_{cc}, I_w, and I_{cw}.

Algorithm 1: Sustainable charging policy.

are as follows: the recharging subset $I_c \subseteq I_a$, the candidate recharging subset $I_{cc} \subseteq I_a$, the working subset $I_w \subseteq I_a$, and the candidate working subset $I_{cw} \subseteq I_{id}$. Notably, the B-CAV in I_{cc} may perform the task of either charging or transporting containers, and the B-CAV in I_{cw} may remain idle or transport containers. Considering the remaining energy of the B-CAVs and the future battery requirements of terminal operations, the sustainable charging policy is summarized in Algorithm 1. Note that, this policy only assigns charging tasks to vehicles which are less than the charging threshold and assigns other B-CAVs to the preallocated set. The specific task allocation will be obtained through the IGA mentioned later.

In the algorithm, Steps 1–5 move the B-CAVs with remaining energy less than b_1 from the set I_a into the recharging set I_c; Steps 6–9 move B-CAVs in I_a with the remaining energy less than b_2 into the set I_c to prevent the total remaining energy S_a^{k+1} of the HTS in the next cycle from being less than the remaining threshold S_r; Steps 10–16 aim to identify candidate charging B-CAVs of whose remaining energy are less than b_2 in I_a, which means they may be able to recharge before falling below the lowest threshold b_1. Otherwise, they are put into working subset I_w and wait for the assignment of the next transportation tasks; finally, Step 17 returns the four subsets. The next tasks of B-CAVs in subsets I_c, I_{cc}, I_w, and I_{cw} will be determined according to several factors, specifically, their route distance of executing the next tasks and the state of traffic congestion. Example 1 is used to illustrate Algorithm 1.

Example 1. Assuming that eight B-CAVs now need task assignments, their respective remaining power is presented in Table 1. The remaining power thresholds are set at b_1 = 20% and b_2 = 40%. According to Steps 1–5 of Algorithm 1, B-CAV 1 is dispatched for charging, and B-CAV 6 is dispatched for work. According to Steps 6–9, if the total remaining energy of HTS is less than the remaining threshold, B-CAV 2 is first selected to recharge. Otherwise, Steps 10–16 put B-CAVs 2 and 3 into the candidate recharging subset, allocate B-CAVs 7 and 8 to the candidate working subset, and place B-CAVs 4 and 5 into the working subset.

Table 1: B-CAVs wait assign tasks.

B-CAV	1	2	3	4	5	6	7	8
Remaining energy (%)	18%	27%	33%	67%	82%	100%	100%	100%

4.2. Space-Time Routing Method. Note that if the number of B-CAVs in a road exceeds its storage capacity, congestion tends to occur. Consequently, the routing plan must not only aim at the shortest path but also take into account the storage capacity of planning travel roads. In the yard layout scenario discussed in this paper, there may exist multiple shortest paths between the origin and the destination. To strike a balance between minimizing travel distance and avoiding congestion, we propose a space-time routing method for B-CAVs based on the busy factor and time window, detailed further in Algorithm 2.

Before introducing Algorithm 2, some assumptions are as follows: the B-CAV velocity is constant, B-CAVs are collision-free, the busy factor of each road is set less than or equal to 1, and the task allocation for the set of working vehicles I_{wa} is given, that is, the value of x_{ij} for all $i \in \{1, 2, \ldots, I\}$ and $j \in \{1, 2, \ldots, J\}$ is determined, where I_{wa} is the set of B-CAVs that completed task assignments. The exact task allocation method will be expounded in the next subsection.

Algorithm 2 aims to iteratively generate the route plan of every B-CAV i in I_{wa}. The algorithm starts by employing an empty Tabu list in Step 2, storing prohibited routing plans.

