

## Article

# Optimization of Joint Scheduling for Automated Guided Vehicles and Unmanned Container Trucks at Automated Container Terminals Considering Conflicts

Liangyong Chu <sup>1,2</sup> , Zijian Gao <sup>1</sup>, Shuo Dang <sup>3,\*</sup>, Jiawen Zhang <sup>1</sup> and Qing Yu <sup>1,\*</sup> <sup>1</sup> School of Navigation, Jimei University, Xiamen 361021, China; chuliangyong@163.com (L.C.); zjw1561577580@163.com (J.Z.)<sup>2</sup> Fujian Shipping Research Institute, Xiamen 361021, China<sup>3</sup> Management School, University of Liverpool, Liverpool L69 7ZH, UK

\* Correspondence: s.dang@liverpool.ac.uk (S.D.); qing.yu@jmu.edu.cn (Q.Y.)

**Abstract:** Port development is a critical component in constructing a resilient transportation infrastructure. The burgeoning integration of automated guided vehicles (AGVs) within container terminals, in conjunction with the orchestrated scheduling of unmanned container trucks (UCTs), is essential for the sustainable expansion of port operations in the future. This study examined the influence of AGVs in automated container terminals and the synergistic scheduling of UCTs on port operations. Comparative experiments were meticulously designed to evaluate the feasibility of integrated scheduling schemes. Through the development of optimization models that consider conflict-free paths for both AGVs and UCTs, as well as strategies for conflict resolution, a thorough analysis was performed. Advanced genetic algorithms were engineered to address task-dispatching models. In contrast, the A\* optimization search algorithm was adapted to devise conflict-free and conflict-resolution paths for the two vehicle types. A range of scaled scenarios was utilized to assess the impact of AGVs and UCTs on the joint-scheduling process across various configuration ratios. The effectiveness of the strategies was appraised by comparing the resultant path outcomes. Additionally, comparative algorithmic experiments were executed to substantiate the adaptability, efficacy, and computational efficiency of the algorithms in relation to the models. The experimental results highlight the viability of tackling the joint-scheduling challenge presented by AGVs and UCTs in automated container terminals. When juxtaposed with alternative scheduling paradigms that operate independently, this integrated approach exhibits superior performance in optimizing the total operational costs. Consequently, it provides significant insights into enhancing port scheduling practices.

**Keywords:** unmanned container trucks; joint scheduling; automated guided vehicles; improved genetic algorithm; conflict-resolution strategies



**Citation:** Chu, L.; Gao, Z.; Dang, S.; Zhang, J.; Yu, Q. Optimization of Joint Scheduling for Automated Guided Vehicles and Unmanned Container Trucks at Automated Container Terminals Considering Conflicts. *J. Mar. Sci. Eng.* **2024**, *12*, 1190. <https://doi.org/10.3390/jmse12071190>

Academic Editors: Lingxiao Wu and Shuaian Wang

Received: 6 June 2024

Revised: 10 July 2024

Accepted: 11 July 2024

Published: 16 July 2024



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

## 1. Introduction

Ports are indispensable to the development of the Silk Road Economic Belt, acting as the linchpin that connects economic development with openness under the Belt and Road Initiative. From the perspective of smart city advancement, automated container terminals have become central to the integrated transportation systems in numerous coastal urban centers. Confronted with escalating labor costs and rising service expectations, these terminals are progressively adopting cutting-edge technologies to achieve comprehensive situational awareness and promote the global sharing of big data platforms. These facilities are set to act as essential pathways for the development and enhancement of international logistics networks and the seamless integration of global supply chains. Therefore, improving port operational efficiency, mitigating congestion, and reducing operational costs in specialized settings depend on enhancing efficiency across various aspects of automated terminals.

In China, several emerging smart ports have begun using unmanned container trucks for horizontal container transportation tasks. Unmanned container trucks present significant cost benefits over automated guided vehicles (AGVs) regarding equipment expenditure; however, their economic viability depends on the terminal's operational efficiency. Notably, the Phase IV strategy at the Shanghai Yangshan Port incorporates collaborative transportation with AGVs and unmanned container trucks. AGVs are responsible for intra-port container loading and unloading transfers. In contrast, unmanned container trucks predominantly oversee external transportation routes from the Yangshan Port to the Donghai Bridge and Lingang Logistics Park. The incorporation of unmanned container trucks has become a key area of research in the scheduling of multi-type horizontal transportation equipment within automated container terminals. Given the dynamic trends in smart port development across various sectors and industries, research into the joint scheduling of AGVs and unmanned container trucks in automated container terminals provides valuable insights for optimizing port resource utilization.

The assessment of operational efficiency and cost reduction in automated container terminals extends beyond the singular focus on AGV operations. Scholars are increasingly investigating cooperative operations among diverse equipment types within ports. Comprehensive research on the joint scheduling of AGVs and unmanned container trucks considers factors that impact port efficiency both within and outside terminals. It formulates models to analyze the optimal ratio of AGV and unmanned container truck configurations to minimize total time costs and compares the advantages of overall terminal scheduling against scheduling that is solely terminal-centric. The joint scheduling of AGVs and unmanned container trucks introduces a novel business paradigm for automated container terminals. The improvements in efficiency derived from joint scheduling are expected to propel sustainable port development and establish a more holistic and superior mode of port service delivery.

The subsequent sections of this paper are organized as follows: Section 2 reviews the pertinent research on AGVs and unmanned container trucks. Section 3 elucidates the factors and models considered in this study. Section 4 details the heuristics and algorithms applied to solve the proposed model. Section 5 presents the case studies. Finally, Section 6 encapsulates the conclusions drawn from this research.

## 2. Literature Review

Most studies concerning the scheduling of horizontal transportation equipment in container terminals have primarily concentrated on task assignments and path planning for container trucks, with the considerations varying based on the type and complexity of the ports. As ports progress in their developmental stages, research into AGV scheduling issues commenced with replacing container trucks (tractors) with AGVs for horizontal transportation. The introduction of automated quay cranes (AQC) and automated rail-mounted gantry cranes (ARMG) has spurred further investigation into multi-resource coordination scheduling for automated container terminals. Given the distinctive features of AGVs, which employ embedded magnetic guidance for fixed-route navigation and are powered by new energy sources, recent research on AGV scheduling has begun incorporating factors such as conflict congestion, energy consumption costs, and charging strategies.

### 2.1. Research on the Scheduling of Various Types of Container Trucks in Automated Container Terminals

The existing literature predominantly addresses AGV scheduling challenges, encompassing task assignment and optimization problems such as path optimization and congestion management. Due to the varied complexities inherent in actual port operations, AGV scheduling is subject to the influence of multiple factors, resulting in diverse scheduling approaches.

#### (1) Analysis of simulation- and algorithm-based scheduling optimization models

Luo et al. [1] proposed an AGV scheduling optimization model to minimize berth waiting times during unloading and loading processes, employing a genetic algorithm (GA)

to address large-scale instances. Zaghdoud-R et al. [2] tackled AGV path-planning, task assignment, and scheduling issues utilizing a hybrid algorithm combining Dijkstra and GA, demonstrating adaptability to different scenarios and robust convergence capabilities. Tao et al. [3] devised a mathematical model for scheduling container trucks (traditional trailers), accounting for empty scheduling and location balance. They employed a discrete-time approach to determine fitness function factors, integrating it with an evolutionary search algorithm to enhance vehicle service time efficiency and reduce empty vehicle rates. Hu et al. [4] focused on AGV scheduling by incorporating container clustering operations to minimize empty travel distances, thereby achieving shorter working times. They compared the performance of fuzzy membership and Pareto functions in transforming multi-objective functions into single-objective function models for optimal solution finding, validating that Pareto functions effectively reduce empty travel distances and expedite completion times. In a separate study, Hu et al. [5] modeled vehicle-scheduling and storage-allocation problems when Automatic Lifting Vehicles (ALVs) and AGVs alternately operated at terminals. They proposed a three-stage decomposition method leveraging particle swarm optimization and greedy algorithms to address the model, demonstrating the adaptability of the algorithms in model resolution and their efficacy in reducing vehicle operation costs.

## (2) AGV path conflict scheduling optimization models

Yue et al. [6] addressed AGV delay issues arising from uncertainties related to quay-crane wait times. They proposed a two-stage hybrid model to minimize transportation costs by integrating Dijkstra's and Q-learning algorithms to tackle the problem. Additionally, they introduced a graph-based conflict-avoidance strategy to resolve operational conflicts among AGVs. Su [7] designed a two-tier objective function model to minimize both the task completion time and the shortest AGV travel distance, thereby mitigating AGV path conflicts. They developed adaptive algorithms based on this model and validated their effectiveness through case studies. These studies commonly utilized shortest-path algorithms to address AGV conflicts, leveraging graph theory-related algorithms to not only plan the shortest routes but also prevent operational conflicts related to the specific nature of AGV travel paths.

## (3) Energy saving and carbon emissions in scheduling optimization models

Traditional fuel-powered container trucks (trailers) operating at ports inevitably contribute to environmental pollution due to exhaust emissions. Chen et al. [8] addressed the carbon-emission issue of container trucks at ports by establishing a dual-objective model to reduce the truck waiting time to meet emission-reduction goals. With the increasing adoption of new energy vehicles, such as AGVs and unmanned container trucks at automated terminals, carbon emissions from AGV operations have been minimized. However, energy conservation has emerged as a new challenge in scheduling AGVs, unmanned container trucks, and other vehicles at automated terminals. More studies focusing on energy savings and emission reduction at automated terminals often explore this issue in conjunction with overall terminal resource coordination. Research into optimizing resource coordination at automated terminals not only effectively reduces carbon emission costs but also holds significant importance for enhancing resource efficiency and cutting costs at terminals.

## 2.2. Research on Cooperative Scheduling of Multiple Equipment Resources in Automated Container Terminals

Cooperative scheduling problems involving multiple equipment resources in automated container terminals primarily revolve around quay cranes, yard cranes, and horizontal transportation equipment. These problems entail modeling equipment task sequences and considering actual operational scenarios to design various types of algorithms for resolution.

