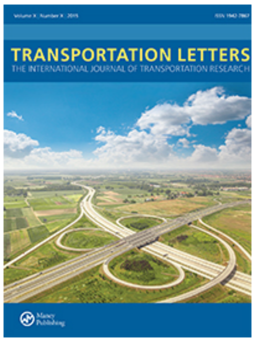

Taylor & Francis
Taylor & Francis Group

Transportation Letters
The International Journal of Transportation Research

ISSN: 1942-7867 (Print) 1942-7875 (Online) Journal homepage: https://www.tandfonline.com/loi/ytrl20

Integrated scheduling in automated container terminals considering AGV conflict-free routing

Ji Shouwen, Luan Di, Chen Zhengrong & Guo Dong

To cite this article: Ji Shouwen, Luan Di, Chen Zhengrong & Guo Dong (2020): Integrated scheduling in automated container terminals considering AGV conflict-free routing, Transportation Letters, DOI: 10.1080/19427867.2020.1733199

To link to this article: https://doi.org/10.1080/19427867.2020.1733199

Published online: 27 Feb 2020.

TRANSPORTATION LETTERS
https://doi.org/10.1080/19427867.2020.1733199

Integrated scheduling in automated container terminals considering AGV conflict-free routing

Ji Shouwen[a], Luan Di[a], Chen Zhengrong[a] and Guo Dong[b]

[a]School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China; [b]School of Mechanical-Electronic and Vehicle Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China

ABSTRACT

In automated container terminals, effectively scheduling quay cranes (QCs), automated guided vehicles (AGVs), automated stacking cranes (ASCs) and AGV routing are two important problems. This paper studies the combinatorial optimization of two problems in the synchronous loading and unloading operation mode of the automated container terminal. First, a bi-level programming model for integrated scheduling problems is established with the goal of minimizing the completion time of all containers. Secondly, in order to solve the model, we design two bi-level optimization algorithms based on conflict resolution strategy: bi-level adaptive genetic algorithm based on conflict resolution strategy (CRS-BAGA) and bi-level genetic algorithm based on conflict resolution strategy (CRS-BGA). Finally, the effectiveness of the two optimization algorithms is verified by numerical experiments. Experimental results show that the proposed optimization algorithms are effective. They can coordinate the scheduling of three equipments and solve the conflict congestion problem of AGV.

KEYWORDS

Automated container terminal; integrated scheduling; synchronous loading and unloading; combinatorial optimization; genetic algorithm

Introduction

With the development of economic globalization and international trade, the international shipping market has also developed rapidly. As the main mode of transportation for international shipping, container traffic has grown rapidly. In order to meet the growing demand for container transportation, it is important to improve the efficiency of automated container terminals. Due to the high equipment costs of quay cranes (QCs), automated guided vehicles (AGVs) and automated stacking cranes (ASCs), it is difficult for container terminals to purchase additional operating equipment to increase terminal efficiency. Therefore, the coordinated scheduling of equipment in automated container terminals plays a key role in improving the efficiency of automated container terminals.

Generally, most of the automated container terminals use the mode of operation that the loading operation are completed before a discharging operation. In this single cycling mode of operation, the operation equipment can only load or unload one container during a round trip operation. It is easy to cause the no-load rate of the equipment to be too high during the entire operation. With the continuous maturity of logistics technology, some terminals have gradually tried new modes of operation. Synchronous loading and unloading mode is a new mode of operation currently adopted by container terminals to improve operational efficiency. This mode of operation enables the equipment to load or unload two containers during a round trip operation, significantly improving the efficiency of the container terminal. Therefore, this paper focuses on the integrated scheduling in automated container terminals which under the mode of synchronous loading and unloading operation.

At present, the scheduling and routing of AGV have become one of the main research issues of container terminals. As an important equipment for connecting QC and ASC, the efficiency of AGV affects the entire container terminal operating system. In the actual operation process, due to many uncertain factors in the working environment, there are some problems in AGV scheduling and path planning, including conflicts and congestion problems during AGV operation. Therefore, in order to solve this problem, we design two optimization methods based on conflict resolution strategy.

As two types of sub-problems in container terminal operations research, only optimizing one type of sub-problem may not be optimal for the entire system. Therefore, we combine the integrated scheduling problems of QC, AGV, ASC, and AGV path planning, to study the combinatorial optimization of the two problems. In this work, we consider the operation mode of synchronous loading and unloading, study the combinatorial optimization of QC, AGV, ASC, and AGV path planning, and solve the conflict and congestion problems of AGV path. This was not considered in previous studies. We have developed two optimization algorithms based on conflict resolution strategies to solve the comprehensive scheduling problem.