Input: $G = (N, E)$, P, I_{wa}, and x_{ij} for all $i \in \{1, 2, \ldots, I\}$ and $j \in \{1, 2, \ldots, J\}$;
Output: the route plan of B-CAVs in I_{wa};
(1) **foreach** i in I_{wa} **do**
(2) the initial Tabu list of B-CAV i is empty;
(3) **while** True **do**
(4) use Dijkstra algorithm to calculate a shortest route r_i of i for executing task j and r_i should not in i route Tabu list;
(5) calculate the new busy factors matrix $P^{'}$;
(6) **if** the length of r_i is not larger than any route in B-CAV i Tabu list **then**
(7) **if** any element of $P^{'}$ is larger than 1 **then**
(8) put r_i into the Tabu list of B-CAV i;
(9) **else**
(10) $P = P^{'}$, put i into $I_{wa}^{'}$;
(11) break;
(12) **else**
(13) use Dijkstra algorithm to calculate a shortest route r_i of i for executing task j;
(14) put i in $I_{wa}^{'}$;
(15) calculate the new busy factors matrix $P^{''}$;
(16) **foreach** $i^{'}$ in $I_{wa}^{'}$ **do**
(17) calculate a new shortest route of $i^{'}$;
(18) calculate the new busy factors matrix $P^{'''}$;
(19) **if** any element of $P^{'''}$ is larger than 1 **then**
(20) continue;
(21) **else**
(22) $P = P^{'''}$;
(23) break;
(24) break;
(25) **return** routing plan of all i in I_{wa}.

Algorithm 2:Space-time routing method.

Steps 4-5 generate a new routing plan for the current B-CAV and update the busy factors matrix P. If the length of the routing plan is not greater than that of the plan in the Tabu list, Steps 6–11 adopt the result as the current B-CAV routing plan, thereby ensuring that no element of matrix P exceeds 1. Otherwise, the routing plan will be put into the Tabu list of the current B-CAV. If the length of the routing plan is larger than the plan in the Tabu list, then enter a replace loop. In Steps 12–24, by replacing the previous vehicle's routing plan, the busy factor constraints are satisfied for all route plans. Ultimately, Algorithm 2 returns all routing plans of all B-CAVs in set I_{wa}.

Example 2. Suppose the maximum storage capacity of a parallel block road is five and the maximum storage capacity of a perpendicular block road is two. As shown in Figure 2, three B-CAVs deliver unloading containers from quayside vessel A to the yard blocks C_3, C_6, and C_{19}, respectively. According to Algorithm 2, B-CAVs with destinations C_3 and C_6 generate routing plans R_1 and R_2, respectively. However, when planning a route for B-CAV with the destination C_{19}, there is only one shortest route R_3. We depict the current busy factors matrix is shown in Figure 4; obviously, the busy factor of road 23 is $p_{23,7}$ at the 7th time window, and $p_{23,7}$ exceeds 1. Thus, according to Steps 12–24 of Algorithm 2, the route of the B-CAV with destination C_3 is adjusted to $r_1^{'}$, as depicted in Figure 2, which aims to avoid possible traffic jams in the road network.

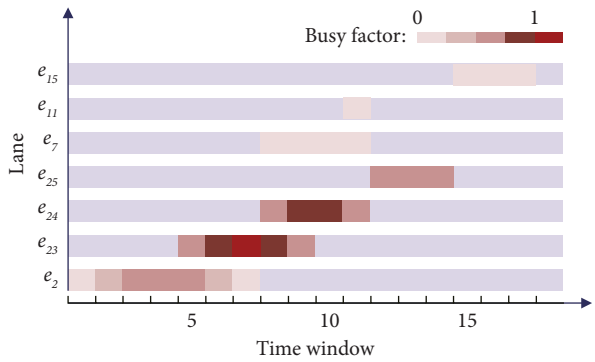

Figure 4: Busy factor of roads in Example 2.

4.3. Improved Genetic Algorithm. The integrated scheduling of transport task assignment, B-CAV route planning, and battery management of HTS results in a vast computation. Meta-heuristic algorithms (e.g., ant colony algorithm, particle swarm optimization, and genetic algorithm) [29–32] are broadly applicable search techniques. They are quite suitable for obtaining accurate or acceptable accuracy within a given time. GA is powerful in solving scheduling problems. Consequently, this work develops an IGA to find the optimal scheduling plan of HTS, the procedure of which is described in Figure 5, wherein transport tasks and the result of the sustainable charging policy are encoded into the chromosomes and the travel distance of the space-time routing result is the fitness evaluation parameter. In addition, we