### (1) Equipment interactions in cooperative scheduling

In the operational phase of automated terminals, due to the interconnected task processes, models are frequently devised to minimize the shortest completion time for

containers. Lu et al. [9] delved into the problem of multi-resource cooperative scheduling under uncertain conditions and proposed a cooperative scheduling model aimed at minimizing the operation time. They employed a particle swarm algorithm to solve this uncertain model and noted its superior stability and performance in case studies. Yang et al. [10] established a bi-level planning model to minimize the maximum completion time for solving multi-resource cooperative scheduling issues. They compared the rolling horizon process algorithm (RHP) with the congestion prevention strategy bi-level genetic algorithm (CRP-BGA) regarding algorithm convergence and cost calculations, affirming the latter's effectiveness in addressing cooperative scheduling problems. Additionally, alongside the earlier approaches, research on modeling specific equipment relationships offers solutions to resource coordination scheduling issues. Homayouni et al. [11] examined the placement of containers and the operational status of terminal equipment to tackle the optimization problem of resource coordination scheduling for quay cranes, yard cranes, and AGVs. They demonstrated the superiority of genetic algorithms over simulated annealing algorithms in solving this problem, validating the feasibility of enhancing terminal work efficiency through resource coordination scheduling. Kizilay et al. [12] accounted for the safety and interference effects of inbound and outbound containers on terminal cranes, resolving multi-resource cooperative scheduling problems using a two-stage optimization CP model and algorithm. Zhou C et al. [13] considered the dynamic movement of yard cranes and container trucks, establishing a cooperative optimization model for yard cranes and container trucks. They utilized a two-stage heuristic algorithm based on tabu search to address the model, comparing and analyzing the superiority of tabu search over other benchmark algorithms in dynamic scenarios and assessing the model's effectiveness in reducing the working time. Yue et al. [14] researched the cooperative scheduling optimization problem of dual-trolley quay cranes and AGVs, proposing a two-tier, dual-layer model to maximize customer satisfaction and minimize AGV delays. They introduced a two-stage optimization algorithm to solve the model, validating the effectiveness of this method in configuring scheduling plans based on customer satisfaction.

## (2) Cooperative scheduling problems in terminal loading and unloading

Considering the characteristics of terminal loading and unloading processes, especially the disparities in time and space between automated terminals with novel equipment and conventional container-loading and -unloading processes, the cooperative scheduling problem of different loading and unloading processes emerges as a pivotal research area. Hop et al. [15] studied the problem of minimizing the loading and unloading processes and transportation time of terminal container tasks, advocating the use of an adaptive particle swarm algorithm. They conducted case studies to verify its performance advantages over fixed particle swarm and grey wolf optimization algorithms in time calculations. Xu et al. [16] considered the U-shaped layout and loading and unloading processes of automated terminals, examining the optimization problem of cooperative scheduling of resources for dual-trolley quay cranes, twin-arm rail-mounted yard cranes, and AGV vehicles. They eradicated the interaction waiting time to achieve temporal and spatial synchronization between AGVs and twin-arm, rail-mounted yard cranes. They reduced the task completion time by employing a reinforcement learning metaheuristic genetic algorithm with a reward and penalty mechanism to address the model. They conducted case studies to confirm the universality and effectiveness of this algorithm independent of specific problems, with the model and algorithm yielding satisfactory solutions to the issue of AGV conflicts.

## (3) Consideration of Cooperative Scheduling problems in AGV conflicts

In the area of container truck scheduling, resource coordination scheduling serves as a crucial foundation for AGVs or other types of container trucks during the transportation process. At the same time, conflicts and congestion often arise in horizontal transportation. Thus, comprehensive coordination of terminal resources can not only prevent conflicts and reduce congestion rates but also enhance terminal efficiency. Shou et al. [17] addressed

the issues of AGV conflicts, congestion, and multi-resource cooperative scheduling and devised a dual-layer scheduling model for loading and unloading processes alongside AGV transportation processes involving the same loading and unloading. They applied a dual-layer genetic algorithm and a dual-layer adaptive genetic algorithm, integrating conflict-resolution strategies into top- and bottom-level models, and validated the effectiveness of the algorithms through experiments with previous data, showcasing their adaptability in tackling AGV congestion and cooperative scheduling of automated terminal resource equipment. Similarly, Zhong et al. [18] addressed known task assignments to establish a resource coordination scheduling problem model focusing on path planning, cooperative scheduling, and AGV conflict-deadlock minimization. They devised a particle swarm algorithm based on fuzzy control to address the model. They verified its effectiveness in resolving AGV conflict deadlocks, demonstrating its capacity to enhance the overall efficiency of automated terminals. Cooperative scheduling of port equipment resources encompasses various sub-problems, including path optimization, task assignment, and conflict congestion, all requiring consideration. Solving individual sub-problems in isolation fails to achieve the overarching goals of optimization and efficiency improvement. Thus, formulating a framework for cooperative scheduling of equipment resources lays the groundwork for research in this area, which should be guided by such principles to realize the objective of global optimization.

### *2.3. Research on Cooperative Scheduling of Multiple Types of Automated Container Trucks*

Scheduling problems involving interactions between automated terminals and external container trucks typically manifest in two scenarios: delivery and pickup of loaded containers. The subsequent studies delve into the scheduling problem of external container trucks under the single-flow mode (i.e., external trucks do not return empty containers to the terminal after delivery).

Cui et al. [19] considered a single-flow transportation scenario by devising a two-stage framework to minimize time and operational costs. They ensured adequate task sequences for external trucks and employed a hybrid algorithm combining a large neighborhood search, and tabu search to tackle the problem. Tang et al. [20] addressed a single-flow transportation scenario by modeling the problem to minimize the maximum completion time. They proposed an enhanced particle swarm algorithm with a novel velocity update strategy to address the model, demonstrating the adaptability of heuristic algorithms compared to exact algorithms through experiments.

In addition to the aforementioned modeling approaches, studies in this category also aimed to optimize external truck queue waiting times to achieve the minimum cost objective using time window constraints. Chen et al. [21] considered the problem of vehicle dispatching with time windows (VDTWs) and estimated the queue length of external trucks based on the time-window predictions while minimizing the total cost. They developed a hybrid algorithm combining GA and another improved genetic algorithm (IGA) to solve this problem, thus validating the effectiveness of VDTWs in controlling external truck arrival times and reducing congestion. Nossack et al. [22] formulated a full truck pickup and delivery problem with time windows (FTPDPW), minimizing the operation time of external trucks with time window constraints at both the terminal and customer ends. They devised a two-stage heuristic algorithm to solve the model and confirmed the adaptability of heuristic algorithms over exact algorithms in addressing this problem. Additionally, alongside the utilization of time window constraints, modeling analysis was conducted to minimize the operation time. Chen et al. [23] addressed the external truck queuing problem with random service market distribution by formulating a convex non-linear programming model to minimize total truck turning and waiting times. They employed a two-stage adaptive optimization algorithm to solve the model. Azab et al. [24] considered the long waiting times and harmful emissions of external trucks, establishing a deterministic scheduling model to minimize the external truck turnaround time. They used simulation

experiments to verify the model's performance and established an IoT-based reservation system to significantly enhance terminal efficiency.

Through in-depth studies on the delivery and pickup problems of external trucks, scholars have gradually addressed issues such as typical cross-dock transportation problems at terminals. This problem primarily focuses on optimizing waiting times and costs from external trucks exchanging cargo containers at terminals. Rijal et al. [25] established a scheduling model for external truck scheduling and mixed terminal port operations (simultaneous inbound and outbound operations) based on external truck round trips between terminals and customer sites. They designed an adaptive large-neighborhood algorithm to solve the model, effectively reducing transportation costs compared to sequential methods. Heidari et al. [26] considered uncertainties in external truck round trips between terminals and customer sites, formulating a model to minimize costs under unknown arrival times of external trucks. They developed a MODE algorithm and a nondominated sorting genetic algorithm-II (NSGA-II) hybrid algorithm to solve the model, demonstrating superior performance compared to genetic algorithms based on a random search. Xi et al. [27] considered uncertainties in external truck round trips between terminals and customer sites, introducing the concept of conflicts and minimizing their costs and quantities. They established a two-stage model and designed column and constraint generation algorithms, validating the effectiveness of the model and algorithms through case studies.

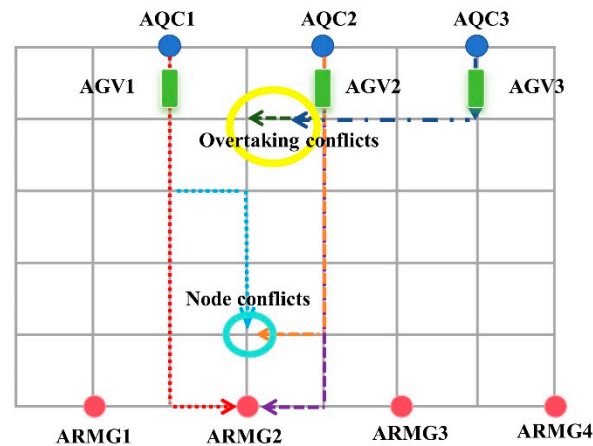
Research on scheduling problems involving various types of automated container trucks in automated terminals has predominantly focused on task sequences, path optimization, and conflict congestion. Studies addressing the cooperative scheduling of multi-resource equipment aim to optimize AGV scheduling and task sequences, considering loading and unloading processes and the interplay among terminal equipment. Scheduling challenges concerning the interactions between external container trucks and terminals primarily concentrate on optimizing waiting times outside terminals and task-order sequences to minimize terminal transportation costs and maximize efficiency. This research emphasizes multi-resource equipment scheduling at terminals, waiting times for external truck queues, AGV operation conflicts and congestion, loading and unloading processes, cost issues, and carbon emissions to further reduce costs and optimize overall efficiency.

### 3. Model

AGVs rely on ground magnetic nail navigation, with their travel paths being fixed. Path conflicts may arise when container tasks are concentrated in specific areas. AGVs are relatively slow in resolving path conflicts, and uneven task assignments can result in early arrival at task nodes, leading to terminal congestion. Therefore, both conflict and congestion are critical factors affecting the efficiency of automated container terminal operations.

#### 3.1. Problem Description

AGVs and unmanned container trucks handle container tasks, with AGVs responsible for seaside container transport and unmanned container trucks for landside container transport. Utilizing an automated container terminal featuring a vertical distribution of shore crane lines and stacking yards provides ample operating space for AGVs on the land side, enabling them to operate effectively in the front working area and forming a barrier with minimal interference from the internal yard and landside operations. As illustrated in Figure 1, AGVs traveling from an automated quay crane (AQC) to an automated rail-mounted gantry (ARMG) have multiple paths within a specified traffic direction at the terminal. The paths of the two AGVs intersect at certain points, potentially resulting in path conflict if both AGVs enter the intersection simultaneously. Common conflicts include overtaking and node conflicts. Adopting conflict-resolution strategies allows AGVs to make decisions based on the encountered conflicts, thus resolving them and completing tasks.



**Figure 1.** The path diagram for AGVs and unmanned container trucks.

This section divides the joint-scheduling optimization problem for AGVs and unmanned container trucks at automated container terminals into two stages. In the first stage, container tasks are allocated, resulting in sequences of tasks to be serviced by AGVs and unmanned container trucks. In the second stage, optimization is applied to the AGV paths to obtain the optimal route. Using the two-stage model, conflict situations are addressed by implementing strategies such as high-priority passage and low-priority parking/waiting when planning AGV paths. These considerations are integrated into the overall joint scheduling cost.

