Computers & Industrial Engineering 209 (2025) 111427

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Integrating multi-equipment scheduling with accurate AGV path planning for U-shaped automated container terminals

Yongsheng Yang [a,*], Shu Sun [a], Yuzhen Wu [a], Junkai Feng [a], Wenying Lu [a], Ling Wu [a], Octavian Postolache [b]

[a] Institute of Logistics Science & Engineering, Shanghai Maritime University, Shanghai, 201306, China
[b] ISCTE-Instituto Universitario de Lisboa and the Instituto de Telecomunicações, Lisbon, 1049-001, Portugal

ARTICLE INFO

Keywords:
Intelligent port logistics
U-shaped automated container terminals
AGV conflict types
Accurate path planning algorithm
Multi-equipment collaborative scheduling
Computational intelligence techniques

ABSTRACT

In the context of the continuous growth of global container transport demand, people are paying more and more attention to the operational efficiency and energy consumption of automated container terminals (ACTs). This study focuses on the scheduling and path planning of Automated Guided Vehicles (AGVs) in complex environments. It aims to address the challenges that arise from direct interaction of multiple equipment in U-shaped ACTs. In this paper, a multi-equipment cooperative scheduling method based on AGV accurate path planning is proposed for the first time, aiming to minimize the total energy consumption of all equipment. Specifically, we establish a two-layer mathematical model for multi-equipment collaborative scheduling and AGV path planning, considering turning and lane-changing. Then, an Adaptive Genetic Algorithm based on the Jaya strategy and an Accurate Path Planning Algorithm are designed to solve this model. Numerical experiments show that the proposed method can significantly improve the calculation speed and reduce the number of path nodes passed by the AGV. This study provides strong support for terminal managers' equipment scheduling strategy and energy consumption optimization strategy.

1. Introduction

The increasingly serious issue of global warming has put forward higher requirements for the energy consumption of port operations. The deployment of automated container terminals (ACTs) plays a key role in reducing operational efficiency, the labor cost of terminals, and the energy consumption of terminals (Tang et al., 2024). ACTs with traditional vertical layout and end interaction, represented by Hamburg's Altenwerder terminal and Shanghai's Yangshan terminal, are playing an increasingly important role in global trade. However, in these layouts, the moving distance of the of yard cranes is too long, and the driving area of the AGV is limited, and the optimization of its efficiency and energy consumption gradually reaches the bottleneck (Kong & Ji, 2024).

To address these issues, the innovative U-shaped ACT layout has been proposed and applied in practice (Li et al., 2021). In a U-shaped ACT, AGVs can travel and stay inside the yard, reducing the moving distance of the double-cantilever rail cranes (DCRCs) and expanding the driving range of AGVs. It theoretically improving the efficiency of wharf operations and reducing the energy consumption of equipment operation. However, AGVs travel into internal yard roads significantly increases their travel distance. In addition, there are both one-way and two-way lanes in U-shaped ACTs, which greatly increases the AGVs queuing, cross-lane turning and lane changing. This situation adds to the complexity of AGV path planning. Additionally, the AGV is directly coupled with the DCRC in U-shaped ACTs, once the AGV path planning is not accurate, it will directly affect the decision and execution of subsequent tasks. Therefore, the AGV path planning method that only considers the linear path can no longer meet the scheduling requirements of U-shaped ACTs. We must develop a more accurate AGV path planning approach to optimize the multi-equipment collaborative scheduling scheme of U-shaped ACTs.

This problem is not only a key academic problem in management science, but also a practical problem that needs to be solved urgently in the unique layout and process of U-shaped ACTs. This research aims to solve the problem of precise AGV path planning and multi-device cooperative scheduling optimization of quay cranes (QCs), AGVs, and DCRCs in U-shaped ACTs. The main contributions are as follows:

(1) Analyzing the conflict situations that AGVs may encounter during horizontal transportation in U-shaped ACTs for the first time. Summarizing and categorizing these conflicts into three main

* Corresponding author.
E-mail address: yangys@shmtu.edu.cn (Y. Yang).

https://doi.org/10.1016/j.cie.2025.111427
Received 14 December 2024; Received in revised form 17 June 2025; Accepted 25 July 2025
Available online 7 August 2025

types: node occupancy conflicts, face-to-face conflicts, and intersection conflicts, covering eight specific situations. And put forward the corresponding solution measures.

(2) Considering the actual operational states of AGVs, such as straight movement, turning, and lane changing, this study introduces an accurate path planning method that combines global and local planning while equipment scheduling. This approach enhances the accuracy and feasibility of path planning results.

(3) Establishing a two-layer model for multi-equipment collaborative scheduling in U-shaped ACTs, focusing on accurate AGV path planning. This model comprehensively considers the interactions between AGV path planning and equipment scheduling. An Adaptive Genetic Algorithm based on the Jaya strategy (JAGA) and an Accurate Path Planning Algorithm (APPA) are designed to enhance the efficiency and reliability of scheduling decisions.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 details the collaborative operation modes of AGVs, conflict type analysis, and the accurate path planning method, along with the proposed two-layer collaborative scheduling model. Section 4 designs applicable algorithms for the model. Section 5 validates the model and algorithm effectiveness through numerical experiments. Section 6 summarizes the research findings and outlines future research directions.

2. Related work

The efficient operation of ACTs relies heavily on the collaborative scheduling of multiple equipment and AGV path planning. In recent years, research by scholars can be summarized into three main areas: multi-equipment collaborative scheduling, AGV conflict-free path planning, and multi-equipment collaborative scheduling that considers AGV path planning.

2.1. Multi-equipment collaborative scheduling problem

As a key device linking marine and land operations, the scheduling optimization of AGVs is crucial for enhancing the overall efficiency and energy consumption of ACTs. Zhong et al. (2020c) established a mixed integer programming model for QCs, AGVs, and yard cranes (YCs), to minimize the loading and unloading times of vessels. They designed adaptive hybrid genetic algorithm and particle swarm optimization algorithm to improve the solving efficiency. Li et al. (2020) focused on resource allocation and integrated scheduling problems of QCs, AGVs, YCs, and external trucks, proposed an improved genetic algorithm based on shared strategy with the goal of minimizing the completion time of all tasks. Lu (2022) developed a comprehensive scheduling model of QCs, AGVs, yard trucks, and automatic stacking cranes based on proportional equity priority, which improved the handling efficiency of both standard and specialized containers.

For U-shaped ACTs, the unique layout and processes make equipment interactions more complex. Li et al. (2021) proposed a mixed scheduling model for DCRCs, AGVs, and external trucks, aiming to minimize makespan. They employed a chaotic particle swarm optimization algorithm with speed control to tackle the scheduling complexity arising from tight equipment interactions. Yang et al. (2023) also investigated the complex scheduling coupling problem in U-shaped ACTs, designing an adaptive co-evolutionary genetic algorithm to enhance scheduling precision. Han et al. (2024) developed a three-objective optimization model to schedule YCs and ETs simultaneously, considering their efficiencies. They proposed a key-time-based method to synchronize the execution of interdependent tasks between equipment, ensuring that the timing of their critical operations is aligned during objective evaluation in each iteration of the NSGA-III. Aiming to minimize container completion time and AGV waiting time simultaneously, Zhang et al. (2024) proposed a mixed integer linear programming model to optimize the scheduling of multiple equipment. They customized a hybrid genetic-cuckoo optimization algorithm with double-point crossover and cuckoo-flight search strategies, which significantly improved efficiency. To maximize the balanced workload of DCRCs and minimize the invalid operation cost of the yard, Wang and Jin (2024) purposed a bi-objective programming model. The model focuses on the collaborative optimization of multi-equipment scheduling and intersection point allocation considering both direct and indirect transshipment modes. Li et al. (2024) proposed a double-cycle strategy for container handling to coordinate the scheduling of QCs, AGVs and YCs and established a mixed integer linear programming model for the maximum completion time of multiple equipment.

However, most of the aforementioned studies overlook the potential conflicts and congestion issues that may arise during AGV path planning They fail to fully consider the impact of AGV path planning on scheduling decisions.

2.2. AGV conflict-free path planning problem

With the increasing number of AGVs in ACTs, the probability of conflicts during operation also rises. To address this issue, many scholars have focused on conflict-free path planning for AGVs. Lu et al. (2021) established a conflict-free path planning model for AGVs, integrating the first-come, first-serve strategy, and proposed a genetic algorithm based on time window to reduce conflicts among AGVs. Chen (2022) designed a collaborative evolutionary genetic algorithm based on dynamic path planning strategy, and introduced road resistance coefficient as a fitness function, successfully alleviating congestion and conflicts in horizontal transport. Zhong et al. (2020) adopted a priority-based speed control strategy combined with Dijkstra's depth-first search algorithm to select the optimal paths through congestion rate testing and conflict timing, thus improving the efficiency of path planning.

Additionally, Hu et al. (2023) used the multi-agent deep deterministic policy gradient method, considering two conflict types of path planning, and improved the intelligence of the algorithm. Yue and Fan (2022) proposed a novel graph-based conflict avoidance strategy, combining Dijkstra and Q-learning algorithms to provide optimal scheduling and path solutions for AGVs. Sun et al. (2022) proposed that conflict-free path planning for AGVs in ACTs requires more factors to be considered, so the modeling and computational process is more complex, and algorithms combined with rules are often used to solve the problem. Liang et al. (2022) address the AGV locking problem by proposing a three-stage integrated scheduling algorithm for AGV route planning. They establish a road network model in the front area of the container port to optimize AGV paths through joint optimization with QCs and yard blocks. Additionally, they propose a speed control strategy to avoid AGV collisions. Lou et al. (2023) presented a digital-twin-driven AGV scheduling and routing framework, aiming to deal with uncertainties in ACT. They proposed an improved artificial fish swarm algorithm Dijkstra for optimal AGV scheduling and routing solutions. Cao et al. (2023) aimed at the AGV dispatching and bidirectional conflict-free routing problem. A bi-level mixed integer programming model is proposed, which fully considers equipment coordination, bidirectional conflict-free routing, and import and export container tasks.

Nevertheless, these studies mainly focus on AGVs traveling along straight paths between nodes, ignoring the influence of curved paths due to turns and lane changes during actual operations, which leads to deviations in path planning results from reality.

2.3. Multi-equipment collaborative scheduling problem considering AGV conflict-free path planning

AGV path planning directly affects the efficiency of horizontal transport in ACTs, subsequently influencing the operations of QCs and

YCs. Therefore, it is significant to simultaneously consider AGV path planning and multi-equipment collaborative scheduling.

Yang et al. (2018), Zhong et al. (2020a) established two-layer planning models for the collaborative scheduling of QCs, AGVs, and YCs with conflict-free paths, aiming to minimize makespan and AGV delay. They developed genetic algorithms based on anti-congestion rules and hybrid genetic algorithms-particle swarm optimization algorithms to enhance solution efficacy. However, due to simplifications of traffic networks, these studies failed to fully consider AGV conflicts under complex road conditions. Guo et al. (2020) proposed a multi-agent system for multi-AGV architecture, combining an improved Dijkstra algorithm and AGV acceleration control methods to minimize AGV blocking rates, but their research was limited to cross conflicts and did not encompass all potential conflict types.

For U-shaped ACTs, Xu et al. (2021, 2022) investigated the collaborative scheduling issues of DCRCs, AGVs, and QCs under various vessel loading and unloading modes, aiming to minimize task processing times. However, their studies primarily focused on cross conflicts without delving into other conflict types AGVs might encounter under bi-directional traffic. Niu et al. (2022) optimized the collaborative scheduling of QCs, DCRCs, AGVs, and external trucks to minimize road conflicts and total energy consumption (TEC), but only considered the case of unidirectional AGV travel. Li et al. (2023) explored equipment scheduling between U-shaped ACTs and railway yards, decomposed the cluster scheduling problem into specific subsets,aiming to minimize container transportation time through conflict-free path planning for IGVs. Liu et al. (2023) optimized YC scheduling, AGV task allocation, and path planning for U-shaped ACTs, but also only considering unidirectional AGV travel.