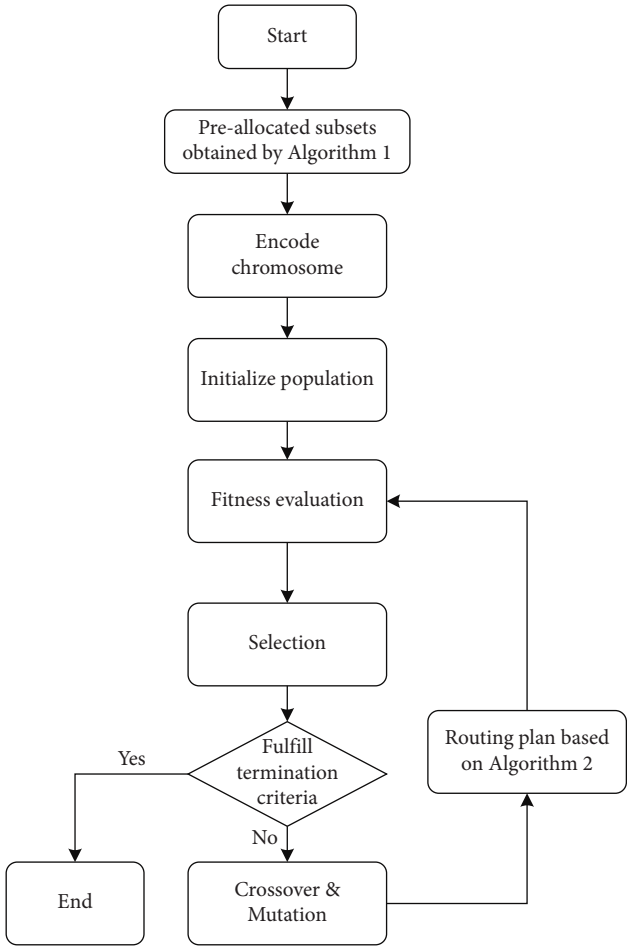

Figure 5: Flowchart of IGA.

redesigned the structure of chromosomes and specifically modified the crossover mutation rules; the details of the algorithm are introduced below.

4.3.1. Encoding of Chromosome and Initialization of the First Generation. Chromosomes are encoded to signify the task schedule of B-CAVs in preallocated subsets, where each chromosome exhibits a three-tier structure, comprising three distinct subchromosomes. Each subchromosome contains genes representing the vehicle number, the unloading task number, and the loading task number, as depicted in Figure 6. In particular, this work divides each subchromosome into two parts: the work part and the exchange part.

The work part is positioned at the former of the subchromosome and the exchange part is in the latter place. These two parts depend on the result calculated by Algorithm 1. The work part represents the B-CAVs in the work set I_w, and the exchange part represents B-CAVs in the set candidate charging set I_{cw} and the candidate work set I_{cc}. Notably, the unique genes 0 can only appear in the exchange part of the last two subchromosomes at the same time, and it represents different tasks according to the corresponding B-CAV. For candidates charging B-CAV, gene 0 denotes the

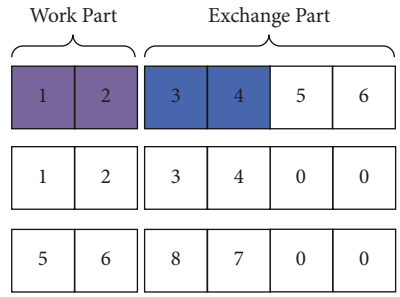

Figure 6: An example of a chromosome.

next task is recharging, and for the candidate working B-CAV, gene 0 indicates an idle state. Similarly, the B-CAVs task schedule can be derived by decoding the chromosome. An example of chromosome coding is shown in Example 3.

Example 3. The SOC and allocate tasks of preallocate B-CAVs are displayed in Table 2. B-CAVs 1 and 2 are allocated to the working part, while B-CAVs 3, 4, 5, and 6 are assigned to the exchange part. B-CAVs 5 and 6 are candidate working vehicles, and their corresponding genes in the last two subchromosomes are denoted as 0. Finally, the encoded chromosome based on Table 2 is shown in Figure 6.

Once a chromosome is encoded according to the tasks and remains SOC of B-CAVs, an initial population which comprised by a set of chromosomes can be constructed. This is accomplished by keeping the first subchromosome unchanged and randomly altering the gene distribution in the remaining two subchromosomes.

4.3.2. Fitness Evaluation and Selection. The fitness function is used to evaluate the quality of chromosomes. Tasks assigned to the B-CAVs are determined by decoding chromosomes and Algorithm 2 is employed to compute the space-time route plan for the B-CAVs. Then, the objective function utilized to derive the minimum travel distance as fitness function value. The roulette method is implemented to select chromosomes after the fitness of all chromosomes has been calculated. The probability of each individual being selected is equal to the proportion of its fitness value to the total fitness value of the entire population.