### 3.2. Model Assumptions

- (1) Terminal handling equipment with AGVs and unmanned container trucks operating on individual containers are averaged over time;
- (2) Unmanned container trucks continue transportation from the yard to the logistics park for the entire ship unloading task. Unmanned container trucks wait for task instructions without considering dispatching entry times;
- (3) Unmanned container trucks travel unidirectionally depending on the flow direction of the terminal lanes, considering queue congestion issues and ignoring path conflicts;
- (4) During the unloading process of a ship's imports, the positions of the containers in the vessel and yard are not considered. The unloading sequence of the container tasks is randomly determined;
- (5) Both AGV and unmanned container trucks have sufficient battery levels, and the battery capacity allows them to complete their entire transportation tasks.

### 3.3. Joint-Scheduling and Task-Allocation Model

The task assignment model is similar to the scenario considered in Section 4. The primary approach involves allocating container tasks to AGVs and unmanned container trucks based on the terminal operating rules. Both AGVs and unmanned container trucks meet the standard requirements for container loading. The parameters, variables, and sets are listed in Table 1.

The task-dispatch model primarily illustrates the operational interactions between the loading/unloading equipment and the horizontal transportation equipment throughout the container-transportation process. This determines the service sequence of each piece of equipment during container unloading. Simultaneously, AGVs and unmanned container trucks achieve joint scheduling, utilizing containers as the medium. Through continuous optimization of scheduling tasks, the overall task time at the terminal is ultimately reduced, leading to optimized objectives in terms of cost.

**Table 1.** Parameters, variables, and sets of the joint-scheduling and task-allocation model.

Notation	Sets
$L$	The set of sequential numbers for unloading container task orders, $L = \{1, 2, \dots, l\}$ ;
$Q$	The set of quay cranes, $Q = \{1, 2, \dots, q\}$ ;
$D$	The set of yard container areas, $D = \{1, 2, \dots, d\}$ ;
$A$	The set of all AGVs, $A = \{1, 2, \dots, a\}$ ;
$K$	The set of all unmanned container trucks, $K = \{1, 2, \dots, k\}$ ;
$Z$	The set of terminal gates, $Z = \{1, 2, \dots, z\}$ ;
$W$	The set of container areas in the logistics park, $W = \{1, 2, \dots, w\}$ ;
$P$	The number of buffer racks in a single container area, indicating the quantity of buffer racks available for placing containers;
$T_{lq}^r$	The scheduled time for quay crane $q$ to handle the $l$ -th container;
$T_{lq}^R$	The actual time for quay crane $q$ to handle the $l$ -th container;
$T_{al}$	The time when the $l$ -th container-handling task begins for the $a$ -th AGV at either the quay crane or in the yard;
$T_{dl}$	The time when the $l$ -th container is placed on the buffer rack in yard $d$ ;
$\alpha_l$	The average operation time for quay crane unloading container $l$ ;
$\beta_l$	The average operation time for the yard crane to extract container $l$ and place it at the destination position;
$\gamma_l$	The average operation time for unloading container $l$ from the unmanned container truck to the yard in the logistics park;
$\alpha_{l,l+1}$	The time taken for the next container task, after handling container $l$ , to be processed at the quay crane;
$\beta_{l,l+1}$	The time taken for the next container task, after handling container $l$ , to be processed in the yard;
$[ET_{lc}, LT_{lc}]$	The time window during which the AGV places container $l$ on the buffer rack in the yard's ARMG area;
$t_{al}$	The time taken for the $a$ -th AGV to transport container $l$ to its destination;
$t_{al,al+1}$	The time taken for the $a$ -th AGV to travel empty from the delivery point of container $l$ to the next task point at $l + 1$ ;
$t_{aql}^m$	The waiting time for the $a$ -th AGV when transporting container $l$ to quay crane $q$ ;
$t_{adl}^m$	The waiting time for the $a$ -th AGV when transporting container $l$ to yard crane $d$ (under the buffer rack);
$t_{kdl}^m$	The waiting time for the $k$ -th unmanned container truck when retrieving container $l$ from quay crane $d$ ;
$t_{kzl}^m$	The waiting time for the $k$ -th unmanned container truck when retrieving container $l$ at gate $z$ , including the queuing time;
$T_{kl}$	The arrival time of the $k$ -th unmanned container truck at the yard to retrieve container $l$ ;
$T'_{dl}$	The time when the yard crane handles the $l$ -th container;
$T_{zkl}$	The time when the $k$ -th unmanned container truck, loaded with the $l$ -th container, exits the gate;
$T_{wkl}$	The time when the $k$ -th unmanned container truck unloads the $l$ -th container in the logistics park;
$t_{kl}$	The average time taken by the $k$ -th unmanned container truck to transport container $l$ to its destination;
$t_w$	The average operation time for loading and unloading containers in the logistics park;
$t_{zw}$	The average travel time on the bridge for an unmanned container truck after leaving the terminal gate;
$M$	A large positive integer.

The decision variables of the model are as follows:

- $x_{al}$ : A binary variable that equals one if container  $l$  is assigned to the  $a$ -th AGV; otherwise, it equals zero.
- $y_{ql}$ : A binary variable that equals one if container  $l$  is handled by quay crane  $q$ ; otherwise, it equals zero.

$z_{dl}$ : A binary variable that equals one if container  $l$  is handled by yard crane  $d$ ; otherwise, it equals zero.

$x_{al,al+1}$ : A binary variable that equals one if the  $a$ -th AGV continues to work on container  $l + 1$  after delivering container  $l$ ; otherwise, it equals zero.

Objective:

$$f_{\min} = f_1 + f_2 + f_3 + f_0 \quad (1)$$

$$f_1 = \varepsilon_1 \sum_{a \in A} \sum_{l \in L} (t_{al} + t_{al,al+1}) \quad (2)$$

$$f_2 = \varepsilon_2 \left( \sum_{l \in L} \sum_{a \in A} \sum_{q \in Q} t_{aq}^m + \sum_{l \in L} \sum_{d \in D} \sum_{q \in Q} t_{adl}^m \right) \quad (3)$$

$$f_3 = \varepsilon_2 \left( \sum_{k \in K} \sum_{l \in L} \sum_{d \in D} t_{kdl}^m + \sum_{k \in K} \sum_{l \in L} \sum_{z \in Z} t_{kzl}^m \right) \quad (4)$$

Equation (1) represents the minimization of the overall cost of the joint scheduling problem, where  $f_0$  denotes the fixed operational cost. Equation (2) aims to minimize the total transportation cost  $f_1$  for AGVs, with  $\varepsilon_1$  as the coefficient for the total transportation cost. Equation (3) accounts for the penalty cost  $f_2$  associated with the AGV waiting for the AQC operation and under the buffer rack, where  $\varepsilon_2$  is the penalty cost coefficient. Equation (4) represents the penalty cost  $f_3$  for unmanned container trucks waiting under the ARMG at the landside and queuing at the gate, where  $\varepsilon_2$  is the penalty cost coefficient.

The objective is subject to the following constraints:

$$\sum_{L=1}^l x_{al} = 1, \forall a \in A \quad (5)$$

$$\sum_{L=1}^l x_{kl} = 1, \forall k \in K \quad (6)$$

$$\sum_{A=1}^a x_{al} = 1, \forall l \in L \quad (7)$$

$$\sum_{A=1}^a x_{al,a(l+1)} = 1, \forall l \in L \quad (8)$$

$$\sum_{K=1}^k x_{kl} = 1, \forall l \in L, \forall k \in K \quad (9)$$

$$\sum_{K=1}^k x_{kl,k(l+1)} = 1, \forall l \in L, \forall k \in K \quad (10)$$

$$\sum_{Q=1}^q y_{ql} = 1, \forall l \in L \quad (11)$$

$$\sum_{D=1}^d z_{dl} = 1, \forall l \in L \quad (12)$$

$$T_{(l+1)q}^R = \max \left[ T_{lq}^r + \alpha_{l,l+1}, T_{(l+1)q}^R \right], \forall l \in L, \forall q \in Q \quad (13)$$

$$T_{lq}^R = \max \left[ T_{al}, T_{lq}^r \right], \forall l \in L, \forall q \in Q, \forall a \in A \quad (14)$$

$$T_{lq}^r \leq T_{lq}^R, \forall l \in L, \forall q \in Q \quad (15)$$

$$T_{(l+1)q}^r - T_{lq}^r \leq T_{(l+1)q}^R - T_{lq}^R, \forall l \in L, \forall q \in Q \quad (16)$$

$$ET_{dl} = T_{dl} - p\beta_{l,l+1}, LT_{dl} = T_{dl}, \forall l \in L, \forall d \in D \quad (17)$$

$$ET_{dl} \leq T_{dl} + t_{al} \leq LT_{dl}, \forall l \in L, \forall d \in D, \forall a \in A \quad (18)$$

$$t_{adl}^m = \max\{ET_{dl} - T_{al} - t_{al}, 0\}, \forall l \in L, \forall d \in D, \forall a \in A \quad (19)$$

$$t_{aqd}^m = \max\{T_{lq}^R - T_{al}, 0\}, \forall l \in L, \forall q \in Q, \forall a \in A \quad (20)$$

$$T_{al} + t_{al} + t_{adl}^m + t_{al,a(l+1)} + t_{ad(l+1)}^m \leq T_{a(l+1)} + M(1 - x_{al,a(l+1)}), \forall l \in L, \forall d \in D, \forall a \in A \quad (21)$$

$$T_{al} + t_{al} + t_{adl}^m + t_{al,a(l+1)} + t_{aq(l+1)}^m \leq T_{a(l+1)} + M(1 - x_{al,a(l+1)}), \forall l \in L, \forall d \in D, \forall q \in Q, \forall a \in A \quad (22)$$

$$T_{dl} + \beta_{l,l+1} \leq T'_{dl}, \forall l \in L, \forall d \in D \quad (23)$$

$$t_{kdl}^m = \max\{T'_{dl} - T_{kl}, 0\}, \forall l \in L, \forall d \in D, \forall k \in K \quad (24)$$

$$T'_{dl} + \beta_l \leq T_{kl}, \forall l \in L, \forall d \in D, \forall k \in K \quad (25)$$

$$T_{zk(l+1)} = t_{kzl}^m + T_{zkl}, \forall l \in L, \forall z \in Z, \forall k \in K \quad (26)$$

$$T_{wkl} = T_{zkl} + t_{zwl}, \forall l \in L, \forall w \in W, \forall k \in K, \forall z \in Z \quad (27)$$

