

## Research Article

# Cooperative Scheduling of AGV and ASC in Automation Container Terminal Relay Operation Mode

Qinglei Zhang,<sup>1</sup> Weixin Hu ,<sup>2</sup> Jianguo Duan,<sup>1</sup> and Jiyun Qin<sup>1</sup>

<sup>1</sup>China Institute of FTZ Supply Chain, Shanghai Maritime University, Shanghai 201306, China

<sup>2</sup>Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China

Correspondence should be addressed to Weixin Hu; 2116957588@qq.com

Received 4 May 2020; Revised 18 October 2020; Accepted 2 November 2020; Published 18 January 2021

Academic Editor: Thomas Hanne

Copyright © 2021 Qinglei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The key problem of operation optimization for automated container terminal is the coordinated scheduling of automated quay crane (QC), automated guided vehicle (AGV), and automated stacking crane (ASC). In order to solve this problem, this paper proposed a new method to optimize the scheduling of ASC and AGV. In the automated container terminal, each container block is equipped with twin ASCs. At the same time, buffer zones are set at both ends of the container block to achieve the decoupling operation among ASC, AGV, and container truck. Considering the buffer capacity constraint and twin ASC operation interference, and introducing the design of handshake area, this paper developed a collaborative scheduling model of AGV and ASC in automatic terminal relay operation mode. This model is designed based on the genetic algorithm. The model aims to minimize the AGV waiting time and the ASC running time. The results indicated that the introduction of handshake area can effectively reduce the ASC operation interference by 10.56% on average. Also, it can be found that increasing buffer capacity can reduce the waiting time of AGV by about 4.25% on average, and the effect of relay operation is more obvious in large-scale operation. It was proved that buffer zone and handshake area can reduce task delay time and improve coordination between AGV and ASC.

## 1. Introduction

In recent years, with rapid increase in container tasks and continuous expansion of ship scale, the limited handling capacity of traditional container terminals can hardly meet the growing business demand. Therefore, safe, intelligent, and green automated terminals are becoming a trend in the development of container terminals.

Figure 1 presents a schematic diagram of an automated container terminal. As shown in Figure 1, the automated container terminal includes three types of automated operation equipment: quay cranes (QCs), automated guided vehicles (AGVs), and automated stacking cranes (ASCs). In the automatic terminal, the QCs are located at the quayside, used for loading containers onto ships and unloading containers from ships. The AGVs are assigned the path by the vehicle control system (VCS) to transfer containers between the quayside and the container yard. ASCs are used for handling containers in the container yard.

In addition, the automated container terminal usually sets a buffer zone (i.e., I/O point support area or container exchange area) on both sides of the container block. The buffer zone is used for loading and unloading the target container with the AGVs and container trucks. This area can avoid the AGVs and the container trucks entering the container block, which help to reduce the operation time.

Figure 2 presents the transport process of the container at the port. In the automatic container terminal, containers stored in the yard can be divided into three categories: import containers, export containers, and transfer containers. There are two operation modes for containers: the loading process and the unloading process.

In the unloading process, the containers are first arrived at the destination port along with the container ships. After receiving the unloading instruction, the QC grabs the container and places it on the surface of AGV firstly. Then, the target container is transported to I/O point by the AGV and temporarily stored on the AGV-mate. The AGV-mate is

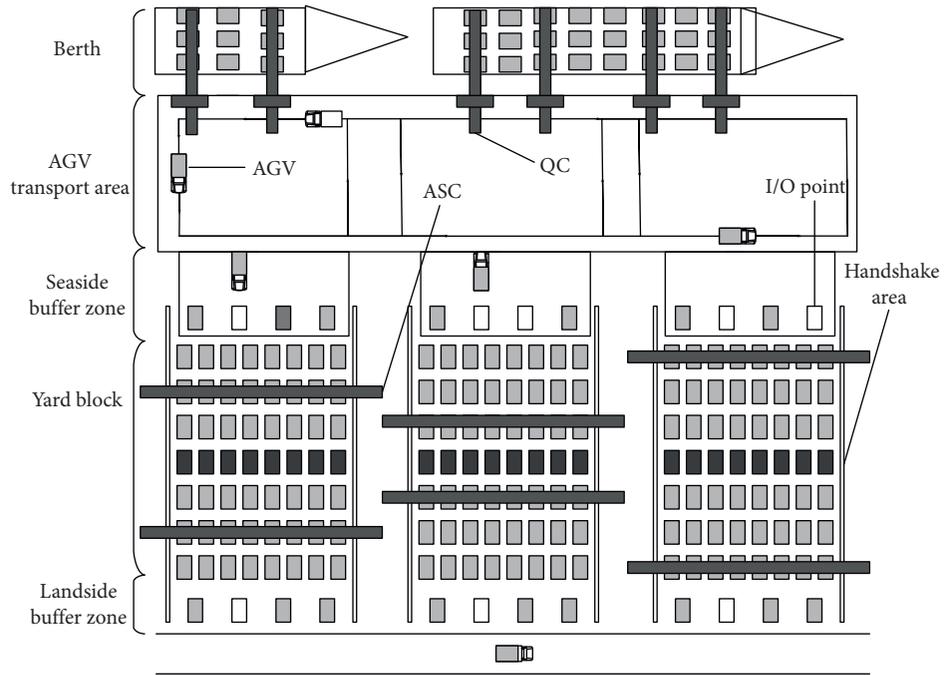


FIGURE 1: Typical layout of automation container terminal, mainly including berth, automated guide vehicle (AGV) transportation area, seaside buffer zone, yard blocks, and landside buffer zone.

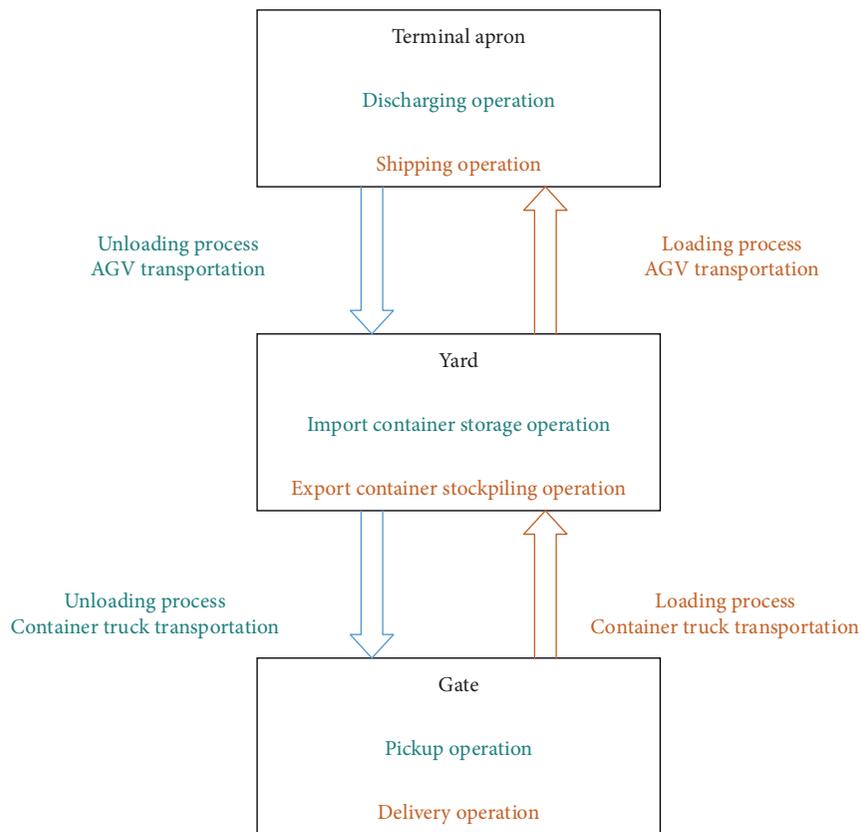


FIGURE 2: Flowchart of a port import and export operation process. The blue font represents the unloading process, and the brown font represents the loading process.

an auxiliary equipment of AGVs and located in the buffer zone. This equipment is used for reducing the waiting times associated with ASC and AGV. When the ASC receives the instruction, it grabs the container from the AGV-mate and places it on the designated location of the yard, waiting for customers to pick up.

In contrast, in the loading process, the containers are first transported to the storage yard by the container truck arranged by the customer, and then stored. After the destination ship arrives at the port for docking, ASC receives the shipping instruction, grabs the container from the storage yard, and places it on the AGV-mate. Then, the container is waiting for the AGV taking away, transporting to the corresponding berth. Finally, the container is loaded onto the ship by the QC.

The loading and unloading operation of the automated container terminal is important links in the operation management of the automated terminal. The efficiency of the loading and unloading operation largely depends on the efficiency of collaboration between various kinds of equipment. At the same time, unreasonable allocation and scheduling plan will cause low utilization rate of container terminal resources and operation delay. The aim of the research on collaborative scheduling optimization of loading and unloading resources in container terminals is to schedule and allocate resources for each ship in a reasonable and efficient method. This can help to accomplish containers loading and unloading tasks on ships in a coordinated manner, effectively avoid the waiting time between equipment, and finally improve the utilization rate of container terminal resources.

In recent years, the storage yard, limited by operation resources such as storage space, yard crane (YC), AGV, etc., is becoming a new bottle neck in port operation. For ASCs and AGVs, how to achieve the cooperative scheduling between ASCs and AGVs is of great significance for reducing equipment waiting time of and improving terminal operation efficiency. Therefore, combined with the actual situation of automated terminals, this paper studies the cooperative scheduling problem of AGVs and twin ASCs in automation container terminal relay operation mode. The purpose is to improve the coordination between different devices and optimize the job sequence of AGV and ASC, so as to minimize the waiting time of AGV and the running time of ASC.

The main contribution of this study is to provide an optimization method to deal with ASC and AGV synthetically for the cooperative scheduling problem of AGVs and ASCs, which considers the loading and unloading operations and the interferences between the ASCs. In addition, we also provide a three-stage genetic algorithm to solve the collaborative scheduling problem.