4.3.3. Crossover and Mutation. The new chromosomes are generated through crossover and mutation operations. A multipoint crossover method is adopted in the crossover operation as illustrated by Figure 7. The crossover only considers the last two subchromosomes of the chromosome, and each part of the chromosome must contain at least one crossover point. The steps of the crossover are as follows: (1) select a series of random points from the work part of a chosen parent and exchange them with the corresponding location of another parent to form the work part of two new child chromosomes; (2) select a series of random points from exchange part of the first parent and exchange them with the corresponding location of second parent to form the exchange part, respectively; and (3) conflict detection is

Table 2: The remaining power and allocated tasks of B-CAVs.

B-CAV	1	2	3	4	5	6
B-CAV SOC	67%	82%	33%	39%	100%	100%
B-CAV set	I_w	I_w	I_{cc}	I_{cc}	I_{cw}	I_{cw}
Unloading tasks	1	2	3	4	0	0
Loading tasks	5	6	7	8	0	0

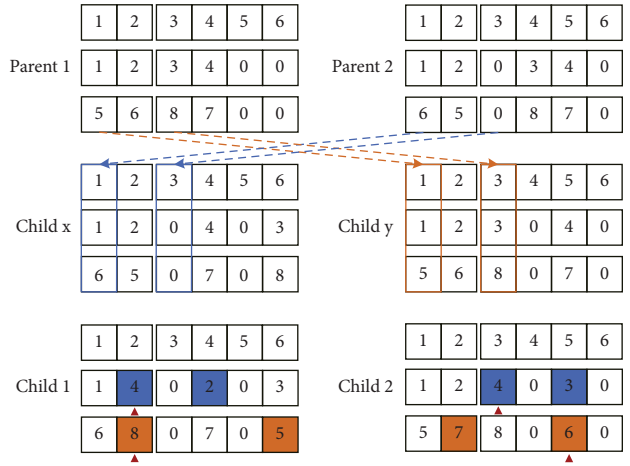

Figure 7: An example of crossover and mutation operations for chromosomes.

performed after the crossover operation to eliminate code exceptions. For example, let us assume two parent chromosomes, parent1 and parent2, as shown in Figure 7, and the results of applying the crossover operation are child x and child y.

In addition, the IGA employs a method of random mutation to modify the sequence of the chromosome genes, thereby ensures the genetic diversity of the population. Particularly, gene mutation only occurs in the same subchromosome and only affects genes encoding nonzero values. An instance of a random mutation is depicted in Figure 7; the mutation genes are marked by red triangles.

In comparison to the GA presented in [1], IGA employs a partition method to address the issue of invalid task assignments to B-CAVs, which is caused by the randomness of GA. In IGA, the partitioning of genes into distinct sections within chromosomes is based on their corresponding B-CAVs preallocated subsets. Furthermore, an essential feature of IGA's crossover and mutation processes mandates that the unique gene 0 is strictly excluded from the work section of the chromosome, thus more effectively for searching the optimal solution.

4.3.4. Stopping Criterion. After a cycle of crossover, mutation, and selection operations, a new batch of populations will be obtained. To balance the time of search computation and attain an approximately optimal solution, this study introduces two optional termination conditions: (1) if the maximum number of generations permitted for IGA evolution exceeds a given integer M_g and (2) if the standard deviation of the fitness values in the current generation is below a small given value. The whole process of the IGA is shown in Figure 5. Finally, the optimal chromosome obtained by the abovementioned method can be translated into practical scheduling plans for the HTS.

5. Simulation Results and Discussion

In this section, several experiments are designed to assess the efficacy of the proposed approach. The test network is from the upgraded traditional container terminal, as illustrated in Figure 2. Several other methods are compared with our method: the exact enumeration algorithm (EEA), which obtains optimal solutions via precise search operations; the nearest assignment algorithm (NAA) [26], which is commonly used for scheduling transportation tasks in container terminals; and the GA used in work [33] to solve AGV scheduling problems within ACT.

To valid the computational efficiency of the proposed method, comparisons are made with EEA, NAA, and GA for small-scale problems. For larger-scale scenarios, the proposed method is primarily compared against NAA and GA. When NAA processes the transportation instruction, it will check the full B-CAV list under the idle status and then assign the nearest B-CAV to handle the corresponding container. A B-CAV is referred as idle if it is neither handling containers nor being recharged. The computational experiments are performed by the microscopic traffic simulation tool SUMO [34] and Python program on a computer with an Intel (R) Core (Tm) CPU@ i7-10710 3.40 GHz and 16 GB of RAM running the Windows 10 operation system.