In summary, existing research simultaneously considering AGV path planning and multi-equipment collaborative scheduling, but still has the following shortcomings:

(1) Most studies assume AGVs travel in a straight line between nodes, neglecting curved paths due to turns and lane changes under actual vehicle states. This lack of accurate AGV path planning methods does not align with the actual operating conditions of terminal AGVs.
(2) In the complex traffic network of U-shaped ACTs, AGVs may encounter a wider variety of conflict situations. The types of conflicts AGVs are only partially considered and have not been thoroughly analyzed or considered.
(3) Most studies primarily focus on operational efficiency, lacking research on equipment energy consumption, particularly the energy consumption of AGVs in U-shaped ACTs.

This paper aims to address the shortcomings of existing research by providing new theory and method support for multi-equipment collaborative scheduling and accurate AGV path planning in U-shaped ACTs. It also offers practical guidance for optimizing energy consumption in these terminals.

3. Problem description and model establishment

In this section, we first introduce the operating mode of U-shaped ACTs, then analyze the potential conflict types that AGVs may encounter in path planning. Following this, we propose an improved AGV accurate path planning method, and finally establish a two-layer optimization model to achieve overall scheduling optimization.

3.1. Problem description

In U-shaped ACTs, QCs, AGVs, and DCRCs work collaboratively to complete container loading and unloading tasks. AGVs are responsible for horizontal transportation connecting QCs and DCRCs. Their operational area included not only the terminal's horizontal transport zone but also extends into the yard, where they interact with DCRCs at the yard's side, as shown in Fig. 1.

AGVs need to perform conflict-free path planning based on the positions of QCs and DCRCs to reach the loading and unloading points for interaction in a timely manner, thereby avoiding congestion. The results of AGV path planning influence the operational sequence of equipment, while the scheduling scheme of the equipment, in turn, affects the specifics of AGV path planning, as shown in Fig. 2.

3.2. Conflict analysis in AGV path planning

Since AGVs operate in complex traffic networks, they may encounter a variety of conflict situations. Assuming AGVs travel at a constant speed without the risk of rear-end collisions, AGVs in U-shaped ACTs may experience the following eight types of conflicts based on their travel direction, position, and operational status:

(1) Two AGVs simultaneously turning into a bidirectional lane that can only accommodate one AGV, leading to a conflict due to opposite directions.
(2) A conflict occurs when another AGV is stationed in the lane position ahead of the AGV's forward direction.
(3) A conflict occurs when another AGV stops in the lane position that the AGV is about to change into.
(4) A conflict arises when two AGVs turn into different lanes at a crossing position.
(5) A conflict occurs when two AGVs change lanes from their respective lanes into different lanes at a crossing position.
(6) A conflict arises when two AGVs move straight in their respective lanes but turn or change lanes into the same lane.
(7) A conflict occurs when one AGV moves straight while another enters the lane straight ahead through turning or changing lanes.
(8) A conflict arises when one AGV moves straight while another turns at a crossing position to enter a different lane.

These eight conflict situations can be categorized into three types: face-to-face conflicts, node occupancy conflicts, and intersection conflicts. Face-to-face conflicts occur when two AGVs travel towards each other in the same lane, resulting in overlapping paths at the next position, as shown in conflict situation ① in Fig. 3. Node occupancy conflicts arise when one AGV approaches another AGV that is stationary, with the distance between the two AGVs being less than the safe distance, illustrated in conflict situations ② and ③ in Fig. 3. Intersection conflicts occur when two AGVs from different directions pass through the same intersection node simultaneously, potentially leading to collisions, as depicted in conflict situations ④ to ⑧ in Fig. 3.

To resolve these conflicts, conflict detection and avoidance strategies must be introduced into the path planning process. By analyzing the AGV operating environment, potential conflict points can be pre-identified and avoided during path planning.

3.3. Improved AGV accurate path planning method

Based on the operational processes and layout of U-shaped ACTs, we have constructed a node network for AGVs, as shown in Fig. 4. This network considers various vehicle states including straight travel, turning, and lane changing. In this network, QC 1, QC 2, QC 3, and QC 4 represent the loading/unloading points below the QCs. Nodes 131 and 141 are the entry points for AGVs into the yard. Nodes 132 and 142 are the exit points from the yard; and the nodes adjacent to the yard are the loading/unloading points beneath the DCRCs. Solid black lines indicate straight paths, while dashed black lines represent paths where turning and lane changing are possible. AGVs complete transportation tasks based on the planned paths. Next, we introduce a curve interpolation algorithm for local path planning, which derives curved paths for turns

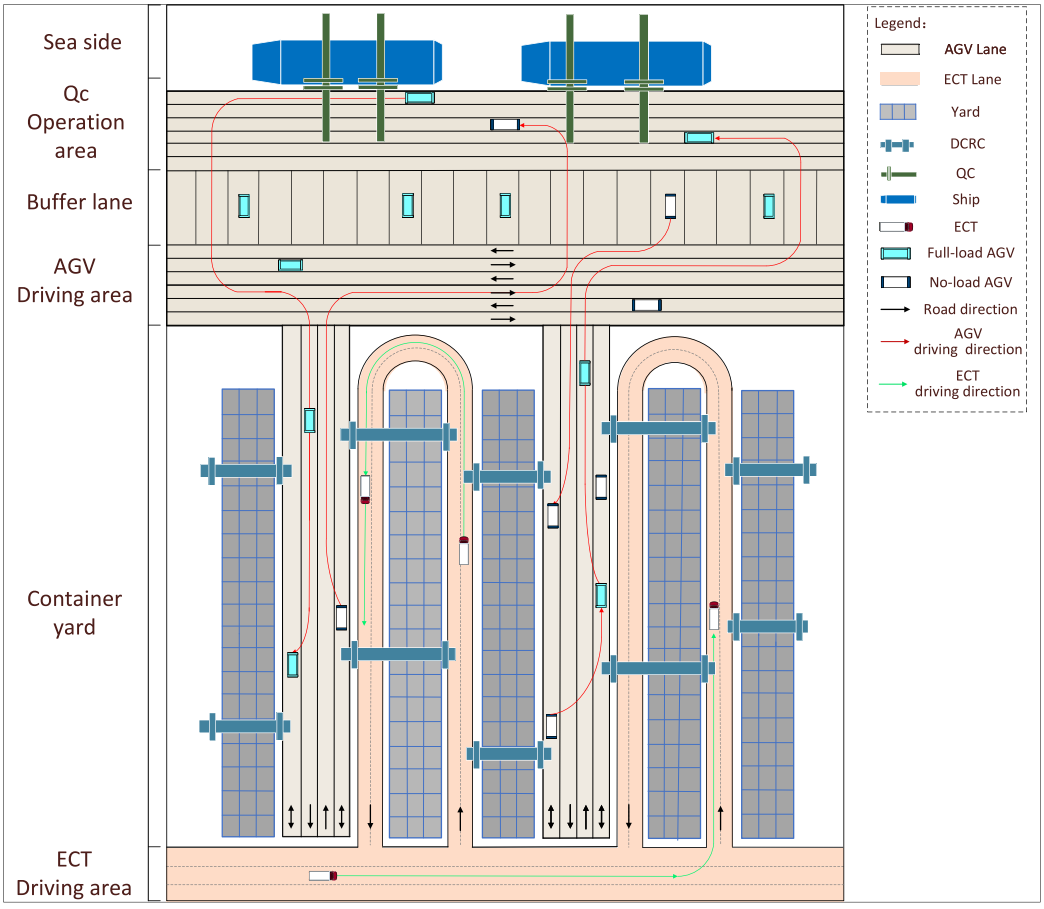

Fig. 1. U-shaped ACTs layout and AGV transportation process.

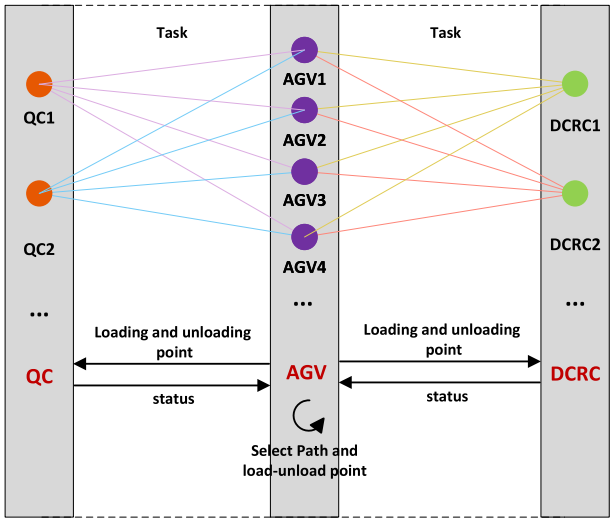

Fig. 2. Relationship between QC, AGV and DCRC in U-shaped ACTs.

or lane changes between nodes, ensuring the feasibility of accurate AGV path planning. By inputting the AGV's desired state, combined with the starting and ending position information for trajectory planning, we use a fifth-degree polynomial for curve fitting, as Eq. (1). Here, $x(t)$ and $y(t)$ represent the variations along the x-axis and y-axis, a and b are the relevant coefficients, and t denotes time. Finally, we store the straight or curved distances between nodes in a distance matrix for use by the algorithm.

$$\begin{cases} x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \end{cases} \tag{1}$$

3.4. Model establishment

To achieve optimization of multi-equipment collaborative scheduling, this study designs a two-layer mathematical model to address the problem. The collaborative scheduling of QCs, AGVs, and DCRCs forms the upper layer model, while the accurate path planning of AGVs serves as the lower layer model, aiming to minimize the TEC of multiple equipment.

3.4.1. Assumptions

(1) Multiple AGVs correspond to multiple QCs and do not serve specific QCs exclusively.
(2) AGVs travel at a constant speed.
(3) The influence of loaded and unloaded conditions on the energy consumption of QCs and DCRCs is not considered.
(4) All containers are standardized at 40 ft in size.
(5) The task list and execution sequence for each QC are fixed.
(6) The operating area of each DCRC is fixed and it only serves a single designated block.

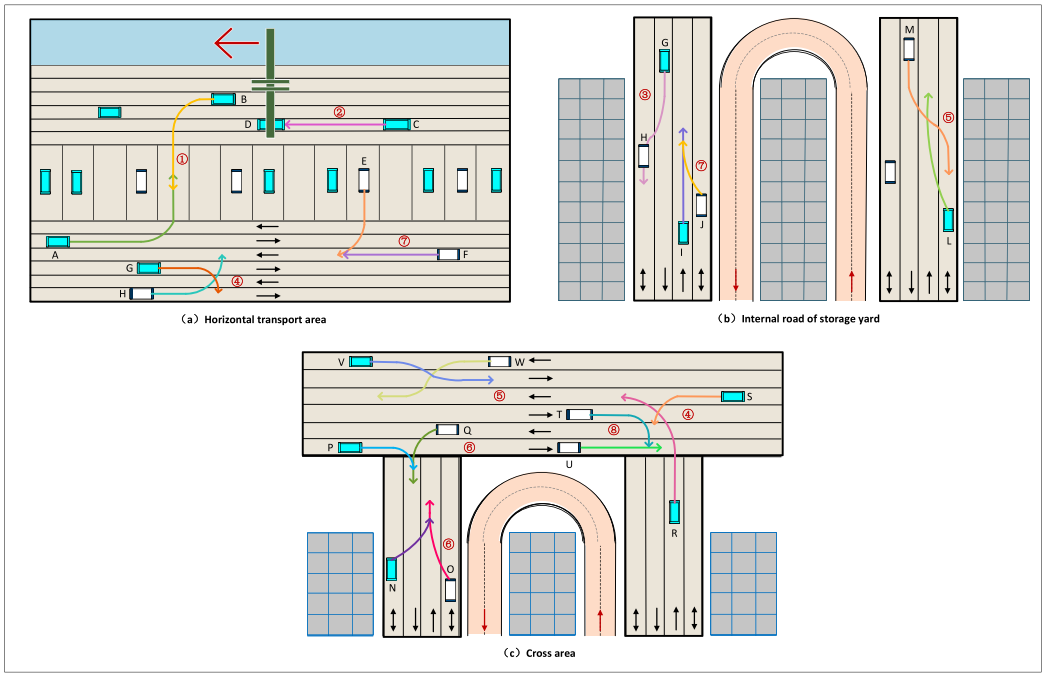

Fig. 3. AGV path conflict of U-shaped ACTs.