## 2. Literature Review

After the AGV was put into application in 1955, there are lots of previous research studies about AGV scheduling. Durrant-Whyte [1] firstly studied the application of AGV in container terminals, and proposed an AGV system for port

transportation of the International Organization for Standardization (ISO) standard cargo containers. Kim and Bae [2] proposed a look-ahead dispatching method for AGVs reducing the delay of ship operation. Bish et al. [3] developed an inverse greedy algorithm to optimize the travel path for AGVs importing and exporting containers. Angeloudis and Bell [4] studied the allocation of automatic navigation operations at container terminals under various uncertain conditions (such as emergency maintenance operation, safety stop, mislabeling, and container loss), and proposed an AGV scheduling method, which can carry out real-time control under uncertain conditions. Cai et al. [5] proposed a Rescheduling New arrival Jobs (RNJ) policy and a Rescheduling Combination of new and unexecuted Jobs (RCJ) policy to solve the problem of autonomous straddle carriers scheduling under the background of uncertainty (such as arrival of new tasks at automated terminals or occurrence of emergencies). Radhia et al. [6] proposed a hybrid method based on Dijkstra algorithm, genetic algorithm, and heuristic algorithm to study the task assignment problem of AGVs. Malopolski [7] developed a new method to describe one-way, two-way or multilane flow field for AGV collision and deadlock problems.

For the twin ASC scheduling study, Nguyen and Kim [8] established a mixed integer programming model to solve the priority loading and unloading order of the target container based on the buffer capacity constraint and time window constraint. Saini et al. [9] constructed a two-level random model to analyze the performance of twin YCs. It was found that the interference effect between cranes would reduce the handling capacity of cranes by about 35%, and the interference delay increase with the number of shelves in the stack. Briskorn and Angeloudis [10] developed an effective graphic model and strong polynomial algorithm to solve the scheduling problem of twin cranes and cross cranes in the automatic terminal. Lu and Wang [11] established an interference model by analyzing the time overlap among different container handling tasks, and designed a particle swarm optimization algorithm based on the graph theory model to solve the scheduling problem of twin ASCs. Hu et al. [12] developed an interference relationship matrix between twin ASCs by analyzing the minimum time interval among different container handling tasks. Based on this matrix, two mixed integer linear programming (MILP) models and a nonlinear model are developed to solve the scheduling problem of twin ASCs without interference.

With the deepening of the research on each subsystem of the port, the demand for collaborative scheduling is becoming more and more obvious in both the academic circle and port enterprises. Integrated scheduling takes the optimization of each link as a part of the whole system, and improves the operation efficiency of the whole system through the seamless connection between links. Different from integrated scheduling, the core of collaborative scheduling is to consider the interaction between different systems, and to achieve the overall system efficiency through the coordination among subsystems. Meersmans and Wagelmans [13] tried to integrate the scheduling of AGVs, QCs, and YCs into the automatic container terminal. A

branch and bound algorithm and a beam search algorithm were developed to minimize the total scheduling span. Xinet al. [14] proposed a method of dispatching QCs, AGVs, and YCs to improve the loading and unloading capacity of automated container terminals through energy saving. Luo and Wu [15] proposed an evolutionary heuristic algorithm (the genetic algorithm) to solve the large-scale problem. Luo et al. [16] studied the integration of AGV scheduling and container storage to optimize the unloading process of the automated container terminal. Hu et al. [17] developed two MILP models and developed a three-stage decomposition method based on particle swarm optimization to solve the joint problem of vehicle scheduling and storage distribution in automated container terminal. Zhong et al. [18] developed the integrated scheduling of the QCs, AGVs, and ASCs in automated terminals. Various basic metaheuristic and improved hybrid algorithms were developed to optimize service level. Zhao et al. [19] considered the capacity limitation of the transfer platform on QCs, established collaborative scheduling model for QCs and AGV. The optimal solution to the model is obtained by feedback from the two-stage taboo search process.

For the research of integrated scheduling of traditional terminals, it is mainly the coordinated scheduling of container trucks, QCs, and YCs. Kim et al. [20] grouped the truck in dynamic situation, so as to cooperate with YC operation, improve the working efficiency of YCs, and reduce the interference among different YCs. Lee et al. [21] studied the joint scheduling problem of YCs and container trucks, and solved this problem by the simulated annealing algorithm. Petersing et al. [22] considered the working efficiency of the whole system, implemented a global expansion of local problems, and made the research of the problem more integrated. In addition, some previous studies focus on the collaborative scheduling of quayside resources. Imai et al. [23] studied the joint scheduling problem of berth allocation and QC allocation. By using the genetic algorithm, a satisfactory solution has been obtained. Liang et al. [24] established a joint optimization model of berth allocation and QC scheduling to minimize the processing time, waiting time, and delay time of each ship.

From previous studies, it was found that the cooperative scheduling of the ASC and the AGV in the automated terminal has not been fully explored. Previous studies mainly focus on the collaborative problem of vehicle scheduling, storage allocation in the automated terminal, and collaborative problem of berth allocation and QC scheduling. In most of the existing literatures on cooperative scheduling of QCs, AGVs, and ASCs, the cooperation between the ASCs is ignored, and the landside ASC passively caters to the operation of the seaside ASC.

In addition, previous studies on collaborative scheduling of loading and unloading equipment in automated container terminals have the following problems. Firstly, the operation requirements of landside ASC have not been considered, which separated the cooperative relationship between the twin ASCs, resulting in the operation delay of landside ASC. Secondly, ASCs are often regarded as a kind of auxiliary equipment, which requires the ASCs to follow the operation

order of QCs and give priority to the quayside operation requirements, while ignoring the ability of ASC reasonable scheduling to improve the operation efficiency of the whole terminal.

Therefore, this paper will take ASC as the main research object, focusing on the cooperative scheduling of ASC and AGV under the condition of limited buffer capacity, and twin ASCs interfere with each other. Combined with the actual situation of the automatic terminal, this paper adopts the dual-cycle strategy (loading operation and unloading operation are considered at the same time) and establishes the cooperative scheduling model of AGV and ASC under the ASC relay operation mode of automatic container terminal.

### 3. Mathematical Model

*3.1. Problem Description.* Yard, as the place for container changing, plays the key role in the automatic terminal. The storage yard consists of several blocks, and a buffer zone is set on both sides of the storage yard, which is used for the container handover between AGVs and ASCs. In the loading operation, the AGVs can be allowed to leave, after AGVs arrive at the buffer zone and are loaded the container by the AGV-mate. In the unloading operation, the AGVs can be allowed to leave without waiting for ASCs, after AGVs arrive at the buffer zone and unloaded the container by the AGV-mate. However, when there is no free buffer area, AGVs must wait outside the buffer. This would have a negative effect on the completion of tasks and the service time of subsequent tasks.

In the automatic container terminal, the loading and unloading processes are carried out simultaneously. In order to ensure that the model proposed is more accurate and closer to the reality, the dual-cycle strategy is adopted in this paper. In the dual-cycle operations, there are four transportation modes for an AGV completing two consecutive deliveries:

- (i) Mode 1: the AGV performs the import container after completing the import container
- (ii) Mode 2: the AGV performs the export container after completing the import container
- (iii) Mode 3: the AGV transports the import container after completing the export container
- (iv) Mode 4: the AGV transports the export container after completing the export container

Figure 3 presents the preparation process between two consecutive deliveries of AGV under different operation modes. In Figure 3, the solid line represents the heavy-load process of AGV, the dotted line represents the no-load process of AGV, and the horizontal line represents the moving process of AGV between the two QCs or two container blocks.

Having described the operation process of the AGV, the ASC operation process is needed to be discussed. In the automatic container terminal, each container block usually has two ASCs, and each ASC can only operate in its own half

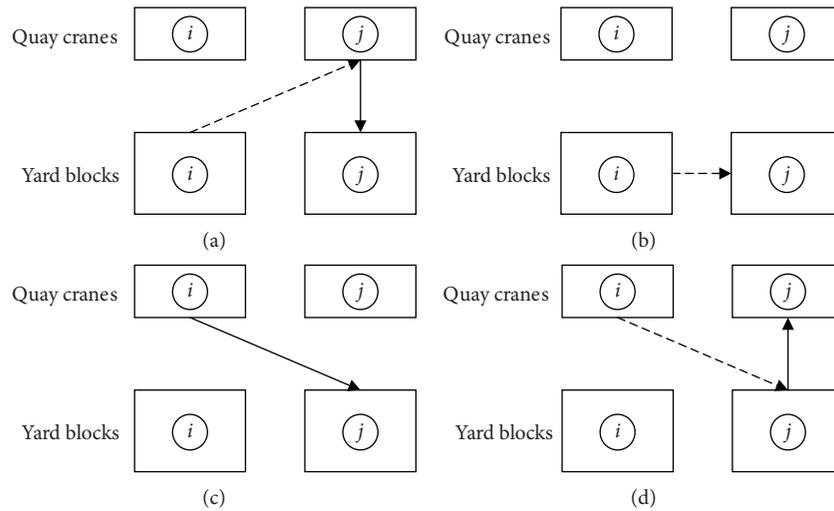


FIGURE 3: Continuous operation flowchart of AGV in the dual-cycle operations: (a) mode 1; (b) mode 2; (c) mode 3; (d) mode 4.

container area. Since twin ASCs of the same model (mechanical performance, specification, and size) are running on the same track, twin ASCs often interfere with each other. For example, when the container near the landside needs to be exited, the landside ASC needs to wait outside the safety zone. After the seaside ASC completes its loading work, the landside ASC can continue its work. This would greatly reduce the work efficiency of twin ASCs. Therefore, the operation of some remote containers usually needs twin ASC relay to complete.

In order to further effectively alleviate the operation interference of twin ASCs, handshake area design is introduced in this paper. The handshake area is attached to the rails of the cranes, which is shown in Figure 4. In addition, the following rules are adopted in the scheduling process of twin ASCs: with the yard center as the boundary, container tasks near the seaside are independently completed by the seaside ASC, and container tasks near the landside are independently completed by the landside ASC. For container task needed to cross over handshake area, this container task is split into the main task and the secondary task according to the sequence of operation. This task is completed by the ASC relay on both sides.