5.1. Parameters Settings. The following experimental parameters were established for the container terminal yard, which is equipped with 120-yard subblocks and 16 charging stations:

(1) The number of containers varies from 4 to 1200, where the scheduling problem with 4–16 containers is classified as a small-sized problem and those with 100–1200 containers are classified as a large-sized problem in the experiment. The origin and destination locations for transportation tasks are randomly generated across all areas of the terminal yard.

(2) The quantity of B-CAVs ranges from 2 to 36, while the number of QCs varies between 1 and 6. It is assumed that all operation times follow a uniform distribution. The processing time of each QC on these containers follows uniform distribution U (100, 150) seconds.

(3) The genetic algorithm (GA) parameters are set based on preliminary tests: a crossover rate of 0.8, a mutation rate of 0.01, a population size of 100, and a maximum of 100 generations.

5.2. Results and Discussion. First, we consider small-scale problems employing the proposed approach, EEA, NAA, and GA. At the beginning, all B-CAVs of the considered scenario are fully charged. Furthermore, we set the number

Table 3: The small-scale experimental results.

No.	N_Q	N_V	N_C	*EEA*		*NAA*		*GA*		*IGA*	
				Computation time (s)	OFV (km)	Computation time (s)	OFV (km)	Computation time (s)	OFV (km)	Computation time (s)	OFV (km)
1	1	2	4	0.07	6.13	—	6.23	0.10	6.13	0.10	6.13
2	2	2	4	0.18	5.35	—	6.26	0.73	5.35	0.34	5.35
3	3	3	6	0.91	8.16	—	9.12	2.31	8.16	1.98	8.16
4	3	3	9	14.32	14.57	—	15.27	6.00	14.61	3.77	14.61
5	3	4	10	25.83	14.12	—	15.23	6.21	14.13	4.26	14.13
6	4	4	10	20.90	10.81	—	10.84	7.62	10.84	5.32	10.86
7	4	5	10	90.72	12.53	—	12.55	7.01	12.56	5.14	12.77
8	5	5	10	126.42	13.52	—	13.98	7.19	13.58	5.76	13.65
9	5	7	15	612.1	18.93	—	19.63	10.64	18.63	7.87	18.76
10	6	6	16	1510.23	16.26	—	16.65	13.75	16.25	9.34	16.23
11	6	8	24	2611.23	21.98	—	24.43	29.45	22.13	23.95	22.16

of QC $N_Q = 1, \ldots, 6$, the number of B-CAVs$N_V = 2, \ldots, 8$, and the number of container transport tasks $N_C = 4, \ldots, 24$. The objective function value (OFV) and computation time are compared and present in Table 3. From the result, we can figure out that both the EEA and IGA can identify superior optimal solutions. However, as shown in Table 3, the EEA exhibits a longer running time compared to both the GA and the IGA, especially as the number of QCs and B-CAVs increases. Although the NAA has the shortest calculation time, it cannot obtain the optimal solution in some instances. Table 4 illustrates the route result of experiment No. 3 in Table 3. Due to the nearest allocation principle, B-CAV 1 and B-CAV 2 transport only one container in the first cycle between the seaside and the terminal yard. Consequently, this increase in empty transport travel obviously escalates the total travel distance. As the number of transport containers increases, the computation time of the GA also progressively increases compared to the IGA. This is attributable to the novel chromosome structure of the IGA, which prevents generation of the infeasible solutions, thereby enabling efficient acquisition of optimal solutions.

Table 5 displays the parameters and outcomes of the NAA, GA, and IGA in large-scale problems. From these results, it is evident that the OFVs of IGA surpass those of GA and NAA methods, particularly when addressing complex issues. The results demonstrate IGA can get a more cost-effective scheduling plan. Simultaneously, the average travel distance of a B-CAV for transporting a container of NAA is longer than that of the proposed method, which means a single container transport consumes more energy. Furthermore, the comparison of congestion time reveals that our proposed space-time routing method can effectively mitigate container terminal congestion compared to the NAA, which is helpful to increase the efficiency of the terminal. Nevertheless, due to the traffic disturbances and the number of vehicles nearing the system capacity limit, complete avoidance of congestion remains challenging. Based on the above analyses, it can be concluded that the method proposed in this paper is capable of identifying near-optimal solutions of the minimum total travel distance, outperforming both the EEA and the NAA. When compared with the GA, the IGA exhibits faster computational speed.