Constraints (5) and (6) stipulate that each AGV and unmanned container truck can only complete one container task at a time. Constraints (7)–(10) specify that each container task can only be executed by either an AGV or an unmanned container truck. Constraints (11) and (12) specify that each container task must be handled by either a quay or yard crane. Constraint (13) denotes the operation time for quay cranes' container-handling tasks other than the initial task, while constraint (14) represents the actual operation time for the quay crane unloading the container. Constraint (15) illustrates the relationship between the actual operation time and the planned operation time for a quay crane. Constraint (16) ensures that the operation interval for the same quay crane between adjacent tasks meets the planned operation interval. Constraint (17) guarantees that the actual operation time for the AGV transporting unloaded containers to the buffer rack complies with the time-window constraint. Constraint (18) ensures that the time required for the AGV to extract and transfer containers meets the time-window constraint. Constraint (19) reflects the waiting time for the AGV to deliver the unloaded containers under the quay crane. Constraint (20) represents the waiting time for the AGV to extract the unloading containers under the quay crane, and Constraints (21) and (22) outline the time constraints for the AGV to handle the next transfer or unloading container after delivering the unloading containers. Constraint (23) ensures that the time required for the yard crane to handle the containers leaving the terminal complies with the time-window constraint. Constraint (24) denotes the waiting time for an unmanned container truck to extract containers from the yard crane terminal. Constraint (25) ensures that the time when the unmanned container truck begins handling containers leaving the terminal meets the time-window constraint. Constraint (26) defines the time at which an unmanned container truck exits the terminal gate. Constraint (27) represents the time at which unmanned container trucks unload the containers in a logistics park.

### 3.4. Path-Optimization Model Considering Conflict Strategies

The conflict-resolution strategy adopted in the path-optimization model prioritizes vehicle task sequences, allowing the preceding vehicle to proceed while subsequent vehicles wait. The parameters, variables, and sets of the model are presented in Table 2.

**Table 2.** Parameters, variables, and sets of the path-optimization model considering conflict strategies.

Notation	Sets
$N$	The set of various operation points in the automated container terminal, including quay cranes, buffer racks in the yard, and intersection points in the AGV driving process. Here, $i$ and $j$ are node identifiers within this set.
$V$	The set of AGV quantities in the automated container terminal, $v = 1, 2, \dots, V$ ;
$F$	The set of quay crane unloading nodes and logistics park unloading nodes, $f = 1, 2, \dots, F$ ;
$r_{ij}$	The lines connecting the nodes in the AGV travel path;
$B$	The set of nodes for unmanned container truck operations, including the yard gantry crane operation points, terminal gate, and logistics park;
$U$	The set of unmanned container trucks, $u = 1, 2, \dots, U$ ;
$G$	The set of all horizontal transport vehicles, $G = U \cup V$ , $g = 1, 2, \dots, G$ ;
$T_{ijv}$	The time at which the AGV arrives at node $j$ ;
$d_{ij}$	The distance from node $i$ to node $j$ ;
$v_0$	The average travel speed of the AGV;
$T_{ijv}^{in}, T_{ijv}^{out}$	The time at which the AGV enters or leaves the path;
$T_{ijg}^{in}, T_{ijg}^{out}$	The time at which all horizontal transport vehicles enter or leave the path;
$RL_{ij}$	The number of AGVs that the current path can accommodate;
$H_0$	The distance between the front of the AGV and a node (assuming the AGV maintains a maximum safe distance that is 2 times its body length, $H_0 \geq 2H$ )
$H$	The length of the AGV's body;
$gt_{ijv}$	The total time it takes for a container to travel from the quay crane to the buffer stand;
$rt_{ijv}$	The safe time interval required between two AGVs passing through the same node;
$t_{ij}$	The time it takes to traverse a specific path.

The decision variables for the model are as follows:

- $x_{ijg}$ : A binary variable that equals one if the  $g$ -th horizontal transport vehicle (including the unmanned container truck and AGV) passes from node  $i$  to node  $j$ ; otherwise, it equals zero.
- $x_{ijv}$ : A binary variable that equals one if the  $g$ -th AGV passes from node  $i$  to node  $j$ ; otherwise, it equals zero.
- $x_{iju}$ : A binary variable that equals one if the  $u$ -th unmanned container truck passes from node  $i$  to node  $j$ ; otherwise, it equals zero.

Objective:

$$f_{road} = \min \varepsilon_3 \sum_{g \in G} \sum_{i \in N} \sum_{j \in N} x_{ijg} t_{ij} \quad (28)$$

Equation (28) regards the minimum transportation cost as the objective function, with time being the primary variable for path optimization. Its goal is to minimize the total transportation duration in order to achieve the lowest transportation cost, where  $\varepsilon_3$  is the transportation cost coefficient.

The objective is subject to the following constraints:

$$\sum_{i \in N} \sum_{v \in V} x_{ijv} = 1, \forall j \in N \quad (29)$$

$$\sum_{j \in N} \sum_{v \in V} x_{ijv} = 1, \forall i \in N \quad (30)$$

$$\sum_{v \in V} x_{ijv} = 1; \forall i \in N, j \in N \quad (31)$$

$$\sum_{i \in B} \sum_{u \in U} x_{iju} = 1, \forall j \in B \quad (32)$$

$$\sum_{j \in B} \sum_{u \in U} x_{iju} = 1, \forall i \in B \quad (33)$$

$$\sum_{u \in U} x_{iju} = 1; \forall i \in B, j \in B \quad (34)$$

$$\sum_{i \in N} \sum_{j \in N} x_{iju} \leq 0, \forall u \in U \quad (35)$$

$$\sum_{i \in B} \sum_{j \in B} x_{ijv} \leq 0, \forall v \in V \quad (36)$$

$$\sum_{i \in N \cup B} x_{ijg} = \sum_{i \in N \cup B} x_{jig}, \forall j \in (N \cup B) \cap (C_{RF}), g \in G \quad (37)$$

$$x_{ijg}r_{ij} \neq x_{ijg}r_{ji} \neq 0, \forall i \in (N \cup B), j \in (N \cup B), g \in G \quad (38)$$

$$x_{ijg} \neq x_{jig}, \forall i \in N, j \in N, g \in G \quad (39)$$

$$P_{(m,n)} = P_m \cap P_n \quad (40)$$

$$PAT_{ijv} = \frac{H + H_0}{v_0}, \forall i \in N, j \in N, v \in V \quad (41)$$

$$|INT_{ijv} - INT_{ijv'}| < \min\{PAT_{ijv}, PAT_{ijv'}\}, \forall i \in N, j \in N, v \in V \quad (42)$$

$$PAT_{ijv} < PAT_{ijv'} \quad (43)$$

$$\varepsilon_{ijv} = \text{Ceil}\left(\frac{PAT_{ijv} - PAT_{ijv'}}{\max\{PAT_{ijv}, PAT_{ijv'}\}}\right) \quad (44)$$

$$DT = \frac{0 - v_0}{-a_0} \quad (45)$$

$$|INT_{ijv} - INT_{ijv'}| > DT, \forall i \in N, j \in N, v \in V \quad (46)$$

$$UT = \frac{v_0 - 0}{a_0} \quad (47)$$

$$t_{ijv}^m = (DT + UT - |INT_{ijv} - INT_{ijv'}|) \bullet (1 - \varepsilon_{ijv}) \quad (48)$$

$$R_{(m,n)} = R_{ijv}^m \cap R_{ijv}^n \quad (49)$$

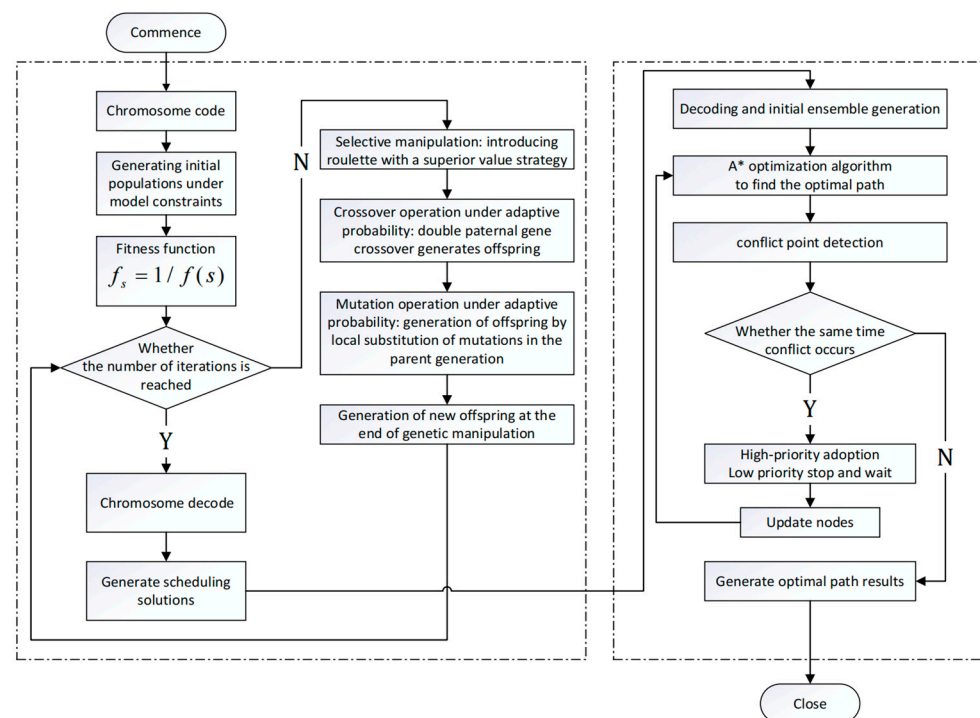
$$RL_{ij} \leq \text{Round}\left(\frac{r_{ij}}{H + H_0}\right), \forall i \in N, j \in N \quad (50)$$

Constraints (29) and (30) stipulate that during the transportation process of containers by the  $v$ -th AGV, only one path from node  $i$  to node  $j$  can be selected. Constraint (31) ensures that each container task is completed by only one AGV. Constraints (32) and (33) express that during the transportation process of containers by the  $b$ -th unmanned container truck, only one path from node  $i$  to node  $j$  can be selected. Constraint (34) ensures that each container task is completed by only one unmanned container truck. Constraints (35) and (36) assert that the operation areas of AGVs and unmanned container trucks do not interfere with each other. Constraint (37) represents the flow balance constraint, ensuring the flow balance of entering and leaving nodes by relaxing the nodes other than quay crane unloading and logistics park unloading. Constraint (38) indicates that the travel routes between nodes are unidirectional guide routes, meaning that the planned routes in the form of AGVs remain unchanged for the current task. Constraint (39) similarly states that the travel routes between nodes are unidirectional guide routes, ensuring the stability of AGV routes planned for the current task. Constraint (40) is used for determining AGV conflict nodes. Constraints (41) and (42) determine AGV path conflicts when the time difference between two AGVs passing through the same node is less than the time taken by

the faster AGV to enter and completely exit the node. Constraints (43) and (44) represent the AGV priority passing strategy. According to the principle of prioritizing the AGV with the shortest time to pass through the node, when a conflict is detected between two AGVs passing through the same node, the instruction for priority passing is provided based on the AGV with the shortest time to pass through the node in the current state. Here,  $\varepsilon_{ijv}$  is the ceiling function. When  $\varepsilon_{ijv}$  is 1, it indicates that the vehicle has higher priority and should pass first; when  $\varepsilon_{ijv}$  is 0, it is the opposite. Constraint (45) represents the time it takes for the following vehicle to decelerate to 0 while parking/waiting. Constraint (46) states that the time required to decelerate to 0 must be less than the time difference between successive vehicles arriving at the node. Constraint (47) represents the time it takes for the following vehicle to accelerate to the original speed. Constraint (48) represents the parking/waiting time for the following vehicle. Constraint (49) determines AGV congestion sections, divided into congestion due to node conflicts and congestion caused by path planning. Constraint (50) ensures that the number of AGVs traveling on the path maintains a safe distance, where Round() is the rounding-down function.