The main contributions of this paper are as follows:

(1) This paper studies the more general ‘QC-AGV-ASC’ terminal layout and operation flow under the mode of synchronous loading and unloading operation. A new bi-level programming model for ‘QC-AGV-ASC’ integrated scheduling problems was established.
(2) According to the model characteristics, two optimization algorithms were designed to solve the problem of integrated scheduling problem and AGV path conflict.
(3) This paper improved the genetic algorithm. Elitism preservation strategy was introduced to ensure the global convergence of the algorithm. Tabu list was introduced to prevent cycle search and improved the adaptive genetic algorithm. And catastrophe operator was introduced to avoid premature convergence.

CONTACT Chen Zhengrong 17120785@bjtu.edu.cn School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

This paper is structured as follows. Section 2 reviews previous literature on container terminal equipment integration scheduling and AGV path planning. Section 3 presents a bi-level programming model formulated for the integrated scheduling problem. Section 4 proposes a bi-level adaptive genetic algorithm (CRS-BAGA) based on the conflict resolution strategy and a bi-level genetic algorithm based on conflict resolution strategy (CRS-BGA). Numerical experiments are conducted in Section 5 to evaluate the effectiveness of the proposed algorithm. Section 6 summarizes the content of this article and proposes future research work.

Literature review

In recent years, there have been many studies on container terminal equipment. There are many sub-problems in the study of container terminals. Here we provide a brief review of existing studies related to scheduling problems of container terminal equipment and path planning problem of AGVs.

The studies on the scheduling problems of automatic container terminal equipment mainly focus on improving the overall operation efficiency of the terminal. The operation between different types of equipment are optimized by considering the actual constraints in the scheduling process. Existing research provides many viable methods for efficient scheduling of container terminal equipment. However, most of the literature does not consider the conflict and congestion problems of AGVs. In terms of operation process, only single cycling is considered. Lau and Zhao (2008) studied the integrated scheduling problem of QC, AGV, and ASC, and proposed a multi-layer genetic algorithm to solve the problem (Lau and Zhao 2008). Cao et al. (2010) studied the integrated scheduling problem of yard cranes and yard trucks, established a mixed-integer programming model, and proposed two solving methods based on Benders (Cao et al. 2010). Tang, Zhao, and Liu (2014) considered the actual constraints such as the priority relationship between containers, the safety distance of the shore bridge and the waiting time of the equipment, in order to minimize the total completion time of the operation. An improved particle swarm optimization algorithm was designed to solve the model (Tang, Zhao, and Liu 2014). Chen, Langevin and Lu (2013) designed a three-stage algorithm to solve the integrated scheduling problem of QCs and trucks (Chen and Langevin 2013). Lu and Le (2014) studied the comprehensive scheduling problem of container terminals under uncertainties. A scheduling optimization method based on particle swarm optimization was proposed (Lu and Le 2014). Xin, Negenborn, and Lodewijks (2014) proposed an energy-saving control method for QC, AGV and ASC scheduling (Xin, Negenborn, and Lodewijks 2014a). He, Huang, Yan and Wang (2015) established a mixed-integer programming model for integrated planning of QCs, ITs, and YCs, in order to minimize the total departure delay time and energy consumption of the ship, and designed a hybrid algorithm of genetic algorithm and particle swarm optimization. (He et al. 2015). Kaveshgar and Huynh (2015) considered the practical constraints of container priority, QCs mutual interference, and safety distance, established a mixed-integer programming model, and designed a genetic algorithm and greedy algorithm mixed algorithm to solve the model (Kaveshgar and Huynh 2015). Luo, Wu and Mendes (2016) considered the integrated scheduling problem of AGV and container storage location allocation in the unloading process. They established a mixed-integer programming model and designed a genetic algorithm to solve (Luo, Wu, and Mendes 2016). Chang, Zhu, Yan and Wang (2017) studied the integrated scheduling problems of GCs, IT, and YCs. They not only considered train loading and unloading, but also considered GC interference and safety margins, GC and YC travel time and buffer area, and proposed a multi-layer genetic algorithm to solve this problem (Chang et al. 2017). Zheng et al. (2019) studied the optimization of the QCs and ITs in the unloading process, and established a mixed-integer programming model. They developed a particle swarm optimization algorithm to solve this model (Zhen et al. 2016).