To verify the suitability of the proposed battery management policy, two types of charging policies are tested in this work. The first, referred to as the conservative charging policy, is derived from [26]. Under this policy, a B-CAV will continuously work until its battery SOC declines to a safety threshold that is set to 20%. The second policy is the sustainable charging policy proposed in this work. In the simulation, we set the QC and container numbers of every ship to the same value. Additionally, the interval time between the loading and unloading processes of the vessels was set to one hour. Figure 8 presents the contrasts between these two policies. Over the course of four distinct experiments, depending on the amount of loading and unloading, the sustainable charging policy reserves more power than the conservative charging policy, and this is related to the amount of future loading and unloading. The results are evident that the sustainable charging policy manages to maintain sufficient energy, aligning with prospective work requirements.

The charging policy also significantly affects the performance of the system. If the number of transportation tasks increases, the total processing time of the terminal will increase, and the number of vehicles required by different charging policies is also different. We make a comparative analysis of the required number of B-CAVs in different transport demand scenarios. We adjust the QC number $N_q = 2, \ldots, 6$ until the cumulative transportation time reaches 12 hours and set the subsequent vessel's transportation task number to zero. As illustrated in Figure 9, in all cases and in comparison to a conservative charging method, our proposed scheduling policy successfully achieved the same volume of transportation tasks with fewer B-CAVs. Notably, as the QC number ascends from 2 to 5, the average number of B-CAVs required per QC witnessed a reduction from 5 to about 3.8. Thus, the proposed integrated scheduling method manifests substantial potential in considerably diminishing the operational expenses of the terminal.

Table 4: The route results of the small-scale experiment No. 3.

V	*Routing plan*	
	NAA	IGA
1	$Q4 \longrightarrow B27 \longrightarrow Q4 \longrightarrow B4 \longrightarrow Q4$	$Q4 \longrightarrow B27 \longrightarrow B12 \longrightarrow Q5$
2	$Q4 \longrightarrow B12 \longrightarrow Q4 \longrightarrow C9 \longrightarrow Q4$	$Q4 \longrightarrow B4 \longrightarrow C9 \longrightarrow Q4$
3	$Q5 \longrightarrow C18 \longrightarrow D24 \longrightarrow Q5$	$Q5 \longrightarrow C18 \longrightarrow D24 \longrightarrow Q5$

Table 5: The large-scale experimental results.

			NAA			*GA*			*IGA*		
No.	N_Q	N_C	OFV (km)	Congestion time (h)	Average travel (km)	OFV (km)	Congestion time (h)	Average travel (km)	OFV (km)	Congestion time (h)	Average travel (km)
1	2	200	250.31	0	1.30	244.22	0	1.17	236.65	0	1.18
2	2	400	517.35	0.61	1.27	464.03	0	1.16	461.23	0	1.15
3	3	500	619.79	1.89	1.22	573.56	0	1.11	555.19	0	1.11
4	3	600	773.33	2.12	1.27	713.75	0.02	1.15	693.93	0.01	1.14
5	4	800	970.47	3.67	1.24	934.43	0.04	1.14	900.37	0.05	1.13
6	4	1000	1275.55	3.56	1.22	1154.75	0.14	1.15	1105.05	0.12	1.13
7	5	1000	1283.86	4.21	1.22	1187.46	0.33	1.13	1110.26	0.34	1.14
8	5	1200	1569.26	4.67	1.26	1396.83	0.47	1.17	1372.06	0.45	1.15
9	6	1200	1543.84	4.35	1.21	1401.02	0.61	1.16	1321.64	0.56	1.16
10	6	1500	1826.94	5.71	1.32	1695.13	0.79	1.25	1615.44	0.72	1.23

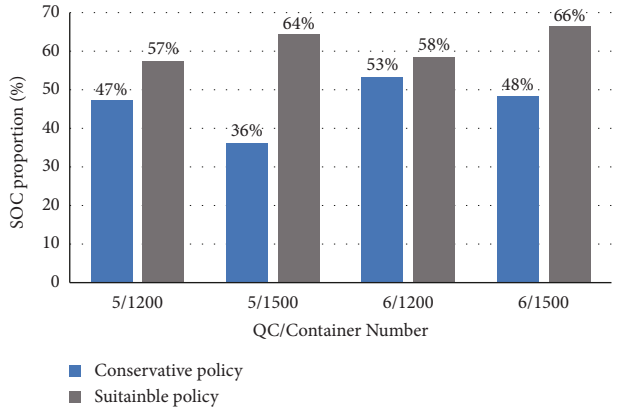

Figure 8: The comparison of the remaining SOC.