Figure 4 presents the ASC relay operation process. In this study, the relay operation mode is divided into two stages. The first stage is that one ASC transports the container from the initial position to the temporary position in the handshake area. The second stage is that the other ASC transports the container from the temporary position in the handshake area to the target position of the container. When the AGV carries the container into the buffer zone, the seaside ASC loads the container from the buffer zone and unloads it to the temporary location of the handshake area. Then, the landside ASC loads the container from the temporary location in the handshake area and unloads it to the storage location of the container.

The dual-cycle operations are considered in this paper. In the dual-cycle operation, unloading and loading

operations are not necessary taken place at the same time in the same cycle. During the unloading process, the container is first unloaded on the AGV by the QC, and the container is transported to the buffer zone by the AGV. If the buffer zone capacity is not full, the AGV enters the buffer zone and unloads the container; otherwise, the AGV waits. After the AGV completes the import container, if the container needs relay operation, first, the seaside ASC will load the container from the buffer zone and unload it to the handshake area. Then, the landside ASC will load the container from the handshake area and unload it to the designated storage location. Otherwise, the seaside ASC will complete the container independently. At this time, AGV performs the next container until all containers have been completed. The loading process is executed in the reverse order of the unloading process.

3.2. Model Hypothesis. Considering the actual situation of the automated terminal, each container to be operated is regarded as a task, and the following assumptions are proposed:

- (1) AGVs travel in the same direction at a constant speed and choose a variety of routes, without considering the mutual interference in AGV transport
- (2) Storage yard location of each task container has been given
- (3) AGVs can only transport one container at a time
- (4) Loading and unloading time of ASC in buffer zone is ignored, and the processing time of container task remains constant
- (5) The AGV does not have to wait for the quayside crane operation, which can be operated immediately upon the arrival of the AGV transporting container
- (6) Twin ASCs cannot cross each other, and the case of overturning has not been considered

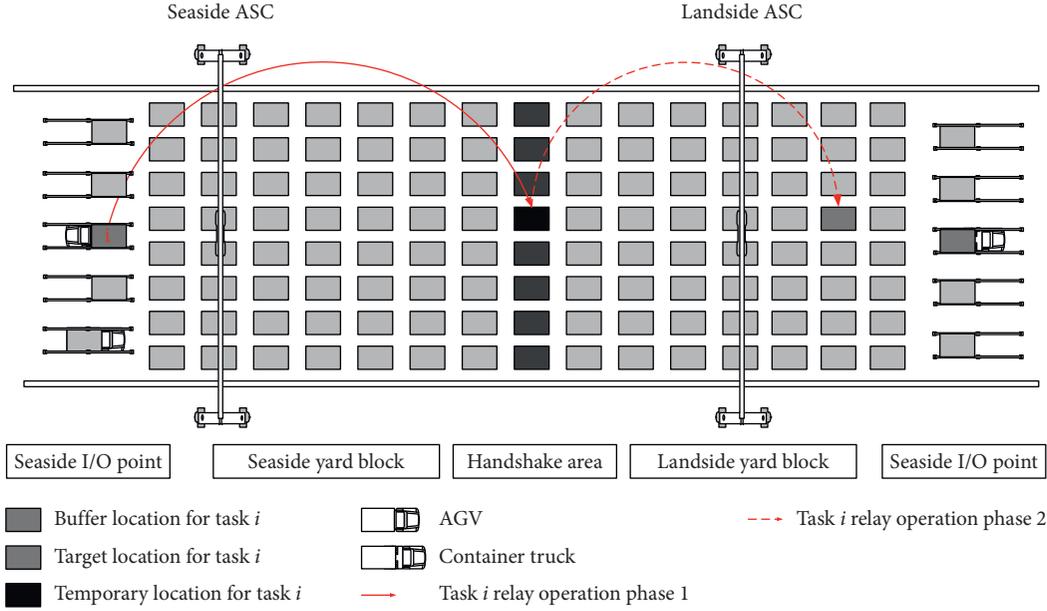


FIGURE 4: Schematic diagram of ASC relay operation.

3.3. *Model Parameters.* To facilitate modelling, the following parameters are defined:

3.4. *Objective Function.* The job scheduling of AGV is the key problem for the terminal operation system. Its scheduling will be restricted by other (such as QC and ASC) links and factors in the system. In the research of the AGV scheduling optimization for the automatic terminal, it usually aims to achieve the shortest waiting time. At the same time, in the operation of automatic terminal, the waiting time of the AGV is also the key factor for the twin ASC scheduling. Therefore, this paper chooses the AGV waiting time and the ASC running time as optimization objectives to improve the overall efficiency of the automation terminal.

In the unloading operation, when the AGV reaches the buffer zone, it needs to judge whether there is residual capacity in the buffer. If there is no residual capacity, the AGV needs to wait in the queue. Similarly, in the loading operation, when ASC completes the task, it needs to judge whether there is residual capacity in the buffer. If there is no residual capacity, ASC needs to wait. However, the waiting of AGV and ASC will affect the completion of the task and the subsequent task service time; if AGV-mate is reserved for each task in the buffer zone, the waiting time of other AGVs will be increased. Therefore, this paper calculates the time window of each container in the buffer zone, and schedules AGV flexibly through the time window to ensure the efficient operation of AGV.

Equation (1) is the objective function for minimizing the waiting time of AGV and the running time of ASC:

$$\min F = f_1 + f_2, \quad (1)$$

where  $f_1$  represents the sum of the AGV waiting times for each task phase, including the waiting time of AGV at the

time of import and export;  $f_2$  indicates the longest running time in twin ASCs to ensure all tasks of ASC are completed.

Equation (2) shows the sum of the differences between the time when all AGVs leave the I/O point support area in the planned time and the time when they arrive at the seaside I/O point support area. This is the sum of the waiting time of AGVs in the whole planned time:

$$f_1 = \sum_{i=1}^U m_i u_i^2 (Z_i - T_i^h) + \sum_{i=1}^L c_i u_i^1 (T_i^E - T_i^h), \quad (2)$$

where  $\sum_{i=1}^U m_i u_i^2 (Z_i - T_i^h)$  indicates the waiting time of AGV at the time of export and  $\sum_{i=1}^L c_i u_i^1 (T_i^E - T_i^h)$  indicates the waiting time of AGV at the time of import:

$$f_2 = \max_{a \in A} \sum_{i=1}^n \left\{ 2t_i + t_{i-1i} + \frac{\alpha [n_i ((B+1)/2 - h_i^a) + (1-n_i)h_i^a]}{V_s} + \frac{(1-\alpha) [n_i (h_i^a - (B+1)/2) + (1-n_i)(|B| - h_i^a)]}{V_s} \right\}. \quad (3)$$

Equation (3) shows ASC scheduling time. When relay operation is required in  $i$  task, the running time of ASC is determined by the sum of ASC loading and unloading time, ASC moving time from the end position of the previous task to the starting position of the current task, and ASC loading running time from the starting position to the relay position or ASC loading running time from the relay position to the starting position. If task  $i$  does not require relay operation, the ASC running time is determined by the sum of the ASC loading and unloading time in the task, the ASC moving time from the end position of the previous task to the starting position of the current task, and the running time when the ASC loads the task to the target location. In the model, we call the starting position and the end position as the target loading and unloading bay of ASC.

---

<i>Sets</i>	
$L$	The set of import containers
$U$	The set of export containers
$N$	The set of all the containers, i.e., $N = L \cup U$
$R_1$	The set of main tasks requiring relay operation
$R_2$	The set of secondary tasks requiring relay operation
$R$	The set of tasks requiring relay operation
$B$	The set of bays, $B = \{1, 2, \dots,  B \}$
$G_{it}$	The set of tasks waiting for operation in the buffer at the time $t$ in $i$ task
$K$	The set of AGVs
$A$	The set of ASCs in container block, $A = \{1, 2\}$
<i>Variables</i>	
$T_i^s$	The start time of task $i$ , when the QC unloads the container onto the AGV
$T_i^h$	The arrival time of AGV transporting task $i$ to buffer
$T_i^E$	The time when task $i$ is unloaded to the AGV-mate
$W_i$	The time window of task $i$ is stored in the buffer
$d_{ij}$	The preparation time for loading task $j$ to the buffer after the AGV completes task $i$
$q$	The loading and unloading time of a container by the QC
$d_0$	The no-load driving time of the AGV
$d_1$	The heavy-load driving time of the AGV
$T_{gi}$	The waiting time of the AGV $g$ in the $i$ task
$T_{max}$	AGV maximum waiting time
$C_i$	Remaining available capacity of buffer in $i$ task
$c_i$	A 0-1 variable, which equals 1 when there is no remaining available capacity in $i$ task, otherwise 0
$m_i$	A 0-1 variable, during the loading process, which equals 1 when AGVs arrive at buffer zone and wait for ASC to complete task $i$ , otherwise 0
$h_i^1$	Target loading and unloading bay of seaside ASC in $i$ task
$h_i^2$	Target loading and unloading bay of landside ASC in $i$ task
$V_s$	Number of bays passed by ASC per unit time
$t_i$	The dropping container or picking container time of ASC in $i$ task
$t_{ij}$	The ASC moving time from the end position of task $i$ to the start position of task $j$
$Z_i^a$	The completion time of ASC $a$ in $i$ task
$u_i^a$	A 0-1 variable, which equals 1 when the container is loaded and unloaded by ASC $a$ at the time $t$ in the $i$ task, otherwise 0
$g_{i,b}^a$	In the $i$ task, at time $t$ , ASC $a$ is in the bay $b$
$\tau$	Safe distance between twin ASCs
$M$	A large enough positive number
$\alpha$	A 0-1 variable, which equals 1 when $a = 1$ , otherwise 0
$n_i$	A 0-1 variable, which equals 1 when relay operation is selected in task $i$ , otherwise 0
<i>Decision variables</i>	
$X_{ij}$	A 0-1 variable, which equals 1 when the same AGV executes the next task $j$ after the AGV completed task $i$ , otherwise 0
$Y_{ij}^a$	A 0-1 variable, which equals 1 when the same ASC executes the next task $j$ after the ASC completed task $i$ , otherwise 0