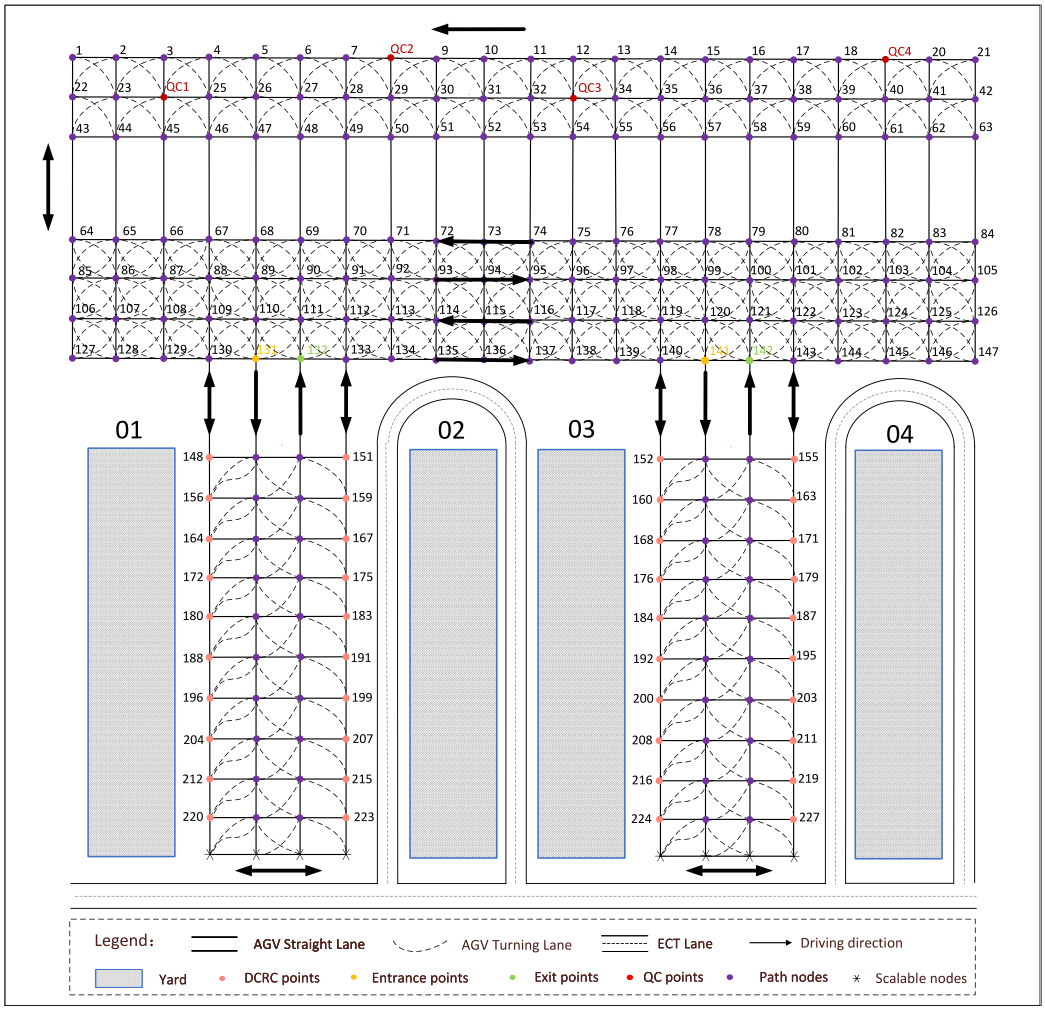

Fig. 4. AGV path conflict of U-shaped ACTs.

3.4.2. Model parameters

(1) Set

U	Set of import containers.$U = \{1, 2, 3 \cdots, i, j \cdots\}$
V	Set of AGVs.$V = \{1, 2, 3 \cdots, c \cdots\}$
Y	Set of DCRCs. $Y = \{1, 2, 3 \cdots, m \cdots\}$
Q	Set of QCs.$Q = \{1, 2, 3 \cdots, k, l \cdots\}$
$\mathcal{U}_k$	Set of container tasks assigned to QC k
D	Set of yards.$D = \{1, 2, 3 \cdots, d \cdots\}$
B_{db}	Set of blocks in yard, b indicates bay number.$d \in D$
N	Set of all nodes of AGV path.$N = \{1, 2, 3 \cdots, i', j' \cdots\}$
N'	Set of bi-directional nodes of AGV path
W	Set of road sections between AGV nodes $W = \left\{ (i'j') \middle\| i' \to j' : i', j' \in N \right\}$

(2) Parameters

M	A very large positive number
T	The time for trolley operation of DCRC for one container task
v_1	Speed of the AGV
v_2	Speed of the DCRC
φ_m^i	Location of the bay in the yard of the ith container task of the DCRC operation
e_c^o	Energy consumption per hour of empty AGV
e_c^f	Energy consumption per hour of laden AGV.
e_c^w	Energy consumption per hour of an AGV while waiting
e_m^o	Energy consumption per hour of a DCRC while loading
e_m^f	Energy consumption per hour of a DCRC while moving
l_b	Length of one bay
l_v	Length of AGV
l_s	Safety distance of AGV on the same path
$w_{i'j'}$	The distance of the road section $(i'j')$ between node i' to node j'

(3) Decision variables

0–1 variables

β_c^i	If AGV c handles the ith container task, it is 1, otherwise it is 0
α_m^i	If DCRC m handles the ith container task, it is 1, otherwise it is 0
ε_k^{ij}	If QC k continues to process the jth container task after processing the ith container task, it is 1, otherwise it is 0
γ_m^{ij}	If DCRC m continues to process the jth container task after processing the ith container task, it is 1, otherwise it is 0
$z_{c,i'j'}$	If AGV c visits node i' after visiting node j', it is 1, otherwise it is 0
$a_{c,i'j'}$	If AGV c visits the road section $(i'j')$, it is 1, otherwise it is 0
ϑw_c^i	If AGV c arrives at the QC loading/unloading point for task i and waits for the interaction to be completed, it is 1, otherwise it is 0
ωw_c^i	If AGV c arrives at the DCRC loading/unloading point for task i and waits for the interaction to be completed, it is 1, otherwise it is 0
ηp_c^i	If AGV c with task i waits in conflict, it is 1, otherwise it is 0
$\varsigma p_{c,c'}^t$	If AGV c is in conflict with AGV c' at time t

No 0–1 variables

r_k^i	The time when the gantry trolley of QC k is scheduled to start working on the ith container task ($i \in \mathcal{U}_k$).
p_{ck}^i	The time when the AGV c arrives under QC k to pick up the ith container task
u_{kc}^i	The time when the gantry trolley of QC k actually puts the ith container task on the AGV c
h_{cm}^i	The time when the AGV c transports the ith container task to the designated bay to wait for DCRC m to operate
q_m^i	The time when DCRC m moves to the target bay for the ith container task
o_m^i	The time when the DCRC m actually starts loading the ith container task
g_{mc}^i	The time when the DCRC m take the ith container task from the AGV c
f_m^i	The time when the DCRC m take the ith container task to the target bay
l_m^{ij}	Time for the DCRC m to move from the target bay of the previous task i to the target bay of the next task j
$t_{c,i'j'}^i$	Travel time from start position i' to finish position j' for AGV c transporting the ith container task
$t_{in,i'j'}^i$	The time when an AGV transporting the ith container task enters the road section$(i'j')$
$t_{out,i'j'}^i$	The time when an AGV transporting the ith container task leaves the road section$(i'j')$
$t_{c,k}^i$	The time AGV c spends waiting for the completion of interaction of task i at the location of QC k
$t_{c,m}^i$	The time AGV c spends waiting for the completion of interaction of task i at the location of DCRC m
tp_c^i	Waiting time for low-priority AGV c with task i due to conflict occurrence
$T_{c,i'j'}^i$	Time window in the road section $(i'j')$ when the AGV performs container task i

3.4.3. Cooperative scheduling model

Given that environmental protection and reduction of energy consumption have become a global consensus. ACTs not only are capable of operating continuously for 24 h to raise the upper limit of terminal efficiency but also should explore feasible solutions for reducing energy consumption and emissions. The existing research mainly focuses on optimization for the purpose of shortening the maximum completion time or increasing throughput, yet the attention paid to energy consumption is relatively insufficient. This study sets minimizing equipment energy consumption as the core objective of the model, with the hope of providing new quantitative references from the perspective of energy consumption. In order to simulate the complete unloading process of the U-shaped ACTs, the scheduling of QCs, AGVs, and DCRCs need to be combined to minimize the energy consumption of all tasks. Through this process, the optimal task sequence of handling and the scheduling of AGVs can be obtained.

Upper-level objective function:

$$f = \min\left\{E_V + E_Y\right\} \tag{2}$$

Eq. (2) is the objective function of the cooperative scheduling model, which indicates that the TEC of AGVs, and DCRC is minimized.

$$E_V = E_c^o + E_c^f + E_c^w \tag{3}$$

$$E_c^o = e_c^o \sum_{i \in U} \sum_{c \in V} t_{c,i'j'}^i \beta_c^i \quad \forall i' \in B_{db}, j' \in Q \tag{4}$$

$$E_c^f = e_c^f \sum_{i \in U} \sum_{c \in V} t_{c,i'j'}^i \beta_c^i \quad \forall i' \in Q, j' \in B_{db} \tag{5}$$

$$E_c^w = e_c^w \sum_{i \in U} \sum_{c \in V} \left(t_{c,w}^i + tp_c^i\right) \tag{6}$$

$$E_Y = e_m^o \sum_{i,j \in U} \sum_{m \in Y} l_m^{ij} \gamma_m^{ij} + e_m^f \sum_{i \in U} \sum_{m \in Y} T_3 \alpha_m^i \tag{7}$$

Eq. (3) represents the TEC of AGVs. Eq. (4) represents the unloaded energy consumption of AGVs. Eq. (5) represents the laden energy

consumption of AGVs. Eq. (6) represents the energy consumption of AGVs waiting for the QC and DCRC. Eq. (7) represents the TEC of DCRCs, including the energy consumption for the movement and the energy consumption of the loading of DCRCs.

Uniqueness constraints:

$$\sum_{i\in U}\beta_c^i = 1 \quad \forall c \in V \tag{8}$$

$$\sum_{i\in U}\alpha_m^i = 1 \quad \forall m \in Y \tag{9}$$

$$\sum_{i\in U}\sum_{c\in V}\theta_c^{ij} = 1 \quad \forall j \in U \tag{10}$$

$$\sum_{i\in U}\sum_{m\in Y}\gamma_m^{ij} = 1 \quad \forall j \in U \tag{11}$$

Eq. (8) represents the fact that only one AGV can operate each container task. Equation (9) represents that only one DCRC can operate each container task. Eqs. (10), and (11) ensure that there is exclusively one next task after the completion of the previous task by the same equipment.