There are more relevant literatures about the path planning problem of AGVs. They solved the conflict congestion problem of the AGV path and helped our research. However, they have ignored the impact of other devices on AGVs. Rajeeva et al. (2003) developed an efficient deadlock prediction algorithm and implemented the proposed method by using Automod simulation software (Moorthy et al. 2003). Nishi, Ando and Konishi (2006) established 143 nodes of the transport system to implement dynamic scheduling of AGV to select the best path for AGV (Nishi, Ando, and Konishi 2006). Nishi, Hiranaka and Grossmann (2011) divided the scheduling and path problems of AGV into two levels, the upper level is the task allocation and scheduling problem, the lower level is the path planning problem, and the corresponding solution is proposed (Nishi, Hiranaka, and Grossmann 2011). Xin, Negenborn, and Lodewijks (2014) proposed a new two-stage energy-aware control method. The higher-level controller determined the schedule of the operating equipment, and the lower-level controller determined the collision-free path of the AGV (Xin, Negenborn, and Lodewijks 2014b). Miyamoto and Inoue (2016) considered the vehicle capacity limitation, machine buffer, and other factors, and described the AGV scheduling and path planning problem as an integer programming model. They used local and random search methods to solve (Miyamoto and Inoue 2016). Roy, Gupta, and Koster (2016) used a nonlinear traffic flow model to determine the appropriate number and effective speed of AGV, to solve the congestion problem of AGV path planning (Roy, Gupta, and De Koster 2015). Waldemar (2018) used a square topology to describe the transportation network and proposed an AGV collision and deadlock prevention method based on chains of reservations (Małopolski. 2018). Li et al. (2018) studied the influences of travel time uncertainty on AGV conflict and path planning. They proposed a method of dynamic adjustment (Jun-Jun et al. 2018).

The above two problems are common sub-problems in automated container terminal research. In fact, two sub-problems are tightly interconnected. Optimizing only one sub-problem may not be an overall optimization. To the best of our knowledge, few literatures have comprehensively considered two sub-problems and studied the combinatorial optimization problem of two sub-problems. Yang, Zhong, and Dessouky et al. (2018) did a related study. They studied the integrated scheduling problems of QC, AGV, and ARMG and AGV path planning. A bi-level programming model was established with the goal of minimizing makespan. They proposed a bi-level General Algorithm based on the preventive congestion rule to solve the model (Yang, Zhong, and Dessouky 2018). Their research has inspired us. The different work and innovative points between our research and their research are as follows.

Firstly, we consider the integrated scheduling of QC and AGV with another operating equipment, which causes the workflow to change. Yang et al. studied the integrated scheduling of 'QC-AGV-ARMG'. In addition, they also considered the AGV-mate. AGV-mate is an auxiliary equipment to assist the AGV in loading or unloading containers. For example, during the unloading process, AGV arrives at the specified block and AGV-mate unloads the container from AGV. Then, the AGV leaves, and it does not need to wait for ASC. Subsequent operations are completed by the AGV-mate and ARMG. In addition, in their studies, QC can place the container in advance on the specified block. But in our research, we did not consider these. AGV need to transfer container directly with QC or ASC. Compared to the research by Yang et al., our research requires higher equipment coordination. We did not consider the extra equipment. In other words, Yang et al. studied a specific container terminal layout and operation process, and we studied the more general container terminal layout and operation process.

Secondly, the AGV running road network is different. They studied the auxiliary roads which in the AGV running road network, are bidirectional single lanes. We studied the auxiliary roads which in the AGV running road network, are unidirectional single lanes. Due to the different terminal layout, operational processes, and the AGV running road network, we need to establish a new bi-level programming model. And our new model is more general and more suitable for most container terminals research.

Thirdly, we adopted a traffic control strategy which conforms actual needs. Yang et al. solved the AGV congestion problem by limiting the auxiliary road density and re-planning the path. And we use the waiting strategy as a conflict resolution strategy.

Finally, in order to solve the new bi-level programming model, we designed two bi-level optimization algorithms. Yang et al. used the genetic algorithm. But the genetic algorithm has the disadvantages of weak global searching ability and easy to fall in local best. Therefore, we will improve the genetic algorithm and propose two bi-level optimization algorithms based on conflict resolution strategy: bi-level adaptive genetic algorithm based on conflict resolution strategy and bi-level genetic algorithm based on conflict resolution strategy.