---

3.5. Constraint Condition.

$$T_j^h - (T_i^h + d_{ij}) \geq M(X_{ij} - 1), \quad i, j \in N, \quad (9)$$

$$\sum_{i=1}^n X_{ij} = 1, \quad \forall j \in N, \quad (4)$$

$$\sum_{j=1}^n X_{ij} = 1, \quad \forall i \in N, \quad (5)$$

$$\sum_{i=1}^n Y_{ij}^a = 1, \quad \forall j \in N, \forall a \in A, \quad (6)$$

$$\sum_{j=1}^n Y_{ij}^a = 1, \quad \forall i \in N, \forall a \in A, \quad (7)$$

$$T_{gi} \leq T_{max}, \quad \forall i \in N, \quad (8)$$

$$d_{ij} = \begin{cases} d_0 + d_1 + q, & \text{mode 1,} \\ 0, & \text{mode 2,} \\ q + d_1, & \text{mode 3,} \\ d_0, & \text{mode 4,} \end{cases} \quad (10)$$

$$\sum_N Z_i^1 \geq |N|, \quad i \in N, \quad (11)$$

$$T_{gi} \geq MC_i, T_{gi} > 0, \quad i \in L, \quad (12)$$

$$T_{gi} \geq Mm_i, T_{gi} > 0, \quad i \in U, \quad (13)$$

$$M(1 - c_i) \geq C_i, \quad i \in N, \quad (14)$$

$$T_i^E \in W_i = \left[ (c_i \min\{Z_{j-1} + t_{j-1}\} + (1 - c_i)T_i^h), Z_{i-1} \right], \\ j < i, j \in G_t, i \in L, \quad (15)$$

$$T_i^E \in W_i = \left[ Z_i, \min\{Z_j, T_i^h\} \right], \quad i < j, j \in L \cup U, i \in U, \quad (16)$$

$$g_{t,b}^1 < g_{t,b'}^2, b' \neq b, b, b' \in B, \quad i \in N, a \in A, \quad (17)$$

$$\sum_N u_{t_i}^a \leq |N|, \quad i \in N, a \in A, \quad (18)$$

$$g_{t,b}^a \neq g_{t,b'}^a, \quad b' \neq b, b, b' \in B, i \in N, a \in A, \quad (19)$$

$$M(1 - g_{t,b}^a) \leq u_{t_i}^a, \quad b \in B, i \in N, a \in A, \quad (20)$$

$$Z_j^1 \geq \sum_{j=1}^n 2t_j + t_{ij} + \frac{[n_j((B+1)/2 - h_j^1) + (1 - n_j)h_j^1]}{V_s}, \\ i, j \in N, \quad (21)$$

$$Z_j^2 \geq \sum_{j=1}^n 2t_j + t_{ij} \\ + \frac{[n_j(h_j^2 - (B+1)/2) + (1 - n_j)(|B| - h_j^2)]}{V_s}, \\ i, j \in N, \quad (22)$$

$$Z_j^1 \geq 2t_j + Z_i^1 + t_{ij} + M(1 - Y_{ij}^1) \\ + \frac{[n_j((B+1)/2 - h_j^1) + (1 - n_j)h_j^1]}{V_s}, \quad i, j \in N, \quad (23)$$

$$Z_j^2 \geq 2t_j + Z_i^2 + t_{ij} + M(1 - Y_{ij}^2) \\ + \frac{[n_j(h_j^2 - (B+1)/2) + (1 - n_j)(|B| - h_j^2)]}{V_s}, \quad i, j \in N, \quad (24)$$

$$Z_i^1 \leq Z_j^2 - \left( 2t_j + t_{ij} + \frac{(h_j^2 - (B+1)/2)}{V_s} \right), \\ i \in R_1, j \in R_2, i, j \in R_1 \cap R_2, l \in N, \quad (25)$$

$$Z_i^2 \leq Z_j^1 - \left( 2t_j + t_{ij} + \frac{((B+1)/2 - h_j^1)}{V_s} \right), \\ i \in R_1, j \in R_2, i, j \in R_1 \cap R_2, l \in N, \quad (26)$$

$$\sum_N n_i \leq |N|, \quad i \in N. \quad (27)$$

Constraint (4) indicates that AGV has one container transportation task before starting the container

transportation task. Constraint (5) indicates that AGV has one container transportation task after completing one container transportation task. Constraint (6) indicates that ASC has one container loading or unloading task before starting to perform the container loading or unloading task. Constraint (7) indicates that ASC has one container loading or unloading task after completing one container loading or unloading task. Constraint (8) indicates that the waiting time of AGV is less than the maximum waiting time. Constraint (9) shows the time relationship between two consecutive tasks transported by the same AGV and arriving at the buffer. Constraint (10) indicates the preparation time for loading task  $j$  to the buffer after the AGV completed task  $i$ , and there are four cases. Constraint (11) ensures that the target container in the unloading operation of the marine vessel can be unloaded and enter the target shell position. Constraint (12) ensures that the remaining available capacity of the support area must be 0 at the current time when the AGV is waiting in the seaside I/O point support area during the unloading operation. Constraint (13) ensures that the current task is not completed when the AGV is waiting in the seaside I/O point support area during the packing operation. Equation (14) constrains the set 0-1 variable and shows that when there is no residual available capacity in the seaside I/O point support area,  $c_i = 1$ . Constraint (15) indicates that, during unloading operation, the earliest start time of the time window of task  $i$  is the earliest time when the task  $i$  is unloaded in the buffer by AGV, and the latest start time is the time when the task  $i$  is loaded from the buffer by ASC. Constraint (16) indicates that, during loading operation, the earliest start time of the time window of task  $i$  is the earliest time when the task  $i$  is unloaded in the buffer by ASC, and the latest start time is the time when the task  $i$  is loaded from the buffer by AGV. Constraint (17) ensures that ASCs do not collide with each other and there is a safe distance  $\tau$  between the twin ASCs. Constraint (18) indicates that ASCs can only perform loading and unloading tasks once at most. Constraint (19) indicates that ASCs can only be in one bay at a certain time. Constraint (20) ensures that the ASC is in the target bay corresponding to the task during loading and unloading. Constraints (21) and (22), respectively, indicate that the actual time for seaside and landside ASC to complete task  $i$  must be greater than or equal to the current task planned completion time. Constraints (23) and (24) show the time sequence relationship of the two adjacent tasks of the seaside and the landside ASC, respectively. Constraints (25) and (26) indicate the time sequence relationship between the main task and the secondary task of the container requiring relay operation. Constraint (27) indicates that the number of relay operations will not be greater than the total number of loading and unloading tasks.

#### 4. Algorithm Design

The optimization of comprehensive scheduling of container handling and transportation equipment is critical for port operation management. However, the integrated scheduling is an NP-hard problem. With the problem size increases, the computational complexity will increase exponentially. The

genetic algorithm can control the collection process adaptively to obtain the approximate optimal solution. It has obvious advantages in solving such NP-hard problems. Therefore, a meta heuristic algorithm (i.e., the genetic algorithm) is proposed to find the best solution of the integrated scheduling problem. The flow of the genetic algorithm is as shown in Figure 5.

In the cooperative scheduling problem of ASC and AGV, the following problems need to be solved: the problem of ASC job interference, the determination of the time window for each unloading and loading task, and the problem of job sequence conflict between ASC and AGV. Based on the above problems, a three-stage genetic algorithm is designed to solve the cooperative scheduling problem of AGV and ASC in automatic container terminals.

The first stage is the scheduling of ASC, which mainly solves the job interference in the plan by avoiding, and obtains a group of feasible ASC scheduling schemes. The second stage is the scheduling of AGV. According to the scheduling plan of the first stage, the optimal AGV scheduling scheme and the time window of each unloading and loading task are searched. The upper limit of the time window is the time when the task is stored in the AGV-mate, and the lower limit is the time when the task is loaded out of the buffer zone. Due to the limited buffer capacity, AGV may need to wait in queue. If  $T_{gi} \geq T_{\max}$ , the job sequence of ASC will be adjusted, and the time window of this task and subsequent tasks will be rearranged. The third stage is the calculation of fitness function. Due to the conflict between ASCs and AGVs, not every chromosome can provide a feasible solution. Therefore, in this stage, the infeasible solution is repaired through cycle iteration. As shown in Figure 6, first of all, the scheduling plans of ASC in the first and second stages are compared to judge whether it is feasible solution. If not, it finds the conflict task  $i$  through triple detection, gets the feasible scheme of the task  $i$  and the preceding task through cyclic iteration, and then traverses all the subsequent tasks to get the final feasible solution. The coordinated scheduling scheme of ASC and AGV is further optimized through the search process of the three-stage genetic algorithm.

**4.1. Chromosome Encoding and Decoding.** In this paper, multilayer chromosome and real number coding are used for coding. An arrangement and combination of all task numbers are taken as the first-layer chromosome which represents the operation sequence of the tasks. The second-layer chromosome is an ASC segment, and the gene in the ASC segment corresponds to the ASC number. The third-layer chromosome is an AGV segment, and the gene in the AGV segment corresponds to the AGV number.

Figure 7 shows an example of this coding system, where the chromosome length is  $N + R_1$ . When numbering the task sequence, it is necessary to split the tasks that need relay operation according to the splitting principle. Assuming that the initial task scale is  $n$ , task  $i$  and  $n + i$  represent the primary and secondary task separated from the same initial task  $i$ . For example, when  $n = 4$ , task 2 and 3 are relay tasks,

and handshake bay is 11, then task sequence after splitting is (1, 2, 3, 4, 6, 7), as shown in Table 1.

In present paper, it is assumed that there are 2 ASCs, and each ASC has 6 container tasks. It is also made an assumption that there are 8 relay tasks, and 3 AGVs are used for horizontal transportation. In the chromosome coding system, the gene of ASC and AGV segments corresponds to the initial task, and the corresponding container operation sequence of ASC and AGV can be obtained by decoding in turn according to the task sequence in the task segment. According to Figure 7, the decoding results show that the job sequence of seaside ASC is (1, 3, 4, 6, 8, 10, 14, 17, 19, 23); the job sequence of landside ASC is (2, 5, 7, 9, 11, 12, 13, 16, 20, 22); the job sequence of agv1 is (4, 14, 23); the job sequence of agv2 is (3, 10, 17, 19); and the job sequence of agv3 is (1, 6, 8).