Temporal coordination constraints:

$$p_{ck}^i \le r_k^i, \quad \forall k \in Q,\ i \in \mathcal{U}_k,\ c \in V \tag{12}$$

$$u_{kc}^i + t_{c,i'j'}^i \le h_{cm}^i \quad \forall i \in \mathcal{U}_k, k \in Q, c \in V, m \in Y, i' \in Q, j' \in B_{db} \tag{13}$$

$$h_{cm}^i \le o_m^i + M\left(1-\alpha_m^i\right) \quad \forall i \in U, m \in Y, c \in V \tag{14}$$

$$o_m^i = \max\left\{q_m^i, h_{cm}^i\right\} \quad \forall i \in U, m \in Y, c \in V \tag{15}$$

$$o_m^i + T \le g_{mc}^i \quad \forall i \in U, c \in V, m \in Y \tag{16}$$

$$g_{mc}^i + T \le f_m^i \quad \forall i \in U, c \in V, m \in Y \tag{17}$$

$$g_{mc}^i + t_{c,i'j'}^j \le p_{cl}^j + M(1-\theta_c^{ij}) \quad \forall i,j \in U, c \in V, l \in Q, m \in Y, \forall i' \in B_{db}, j' \in Q \tag{18}$$

$$t_{c,k}^i = u_{kc}^i - p_{ck}^i \quad \forall i \in U, k \in Q, c \in V \tag{19}$$

$$t_{c,m}^i = g_{mc}^i - h_{cm}^i \quad \forall i \in U, m \in Y, c \in V \tag{20}$$

$$l_m^{ij} = l_b \cdot \left|\varphi_m^i - \varphi_m^j\right| / v_2 \quad \forall i,j \in U, m \in Y \tag{21}$$

$$q_m^j \ge f_m^i + l_m^{ij}\gamma_m^{ij} \quad \forall i,j \in U, m \in Y \tag{22}$$

Eq. (12) represents the relationship between the moment when the container task i is placed on AGV c by the DCRC m and the moment when AGV c arrives at the position beneath the DCRC m. Equation (13) represents the relationship between the time when the AGV c starts to transport the container$i$ and the time when the AGV c transports the container i to the target bay. Eq. (14) represents the relationship between the time when the DCRC m actually starts working on the container$i$ and the time when the AGV c arrives at the target bay. Equation (15) represents that the time when the DCRC m actually starts to work on the container i should be the more significant value between the time when the DCRC m arrives at the target bay and the time when the AGV c arrives at the target bay. Equation (16) represents the relationship between the time when the DCRC m actually starts the loading operation and the time when the DCRC m removes the container i from the AGV c. Eq. (17) represents the relationship between the time when the DCRC m takes the container$i$ from the AGV c and the time when the current task i ends. Eq. (18) represents the relationship between the time the AGV c completes the last container task i and the start time of the next container task j. Eqs. (19) and (20) represent the waiting time of the AGV c under the QC k and DCRC m, respectively. Eq. (21) represents the time for the DCRC m to move from the previous position i to the next j. Eq. (22) represents the time relationship between the two tasks i, j of the DCRC m in continuous operation.

$$r_k^i, u_{kc}^i, p_{ck}^i, h_{cm}^i, g_{mc}^i, q_m^i, o_m^i, f_m^i \ge 0 \quad \forall i \in U, k \in Q, c \in V, m \in Y \tag{23}$$

Eq. (23) represents the range of relevant time parameters.

3.4.4. Path planning model

Based on the sequence of scheduling tasks decided by the upper layer model, the lower layer model plans accurate paths of AGVs to operate the tasks and feeds the AGV travel time back to the upper layer model for solving. The path planning model is as follows.

Lower-level objective function:

$$t_{c,i'j'}^i = \min \sum_{i',j'\in N}\beta_c^i\left(w_{i'j'}a_{c,i'j'}/v_1\right) \quad \forall i \in U, c \in V \tag{24}$$

Eq. (24) is the objective function of the lower level, and the objective is the shortest time traveled by the AGV c to transport the task from the start point to the end point after accurate path planning.

Uniqueness constraints:

$$\sum_{i'=1}^{N}\sum_{j'=1}^{N} z_{c,i'j'} = 1 \quad \forall c \in V \tag{25}$$

$$a_{c,i'j'} \le z_{c,i'j'} \quad \forall i', j' \in N, c \in V \tag{26}$$

Eq. (25) represents that AGVs do not visit the same path repeatedly during traveling to avoid creating loops. Eq. (26) represents that the selection of a road section presupposes the existence of a path between nodes.

Direction and safety distance constraints:

$$w_{i'j'} \ne w_{j'i'} \quad \forall i', j' \in \left(N - N'\right) \tag{27}$$

$$w_{i'j'} = w_{j'i'} \quad \forall i', j' \in N' \tag{28}$$

$$l_s + l_v > \left|t_{in,i'j'}^i - t_{in,i'j'}^j\right| \cdot v_1 \quad \forall\left(i'j'\right) \in W, i,j \in U \tag{29}$$

$$l_s > \left|t_{in,i'j'}^i - t_{in,j'i'}^j\right| \cdot v_1 \quad \forall\left(i'j'\right),\left(j'i'\right) \in W, i,j \in U \tag{30}$$

Eqs. (27) and (28) represent the unidirectional and bi-directional paths of AGVs, respectively. Eq. (29) represents the safe distance between AGVs traveling in the same direction. Eq. (30) represents the safe distance between AGVs traveling in opposite directions.

Conflict detection and avoidance constraints:

$$T_{c,i'j'}^i = \left(i, c, t_{in,i'j'}^i, t_{out,i'j'}^i\right) \quad \forall\left(i'j'\right) \in W, i \in U, c \in V \tag{31}$$

$$T_{c,i'j'}^i \cap T_{c',i'j'}^j = \varnothing \quad \forall\left(i'j'\right) \in W, i,j \in U, c,c' \in V \tag{32}$$

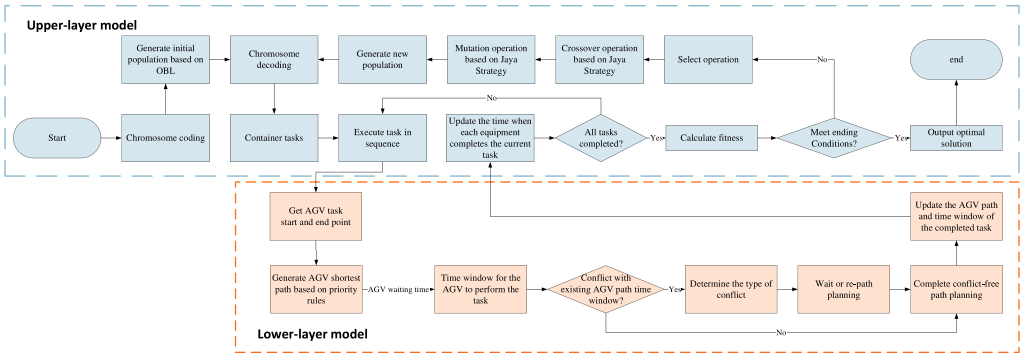

Fig. 5. Algorithm flow.

$$T^{i}_{c,i'j'} \cap T^{j}_{c',j'i'} = \varnothing \quad \forall (i'j'), (j'i') \in W, i, j \in U, c, c' \in V \tag{33}$$

$$t^{i}_{out,i'j'} = t^{i}_{in,i'j'} + \left(w_{i'j'} a_{c,i'j'} / v_1 \right) + t^{i}_{c,k} \vartheta w^{i}_{c} + t^{i}_{c,m} \omega w^{i}_{c} + t^{i}_{c,p} \eta p^{i}_{c} \quad \forall (i'j') \in W, i', j' \in N, i \in U, c \in V \tag{34}$$

$$tp^{i}_{c} = t^{i}_{out,i'j'} - t^{j}_{in,i''j''} \quad \forall (i'j'), (i''j'') \in W, i, j \in U, c \in V \tag{35}$$

$$\begin{cases} a_{c,i'j'} = 1, a_{c',i'j'} = 0 & if\, \varsigma p^{t}_{c,c'} = 1 \\ a_{c,i'j'} = 0, a_{c',i'j'} = 1 & if\, \varsigma p^{t}_{c,c'} = 1 \end{cases} \quad \forall c, c' \in V, i, j \in U, i', j' \in N \tag{36}$$

$$\begin{cases} a_{c,i'j'} = 1, a_{c',j'i'} = 0 & if\, \varsigma p^{t}_{c,c'} = 1 \\ a_{c,i'j'} = 0, a_{c',j'i'} = 1 & if\, \varsigma p^{t}_{c,c'} = 1 \end{cases} \quad \forall c, c' \in V, i, j \in U, i', j' \in N \tag{37}$$

Eq. (31) represents the time window on the road section $(i'j')$ during task i of the AGV operation. Eqs. (32) and (33) represent the detection of overlapping time windows for two AGVs on a roadway section that can be traveled in one and two directions, respectively. Eq. (34) represents that the time when an AGV leaves a road section equals the sum of the time it enters the roadway section and the AGV's travel time and waiting time on the road section. Eq. (35) represents that the time that the AGV with low priority waits due to conflict is equal to the difference between the time when the AGV with high priority leaves the target location and the time when the AGV with low priority arrives at the location section outside the safety distance. Eqs. (36) and (37) represent that if there is a time conflict at time t, the AGV with low priority cannot enter the road section $(i'j')$ at time t. If it is a bi-directionally travelable road section, it cannot enter the road section $(i'j')$ or $(j'i')$.

4. Algorithm

For the established two-layer model, this study designs the Adaptive Genetic Algorithm Based on the Jaya Strategy (JAGA) to solve the upper-layer model. It adopts the AGV Accurate Path Planning Algorithm (APPA) to solve the lower-layer model to obtain the optimal cooperative scheduling scheme and the conflict-free shortest path. The flow of the algorithm is shown in Fig. 5, The pseudocode of the algorithm is presented in Appendix A.

AGV	1	1	1	2	2	2	3	3	3
QC	2	1	3	1	3	3	2	2	1

Fig. 6. Chromosome coding.

4.1. Upper-layer model algorithm

As a global optimization heuristic algorithm, the genetic algorithm is widely used to solve ACTs' multi-equipment cooperative scheduling problem (Shouwen et al., 2021; Xu et al., 2021). However, the problems of insufficient local search ability and slow convergence speed of genetic algorithms are more prominent. In this study, JAGA is designed to improve adaptive genetic algorithms' optimization ability and convergence speed.

4.1.1. Chromosome coding and decoding

Since the container-handling tasks, task sequences for each quay crane (QC), and the target yard blocks and bays for each container are predetermined during preprocessing, this study focuses on how to schedule AGVs to arrive at the assigned QCs at the proper time for container transport. To represent the scheduling plan effectively, a real-number encoding scheme is adopted. The chromosome is designed as a two-row matrix, with its length equal to the number of tasks. The first row indicates which AGV executes each task; the second row specifies the corresponding QC for each task. The chromosome coding schematic is shown in Fig. 6. After decoding, for AGV 1, the decoded task sequence is as follows: first, it serves QC 2; next, it proceeds to QC 1; then, it handles a task at QC 3. The AGV performs these transport tasks in the specified order, moving between QCs accordingly.

4.1.2. Fitness function

The considered model in this work takes the minimization of the total energy consumption of the multi-equipment as the objective function, so its reciprocal is taken as the fitness function, and the specific fitness function is as follows.

$$\max F = \frac{1}{f} \tag{38}$$

4.1.3. Initialize the population

The population generated using a stochastic strategy has an unknown convergence rate since no estimation has been performed. Therefore, this study uses the Opposition-Based Learning (OBL) strategy to initialize the population. The concept of OBL strategy was introduced by Tizhoosh (2005). Its main idea is to select the better of the two, the forward and the reverse solutions, as the newly generated solutions. Later, Rahnamayan et al. (2008) suggested that using the OBL strategy for initial population optimization can accelerate the convergence

and experimentally verified it with various of complex benchmark functions.