#### 4. Algorithm Design

In this study, an improved genetic algorithm (IGA) was utilized to solve the task allocation model and derive container task assignment results. To address the path optimization model, AGV paths were converted into a node matrix, where matrix coordinates were assigned as path time values. An A\* optimization search algorithm was integrated to generate the optimal path. The comparison of features of algorithms relevant to solving this problem is documented in the Appendix A. The overall algorithmic process is depicted in Figure 2.



**Figure 2.** Flowchart of the staged algorithm.

##### 4.1. Solving the Task-Allocation Model Using Improved Genetic Algorithm

Step 1: Utilize dual-layer real-number encoding to generate double-chromosome matrix sequences. The number of chromosomal genes corresponds to the container tasks to be performed. The first chromosome denotes the AGV numbers for container tasks, while the second chromosome represents the number of unmanned container trucks. Figure 3

illustrates the distribution of ten container tasks among three AGVs and five unmanned container trucks.

Task number	1	2	3	4	5	6	7	8	9	10
AGV Task	2	3	1	2	1	3	2	1	3	2
Unmanned Collector Task	1	5	1	4	2	3	2	1	5	3

**Figure 3.** Chromosome coding diagram.

Step 2: Implement a random-generation method to create the initial population, ensuring diversity within the population while adhering to Constraints (5)–(12). An initial population is generated in appropriate quantities at each gene position within the double chromosome;

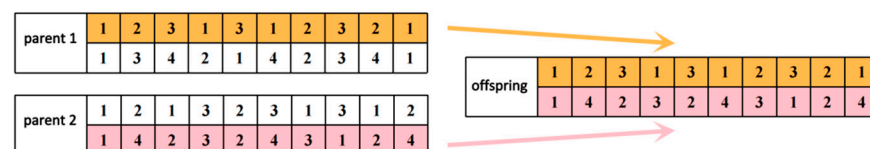
Step 3: Execute the task process based on unloading conditions to minimize the total transportation costs as the objective function. The reciprocal of the objective function is adopted as the fitness function. Excellent chromosomes with high fitness values are selected for reproduction in the offspring generation. The fitness function is represented by Equation (51):

$$f_s = \frac{1}{f(s)} \quad (51)$$

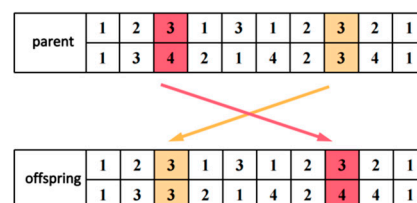
where  $f_s$  is the fitness function and  $f(s)$  is the objective function.

Step 4: Utilize a roulette wheel selection strategy to retain superior individuals, ensuring their exclusion from mutation and crossover operations to prevent loss and maintain optimal overall solutions;

Step 5: Conduct crossover and mutation operations under adaptive probabilities. Exchange the chromosomes of the two parent individuals to generate new offspring under adaptive probability. These offspring inherit the original parent AGV and unmanned truck transport sequences for crossover operations. Mutation operations involve exchanging non-identical gene positions within the same chromosome, yielding new-chromosome offspring in Figure 4.



(a) crossover operation



(b) mutation operation

**Figure 4.** Schematic diagram of the crossover operation.

Step 6: Check whether the algorithm reaches the maximum iteration limit. If the condition is satisfied, the algorithm is halted; otherwise, the computation continues.

#### 4.2. Solving Conflicting Paths Using A\* Optimization Search Algorithm

The A\* search algorithm generates multiple nodes on a two-dimensional plane and connects them to determine the least costly path. It is commonly employed for pathfinding and traversing two-dimensional plane positions. Building upon the performance improvements of Dijkstra's algorithm, the A\* search algorithm offers a more efficient and accurate approach for calculating the shortest routes.

The section employs Equation (52) as the cost function and is represented as  $f(n)$  for the A\* algorithm. Here, the actual cost function  $g(n)$  calculates the basic transportation cost incurred between the current node  $n$  and its child node  $n'$  due to the distance. The heuristic function  $h(n)$  factors in the cost generated by the initial exploration command and ensures that the heuristic result for the current node satisfies  $h(n) \leq c(n, n') + h(n')$ , where  $c(n, n')$  denotes the actual movement cost from node  $n$  to node  $n'$ .

$$f(n) = g(n) + h(n) \quad (52)$$

In this study, the A\* optimization algorithm was employed to resolve conflicting paths, as depicted in Figure 5.

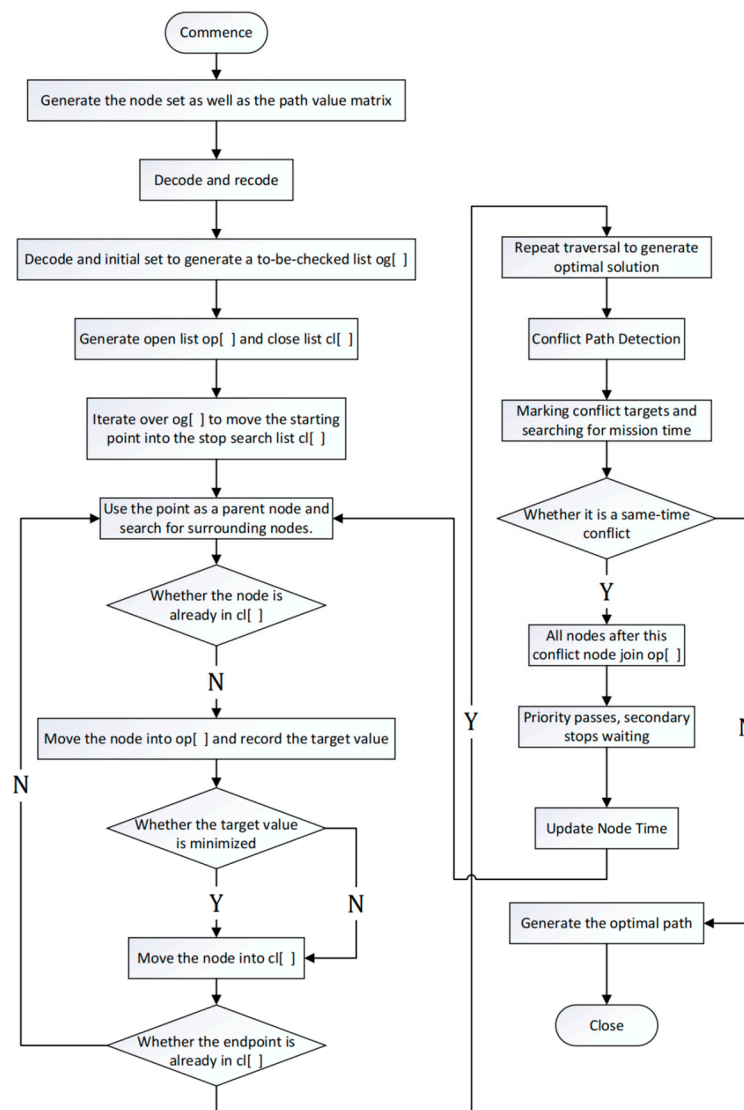
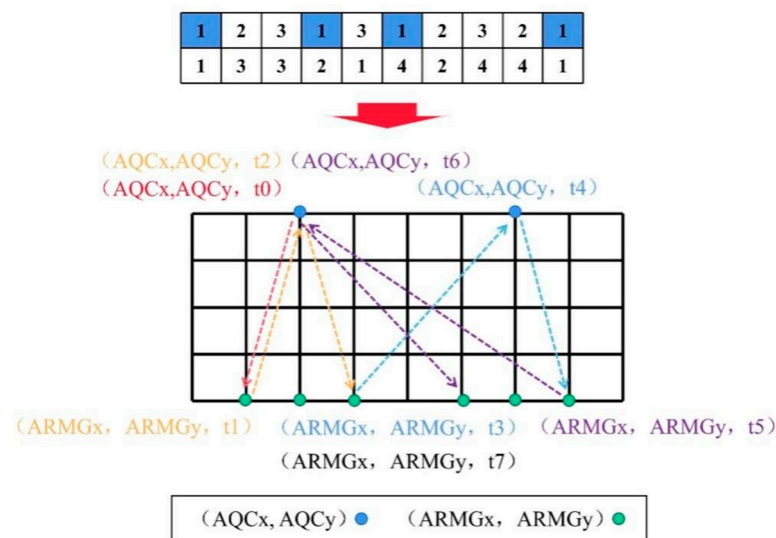


Figure 5. Flowchart of the algorithm to solve path conflicts.