**4.2. Initial Population.** In this paper, the heuristic method is used to generate the initial population, which makes the generated individuals closer to the optimal solution and the evolutionary effect is better:

- (i) Step 1: an initial solution is generated randomly, the seaside ASC gene segment  $\{y_1^1, y_2^1, \dots, y_j^1\}$  and landside ASC gene segment  $\{y_1^2, y_2^2, \dots, y_v^2\}$  are proposed, and then the gene position of the first and second stages of the relay task and the difference  $d$  between two positions are calculated.
- (ii) Step 2: it is judged that whether the difference  $d$  is greater than  $\theta$ ,  $\theta \in [2, N - R]$ . If  $d < \theta$ , the first stage of the relay task is searched forward, and a gene locus is randomly selected and exchanged. The second stage of the relay task is searched backward. A gene locus is randomly selected and exchanged until  $d > \theta$ .
- (iii) Step 3: the remaining relay tasks were selected in turn until all relay tasks meet the conditions, and then a chromosome can be obtained. The above steps are repeated for NP times to generate an initial population of size of NP. The generation process of initial population is shown in Figure 8, where  $\theta = 2$ , and task 4 and task 16 are the first and second phases of the relay task, respectively.

**4.3. Fitness Function and Selection.** In this paper, the sum of the minimum waiting time of AGV and the completion time of twin ASCs is used as the fitness value, and the better individuals are selected by the roulette wheel sampling method for crossover and mutation. Generally speaking, the fitness function is transformed from the objective function, and the selection of fitness function directly affects the convergence speed of the genetic algorithm and whether the optimal solution can be found. The fitness function is given as

$$\text{Fit}(f(x)) = \begin{cases} c_{\max} - f(x), & f(x) < c_{\max}, \\ 0, & \text{others,} \end{cases} \quad (28)$$

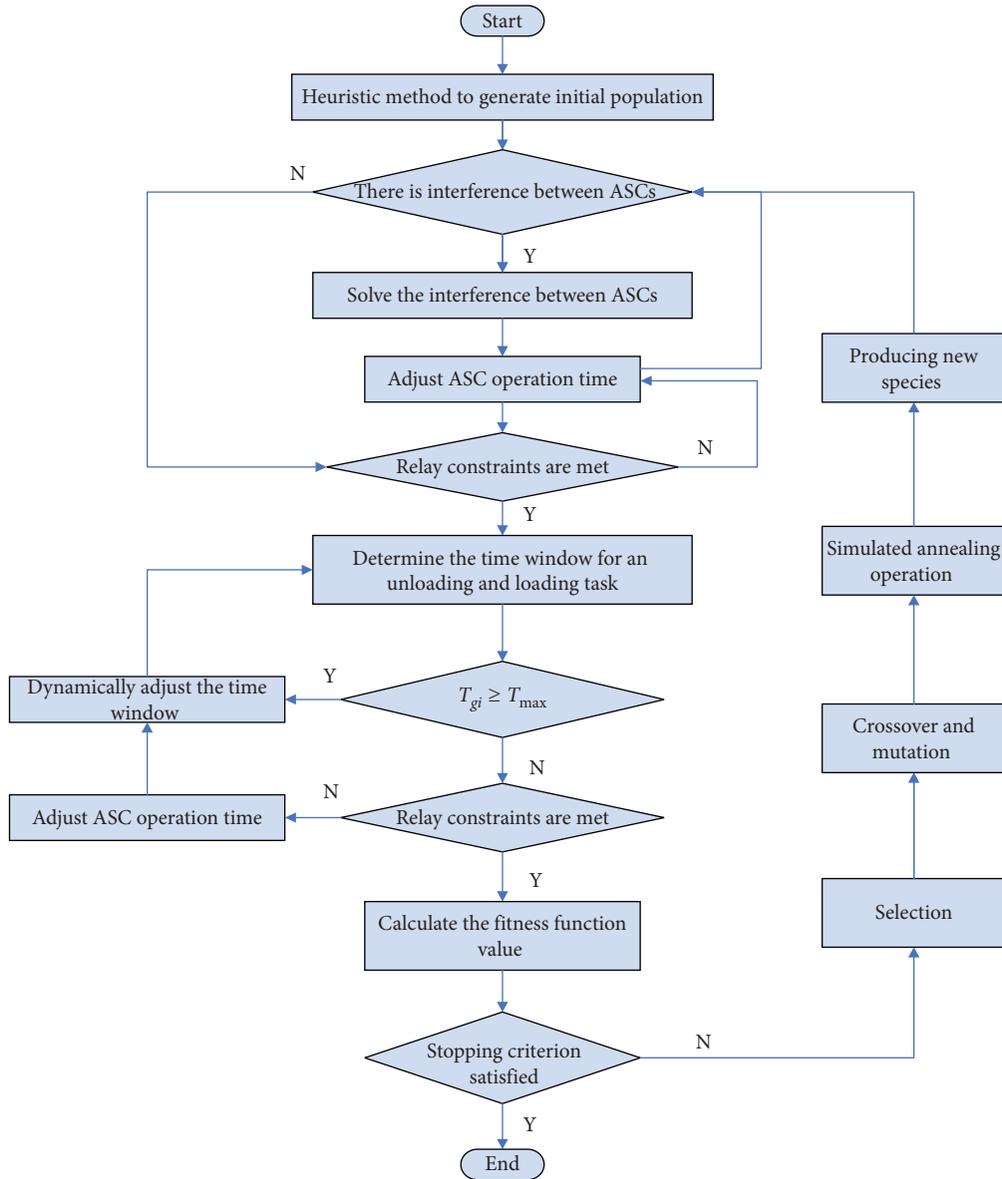


FIGURE 5: Flowchart of the genetic algorithm.

where  $c_{\max}$  is the maximum estimate of  $f(x)$ .

**4.4. Crossover and Mutation.** This paper uses a two-point crossover method to cross the chromosome population. Since the tasks served by ASC are known after task splitting, the whole chromosome is first decomposed into 2 parts according to the ASC number to determine the parent chromosome of crossover operation, and then crossover is performed in two groups. This means that only the AGV segment and task segment need to be crossed. It can be seen from Figure 9 that the number of tasks for seaside ASC after chromosome grouping is 10. According to the crossing principle, two random integers  $r_1$  and  $r_2$  in the  $[1, 10]$  interval are generated. The data between  $r_1$  and  $r_2$  are cross processed, and the unreasonable chromosome after cross is genetically repaired. The process of chromosome crossing

is shown in Figure 9, where  $r_1$  and  $r_2$  are 4 and 7, respectively.

After crossing, the numbers being not repeated in the same chromosome are retained, and the repeated numbers are mapped with the corresponding relationship of the middle segment to eliminate the duplicate numbers.

Mutation operation is an auxiliary method to generate new individuals, which determines the local search ability of the genetic algorithm to a certain extent, maintains the diversity of population, and prevents premature convergence. Because the result of mutation is uncertain, mutation probability  $p_m$  is usually set to a very small number. In this paper, we use the basic mutation method, based on the mutation probability. A parent chromosome is randomly selected, two integers of  $[1, 10]$  are randomly selected as mutation points, and the gene values on the two mutation points are mutated. The mutation operation is shown in Figure 10.

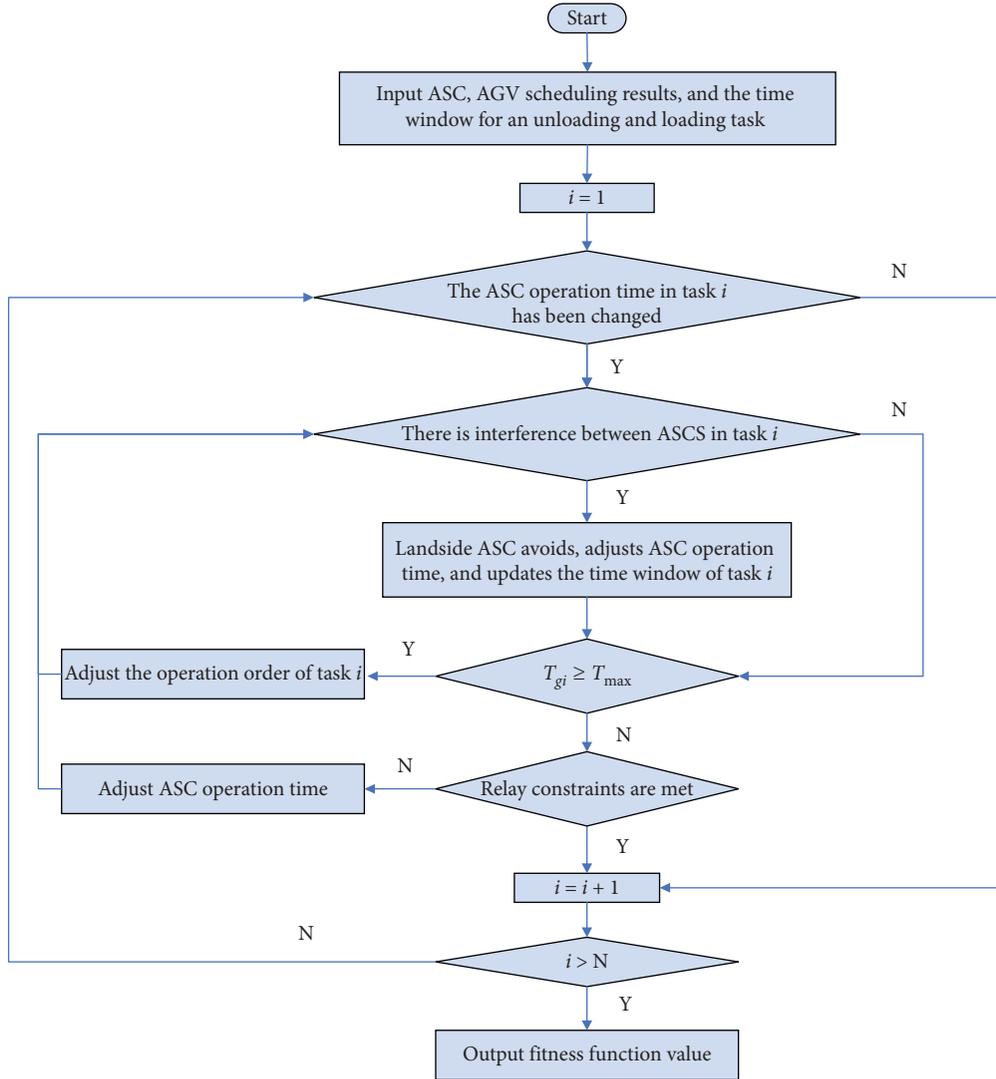


FIGURE 6: The operation flow of repairing infeasible solution.