The specific steps are as follows. First, generating an initial population $P\left(N_P\right)$ uniformly and randomly. Second, a reverse population OP of the current population is generated according to Eq. (38). Again, the fitness of all the individuals $\left\{OP \bigcup P\right\}$ is calculated and ranked in descending order. Finally, the best N_p from concatenating the current and reverse populations are selected to participate in the iteration. The result of this operation may produce infeasible solutions, which are subsequently repaired to feasible solutions.

$$OP_{i,j} = a_j + b_j - P_{i,j}, \forall i = 1, 2, \ldots, N_p, j = 1, 2, \ldots, D \tag{39}$$

In the equation, a_j and b_j represent gene boundaries, respectively. $P_{i,j}$ represents jth gene on the ith chromosome of the randomly generated initial population. $OP_{i,j}$ represents the jth gene on the ith chromosome of the reverse population generated according to the strategy. N_p represents the number of chromosomes in the population. D represents the number of genes in the chromosomes.

4.1.4. Select the operations

In order to prevent some individuals from being selected too repeatedly and falling into a local optimum, a combination of elite selection strategy and roulette is used for the selection operation. First, a certain number of individuals with the highest fitness are selected from the population after cross-mutation to enter the next generation directly. Then, a roulette wheel is used to continue to select from the remaining individuals until the population size of the next generation reaches a preset value.

4.1.5. Crossover and mutation operations

Genetic algorithms operate by crossover and mutation to produce offspring, and their convergence speed and searchability are affected by both crossover and mutation probability. In order to improve the performance of the algorithm, the crossover probability and mutation probability equations in Ref. (Yan et al., 2020) and the adaptive crossover probability and adaptive mutation probability are expressed explicitly in Eqs. (40) and (41).

$$P_c = \begin{cases} K_1\left(1 - \dfrac{\arcsin\left(\frac{F_{avg}}{F_{\max}}\right)}{\pi/2}\right) & \arcsin\left(\dfrac{F_{avg}}{F_{\max}}\right) \geq \pi/6 \\ K_4\left(\dfrac{\arcsin\left(\frac{F_{avg}}{F_{\max}}\right)}{\pi/2}\right) & \arcsin\left(\dfrac{F_{avg}}{F_{\max}}\right) < \pi/6 \end{cases} \tag{40}$$

$$P_m = \begin{cases} K_3\left(1 - \dfrac{\arcsin\left(\frac{F_{avg}}{F_{\max}}\right)}{\pi/2}\right) & \arcsin\left(\dfrac{F_{avg}}{F_{\max}}\right) < \pi/6 \\ K_2\left(\dfrac{\arcsin\left(\frac{F_{avg}}{F_{\max}}\right)}{\pi/2}\right) & \arcsin\left(\dfrac{F_{avg}}{F_{\max}}\right) \geq \pi/6 \end{cases} \tag{41}$$

In the equation, p_c and p_m are the probabilities of crossover and mutation, K_1, K_2, K_3 and K_4 are random numbers between [0,1], $F_{\max}$ and F_{avg} are the maximum and average values of the fitness, respectively.

Jaya algorithm is an optimization algorithm with the basic idea of converging to the optimal solution and staying away from the worst solution (Rao, 2016), which has the advantages of parameter-free operation, fast solution speed, and not quickly falling into the local optimal solution. The equation for chromosome gene update based on the Jaya algorithm is as follows:

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,best,i} - \left|X_{j,k,i}\right|\right) - r_{2,j,i}\left(X_{j,worst,i} - \left|X_{j,k,i}\right|\right) \tag{42}$$

In the equations, $X_{j,best,i}$ represents the jth gene of the chromosome with the largest fitness value after the ith iteration. $X_{j,worst,i}$ represents the jth gene of the chromosome with the smallest fitness value after the ith iteration. $X_{j,k,i}$ represents the jth gene of the chromosome to be crossed over after the ith iteration. $X'_{j,k,i}$ represents the value after updating $X_{j,k,i}$. $r_{1,j,i}$ and $r_{2,j,i}$ is a random number between [0,1].

Select the chromosomes to be crossover according to the crossover probability and combine the Jaya update solution strategy to crossover the chromosome's first layer of task number, as shown in Fig. 7. First, find the chromosome with the most significant fitness value and the chromosome with the smallest fitness value in this iteration process, i.e., the optimal individual and the worst individual. Randomly select the location of the two points of crossover, according to Eq. (42) crossover region of the chromosome genes are updated and rounded, and then the same elements in the chromosome will be set to "0". Finally, other individuals in the population were randomly selected and compared with the updated individuals, and the elements contained in the randomly selected individuals but not in the updated individuals were sequentially replaced by the "0" elements in the latter.

The chromosome to be mutated is selected according to the mutation probability, and the chromosome after two-point mutation is further optimized using the combined Jaya update strategy. First, two different gene positions are randomly selected in the first layer of the chromosome, and the gene elements in these two positions are exchanged. The subsequent steps are similar to the crossover, as shown in Fig. 8.

4.2. Lower-layer model algorithm

4.2.1. Priority rules

The lower-level model generates conflict-free and accurate routes for all AGVs operating in a U-shaped automated container terminal (ACT). Starting from a first-come-first-served (FCFS) policy, the AGV motion direction priority rules are designed as follows:

Rule 1: Lane choice by task position.
If the AGV is to the right of its task's destination, it selects the right-hand lane. If the AGV is to the left of its task's destination, it selects the left-hand lane.
Example: An AGV traveling from the yard toward QC-2—located on its left—stays in the left lane to avoid lane crossing.

Rule 2: Horizontal motion in the QC handling zone.
When the task destination lies under a QC, the AGV must enter the loading point laterally from the node on the right-hand side of the road, complete the hand-off, then exit laterally via the left-hand node and turn vertically into the buffer lane.
Example: The AGV enters the QC bay from the right-side node, performs the lift, exits through the left-side node, and turns vertically to the next task point, keeping the QC area orderly and safe.

Rule 3: Vertical motion in the DCRC yard zone.
When the task destination lies under a DCRC, the AGV approaches vertically from the node above the loading point. After hand-off it may change lane directly and depart.
Example: The AGV drives straight down into the DCRC bay and, without reversing, changes lane to the main road and leaves swiftly.

4.2.2. Conflict resolution strategy

The standard strategy for conflict resolution includes stopping and waiting, re-planning routes, acceleration, and deceleration avoidance. In this study, the conflict resolution strategy combining waiting and re-pathing is designed for U-shaped ACTs' process mode, layout characteristics, and road settings, as shown in Table 1.

AGV node-occupancy conflicts in U-shaped ACTs mainly occurs in the loading and unloading lanes of QCs and the operation lanes in the yards. It means that the AGVs need to maintain a considerable safety distance from each other so as not to interfere with the interactive operations. Therefore, it is more appropriate to set the AGV with low

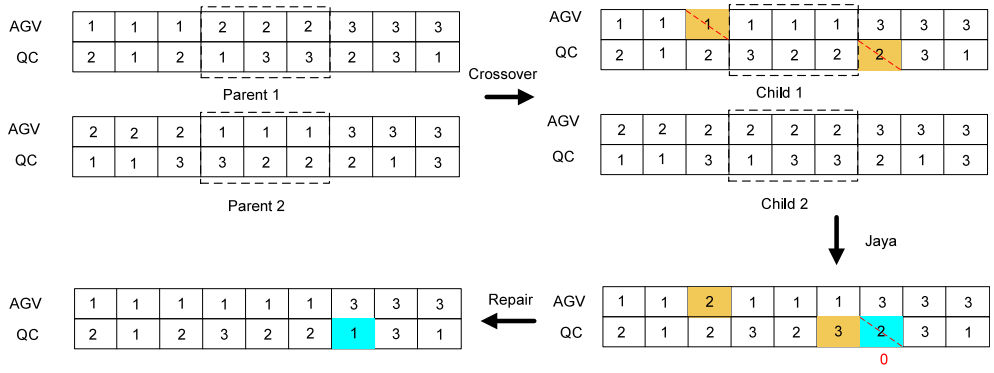

Fig. 7. Chromosome crossover.

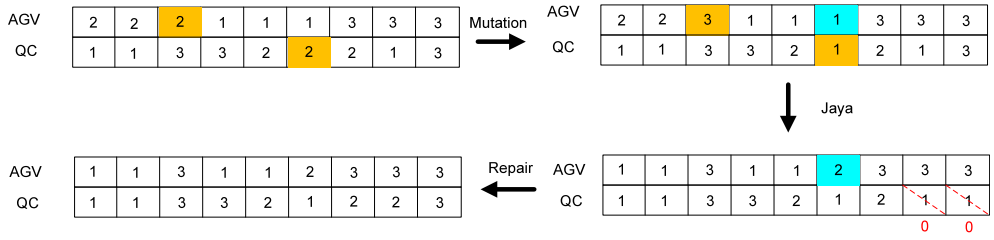

Fig. 8. Chromosome mutation.

Table 1
Conflict Resolution Strategy.

Type of conflict	Strategy
Node-occupancy conflicts	Stop and wait
Head-on conflict	Re-pathing
Cross conflict	Re-pathing

priority to stop and wait outside the safe distance of the same lane until the AGV with high priority leaves.

Head-on conflict and cross conflict mainly occur in the process of AGV traveling. AGV acceleration and deceleration may lead to the AGV traveling in the same direction to catch up with the conflict. Stopping and waiting will lead to road congestion. The number of AGV lanes in the horizontal area and the yard is significant, and there are bidirectional driving lines in U-shaped ACTs, meaning there are multiple shortest paths, so the AGVs with low priority are adopted to re-path the planning. Specifically, the low-priority AGV aborts its current path segment from the conflict point (or the node just before the conflict occurs) and re-runs the APPA path planning algorithm using updated constraints that exclude the blocked/conflicting road segments and time windows. The portion of the route that has already been traveled remains fixed, while only the remaining path from the conflict node to the destination is recalculated. During this recalculation, the lower-priority vehicle immediately re-plans its route to ensure the new path is collision-free, and always selects the shortest feasible detour. This approach ensures that the AGV can bypass the conflict area by selecting an alternative feasible route, reducing the likelihood of prolonged congestion on critical nodes or bidirectional sections.

4.2.3. Algorithm flow of path planning

The specific steps of the accurate path planning algorithm based on priority rules for AGV are as follows:

Step1. Determine the location of the task loading and unloading points using the AGV scheduling scheme obtained after chromosome decoding.

Step2. Initial route generation.
a. Build a candidate set of k shortest paths satisfying Rules 1–3;
b. Choose the shortest path as the initial route;
c. Incorporate real-time traffic and compute travel time.

Step3. Combine the AGV waiting time returned by the upper layer algorithm to generate the time window for the AGV to operate and store this task.

Step4. Detect the conflict between the time window of the AGV operating the task and the time window of the existing path of each AGV.

Step5. Conflict handling.
a. If the conflict is node-occupancy → low-priority AGV waits.
b. If the conflict is head-on or cross → low-priority AGV re-paths (return to Step 2).
c. Then, turn to Step 4. to recheck.

Step6. If there is no time conflict, then directly complete the conflict-free path planning.

Step7. Update the AGV path and time window of the completed task and return it to the upper-layer model algorithm to update the time for each device to complete the current task.

5. Simulation and validation

This section designs experiments to verify the effectiveness of the conflict resolution strategies and the optimization algorithms. These experiments are conducted on a computer with a Windows 11 operating system, 16 GB RAM, and a 2.40 GHz CPU, using MATLAB 2024b for optimization.