- (1) Establishment of two-dimensional plane node set and path assignment: Generate combinations of position nodes and assign path values based on actual operating distances;
- (2) Decoding and data processing to generate the initial set: Utilize the IGA to obtain the scheduling task sequence for decoding. Generate a scheduling program and reprocess the data with a three-point coordinate matrix, where the three-point coordinates correspond to the two-dimensional plane horizontal coordinates, vertical coordinates, and time parameters, respectively. As depicted in Figure 6, the decoding result of the AGV1 task is to service containers 1-4-6-10, which is transformed into the initial time  $t_0$  when the location is  $(AQCx, AQCy)$ . It is then added to the initial time dimension to generate coordinates  $(AQCx, AQCy, t_0)$  in accordance with the AGV operation path. Based on the IGA scheduling result, a new set of coordinates is generated and stored in the search list  $op[]$ ;



**Figure 6.** Schematic diagram of decoding and initial set generation.

- (3) Path search and iterative optimization: (a) Traverse the search list  $op[]$ , find the starting point value, and move it into the stop search list  $cl[]$ . (b) Move the node to the surroundings, transfer the surrounding nodes into  $op[]$ , record the target value and node state after movement, and ignore the nodes that do not meet the conditions. (c) Repeat “Step b” in a loop search until the endpoint is found and added to  $op[]$ , outputting the path optimization results;
- (4) Conflict path resolution: Determine path conflicts based on the obtained coordinates. A path conflict occurs when judging the starting time of the task to establish priority. High priority is given to pass, while low priority stops and waits for the node time to be updated. Invoke the A\* algorithm from the point of conflict to generate subsequent task routes until no conflicting paths remain, reaching the end to generate the optimal path.

## 5. Case Study Analysis

Various scenarios were chosen to assess the feasibility of employing the improved genetic algorithm in conjunction with the A\* algorithm to tackle the joint-scheduling model of AGVs and unmanned container trucks while considering conflict-resolution strategies. Furthermore, comparative experiments were devised to scrutinize the disparities in scheduling methodologies and path-optimization outcomes between conflict-free path algorithms and those integrating conflict-resolution strategies, as well as to evaluate the efficacy of these strategies. The experiments were conducted on a computer equipped with an Intel(R) Core(TM) i5-10400 CPU @ 2.90 GHz processor and 32 GB of RAM, utilizing example programs within the Python framework PyCharm version 2023.

### 5.1. Experimental Layout and Parameters

For this study, the layout of the automated container terminal at Yangshan Port Phase IV served as a benchmark. During the path-optimization phase, a node simulation layout, depicted in Figure 7, was generated. The operational area spans 300 m in length and 120 m in width at the front operating terminal. The AGV driving range encompasses 120 nodes, encompassing quay crane operation nodes, yard crane operation nodes (representing yard buffer rack positions), and AGV turn nodes. One-way lanes are designated for the loading and unloading areas, whereas the buffer lanes feature two-way lanes facilitating AGV reversal and turning. The driving area includes two-way lanes for AGV maneuvering, encompassing turning, reversing, and turning around. Notably, the driving area for unmanned container trucks excludes complex path optimization and is thus not simulated with nodes.

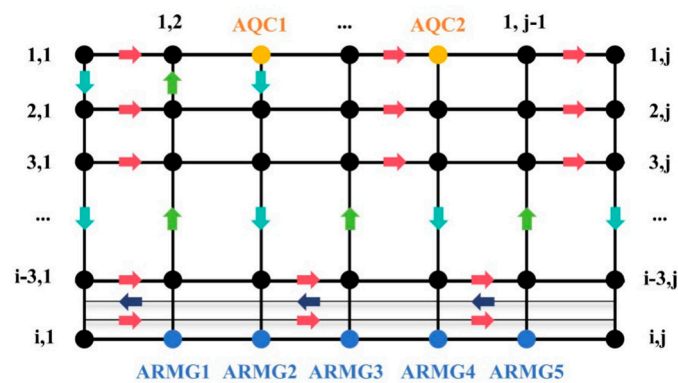


Figure 7. Schematic of the node layout in the AGV operation area.

Other experimental parameters were set, as shown in Table 3.

Table 3. List of other parameters.

Notation	Parameterization	Notation	Parameterization
$L$	10–200	$\alpha_l$	100 s
$Q$	2	$\beta_l$	120 s
$D$	3–6	$\gamma_1$	50 s
$A$	3–12	$t_{kl}$	200 s
$K$	3–12	$P$	2
$Z$	6	$v_0$ (unloaded)	2 m/s
$W$	6	$v_0$ (reloading)	1 m/s

### 5.2. Algorithm Validation and Result Analysis

In this section, 20 different sets of calculations were generated to validate the effectiveness of the improved genetic algorithm (IGA) and A\* optimization search algorithm employed in this study. Additionally, a standard genetic algorithm (GA) was used as a control to compare path-optimization performance. The path-optimization tasks in this example were executed using the A\* algorithm, while the control experiment utilized a GA. The average values from ten program runs were recorded for each algorithm. Table 4 presents the parameter settings for the algorithms.

Table 5 presents the experimental results, where  $f$  represents the value of the objective function calculated by the GA algorithm and  $f^*$  represents the value of the objective function calculated by the IGA + A\* algorithm. The difference degree was computed using the formula. The average computing time of the GA algorithm is 66.55 s, while that of the IGA + A\* algorithm is 64.01 s. The disparity between the two algorithms' computing times is negligible, with the IGA + A\* algorithm being only 0.04% lower than the GA algorithm. Additionally, when comparing the objective function values of the two algorithms, the average difference between the GAP values of the IGA + A\* algorithm

and the GA algorithm is 3.21%. This indicates that the IGA + A\* algorithm exhibits superior optimization capabilities compared to the GA algorithm. Therefore, the results of the IGA + A\* algorithm utilized in this example demonstrate satisfactory performance and robust optimization abilities, thereby verifying the algorithm's effectiveness in solving the model.  $T$  denotes the runtime of the computer algorithm when using the GA algorithm to complete the test case;  $T^*$  denotes the runtime of the computer algorithm when using the IGA + A\* algorithm to complete the test case.

**Table 4.** Algorithm parameter list.

Parameter Object	Parameter Value	Parameter Meaning
Pop_size	100	IGA population size
Iter_Max (small, large)	100, 200	Maximum number of iterations for different sizes of IGAs
$P_c, P_m$	Adaptation values	IGA crossover, mutation probability
$\varepsilon_1$	0.8 CNY/s	Base transportation cost factor
$\varepsilon_2$	0.3 CNY/s	Penalty cost factor

**Table 5.** Comparison of the joint-scheduling optimization algorithms while considering conflicts.

Serial Number	Number of Tasks	Number of AGVs	Number of Unmanned Collector Trucks	Configuration Ratio	GA		IGA + A*		GAP%
					$f$	$T$	$f^*$	$T^*$	
1	10	5	5	1.00	2495	10.64	2357	8.03	5.85%
2	10	7	7	1.00	2464	10.91	2337	8.74	5.43%
3	10	7	5	1.40	2460	10.67	2342	8.94	5.04%
4	10	5	7	0.71	2499	10.73	2357	8.98	6.02%
5	20	7	7	1.00	3879	20.30	3744	17.47	3.61%
6	20	12	12	1.00	3717	21.02	3604	19.86	3.22%
7	20	7	12	0.58	3702	20.46	3599	18.97	2.86%
8	20	12	7	1.71	3789	20.49	3629	18.42	4.41%
9	50	9	9	1.00	9204	47.03	8995	40.41	2.32%
10	50	12	12	1.00	9197	48.26	8976	41.97	2.46%
11	50	9	12	0.75	9291	48.39	9001	41.71	3.22%
12	50	12	9	1.33	9374	49.97	9014	41.83	3.99%
13	100	9	9	1.00	18,049	85.88	17,697	73.49	1.99%
14	100	12	12	1.00	17,991	94.45	17,603	81.51	2.20%
15	100	9	12	0.75	17,997	94.03	17,699	81.25	1.68%
16	100	12	9	1.33	18,006	93.97	17,710	82.54	1.67%
17	200	9	9	1.00	38,321	185.71	37,463	171.41	2.29%
18	200	12	12	1.00	38,002	184.36	37,298	170.57	2.12%
19	200	9	12	0.75	38,079	186.02	37,254	172.79	2.21%
20	200	12	9	1.33	37,902	184.25	37,297	172.44	1.62%

At the same scale of container tasks, an increase in the number of AGVs and unmanned container trucks resulted in a noticeable decrease in the objective function value, for instance, in Experiments 1 and 2, 5 and 6, 9 and 10, and 17 and 18. Similarly, with the same scale of container tasks, as the ratio of AGVs to unmanned container trucks approached 1, the objective function value was lower than that in cases where the ratio was significantly different. For example, in Experiments 10–12, with a task scale of 50, the objective function values obtained with a ratio of 0.75 and 1.33 were higher than those obtained with a ratio of 1. This further demonstrates that the ratio and quantity of AGVs and unmanned container trucks affect the objective function.

### 5.3. Validation of Strategy Effectiveness

To validate the effectiveness of the conflict-resolution strategy employed in this study for joint-scheduling problems, a validation group was established with 10 container tasks,

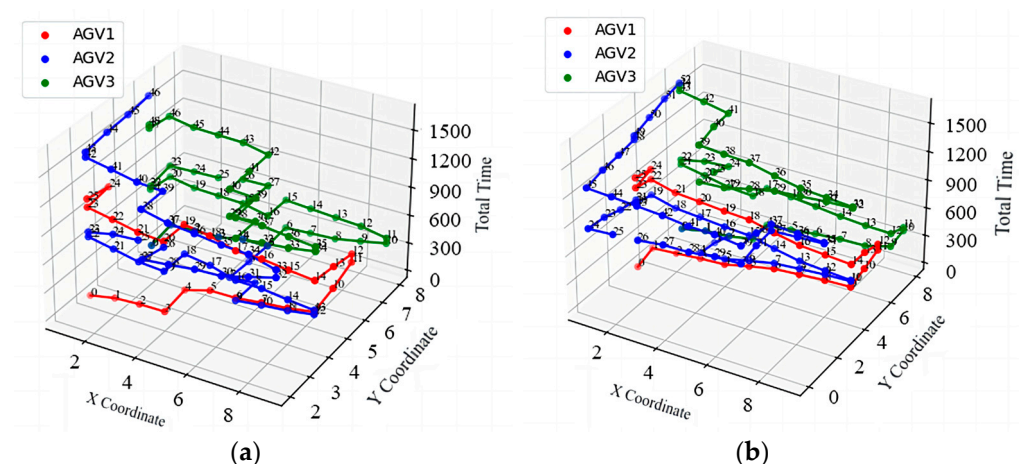
three AGVs, and three unmanned container trucks. The task allocation results for the AGVs are presented in Table 6.