Task sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16	17	19	20	22	23
ASC number	1	2	1	1	2	1	2	1	2	1	2	2	2	1	2	1	1	2	2	1
AGV number	3	0	2	1	0	3	0	3	0	2	0	0	0	1	0	2	2	0	0	1

FIGURE 7: Chromosome encoding and decoding diagram.

TABLE 1: Example of the task number.

	Original task				Task after splitting					
Task sequence	1	2	3	4	1	2	6	3	7	4
Origin	0	0	18	9	0	0	11	18	11	9
Destination	5	16	0	0	5	11	16	11	0	0

4.5. Simulated Annealing Operation. Because the traditional genetic algorithm is easy to fall into the local optimal problem, this paper introduces the idea of simulated annealing in the genetic algorithm. After mutation, the new

individuals are judged through the metropolis criterion to decide whether to accept it or not:

$$p_i(i \rightarrow j) = \begin{cases} 1, & \text{Fit}(j) \leq \text{Fit}(i), \\ \exp\left(\frac{\text{Fit}(i) - \text{Fit}(j)}{T}\right), & \text{else,} \end{cases} \quad (29)$$

where  $\text{Fit}(i)$  is the fitness before cross mutation,  $\text{Fit}(j)$  is the fitness after cross mutation,  $T$  is the current temperature,

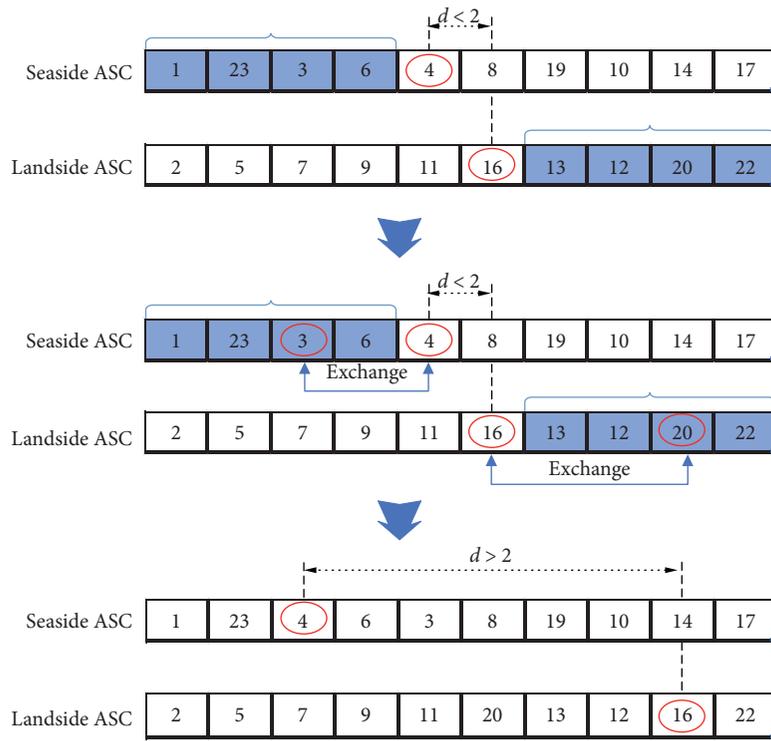


FIGURE 8: The generation process of initial population.

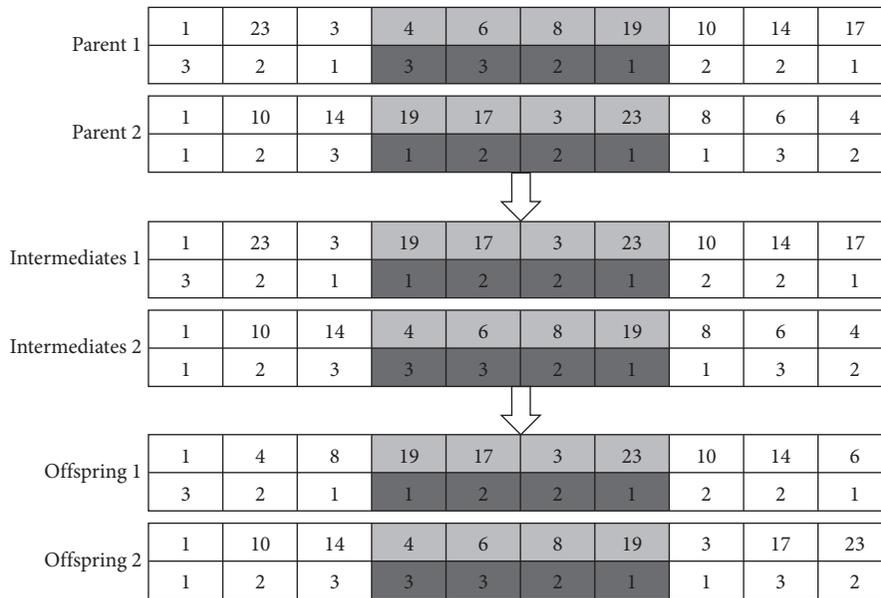


FIGURE 9: The process of seaside ASC chromosome crossing.

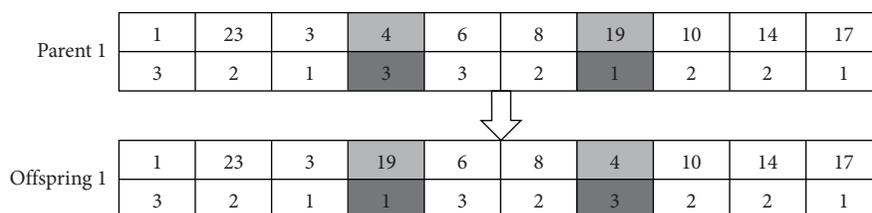


FIGURE 10: The process of seaside ASC chromosome mutating.

and  $T$  decreases according to the cooling rate with the evolution.

## 5. Results and Discussion

In this paper, the genetic algorithm was used to solve the problem of the ASC and the AGV cooperative scheduling under the condition of limited buffer capacity and twin ASC job interference. The superiority of the model and the effectiveness of the algorithm were verified through the comparative analysis of with and without handshake area, and the performance comparison of the algorithm and a sensitivity experiment. The related parameters are set as follows: the no-load running speed of AGV is set to be 4 m/s, and the heavy-load running speed is set to be 2 m/s; the start time of AGV tasks follows the sequence with the first term of 0 and tolerance of 50; the loading and unloading time of a container by ASC is set to be 200 s, the moving speed of ASC is set to be 3.5 s/bay.

*5.1. Comparative Analysis of with and without Handshake Area.* According to the operating experience of existing automatic terminals, the handshake area or temporary storage area is widely used. The handshake area design is often an efficient scheduling scheme for the twin YC of the automatic container terminal. Therefore, in the cooperative scheduling of AGV and ASC under the relay operation mode, this paper compared cooperative scheduling with and without a handshake area to verify the feasibility of handshake area.

In this paper, according to the actual operation tasks of the relay gantry crane in the automatic container terminal of the Yangshan deepwater port phase IV project, which is located in Shanghai, China, an experimental example of 20 tasks had been designed. It is assumed that there are 39 bays in each block of the yard, and bay 20 is initially assumed to be handshake area. The seaside I/O point lies in the 0<sup>th</sup> bay, while the landside I/O point lies in the 40<sup>th</sup> bay. The distribution of specific tasks is shown in Table 2. The results of the simulation are shown in Tables 3 and 4. The result of Table 3 showed that the scheduling scheme without handshake area design is better under a small number of tasks. The result of Table 4 showed that the scheduling scheme with or without handshake area design.

Figures 11 and 12 show ASC operation path diagrams under two schemes, in which the horizontal axis represents bay and the vertical axis represents time. With handshake area, ASC workload on both sides is relatively balanced. The seaside ASC takes 4423 s, and the landside ASC takes 4360 s. Without handshake area, the seaside ASC takes 4413 s and landside ASC takes 5003.5 s. There is approximately 12% difference in task completion time between ASCs on both sides. This would cause the workload of ASCs on both sides to be mismatched.

According to the scheduling results in Table 3, for the same 20 task data, the scheduling scheme without handshake area design is much better than that with handshake area design. The reason is that the ASC workload on both sides is unbalanced under the design of no handshake area, and the

TABLE 2: Bay-location distribution of tasks.

Task number	Initial position	Target location	Task type	Relay or not
1	0	10	L	No
2	25	40	L	No
3	0	29	L	Relay
4	36	0	U	Relay
5	0	27	L	Relay
6	19	0	U	No
7	22	40	L	No
8	0	30	L	Relay
9	35	0	U	Relay
10	0	26	L	Relay
11	15	0	U	No
12	32	0	U	Relay
13	39	0	U	Relay
14	0	38	L	Relay
15	15	0	U	No
16	40	25	U	No
17	28	40	L	No
18	34	0	U	Relay
19	0	25	L	Relay
20	29	0	U	Relay

TABLE 3: Solution results of two schemes for design with or without handshake area under relay operation mode.

Schemes	$f_1$ (s)	$f_2$ (s)	Objective function value (s)
With handshake area design	4951	4423	9374
Without handshake area design	3957	5003	8960

seaside ASC completes the task too early, resulting in the AGV waiting time far less than the former. In the case of with handshake area design, the maximum time consumption of ASC is 4423 s, while in the case of without handshake area design, the maximum time consumption of ASC is 5003.5 s, and it is 11.60% higher than the former. However, the waiting time of AGV under the case of without handshake area design is about 20% less than the former, which leads to better scheduling scheme in the case of without handshake area design. In summary, the design of without handshake area saves 4.42% of the time.