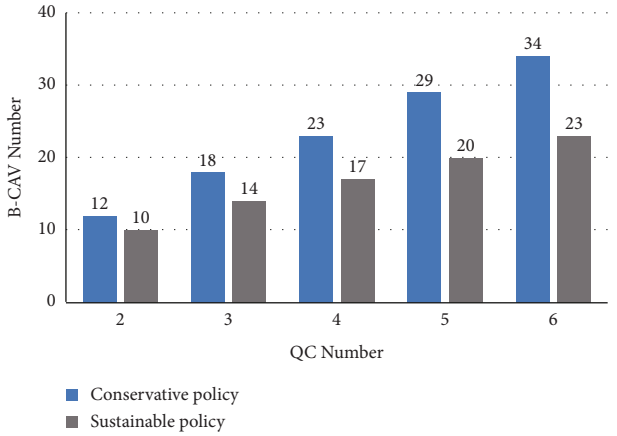

Figure 9: Comparison of the minimum vehicles required for conservative charging policy and sustainable charging policy.

6. Conclusion

In an endeavor to enhance the operational efficiency of the HTS at a container terminal equipped with B-CAVs, we propose an integrated scheduling approach that considers battery management. A novel sustainable charging policy is designed to balance the charging time and working time of B-CAVs. Furthermore, we develop an IGA that incorporates the charging policy and a space-time routing methodology to generate a reasonable task allocation and a congestion-free routing plan with minimum travel distance. Numerical experiments carried out demonstrate that the proposed method is effective in reducing computing time and travel distance. Importantly, the number of required vehicles is reduced under the premise of ensuring sufficient transportation capacity. Apart from QCs and transport vehicles, the whole container terminal efficiency is also influenced by

the yard crane dispatch policy. Interesting future research directions may include taking the yard cranes and other transportation resources into the terminal system scheduling system.

Data Availability

Simulation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Science and Technology Program of Zhejiang Province of China under Grant no. 2023C01174.

References

[1] Y. Yang, M. Zhong, Y. Dessouky, and O. Postolache, "An integrated scheduling method for agv routing in automated container terminals," *Computers & Industrial Engineering*, vol. 126, pp. 482–493, 2018.

[2] M. Grunow, H.-O. Günther, and M. Lehmann, *Dispatching Multi-Load AGVs in Highly Automated Seaport Container Terminals*, Springer Berlin Heidelberg, Berlin, Germany, 2005.

[3] R. Zaghdoud, K. Mesghouni, S. C. Dutilleul, K. Zidi, and K. Ghedira, "A hybrid method for assigning containers to agvs in container terminal," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 96–103, 2016.

[4] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[5] B. Xu, S. E. Li, Y. Bian et al., "Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 322–334, 2018.

[6] X. T. Yang, K. Huang, Z. Zhang, Z. A. Zhang, and F. Lin, "Eco-driving system for connected automated vehicles: multi-objective trajectory optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7837–7849, 2021.

[7] Z. Wang, P. Amar, E. Garmon et al., "Early findings from field trials of heavy-duty truck connected eco-driving system," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, November 2019.

[8] B. K. Lee, L. H. Lee, and E. P. Chew, "Analysis on high throughput layout of container yards," *International Journal of Production Research*, vol. 56, no. 16, pp. 5345–5364, 2018.

[9] E. K. Bish, "A multiple-crane-constrained scheduling problem in a container terminal," *European Journal of Operational Research*, vol. 144, no. 1, pp. 83–107, 2003.

[10] E. Nishimura, A. Imai, and S. Papadimitriou, "Yard trailer routing at a maritime container terminal," *Transportation Research Part E: Logistics and Transportation Review*, vol. 41, no. 1, pp. 53–76, 2005.

[11] D. Adelman, "Price-directed control of a closed logistics queueing network," *Operations Research*, vol. 55, no. 6, pp. 1022–1038, 2007.

[12] D.-H. Lee, J. X. Cao, Q. Shi, and J. H. Chen, "A heuristic algorithm for yard truck scheduling and storage allocation problems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 5, pp. 810–820, 2009.