**Table 6.** AGV task allocation results.

AGV No.	AGV Tasking Results	AGV Initial Path List
AGV1	8-3-6	(1,3,103) (9,6,941) (1,3,1019) (9,12,1289) (1,3,1404) (9,6,1723)
AGV2	10-2-1-9-7	(1,6,98) (9,4,478) (1,3,500) (9,2,841) (1,3,974) (9,10,1451) (1,3,1501) (9,2,1876) (1,6,98) (9,8,1876)
AGV3	4-5	(1,6,278) (9,8,699) (1,6,1093) (9,2,1649)

In the initial task list for the AGVs, AGV2 was assigned a relatively large number of tasks, resulting in more complex routes. However, the running time for each task was shorter for AGV2 than AGV1 and AGV3, leading to a longer total transportation time for AGV2. Nonetheless, the time difference between the completion of the final tasks by the three AGVs was relatively small, indicating a minor impact on subsequent tasks. The initial task list was generated by the IGA and primarily consisted of point-to-point paths. The path optimization results obtained using the A\* algorithm for this initial list are presented in Tables 5–7. After implementing the path conflict-resolution strategy for AGVs, the results had different coordinates and time dimensions, indicating the absence of conflicts in the generated AGV paths. The optimized AGV path times were close to the initial objective function values obtained from the first scheduling using the IGA, and in some cases, the total time was even shorter. After path optimization, the total objective function decreased by 1.04%, 3.78%, and 1.51% for AGV1, AGV2, and AGV3, respectively. Figure 8 illustrates a spatiotemporal path diagram depicting the generation of conflicting paths in a specific case and the generation of optimal paths using the conflict-resolution strategy.

Before conflict optimization, nodes 6-7-8-9 of AGV1 conflicted with nodes 9-10-11-12 of AGV2, resulting in the temporal and spatial overlap of paths in Figure 8. Conflict optimization entails replanning routes to ensure conflicting paths do not occur within the same timeframe. In summary, the effectiveness of the IGA and A\* algorithms in resolving conflicts was demonstrated. Adopting this strategy can optimize conflicting paths and improve transportation efficiency.



**Figure 8.** Schematic diagram of conflict path optimization results. (a) Before conflict optimization. (b) Conflict optimization.

To further validate this strategy, we compared data with the same configuration ratio but different scales with the conflict-free paths generated by the optimized particle swarm algorithm. The Optimized Particle Swarm Optimization (OPSO) algorithm adopts various

iterative strategies to ensure the final results are conflict-free when solving optimal path problems. By comparing scenarios where the IGA and A\* algorithms, as used in this study, were employed to solve path scenarios with conflicts and scenarios where OPSP was used to directly generate the conflict-free paths, we validated the algorithm's effectiveness. This comparison helps demonstrate how OPSP contributes to conflict resolution in pathfinding tasks alongside the traditional IGA and A\* approaches.

**Table 7.** AGV path optimization results.

Serial Number	Task Allocation	AGV Initial Path List
AGV1	8-3-6	(1,3,103) (1,4,143) (2,4,176) (3,4,190) (3,5,206) (4,5,238) (5,5,251) (6,5,277) (7,5,303) (7,6,322) (8,6,344) (9,6,362) (9,6,381) (8,6,417) (7,6,437) (7,5,449) (6,5,483) (5,5,499) (4,5,522) (3,5,546) (3,4,557) (2,4,588) (1,4,608) (1,3,640) (1,3,663) (1,4,684) (2,4,716) (3,4,730) (3,5,744) (4,5,768) (5,5,791) (6,5,811) (7,5,833) (7,6,856) (7,7,887) (7,8,901) (8,8,918) (8,9,941) (8,10,971) (9,10,994) (9,11,1013) (9,12,1041) (9,12,1067) (9,11,1089) (9,10,1101) (8,10,1131) (8,9,1153) (8,8,1171) (7,8,1196) (7,7,1210) (7,6,1237) (7,5,1262) (6,5,1278) (5,5,1304) (4,5,1319) (3,5,1347) (3,4,1365) (2,4,1386) (1,4,1404) (1,3,1435) (1,3,1463) (1,4,1486) (2,4,1509) (3,4,1535) (3,5,1547) (4,5,1578) (5,5,1603) (6,5,1622) (7,5,1643) (7,6,1663) (8,6,1689) (9,6,1705)
AGV2	10-2-1-9-7	(1,6,98) (1,5,109) (2,5,124) (2,4,132) (3,4,140) (4,4,154) (5,4,178) (6,4,192) (7,4,213) (8,4,225) (9,4,240) (9,4,253) (8,4,265) (7,4,290) (6,4,309) (5,4,320) (4,4,341) (3,4,355) (2,4,371) (1,4,381) (1,3,403) (1,3,418) (2,3,432) (3,3,451) (4,3,468) (5,3,482) (6,3,498) (6,2,520) (7,2,536) (8,2,554) (9,2,575) (9,2,591) (8,2,610) (7,2,624) (6,2,635) (6,3,658) (5,3,676) (4,3,686) (3,3,708) (2,3,726) (1,3,739) (1,3,761) (1,4,774) (1,5,791) (1,6,801) (2,6,819) (2,7,838) (3,7,852) (4,7,863) (5,7,882) (5,8,902) (6,8,914) (7,8,925) (7,9,944) (8,9,965) (8,10,976) (9,10,992) (9,10,1012) (8,10,1028) (8,9,1046) (7,9,1060) (7,8,1074) (6,8,1091) (5,8,1099) (5,7,1121) (4,7,1137) (3,7,1151) (2,7,1175) (2,6,1190) (1,6,1202) (1,5,1218) (1,4,1235) (1,3,1246) (1,3,1268) (2,3,1280) (3,3,1297) (4,3,1317) (5,3,1331) (6,3,1346) (6,2,1369) (7,2,1384) (8,2,1398) (9,2,1421) (9,2,1436) (8,2,1446) (7,2,1472) (6,2,1488) (6,3,1497) (5,3,1516) (4,3,1531) (4,4,1553) (3,4,1563) (2,4,1582) (2,5,1595) (1,5,1610) (1,6,1623) (1,6,1644) (2,6,1659) (2,7,1674) (3,7,1692) (4,7,1710) (5,7,1728) (5,8,1745) (6,8,1757) (7,8,1769) (8,8,1789) (9,8,1805)
AGV3	4-5	(1,6,278) (2,6,297) (3,6,328) (4,6,429) (5,6,424) (5,7,425) (6,7,530) (7,7,584) (8,7,544) (8,8,636) (9,8,640) (9,8,741) (8,8,737) (8,7,760) (7,7,857) (6,7,896) (5,7,919) (5,6,971) (4,6,979) (3,6,1038) (2,6,1058) (1,6,1065) (1,6,1131) (1,5,1175) (2,5,1160) (2,4,1251) (2,3,1279) (3,3,1346) (4,3,1379) (5,3,1429) (6,3,1406) (7,3,1462) (8,3,1479) (9,3,1579) (9,2,1624)

The formula for calculating the difference between the objective function values of the two strategies is  $GAP = (f - f^*)/f^*$ , where  $f$  and  $f^*$  represent the waiting objective value of the conflict-free path scheduling result and the conflict-resolution strategy scheduling result, respectively.

Further details are presented in Table 8.

**Table 8.** A comparative analysis of strategies.

Serial Number	Number of Tasks	Number of AGVs	Number of Unmanned Collector Trucks	Conflict-Free Path		Conflict-Resolution Strategies		GAP%
				$f$	$Wf$	$f^*$	$Wf^*$	
1	10	5	5	2361	279	2357	249	0.17%
2	10	7	7	2363	284	2337	271	1.11%
5	20	7	7	3740	511	3744	504	−0.11%
6	20	12	12	3593	527	3604	515	−0.31%
9	50	9	9	9032	1120	8995	1098	0.41%
10	50	12	12	9009	1349	8976	1246	0.37%
13	100	9	9	17,708	2677	17,697	2554	0.06%
14	100	12	12	17,689	2703	17,603	2740	0.49%
17	200	9	9	37,157	5662	37,463	5599	−0.82%
18	200	12	12	37,327	5832	37,298	5702	0.08%

Based on the analysis of the experimental results, the average difference in the objective function values between the conflict-free path strategy and the conflict-resolution strategy employed in this study was 0.14%. This indicates that the optimization capabilities of the algorithms for the two strategies are similar. Further analysis of the waiting objective values under both strategies revealed that the waiting objective values under the conflict-resolution strategy joint scheduling were consistently lower than those under conflict-free path joint scheduling. Therefore, for the optimization problem of the joint scheduling of AGVs and unmanned trucks, using conflict-free paths and planning paths with conflict resolution strategies can achieve the expected results. Additionally, under the conflict-resolution strategy, the waiting cost for AGVs at the terminal is lower, whereas the transportation cost is higher.

In this study, when conflicts arose in the paths, a conflict-resolution strategy was employed to address the issue. The proposed conflict-resolution strategy model integrates conflict path detection and imposes constraints on AGVs at specific nodes during conflicts, following a strategy of allowing high-priority vehicles to proceed while low-priority ones wait. An enhanced genetic algorithm was developed to generate scheduling solutions. These scheduling results were then decoded to create initial solution sets for the A\* optimization search algorithm to begin the path search iterations. Various scenarios were designed to compare the performance of the algorithms and the effectiveness of the strategies. Based on the experimental findings, the viability of solving the joint scheduling problem of AGVs and unmanned trucks at automated container terminals using the conflict-resolution strategies proposed in this study was confirmed. Moreover, the efficacy of the improved genetic algorithm and A\* optimization search algorithm in solving the model was demonstrated.

## 6. Conclusions

In this study, we conducted a systematic investigation into the operational intricacies of the Shanghai Yangshan Port Phase IV automated container terminal, with the spotlight on the integrated scheduling of AGVs and UCTs. Our research delineated the critical impact of this joint-scheduling approach on augmenting the terminal's operational efficiency. The manuscript unfolds in a coherent manner, beginning with an extensive feasibility analysis. This analysis was meticulously executed through a review and synthesis of the literature, empirical field investigations, and comparative studies. The comparative analysis of the task completion times under various operational modalities—fixed service object mode, independent scheduling, and joint scheduling—provided a robust evidence base supporting the efficacy of the joint-scheduling paradigm. The theoretical constructs developed within this study offer a comprehensive framework that elucidates the structural hierarchies, cost dynamics, and overarching scheduling architecture. These theoretical underpinnings not only validate the feasibility of the joint scheduling approach but also provide a solid foundation for the mathematical model employed. Addressing the path-conflict challenges inherent in joint scheduling, this study introduces a dual-layer joint-scheduling optimization model integrating strategic conflict-resolution mechanisms. Applying an enhanced genetic algorithm for task allocation and an A\* optimization algorithm for pathfinding exemplifies an innovative approach to addressing complex operational challenges. This methodology ensures the identification of optimal paths, the detection of conflict paths, and the strategic updating of node information through conflict-resolution strategies.

The comparative analysis of the proposed algorithms across varying scales of case studies substantiates the effectiveness of the methodologies employed. It confirms the feasibility of utilizing the enhanced genetic algorithm and A\* optimization algorithm to tackle joint-scheduling challenges, especially when considering conflict strategies. Moreover, this study verifies the efficacy of the conflict-resolution strategy through a rigorous examination of the algorithm's path-generation outcomes. The findings of this study not only enhance our understanding of operational efficiencies but also offer a blueprint for future studies seeking to optimize automated terminal operations. Future research could explore inte-

grating real-time data and dynamic environmental factors into the current model, further refining the scheduling paradigm and its applicability in real-world scenarios.