Figure 13 presents a comparison of the results of two scheduling schemes under different task numbers. It can be seen that, without considering other factors, when the number of tasks is small, the result of the scheduling scheme without handshake area design is better. With the increase of the number of containers, the difference between the two scheduling schemes becomes smaller. When the number of tasks exceeds 90, the scheduling scheme with handshake area design is better than the latter. The main reason is that with the increase in task size, the delay time caused by the interference between the ASCs increases gradually, while the waiting time of AGV tends to be equal. When the buffer capacity is increased, the scheduling scheme under the

TABLE 4: Operation process of each kind of equipment under the two schemes with or without handshake area design.

Schemes	Equipment	Work sequence
With handshake area design	AGV1	15-8-1-10-5
	AGV2	13-20-4-9-12
	AGV3	6-18-11-19-3-14
	ASC1	15-6-8-18-1-11-10-13-19-2-0-5-4-3-9-14-12
	ASC2	14-13-7-8-20-2-17-16-10-4-19-9-5-12-3-18
Without handshake area design	AGV1	6-15-11-10-19
	AGV2	1-8-14-3-5
	AGV3	9-4-20-12-18-13
	ASC1	6-15-11-1-8-10-14-19-9-3-4-5-20-12-18-13
	ASC2	16-12-19-20-7-9-8-4-10-3-2-17-18-14-13-5

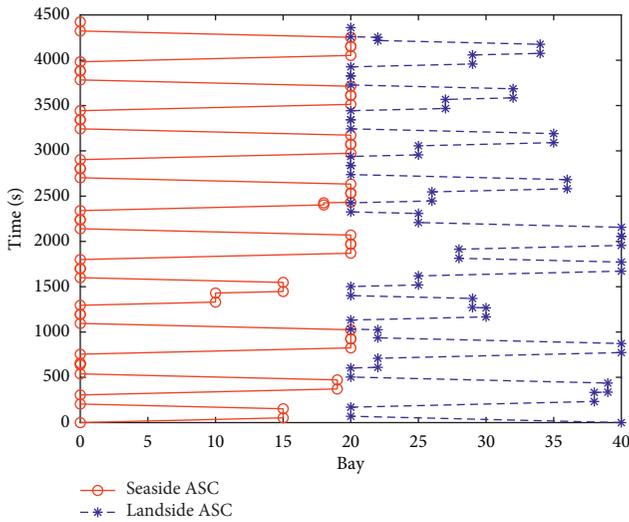


FIGURE 11: ASC scheduling with handshake area design.

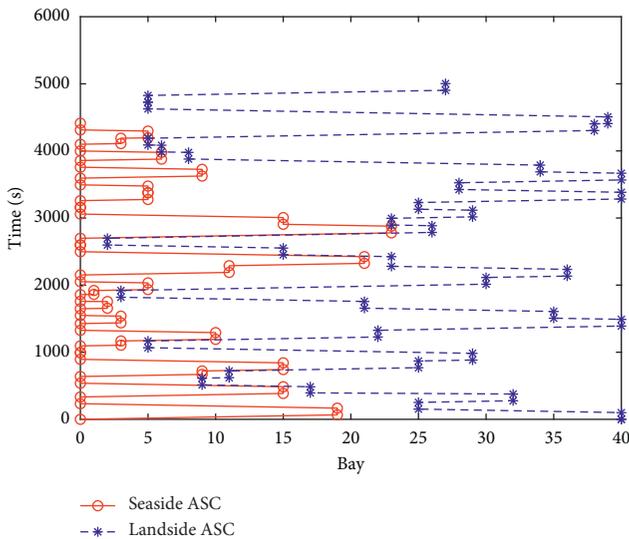


FIGURE 12: ASC scheduling without handshake area design.

handshake area design will be better than the latter earlier, which verified the feasibility of introducing handshake area into the cooperative scheduling model of AGV and ASC under the relay operation mode.

5.2. GA Parameter Experiment. In the genetic algorithm, the parameters have an important influence on the performance of the algorithm. We study the search results of the genetic algorithm under different parameter settings, so as to select the appropriate parameter combination for the problem. Here, we used the data in Table 2. The parameter values of GA are as follows:

$$\text{Crossover rate } p_c = \{0.6, 0.7, 0.8, 0.9\}$$

$$\text{Mutation rate } p_m = \{0.3, 0.4, 0.5, 0.6, 0.7\}$$

$$\text{Population size Pop} = \{100, 200, 300, 400, 500\}$$

Table 5 show the mean of obtained solutions under various crossover rate and mutation rate in 20 size test case. It can be seen that the setting with crossover rate  $p_c = 0.9$  and mutation rate  $p_m = 0.5$  outperforms others for this problem. Figure 14 presents the mean objective values for different population size under  $p_c = 0.9$  and  $p_m = 0.5$  in 20 size test cases. With the increase of population size, it was found that the value of the objective function gradually decreases, but when the population size reaches 500, the decline range is small, so that the setting with crossover rate  $p_c = 0.9$ , mutation rate  $p_m = 0.5$ , and population size  $\text{Pop} = 500$  is the most suitable for this problem.

5.3. Algorithm Contrast. In order to verify the effectiveness and stability of the algorithm, this paper set up two kinds of scale examples for the impact of different numbers of tasks, buffer capacity, and number of AGVs on the objective function. In the small-scale case, we used the genetic algorithm and CPLEX to solve 10 groups of examples, respectively. In the large-scale case, we used the genetic algorithm and particle swarm optimization algorithm to solve 14 groups of examples, respectively. Because there are random errors in the results of the genetic algorithm and particle swarm optimization, in order to reduce the random errors, each case runs 20 times, and its average operation time and average optimal fitness value were recorded. The specific results are shown in Table 6 and 7. The parameters of PSO are set as follows: the population size of PSO is 500, the maximum iteration times is 100,  $c_1 = c_2 = 1.1496$ , and  $\omega = 0.7298$ .

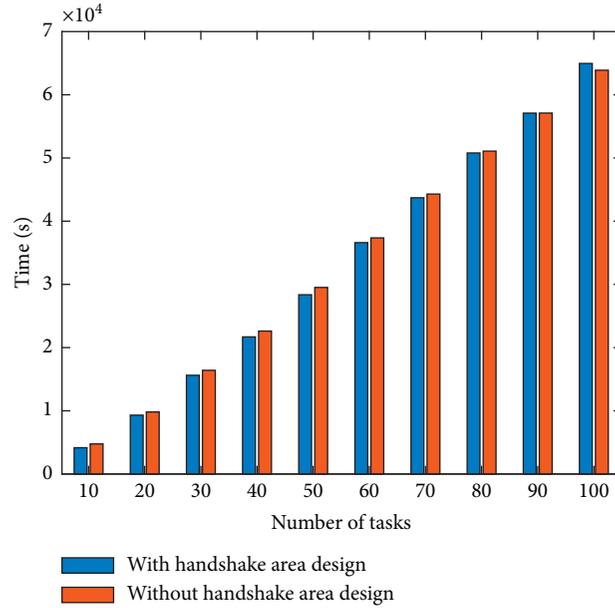


FIGURE 13: Comparison of two scheduling schemes.

TABLE 5: Mean of obtained solutions under various crossover and mutation rates.

$P_c$	$P_m$				
	0.3 (min)	0.4 (min)	0.5 (min)	0.6 (min)	0.7 (min)
0.6	176.23	179.68	171.89	173.84	179.39
0.7	179.59	182.49	171.07	176.84	173.25
0.8	177.46	176.72	171.29	175.62	172.45
0.9	177.01	170.63	168.45	170.05	170.95

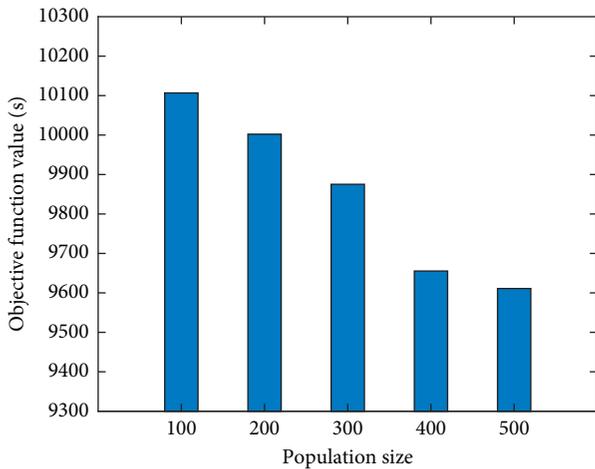


FIGURE 14: Mean objective values for different population size under  $p_c = 0.9$  and  $p_m = 0.5$  in 20 size test cases obtained by the proposed GA.

$$\text{Deviation (\%)} = \frac{\text{GA} - \text{CPLEX}}{\text{GA}} \times 100. \quad (30)$$

$$\text{Deviation (\%)} = \frac{\text{PSO} - \text{GA}}{\text{PSO}} \times 100. \quad (31)$$

It can be seen from Table 6 that when the task size is small, CPLEX can quickly calculate the exact solution. However, with the increase of the number of tasks, the solution time of CPLEX becomes longer and longer. When the number of tasks reaches 25, CPLEX cannot get the exact solution in an acceptable time. The average difference between the optimal fitness value obtained by GA and that by CPLEX is 4.11%. The above results show that the GA is effective for small-scale calculation. From Table 7, it can be seen that, in the large-scale case, with the increase in tasks, the solution obtained by each group of the genetic algorithm has obvious advantages compared with particle swarm optimization. It also found that the running time of the genetic algorithm is relatively short, and the convergence speed is faster. The results verified the stability and effectiveness of the genetic algorithm in solving the collaborative scheduling model of port handling equipment in this paper.

**5.4. Buffer Capacity Analysis.** In this paper, five groups of experiments were set for the effects of different numbers of tasks and buffer size on the objective function, corresponding to the number of tasks at different scales of 10, 20, 30, 40, and 50, respectively. At the same time, four sub-examples were set for each experiment, corresponding to

TABLE 6: Comparison of CPLEX and GA result.