5.1. Parameter settings

Referring to the layout of the U-shaped ACT in the Beibu Gulf, the specific settings for the experimental parameters are as follows:

(1) The number of QC and DCRC is set to 4 each, with the number of AGVs ranging from 5 to 50, and there are 4 blocks, each with a length of 40 bays. This is typical for small to medium-sized ACTs.
(2) The width of the horizontal transport area is 100 m, and its length is 300 m. Each bay in the storage yard is 15 meters long.

Table 2
The energy consumption parameters of the equipment.

Parameter	Value	Parameter	Value
e_c^f	21 kWh/h/ unit	e_k^f	91.24 kWh/h/ unit
e_c^o	14 kWh/h/ unit	e_m^o	90 kWh/h/ unit
e_c^w	9 kWh/h/ unit	e_m^f	25 kWh/h/unit

Table 3
Solution results of the three algorithms at different scales.

Number of Tasks	JAGA (OFV/ CPUT)	AGA (OFV/ CPUT)	BGA (OFV/ CPUT)	PSO (OFV/ CPUT)	ACO (OFV/ CPUT)
20	65.98/ 11.90	69.73/14.77	67.85/13.61	68.17/15.20	69.22/14.54
40	150.36/18.16	155.64/22.11	158.45/21.78	160.03/25.59	159.84/24.57
60	269.87/31.49	285.19/37.23	292.42/35.24	304.51/41.38	288.06/39.89
80	431.28/37.53	460.75/44.62	483.63/43.01	500.98/49.55	470.39/49.26
100	500.42/34.13	530.91/40.48	565.33/39.65	590.14/46.37	545.88/46.24

(3) The horizontal transport area includes 3 AGV operation lanes and 4 AGV express lanes. There are 2 AGV operation lanes and 2 AGV driving lanes between two adjacent blocks in the storage yard.
(4) The length of each AGV is 13 m, with a safety distance of 5 m. The AGV speed is set at 5 m/s, and the DCRC's movement speed is 2 m/s. The energy consumption parameters for the equipment are shown in Table 2.
(5) The DCRC has a loading and unloading time of 30 s.
(6) Using the control variable method through multiple experiments, the population size to 100, the initial crossover and mutation probabilities to 0.7 and 0.3, and the learning Generation Gap to 0.15, respectively. The sensitivity analysis of model parameters is presented in Appendix B, with detailed results shown in Tables B.1–B.4.

5.2. Algorithm comparison

To verify the effectiveness of the proposed model and algorithm, a set of comparative experiments was conducted under different task scales. In the same simulation environment with 10 AGVs, five algorithms—classic Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), AGA (Yan et al., 2020), BGA (Yang et al., 2018), and the proposed JAGA—were used to solve the model. Their performance was compared in terms of the optimal objective function value (OFV, in kWh) and computation time (CPUT, in seconds). All five algorithms employ the same convergence criterion based on JAGA: iteration halts when the improvement over 300 generations falls below 0.01%. Detailed results are presented in Table 3.

Table 3 compares optimal energy consumption (OFV) and computation time (CPUT) for the five methods under scenarios of 20–100 tasks with 10 AGVs. Key observations: JAGA consistently achieves the lowest OFV across all problem sizes, while keeping CPUT under 40 s. For 20 tasks, it finds 69 kWh in just 11.90 s; at 100 tasks, OFV rises to 500 kWh but remains below that of the other algorithms. BGA and AGA see OFV grow markedly with scale and converge far slower than JAGA. At 100 tasks, their OFVs exceed JAGA's by roughly 9% and 16%, respectively, with CPUTs over 40 s. PSO and ACO match classic GA variants at small-to-medium scales but, beyond 80 tasks, suffer from limited search dimensionality: both OFV and CPUT spike, and overall performance falls behind JAGA.

Fig. 9 illustrates the convergence curves of the three algorithms with 20 container tasks and 10 AGVs.

As this figure shown, JAGA drives the objective below 66 kWh within 80 generations and continues a steady descent. By 150 generations, improvements dip under 0.01%, demonstrating high convergence speed and stability. In contrast, BGA and AGA show pronounced oscillations—sensitive to population diversity—while PSO and ACO converge smoothly but to higher final OFVs than JAGA.

A combined look at Tables 8 and 9 confirms that JAGA reaches superior objective values in relatively few generations, whereas AGA/BGA exhibit greater quality volatility at larger scales, and PSO/ACO incur sharply rising CPU times as scale grows. These results validate JAGA's effectiveness and robustness for multi-equipment scheduling and AGV path planning in U-shaped automated container terminals.

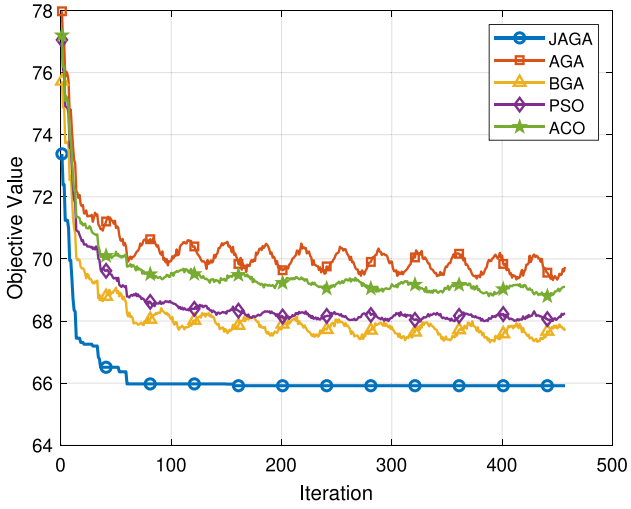

Fig. 9. Performance comparison of different algorithms.

5.3. Validation of conflict resolution strategies

To assess the effectiveness of the APPA, we conducted a small-scale simulation with 15 tasks and 5 AGVs, using the traditional path-planning algorithm (TPPA) for comparison. TPPA relies on a static AGV priority list to perform a single global shortest-path computation, whereas APPA dynamically applies re-routing whenever a conflict is detected.

The TPPA routing results are shown in Fig. 10, where red "×" symbols mark the nine distinct conflict nodes identified around 15 s, 40 s, and 85 s of simulation time. These clusters reveal severe local congestion when using a purely static priority scheme.

Under APPA, higher-priority AGVs proactively avoid conflict zones by recalculating their routes. Fig. 11 presents the final APPA results: all nine previously detected conflicts have been successfully eliminated. Green circles indicate the original conflict locations, clearly demonstrating that dynamic re-planning has resolved them, and the AGV trajectories are entirely conflict-free. The total execution time for APPA's conflict detection and re-planning was 0.17 s, with each re-planning operation averaging 18 ms. This meets the real-time responsiveness requirements of automated container terminals.

Table 4 shows the path planning results for AGV 1 produced by TPPA and by the APPA. It can be seen that APPA reduces the number of passing nodes, significantly improving the efficiency of path planning.

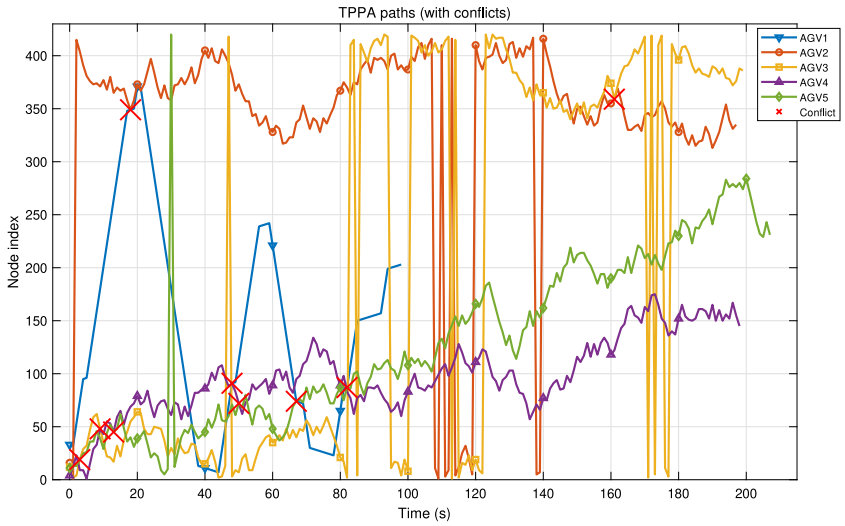

Fig. 10. AGV accurate path planning results without considering conflicts.

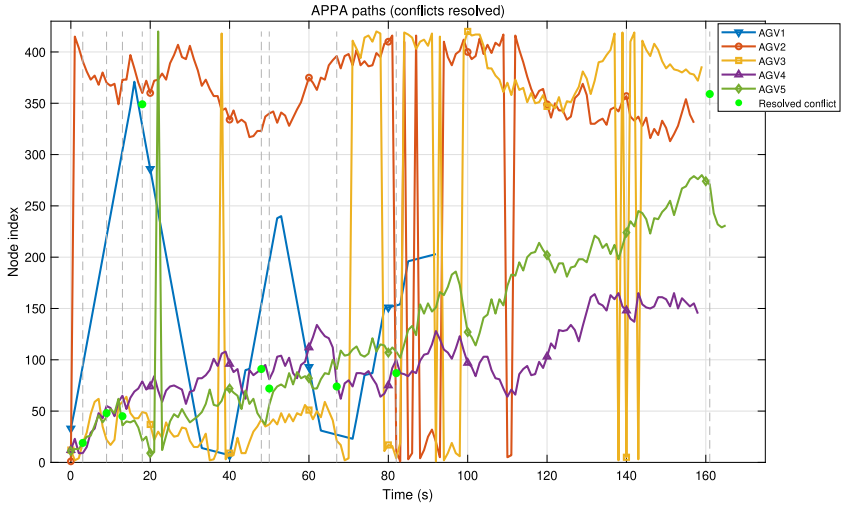

Fig. 11. Conflict-free AGV accurate path planning results.

Table 4
Two path planning results of AGV 1 under TPPA and APPA.

Starting point - End point	TPPA	APPA
QC3 → B4(29)	33-32-53-74-95-96-117-138-159-180-201-222-243-264-285-306-327-348-349-350–371	33-52-73-94-115-136-157-178-199-220-241-262-283-304-325-346-371
B4(29) → QC2	371-370-349-328-307-286-265-244-223-202-181-160-139-118-97-76-55-34-13-12-11-10-9–8	371-349-328-307-286-265-244-223-202-181-160-139-118-97-76-55-34-14-13-12-11-10-9-8
QC2 → B3(12)	8-7-28-49-70-91-92-113-134-155-176-197-218-239-240	8-7-27-48-69-90-91-112-133-154-175-196-217-238-240
B3(12) → QC1	240-241-242-221-200-179-158-137-116-95-74-73-72-51-30-29-28-27-26-25–24	240-219-198-177-156-135-114-93-72-51-31-30-29-28-27-26-25-24
QC1 → B4(7)	24-23-44-65-86-87-108-129-150-151-152-153-154-155-156-157-178-199-200-201-202-203	24-23-43-64-85-86-87-108-129-150-151-152-153-154-175-196-197-198-199-200-201-202-203

5.4. Sensitivity analysis

In ACTs, the numbers of QCs and DCRCs are fixed at design time, whereas the AGV fleet can be scaled to match workload. Focusing on the single objective- minimizing total equipment energy consumption, we examine how varying the number of AGVs affects both energy use and makespan. Simulations were conducted for six task volumes (200, 400, 600, 800, 1000, and 1200 tasks), incrementally increasing AGVs from 20 to 50. Each configuration was run 20 times and averaged. The results appear in Fig. 12. (Detailed data on energy, makespan, and CPU time are listed in Appendix C, with detailed results shown in Tables C.1–C.6.)