As stated above, as for the literatures about scheduling problems of container terminal equipment, most literatures did not consider the conflict problems of AGVs and dual cycling operation process. As for the literatures about path planning problem of AGVs, most literatures did not consider the cooperation between AGVs and other equipment. Two problems are correlated with each other. Therefore, we studied the combined optimization problem of container terminal equipment scheduling problem and AGV path planning problem. This is especially important in the actual operation process.

Model formulation

Problem description

Generally, automated container terminals primarily include three types of operating equipment, QC, AGV, and ASC. Figure 1 shows the layout of an automated container terminal. It is mainly divided into dock parking area, AGV driving area, and stacking area. In the dock parking area, QC are responsible for loading and unloading containers on the ship. In the stacking area, ASC are responsible for the storage and removal of containers. As the interface between the docking parking area and the stacking area, AGV is responsible for the transportation of containers. The place where the AGV transfer containers between QC and ASC is called the transfer point, as shown in Figure 1.

After the ship arrives at the dock parking area of the terminal, QC picks up the container from the ship and transports it to the transfer point for handover with the AGV. Before this, the AGV needs to arrive at the transfer point in advance to wait for the QC. Then, the AGV transports the container to the transfer point with the ASC, which unloads the container to the designated storage location. After that, the AGV receives the instruction of the next task and goes to the corresponding transfer point to wait for the ASC. After the ASC takes the container out of the stacking area and transfers it to the AGV, the AGV transports the container to the transfer point with QC. QC completes the subsequent loading operation. In this process, the AGV can perform loading or unloading tasks of any QC and ASC. In addition, we do not consider the specific allocation problem of containers in the stacking area.

Through the above problems, we can know the time to loading and unloading the container and the initial and target positions of the container. The next step involves the assignment of tasks to containers. After that, we need to plan the path of the AGV. We use nodes to represent inflection points, intersections, and transfer points, and model the road network of Figure 1 as the model diagram shown in Figure 2. Due to the speed of the AGV is constant, the path planning problem can be converted to the shortest path problem of known starting and ending points.

When the number of AGVs increases, the problems of AGV conflict congestion are easy to occur. As shown in Figure 3, there are three conflict types of AGV: node conflict, chasing conflict and reverse conflict. Due to the path is a one-way path that considered in this paper, and the speed of AGV is constant. Therefore, there is no problem of chasing conflict and reverse conflict.

In order to solve the node conflict problem, we need to limit the number of AGVs on the node. In other words, at the same time, only one AGV can exist on the same node. We use the waiting strategy as a conflict resolution strategy to plan AGV conflict-free path. When the AGV runs to a node, it is determined whether the front node has AGV occupation. If exists, the AGV waits at the current node, otherwise it proceeds.

Model hypothesis

(1) QC and ASC can only load or unload one container at a time.

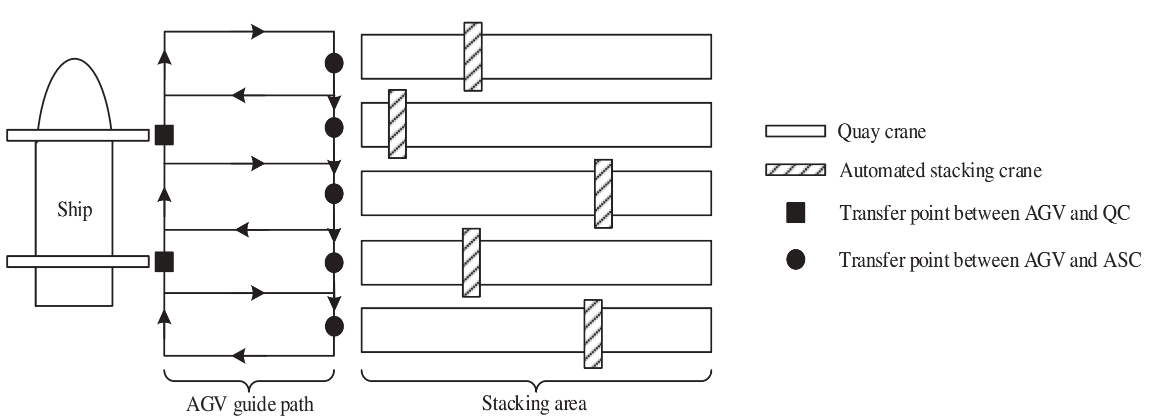

Figure 1. Layout of an automated container terminal.