Sequence	$N * B * K$	CPLEX		GA		Deviation (%)
		$F$ (min)	Time (s)	$F$ (min)	Time (s)	
1	5 * 3 * 3	33.13	0.27	33.96	25.61	2.42
2	10 * 3 * 3	67.77	754.36	69.08	50.25	1.91
3	15 * 3 * 3	85.38	2035	86.76	72.62	1.59
4	20 * 3 * 3	117.55	4319	124.02	90.20	5.25
5	25 * 4 * 4	158.85	6900	163.34	122.25	5.07
6	30 * 4 * 5	201.35	11520	212.94	165.23	5.43
7	40 * 4 * 5	245.30	14999	256.78	198.80	4.47
8	50 * 5 * 5	348.28	18000	363.12	240.56	4.09
9	80 * 6 * 6	740.40	>20000	794.08	510.26	6.76

TABLE 7: Comparison of algorithm results.

Sequence	$N * B * K$	GA		PSO		Deviation (%)
		$F$ (min)	Time (s)	$F$ (min)	Time (s)	
1	10 * 3 * 2	68.57	47.50	69.34	48.60	1.11
2	10 * 3 * 3	69.61	48.50	70.12	49.70	0.73
3	20 * 3 * 3	125.27	87.60	128.32	91.10	2.37
4	20 * 4 * 4	131.99	86.90	154.00	92.30	14.29
5	50 * 3 * 5	482.31	206.40	507.41	210.90	10.11
6	50 * 4 * 5	377.78	212.50	487.81	289.60	22.00
7	80 * 4 * 5	823.04	496.60	857.06	564.20	4.00
8	80 * 5 * 5	797.46	480.90	833.73	494.40	4.35
9	100 * 4 * 6	1014.18	718.30	1098.09	749.00	7.64
10	100 * 5 * 6	987.43	795.00	1079.72	819.00	7.78
11	150 * 5 * 6	1385.71	1673.00	1655.83	1676.50	16.32
12	150 * 5 * 8	1438.43	1720.00	1708.34	1741.00	15.80
13	200 * 4 * 10	1896.8	2950.00	2051.52	2994.00	7.54
14	200 * 5 * 10	1784.4	2992.00	1888.16	3007.00	5.50

TABLE 8: Buffer capacity analysis.

Task quantity	Buffer capacity/objective function				
	3 (min)	4 (min)	5 (min)	6 (min)	7 (min)
10	79.29	73.05	70.93	70.02	69.53
20	163.60	156.47	153.65	151.79	150.29
30	273.40	263.10	257.67	254.70	252.30
40	376.95	363.40	354.73	348.72	345.02
50	491.95	475.53	462.80	454.57	449.27

different buffer capacities of 3, 4, 5, 6, and 7, respectively. The genetic algorithm was used to solve 5 groups of experiments, and the average value of 10 experiments was taken. The specific results are shown in Table 8.

The result of Table 8 showed that when buffer capacity is small, increasing the buffer capacity can reduce the total operation time of each task. This effect is becoming more obvious with the increase in tasks for relay operation. Figure 15 presents the trend of total delay time of different buffer capacities. By comparing the total operation time of different buffers in each group, it was found that increasing the buffer capacity has a decreasing effect on reducing the total operation time.

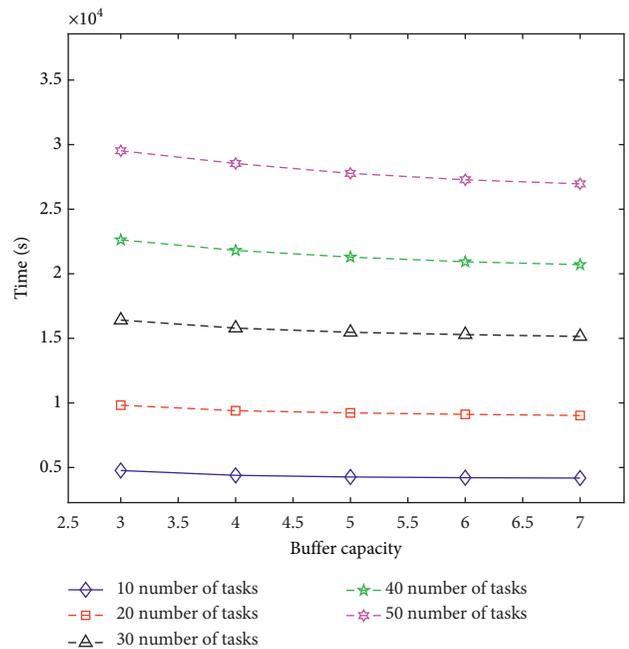


FIGURE 15: Trend of total operation time of different buffers.

## 6. Conclusion

This paper analysed the cooperative scheduling of automated guided vehicle and automated stacking crane in the automated container terminal, and optimized the operation characteristics of the new loading and unloading technology for the automated terminal. By combining the AGV and ASC scheduling problems, the collaborative scheduling model of AGV and ASC under ASC relay operation mode was developed, and a genetic algorithm was designed to solve the problem.

The results showed that, compared without handshake area, the handshake area design can effectively reduce the interference between the twin ASC operations, and has obvious advantages in large-scale operations. The results indicated that the handshake area design can reduce the ASC operation interference time by an average of 10.56%. It was also found that increasing the buffer capacity can reduce the total operation time of each task, and the effect is more obvious with the increase in the number of tasks. It was found that the average operation time decreases by about 2.27% for each buffer increase, where the waiting time of the AGV can be reduced by 4.35% on average. Generally speaking, the setting of buffer zone and handshake area can effectively reduce the operation delay, which can help to improve the overall operation efficiency of the terminal.

## Data Availability

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

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (grant number 51875332) and the Capacity Building Projects of Some Local Universities of the Shanghai Science and Technology Commission (grant number 18040501600).

## References

- [1] H. F. Durrant-Whyte, "An autonomous guided vehicle for cargo handling applications," *The International Journal of Robotics Research*, vol. 15, no. 5, pp. 407–440, 1996.
- [2] K. H. Kim and J. W. Bae, "A look-ahead dispatching method for automated guided vehicles in automated port container terminals," *Transportation Science*, vol. 38, no. 2, pp. 224–234, 2004.
- [3] E. K. Bish, F. Y. Chen, Y. T. Leong, B. L. Nelson, J. W. C. Ng, and D. Simchi-Levi, "Dispatching vehicles in a mega container terminal," *OR Spectrum*, vol. 27, no. 4, pp. 491–506, 2005.
- [4] P. Angeloudis and M. G. H. Bell, "An uncertainty-aware AGV assignment algorithm for automated container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 3, pp. 354–366, 2010.
- [5] B. Cai, S. Huang, D. Liu, and G. Dissanayake, "Rescheduling policies for large-scale task allocation of autonomous straddle carriers under uncertainty at automated container terminals," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 506–514, 2014.
- [6] R. Zaghdoud, K. Mesghouni, S. C. Dutilleul et al., "A hybrid method for assigning containers to AGVs in the dynamic environment of container terminals," *Stud Inform Control*, vol. 24, no. 1, pp. 43–50, 2015.
- [7] W. Malopolski, "A sustainable and conflict-free operation of AGVs in a square topology," *Computers & Industrial Engineering*, vol. 126, pp. 472–481, 2018.
- [8] V. D. Nguyen and K. H. Kim, "A dispatching method for automated lifting vehicles in automated port container terminals," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 1002–1020, 2009.
- [9] S. Saini, D. Roy, and R. de Koster, "A stochastic model for the throughput analysis of passing dual yard cranes," *Computers & Operations Research*, vol. 87, pp. 40–51, 2017.
- [10] D. Briskorn and P. Angeloudis, "Scheduling co-operating stacking cranes with predetermined container sequences," *Discrete Applied Mathematics*, vol. 201, pp. 70–85, 2016.
- [11] H. Lu and S. Wang, "A study on multi-ASC scheduling method of automated container terminals based on graph theory," *Computers & Industrial Engineering*, vol. 129, pp. 404–416, 2019.
- [12] Z.-H. Hu, J.-B. Sheu, and J. X. Luo, "Sequencing twin automated stacking cranes in a block at automated container terminal," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 208–227, 2016.
- [13] P. J. M. Meersmans and A. P. M. Wagelmans, *Effective Algorithms for Integrated Scheduling of Handling Equipment at Automated Container Terminals*, Erasmus Research Institute of Management, Erasmus Universiteit, Rotterdam, Netherlands, 2001.
- [14] J. Xin, R. R. Negenborn, G. Lodewijks et al., "Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 214–230, 2014.
- [15] J. Luo and Y. Wu, "Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 79, pp. 49–64, 2015.
- [16] J. Luo, Y. Wu, A. B. Mendes et al., "Modelling of integrated vehicle scheduling and container storage problems in unloading process at an automated container terminal," *Computers & Industrial Engineering*, vol. 94, pp. 32–44, 2016.
- [17] H. Hu, X. Chen, T. Wang, and Y. Zhang, "A three-stage decomposition method for the joint vehicle dispatching and storage allocation problem in automated container terminals," *Computers & Industrial Engineering*, vol. 129, pp. 90–101, 2019.
- [18] M. Zhong, Y. Yang, Y. Zhou, and O. Postolache, "Adaptive autotuning mathematical approaches for integrated optimization of automated container terminal," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–14, Article ID 7641670, 2019.
- [19] Q. Zhao, S. Ji, D. Guo, X. Du, and H. Wang, "Research on cooperative scheduling of automated quayside cranes and automatic guided vehicles in automated container terminal,"

- Mathematical Problems in Engineering*, vol. 2019, Article ID 6574582, 15 pages, 2019.
- [20] K. H. Kim and Y.-M. Park, "A crane scheduling method for port container terminals," *European Journal of Operational Research*, vol. 156, no. 3, pp. 752–768, 2004.
  - [21] D.-H. Lee, Z. Cao, and Q. Meng, "Scheduling of two-trans-tainer systems for loading outbound containers in port container terminals with simulated annealing algorithm," *International Journal of Production Economics*, vol. 107, no. 1, pp. 115–124, 2007.
  - [22] M. E. H. Petering, Y. Wu, W. Li, M. Goh, and R. de Souza, "Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals," *OR Spectrum*, vol. 31, no. 4, pp. 801–835, 2009.
  - [23] A. Imai, H. C. Chen, E. Nishimura, and S. Papadimitriou, "The simultaneous berth and quay crane allocation problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 5, pp. 900–920, 2008.
  - [24] C. Liang, Y. Huang, Y. Yang et al., "A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 1021–1028, 2009.