Across all task scales, increasing AGVs from 20 to 50 raises total energy consumption. For example, with 1200 tasks, the OFV climbs from 7418 kWh at 20 AGVs to 11200 kWh at 50 AGVs—an average increase of about 750 kWh per additional five vehicles, with larger increments under heavier load.

In the light-to-moderate load range (200–600 tasks), expanding the fleet to 30 AGVs compresses makespan by 19%–24%. Beyond 35 AGVs, further gains vanish or reverse. For 200 tasks, makespan rebounds from 8031 s at 30 AGVs to 9698 s at 35 AGVs due to increased path conflicts.

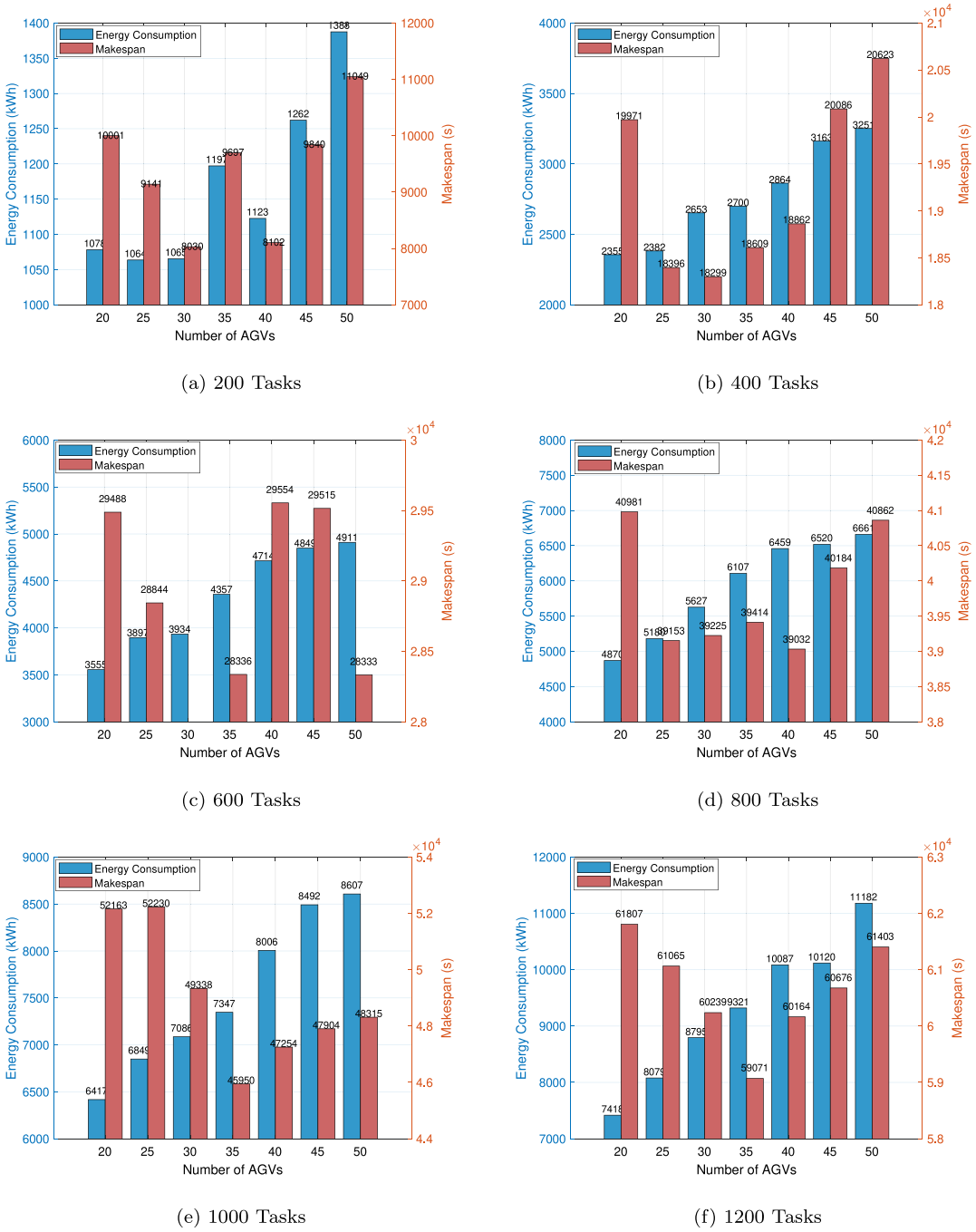

Fig. 12. Comparison of Energy Consumption and Makespan under Different AGV Numbers.

In heavier scenarios (800 tasks and above), the shortest makespan occurs at 35–40 AGVs, but at the cost of over 25% more energy—a clear "time-for-energy" trade-off entering a high-cost region.

Initially, makespan declines sharply as AGVs increase, but after a threshold, adding more vehicles yields diminishing or negative returns, and excessive fleet size can even lengthen makespan owing to more frequent conflicts. For U-shaped ACTs, managers must balance workload demands, energy costs, and service efficiency when sizing the AGV fleet.

All experiments were conducted using MATLAB 2024b on a Windows 11 system with 16 GB RAM and a 13th Gen Intel Core i7-13700H @ 2.40 GHz processor. Under this configuration, the largest test case—comprising 1,200 tasks and 50 AGVs—reached an optimal solution within 240 s. A total of 156 re-planning events were triggered during this test, with an average computation time of 14.8 ms per re-planning and a maximum of 47 ms.

Further testing showed that, when the same case was executed on a platform equipped with an Intel Core i9-14900K @ 3.20 GHz and parallel computing enabled via MATLAB's parpool, the total computation time was reduced to 30 s (28.97 s), demonstrating a significant improvement in computational efficiency. If deployed on server-grade hardware platforms using Intel Xeon or AMD EPYC processors, performance is expected to improve even further, meeting the real-time demands of ACTs.

6. Conclusion

To optimize the collaborative scheduling of QC, AGVs, and DCRCs in U-shaped ACTs, this paper takes AGV path planning as an entry

point, clarifying the scheduling relationships among the three types of equipment in U-shaped ACTs, summarizing three potential conflict types within the AGV road network in U-shaped ACTs. A new accurate AGV path planning method is proposed to address the limitations of traditional AGV path planning approaches. Furthermore, to minimize the TEC, a two-layer mathematical model of accurate AGV path planning and multi-equipment collaborative scheduling is established. The experimental results verify the validity of our model, conflict resolution strategy and algorithm. Compared with traditional methods, our method has faster computation time, enables AGVs to pass through fewer path nodes. Finally, we simulate how the number of AGVs affects the TEC in U-shaped ACTs through simulation experiments, which provides a useful reference for ACT managers.

However, the AGV path planning and scheduling problem in U-shaped ACTs remains highly complex. Although the proposed method achieved promising results in simulations, several challenges must be addressed before it can be applied to real-world U-shaped terminals: (1) U-shaped ACTs allow external trucks to enter the yard, where DCRCs must serve both AGVs and external trucks on both sides. The random arrival of external trucks can greatly affect the availability of DCRCs. Future studies should investigate the impact of these arrivals on the overall system performance. (2) In actual terminal environments, various dynamic traffic patterns exist, including random arrivals and uncertain disturbances. AGV scheduling faces delays caused by multiple sources of data. Therefore, the algorithm needs to be further improved to handle real-time dynamic scenarios effectively. (3) Real terminals involve larger and more complex yards. To validate the practicality of the proposed method, the model must be integrated with simulation software for further testing under realistic conditions. (4) Finally, although energy consumption was the focus of this study, future work will consider efficiency, energy, and cost as joint optimization objectives to provide better decision support for terminal operators.

CRediT authorship contribution statement

Yongsheng Yang: Project administration, Funding acquisition, Conceptualization. **Shu Sun:** Writing – original draft, Data curation. **Yuzhen Wu:** Resources, Methodology, Formal analysis. **Junkai Feng:** Validation, Software. **Wenying Lu:** Validation, Investigation. **Ling Wu:** Writing – original draft, Validation. **Octavian Postolache:** Writing – review & editing, Supervision.

Appendix A. Algorithm pseudocode

Algorithm 1 Adaptive Genetic Algorithm Based on the Jaya Strategy (JAGA)

Require: Task set $\mathcal{T}$, device sets $\{\mathrm{QC}, \mathrm{AGV}, \mathrm{DCRC}\}$, population size N_p, maximum generation $MaxGen$, GA parameters $\{K_1, K_2, K_3, K_4\}$, random numbers $r_{1,j,i}, r_{2,j,i}$, road network $\mathcal{G}$, path–planning parameters **param**.

Ensure: Best chromosome $\mathbf{Chrom}^*$ and minimal total energy $E_{\min}$.

Initialization
Generate a random population $P = \{\mathbf{Chrom}_i\}_{i=1}^{N_p}$.
Generate opposite population OP and merge $P \cup OP$.
Evaluate all individuals via **Call-APPA**$(\cdot)$ (Algo. 2) and select the best N_p to form the initial population.
for $iter = 1$ **to** $MaxGen$ **do**
1) Fitness Evaluation (embedded path planning)
  for all individuals $\mathbf{Chrom}_i$ **do**
    Decode $\mathbf{Chrom}_i \rightarrow$ AGV task sequences.
    $[\mathbf{Path}, \mathbf{T}] \leftarrow$ **Call-APPA**$(\mathbf{Chrom}_i, \mathcal{G}, \mathbf{param})$.
    Compute *true* energy E_i and makespan M_i from $\mathbf{T}$.
    Set fitness $F_i = 1/E_i$.
  end for
2) Elitist Selection
  Copy the best $\lceil 0.2N_p \rceil$ individuals to the next generation.
  Use roulette-wheel selection for the remaining slots.
3) Adaptive Crossover (Jaya update)
  Compute P_c using Eq. (40) of Section 4.1.
  Perform two-point crossover on the task layer; update genes by Jaya rule: $X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}(X_{j,best} - |X_{j,k,i}|) - r_{2,j,i}(X_{j,worst} - |X_{j,k,i}|)$.
  Repair duplicates and fill missing genes sequentially.
4) Adaptive Mutation
  Compute P_m via Eq. (41) and swap two genes on the task layer.
  Apply the same Jaya update to mutated genes.
5) Population Update
  Combine parents and offspring; keep the best N_p individuals.
6) Termination Test
  if convergence criterion satisfied **or** $iter = MaxGen$ **then**
    break;
  end if
end for
return $\mathbf{Chrom}^*$ (best individual) and $E_{\min}$.

Algorithm 2 AGV Accurate Path Planning Algorithm (APPA)

Require: Decoded schedule S (ordered task list for every AGV), road network $\mathcal{G}$, parameters **param**: safe headway d_{safe}, AGV speed v_{agv}, maximum wait $T_{\max}$, # shortest paths k.

Ensure: Conflict-free paths $\{\mathcal{P}_a\}$, time logs $\mathbf{T}$, total energy E, makespan M.

Initial Path Extraction
for each AGV a **do**
  for each task $(s,t) \in \mathcal{S}_a$ **do**
    Find k shortest paths $p_a^{(1\ldots k)}$ in $\mathcal{G}$.
    Select the feasible path $\mathcal{P}_a \leftarrow p_a^{(m)}$ that meets directional rules.
    Compute segment times by v_{agv}; store in event queue.
  end for
end for
Time-Space Conflict Detection
while event queue not empty **do**
  Pop earliest edge-occupancy event $\langle e, [t_1, t_2], a \rangle$.
  if another AGV b occupies e within d_{safe} time gap **then**
    if $\Delta t = t_2^b - t_1 < d_{\mathrm{safe}}/v_{\mathrm{agv}}$ **then**
      Resolve Conflict:
      if $\Delta t \le T_{\max}$ **then**
        Insert wait Δt to lower-priority AGV.
      else
        Re-route lower-priority AGV: choose next best path $p_b^{(m+1)}$, update future events.
      end if
    end if
  end if
end while
Post-processing
Compute travel times $\{T_{\mathrm{trav}}\}$, waiting times $\{T_{\mathrm{wait}}\}$.
$M \leftarrow \max_a\{\mathrm{completion}_a\}$.
Evaluate energy E with Eqs. (3)–(7) of Section 3.4.
return $\{\mathcal{P}_a\}, \mathbf{T}, E, M$.