[13] D. C. Hop, N. Van Hop, and T. T. M. Anh, "Adaptive particle swarm optimization for integrated quay crane and yard truck scheduling problem," *Computers & Industrial Engineering*, vol. 153, Article ID 107075, 2021.

[14] V. D. Nguyen and K. H. Kim, "A dispatching method for automated lifting vehicles in automated port container terminals," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 1002–1020, 2009.

[15] Y. C. Ho and T. W. Liao, "Zone design and control for vehicle collision prevention and load balancing in a zone control agv system," *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 417–432, 2009.

[16] T. Nishi, Y. Hiranaka, and I. E. Grossmann, "A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles," *Computers & Operations Research*, vol. 38, no. 5, pp. 876–888, 2011.

[17] N. Wu and M. Zhou, "Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking," *IEEE*, vol. 12, no. 1, pp. 63–72, 2007.

[18] T. Miyamoto and K. Inoue, "Local and random searches for dispatch and conflict-free routing problem of capacitated agv systems," *Computers & Industrial Engineering*, vol. 91, pp. 1–9, 2016.

[19] H. Fazlollahtabar and M. Saidi-Mehrabad, "Optimal path in an intelligent agv-based manufacturing system," *Transportation Letters*, vol. 7, no. 4, pp. 219–228, 2015.

[20] T. Qin, Y. Du, J. H. Chen, and M. Sha, "Combining mixed integer programming and constraint programming to solve the integrated scheduling problem of container handling operations of a single vessel," *European Journal of Operational Research*, vol. 285, no. 3, pp. 884–901, 2020.

[21] X. Chen, S. He, Y. Zhang, L. C. Tong, P. Shang, and X. Zhou, "Yard crane and agv scheduling in automated container terminal: a multi-robot task allocation framework," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 241–271, 2020.

[22] M. Zhong, Y. Yang, Y. Dessouky, and O. Postolache, "Multi-agv scheduling for conflict-free path planning in automated container terminals," *Computers & Industrial Engineering*, vol. 142, Article ID 106371, 2020.

[23] X. Ma, Y. Bian, and F. Gao, "An improved shuffled frog leaping algorithm for multiload agv dispatching in automated container terminals," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1260196, 13 pages, 2020.

[24] Z. Xing, X. Chen, X. Wang, W. Wu, and R. Hu, "Collision and deadlock avoidance in multi-robot systems based on glued nodes," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 7, pp. 1327–1330, 2022.

[25] A. Skaf, S. Lamrous, Z. Hammoudan, and M.-A. Manier, "Integrated quay crane and yard truck scheduling problem at port of tripoli-Lebanon," *Computers & Industrial Engineering*, vol. 159, Article ID 107448, 2021.

[26] N. Ma, C. Zhou, and A. Stephen, "Simulation model and performance evaluation of battery-powered agv systems in automated container terminals," *Simulation Modelling Practice and Theory*, vol. 106, Article ID 102146, 2021.

[27] X. Xiang and C. Liu, "Modeling and analysis for an automated container terminal considering battery management," *Computers & Industrial Engineering*, vol. 156, Article ID 107258, 2021.

[28] L. Li, Y. Li, R. Liu, Y. Zhou, and E. Pan, "A two-stage stochastic programming for agv scheduling with random tasks and battery swapping in automated container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 174, Article ID 103110, 2023.

[29] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 904–916, 2019.

[30] M. Hamzheei, R. Z. Farahani, and H. Rashidi-Bajgan, "An ant colony-based algorithm for finding the shortest bidirectional path for automated guided vehicles in a block layout," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 1-4, pp. 399–409, 2013.

[31] Y. Wang and X. Zuo, "An effective cloud workflow scheduling approach combining pso and idle time slot-aware rules," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 1079–1094, 2021.

[32] Z. Cao, C. Lin, M. Zhou, C. Zhou, and K. Sedraoui, "Two-stage genetic algorithm for scheduling stochastic unrelated parallel machines in a just-in-time manufacturing context," *IEEE Transactions on Automation Science and Engineering*, vol. 20, pp. 1–14, 2022.

[33] J. Luo and Y. Wu, "Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 79, pp. 49–64, 2015.

[34] P. A. Lopez, M. Behrisch, L. Bieker-Walz et al., "Microscopic traffic simulation using sumo," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, Maui, HI, USA, November 2018.