**Author Contributions:** Conceptualization, L.C., Z.G. and Q.Y.; methodology, L.C., Z.G.; software, S.D. and J.Z.; validation S.D. and J.Z.; formal analysis, S.D. and J.Z.; investigation, L.C., Z.G.; resources, S.D. and J.Z.; data curation, S.D. and J.Z.; writing—original draft preparation, L.C., Z.G.; writing—review and editing, Q.Y.; visualization, S.D. and J.Z.; supervision, Q.Y.; project administration, Q.Y.; funding acquisition, Q.Y. 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 under grant 52201412, National Social Science Fund Key Project of China [grant number 22AZD108]; National Social Science Fund of China [grant number 23&ZD138]; Natural Science Foundation of Fujian Province under grant No. 2022J05067, Fujian Provincial New Think Tank Major Project [grant number 24MZKA20]; National Key Research and Development Program Project of China [grant number 2017YFC0805309]; Ocean and Fisheries Development Special Fund of Xiamen City, Technology Innovation Project for Young Scientists (No. 23ZHZB055QCB39).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data, models, or code generated or used during the study are available from the corresponding author by request.

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

Abbreviations

AGV	Automated Guided Vehicles
UCTs	Unmanned Container Trucks
AQC	Automated Quay Cranes
ARMG	Automated Rail-Mounted Gantry Cranes
GA	Genetic Algorithm
IGA	Improved Genetic Algorithm
ALVs	Automatic Lifting Vehicles
RHP	Rolling Horizon Process algorithm
VDTWs	Vehicle Dispatching with Time Windows
CRP-BGA	Congestion Prevention strategy Bi-level Genetic Algorithm
NSGA-II	Nondominated Sorting Genetic Algorithm-II
OPSO	Optimized Particle Swarm Optimization

Appendix A

	Advantage	Disadvantage
GA	Capable of handling non-linear, non-differentiable objective functions with less sensitivity to the initial solution.	Genetic algorithms typically require more computational resources and time to converge compared to single-point search algorithms.
IGA	By optimizing selection, crossover, and mutation strategies, it is possible to accelerate the convergence process.	May not always find the globally optimal solution.
A-star	Widely used and well-understood, with extensive research and optimization techniques available.	Requires a good heuristic function to perform well; finding such a heuristic can be challenging.
Optimized A-star	Mitigates memory usage concerns by limiting the size of the explored set or dividing the search space into manageable parts.	Specific variants may require additional computational overhead for managing state spaces or integrating with domain-specific constraints.

	Advantage	Disadvantage
Other	Greedy Best-First Search can be effective when a simple heuristic is available, and the graph structure allows for quick expansion of promising paths; Dijkstra's Algorithm guarantees the shortest path in non-negative weighted graphs.	Both may not scale well to very large graphs or complex search spaces without additional optimizations.
Heuristic Algorithms	Heuristics are generally faster than exact solvers because they sacrifice optimality for efficiency, and often easier to implement, and require less computational resources.	It can be challenging to analyze the performance and theoretical guarantees of heuristic algorithms due to their non-deterministic nature.
Solvers	Solvers guarantee finding the optimal solution (if one exists), ensuring the best possible outcome.	Implementing and fine-tuning exact methods can require a deep understanding of algorithmic complexities and optimization techniques. They may struggle with certain types of constraints or problem structures that are not well-suited to the problem-solving approach.

## References

- Luo, J.; Wu, Y. Modelling of Dual-Cycle Strategy for Container Storage and Vehicle Scheduling Problems at Automated Container Terminals. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *79*, 49–64. [\[CrossRef\]](#)
- Zaghoud, R.; Mesghouni, K.; Dutilleul, S.C.; Zidi, K.; Ghedira, K. A Hybrid Method for Assigning Containers to AGVs in Container Terminal. *IFAC Pap.* **2016**, *49*, 96–103. [\[CrossRef\]](#)
- Tao, J.; Qiu, Y. A Simulation Optimization Method for Vehicles Dispatching among Multiple Container Terminals. *Expert Syst. Appl.* **2015**, *42*, 3742–3750. [\[CrossRef\]](#)
- Hu, X.; Guo, J.; Zhang, Y. Optimal Strategies for the Yard Truck Scheduling in Container Terminal with the Consideration of Container Clusters. *Comput. Ind. Eng.* **2019**, *137*, 106083. [\[CrossRef\]](#)
- Hu, H.; Chen, X.; Wang, T.; Zhang, Y. A Three-Stage Decomposition Method for the Joint Vehicle Dispatching and Storage Allocation Problem in Automated Container Terminals. *Comput. Ind. Eng.* **2019**, *129*, 90–101. [\[CrossRef\]](#)
- Yue, L.; Fan, H. Dynamic Scheduling and Path Planning of Automated Guided Vehicles in Automatic Container Terminal. *IEEE-CAA J. Autom. Sin.* **2022**, *9*, 2005–2019. [\[CrossRef\]](#)
- Su, P. AGV Scheduling and Path Planning Considering Opportunistic Charging Process. *Int. Core J. Eng.* **2022**, *8*, 523–538.
- Chen, G.; Govindan, K.; Golias, M.M. Reducing Truck Emissions at Container Terminals in a Low Carbon Economy: Proposal of a Queueing-Based Bi-Objective Model for Optimizing Truck Arrival Pattern. *Transp. Res. Part E Logist. Transp. Rev.* **2013**, *55*, 3–22. [\[CrossRef\]](#)
- Lu, Y.; Le, M. The Integrated Optimization of Container Terminal Scheduling with Uncertain Factors. *Comput. Ind. Eng.* **2014**, *75*, 209–216. [\[CrossRef\]](#)
- Yang, Y.; Zhong, M.; Dessouky, Y.; Postolache, O. An Integrated Scheduling Method for AGV Routing in Automated Container Terminals. *Comput. Ind. Eng.* **2018**, *126*, 482–493. [\[CrossRef\]](#)
- Homayouni, S.M.; Tang, S.H.; Motlagh, O. A Genetic Algorithm for Optimization of Integrated Scheduling of Cranes, Vehicles, and Storage Platforms at Automated Container Terminals. *J. Comput. Appl. Math.* **2014**, *270*, 545–556. [\[CrossRef\]](#)
- Kizilay, D.; Van Hentenryck, P.; Eliyi, D.T. Constraint Programming Models for Integrated Container Terminal Operations. *Eur. J. Oper. Res.* **2020**, *286*, 945–962. [\[CrossRef\]](#)
- Zhou, C.; Lee, B.K.; Li, H. Integrated Optimization on Yard Crane Scheduling and Vehicle Positioning at Container Yards. *Transp. Res. Part E Logist. Transp. Rev.* **2020**, *138*, 101966. [\[CrossRef\]](#)
- Yue, L.; Fan, H.; Ma, M. Optimizing Con Fi Guration and Scheduling of Double 40 Ft Dual-Trolley Quay Cranes and AGVs for Improving Container Terminal Services. *J. Clean Prod.* **2021**, *292*, 126019. [\[CrossRef\]](#)
- Dang, C.H.; Nguyen, V.H.; Truong, T.M.A. Adaptive Particle Swarm Optimization for Integrated Quay Crane and Yard Truck Scheduling Problem. *Comput. Ind. Eng.* **2021**, *153*, 107075.
- Xu, B.; Jie, D.; Li, J.; Yang, Y.; Wen, F.; Song, H. Integrated Scheduling Optimization of U-Shaped Automated Container Terminal under Loading and Unloading Mode. *Comput. Ind. Eng.* **2021**, *162*, 107695. [\[CrossRef\]](#)
- Ji, S.; Luan, D.; Chen, Z.; Guo, D. Integrated Scheduling in Automated Container Terminals Considering AGV Conflict-Free Routing. *Transp. Lett.* **2021**, *13*, 501–513.
- Zhong, M.; Yang, Y.; Dessouky, Y.; Postolache, O. Multi-AGV Scheduling for Conflict-Free Path Planning in Automated Container Terminals. *Comput. Ind. Eng.* **2020**, *142*, 106371. [\[CrossRef\]](#)
- Cui, H.; Chen, S.; Chen, R.; Meng, Q. A Two-Stage Hybrid Heuristic Solution for the Container Drayage Problem with Trailer Reposition. *Eur. J. Oper. Res.* **2022**, *299*, 468–482. [\[CrossRef\]](#)

20. Tang, L.; Zhao, J.; Liu, J. Modeling and Solution of the Joint Quay Crane and Truck Scheduling Problem. *Eur. J. Oper. Res.* **2014**, *236*, 978–990. [[CrossRef](#)]
21. Chen, G.; Govindan, K.; Yang, Z. Managing Truck Arrivals with Time Windows to Alleviate Gate Congestion at Container Terminals. *Int. J. Prod. Econ.* **2013**, *141*, 179–188. [[CrossRef](#)]
22. Nossack, J.; Pesch, E. A Truck Scheduling Problem Arising in Intermodal Container Transportation. *Eur. J. Oper. Res.* **2013**, *230*, 666–680. [[CrossRef](#)]
23. Chen, X.; Zhou, X.; List, G.F. Using Time-Varying Tolls to Optimize Truck Arrivals at Ports. *Transp. Res. Part E Logist. Transp. Rev.* **2011**, *47*, 965–982. [[CrossRef](#)]
24. Azab, A.; Karam, A.; Eltawil, A. A simulation-based optimization approach for external trucks appointment scheduling in container terminals. *Int. J. Model. Simul.* **2020**, *40*, 321–338. [[CrossRef](#)]
25. Rijal, A.; Bijvank, M.; de Koster, R. Integrated Scheduling and Assignment of Trucks at Unit-Load Cross-Dock Terminals with Mixed Service Mode Dock Doors. *Eur. J. Oper. Res.* **2019**, *278*, 752–771. [[CrossRef](#)]
26. Heidari, F.; Zegordi, S.H.; Tavakkoli-Moghaddam, R. Modeling Truck Scheduling Problem at a Cross-Dock Facility through a Bi-Objective Bi-Level Optimization Approach. *J. Intell. Manuf.* **2018**, *29*, 1155–1170. [[CrossRef](#)]
27. Xi, X.; Liu, C.; Yuan, W.; Hay, L.L. Two-Stage Conflict Robust Optimization Models for Cross-Dock Truck Scheduling Problem under Uncertainty. *Transp. Res. Part E Logist. Transp. Rev.* **2020**, *144*, 102123. [[CrossRef](#)]

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