Appendix B. Sensitivity analysis of model parameters

To avoid randomness, only a single parameter is changed when other parameters are unchanged. All tests were averaged over 20 results (see Tables B.1–B.4).

Appendix C. Sensitivity analysis of AGV quantity

See Tables C.1–C.6.

Table B.1
Sensitivity analysis of population size.

Number of Tasks	**60**	**80**	**100**	**120**	**140**
20	67.30	66.64	**65.98**	65.79	66.31
40	153.37	151.86	**150.36**	149.91	151.11
60	275.27	272.57	**269.87**	269.06	271.22
80	439.91	435.59	**431.28**	429.99	433.44
100	510.43	505.42	**500.42**	498.92	503.93

Table B.2
Sensitivity analysis of initial crossover probability.

Number of Tasks	**0.50**	**0.60**	**0.70**	**0.80**	**0.90**
20	67.63	66.64	**65.98**	66.31	66.97
40	154.12	151.86	**150.36**	151.11	152.62
60	276.62	272.57	**269.87**	271.22	273.92
80	442.06	435.59	**431.28**	433.44	437.75
100	512.93	505.42	**500.42**	502.92	507.93

Table B.3
Sensitivity Analysis of Initial Mutation Probability.

Number of Tasks	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**
20	67.30	66.64	66.31	**65.98**	66.64
40	153.37	151.86	151.11	**150.36**	151.86
60	275.27	272.57	271.22	**269.87**	272.57
80	439.91	435.59	433.44	**431.28**	435.59
100	510.43	505.42	502.92	**500.42**	505.42

Table B.4
Sensitivity Analysis of Learning Generation Gap.

Number of Tasks	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**
20	66.51	**65.98**	66.31	66.64	67.17
40	151.56	**150.36**	151.11	151.86	153.07
60	272.03	**269.87**	271.22	272.57	274.73
80	434.73	**431.28**	433.44	435.59	439.04
100	504.42	**500.42**	502.92	505.42	509.43

Table C.1
200 Tasks.

Number of AGVs	OFV (kWh)	Makespan (s)	CPUT (s)
20	1078.19	10 001	30.58
25	1063.54	9141	26.66
30	1065.39	8030	37.56
35	1197.24	9697	35.20
40	1122.61	8102	34.14
45	1262.12	9840	36.73
50	1387.64	11 049	38.65

Table C.2
400 Tasks.

Number of AGVs	OFV (kWh)	Makespan (s)	CPUT (s)
20	2355.45	19 971	41.61
25	2382.08	18 396	48.32
30	2652.51	18 299	71.91
35	2700.19	18 609	62.71
40	2863.88	18 862	68.30
45	3162.63	20 086	74.53
50	3251.48	20 623	67.32

Table C.3
600 Tasks.

Number of AGVs	OFV (kWh)	Makespan (s)	CPUT (s)
20	3554.84	29 488	75.95
25	3896.53	28 844	76.92
30	3933.74	26 807	96.43
35	4356.74	28 336	122.29
40	4714.34	29 554	107.48
45	4848.75	29 515	191.72
50	4910.95	28 333	128.84

Table C.4
800 Tasks.

Number of AGVs	OFV (kWh)	Makespan (s)	CPUT (s)
20	4869.87	40 981	89.97
25	5179.88	39 153	99.00
30	5626.65	39 225	140.17
35	6107.26	39 414	158.30
40	6459.42	39 032	114.98
45	6519.78	40 184	127.03
50	6661.21	40 862	134.92

Table C.5
1000 Tasks.

Number of AGVs	OFV (kWh)	Makespan (s)	CPUT (s)
20	6417.13	52 163	109.32
25	6849.26	52 230	122.51
30	7086.01	49 338	128.27
35	7346.72	45 950	124.02
40	8006.45	47 254	139.29
45	8491.91	47 904	133.87
50	8607.21	48 315	106.34

Table C.6
1200 Tasks.

Number of AGVs	OFV (kWh)	Makespan (s)	CPUT (s)
20	7418.18	61 807	172.79
25	8079.05	61 065	174.11
30	8794.53	60 239	159.38
35	9320.95	59 071	163.59
40	10 087.16	60 164	179.76
45	10 120.14	60 676	200.70
50	11 181.50	61 403	221.24

Data availability

Data will be made available on request.

References

Cao, Y., Yang, A., Liu, Y., Zeng, Q., & Chen, Q. (2023). AGV dispatching and bidirectional conflict-free routing problem in automated container terminal. *Computers & Industrial Engineering*, *184*, Article 109611.

Chen, J. (2022). Dynamic path planning of Multi-AGVs in automated terminal based on co-evolution. *International Core Journal of Engineering*, *8*(2), 284–295.

Guo, K., Zhu, J., & Shen, L. (2020). An improved acceleration method based on multi-agent system for AGVs conflict-free path planning in automated terminals. *IEEE Access*, *9*, 3326–3338.

Han, Y., Zheng, H., Ma, W., Yan, B., & Ma, D. (2024). Integrated scheduling of automated rail-mounted gantries and external trucks in U-shaped container terminals. *IEEE Transactions on Automation Science and Engineering*.

Hu, H., Yang, X., Xiao, S., & Wang, F. (2023). Anti-conflict AGV path planning in automated container terminals based on multi-agent reinforcement learning. *International Journal of Production Research*, *61*(1), 65–80.

Kong, L., & Ji, M. (2024). Mathematical modeling and optimizing of yard layout in automated container terminals. *Expert Systems with Applications*, *258*, Article 125117.

Li, X., Peng, Y., Huang, J., Wang, W., & Song, X. (2021). Simulation study on terminal layout in automated container terminals from efficiency, economic and environment perspectives. *Ocean & Coastal Management*, *213*, Article 105882.

Li, H., Peng, J., Wang, X., & Wan, J. (2020). Integrated resource assignment and scheduling optimization with limited critical equipment constraints at an automated container terminal. *IEEE Transactions on Intelligent Transportation Systems*, *22*(12), 7607–7618.

Li, Y., Sun, Z., & Hong, S. (2024). An exact algorithm for multiple-equipment integrated scheduling in an automated container terminal using a double-cycling strategy. *Transportation Research Part E: Logistics and Transportation Review*, *186*, Article 103565.

Li, J., Yan, L., & Xu, B. (2023). Research on multi-equipment cluster scheduling of U-shaped automated terminal yard and railway yard. *Journal of Marine Science and Engineering*, *11*(2), 417.

Liang, C., Zhang, Y., & Dong, L. (2022). A three stage optimal scheduling algorithm for AGV route planning considering collision avoidance under speed control strategy. *Mathematics*, *11*(1), 138.

Liu, W., Zhu, X., Wang, L., & Wang, S. (2023). Multiple equipment scheduling and AGV trajectory generation in U-shaped sea-rail intermodal automated container terminal. *Measurement*, *206*, Article 112262.

Lou, P., Zhong, Y., Hu, J., Fan, C., & Chen, X. (2023). Digital-twin-driven AGV scheduling and routing in automated container terminals. *Mathematics*, *11*(12), 2678.

Lu, Y. (2022). The optimization of automated container terminal scheduling based on proportional fair priority. *Mathematical Problems in Engineering*, *2022*(1), Article 7889048.

Lu, T., Sun, Z. H., Qiu, S., & Ming, X. (2021). Time window based genetic algorithm for multi-AGVs conflict-free path planning in automated container terminals. In *2021 IEEE international conference on industrial engineering and engineering management* (pp. 603–607). IEEE.

Niu, Y., Yu, F., Yao, H., & Yang, Y. (2022). Multi-equipment coordinated scheduling strategy of U-shaped automated container terminal considering energy consumption. *Computers & Industrial Engineering*, *174*, Article 108804.

Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation*, *12*(1), 64–79.

Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, *7*(1), 19–34.

Shouwen, J., Di, L., Zhengrong, C., & Dong, G. (2021). Integrated scheduling in automated container terminals considering AGV conflict-free routing. *Transportation Letters*, *13*(7), 501–513.

Sun, P. Z., You, J., Qiu, S., Wu, E. Q., Xiong, P., Song, A., Zhang, H., & Lu, T. (2022). AGV-based vehicle transportation in automated container terminals: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *24*(1), 341–356.

Tang, K., Zhang, Y., Yang, C., He, L., Zhang, C., & Zhou, W. (2024). Optimization for multi-resource integrated scheduling in the automated container terminal with a parallel layout considering energy-saving. *Advanced Engineering Informatics*, *62*, Article 102660.

Tizhoosh, H. R. (2005). Opposition-based learning: a new scheme for machine intelligence. In *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-iAWTIC'06)*: *vol. 1*, (pp. 695–701). IEEE.

Wang, X., & Jin, Z. (2024). Collaborative optimization of multi-equipment scheduling and intersection point allocation for U-shaped automated sea-rail intermodal container terminals. *IEEE Transactions on Automation Science and Engineering*.

Xu, B., Jie, D., Li, J., Yang, Y., Wen, F., & Song, H. (2021). Integrated scheduling optimization of U-shaped automated container terminal under loading and unloading mode. *Computers & Industrial Engineering*, *162*, Article 107695.

Xu, B., Jie, D., Li, J., Zhou, Y., Wang, H., & Fan, H. (2022). A hybrid dynamic method for conflict-free integrated schedule optimization in U-shaped automated container terminals. *Journal of Marine Science and Engineering*, *10*(9), 1187.

Yan, C., Li, M., Liu, W., & Qi, M. (2020). Improved adaptive genetic algorithm for the vehicle Insurance Fraud Identification Model based on a BP Neural Network. *Theoretical Computer Science*, *817*, 12–23.

Yang, Y., Sun, S., Zhong, M., Feng, J., Wen, F., & Song, H. (2023). A refined collaborative scheduling method for multi-equipment at U-shaped automated container terminals based on rail crane process optimization. *Journal of Marine Science and Engineering*, *11*(3), 605.

Yang, Y., Zhong, M., Dessouky, Y., & Postolache, O. (2018). An integrated scheduling method for AGV routing in automated container terminals. *Computers & Industrial Engineering*, *126*, 482–493.

Yue, L., & Fan, H. (2022). Dynamic scheduling and path planning of automated guided vehicles in automatic container terminal. *IEEE/CAA Journal of Automatica Sinica*, *9*(11), 2005–2019.

Zhang, X., Hong, Z., Xi, H., & Li, J. (2024). Optimizing multiple equipment scheduling for U-shaped automated container terminals considering loading and unloading operations. *Computer-Aided Civil and Infrastructure Engineering*.

Zhong, M., Yang, Y., Dessouky, Y., & Postolache, O. (2020). Multi-AGV scheduling for conflict-free path planning in automated container terminals. *Computers & Industrial Engineering*, *142*, Article 106371.

Zhong, M., Yang, Y., Sun, S., Zhou, Y., Postolache, O., & Ge, Y. E. (2020). Priority-based speed control strategy for automated guided vehicle path planning in automated container terminals. *Transactions of the Institute of Measurement and Control*, *42*(16), 3079–3090.

Zhong, M., Yang, Y., Zhou, Y., & Postolache, O. (2020). Application of hybrid GA-PSO based on intelligent control fuzzy system in the integrated scheduling in automated container terminal. *Journal of Intelligent & Fuzzy Systems*, *39*(2), 1525–1538.