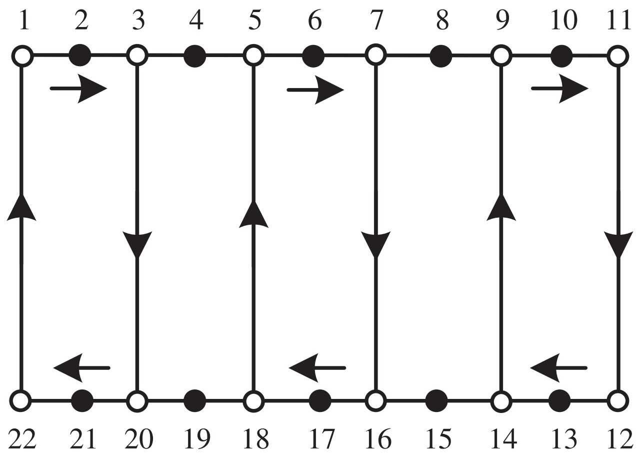

Figure 2. The path of AGV.

(2) AGV can only transport one container at a time.
(3) Interference conflicts between QCs are not considered.
(4) The speed of AGVs is uniform.
(5) ASC cannot cross stacking area.
(6) Each QC and ASC is only responsible for container transfer operations at one transfer point.

Model parameters

(1) Parameters

N: the set of import containers, $n \in N$

L: the set of export containers, $l \in L$

I: the set of all tasks(containers), $I = N \cup^{L}$, $i,\ i' \in I$, task 0 is virtual initial task and task f is virtual last task. The operation time of virtual task is 0

Q: the set of QCs, $q \in Q$

A: the set of AGVs, $a, a' \in A$

B: the set of ASCs, $b \in B$

K: the set of all equipment, $K = Q \cup^{A} \cup^{B}$

M: a large positive number

J: the set of all nodes in the path network, $s,\ m,\ s',\ m' \in J$

G: the directed line segment between the nodes

v: the speed of AGV

d_{sm}: the distance between node s and node m

C: the maximum makespan of all containers

(2) Decision variables

$X^k_{ii'}$: $X^k_{ii'} = 1$, if container i is handled before container i' on the equipment k; $X^k_{ii'} = 0$, otherwise

Y^k_i:$Y^k_i = 1$, if container i is handled by equipment k;$Y^k_i = 0$, otherwise

Z_{sm}: $Z_{sm} = 1$, if AGV get through the connected path of node s and$m$; $Z_{sm} = 0$, otherwise

ts^q_i: the starting time of container i by QC q

te^q_i: the completion time of container i by QC q

ts^a_i: the starting time of container i by AGV a

te^a_i: the completion time of container i by AGV a

ts^b_i: the starting time of container i by ASC b

te^b_i: the completion time of container i by ASC b

$t^a_{e,s}$: the time when AGV a arriving at node s

$t^a_{o,s}$: the time when AGV a leaving from node s

$t^a_{e,sm}$: the time when AGV a arriving at the connected path of node s and m

$t^a_{e,sm}$: the time when AGV a leaving from the connected path of node s and m

Upper model: integrated scheduling model

The purpose of this paper is to improve the overall efficiency of automated container terminals. Therefore, the purpose of the upper model is to minimize the maximum makespan of all containers:

$$\min C \tag{1}$$

Subject to:

$$\sum_{i \in I} X^q_{0i} = 1, \forall q \in Q \tag{2}$$

$$\sum_{i \in I} X^q_{if} = 1, \forall q \in Q \tag{3}$$

$$\sum_{i' \in I} X^q_{i} - \sum_{i' \in I} X^q_{ii'} = 0, \forall i \in I, \forall q \in Q \tag{4}$$

$$\sum_{q \in Q} Y^q_i = 1, \forall i \in I \tag{5}$$

$$te^q_i - ts^q_{i'} \le M(1 - X^q_{ii'}), \forall i, i' \in I, \forall q \in Q \tag{6}$$

$$te^q_{i'} - ts^q_i \le M \cdot X^q_{ii'}, \forall i, i' \in I, \forall q \in Q \tag{7}$$

Constraint (1) is the objective function that defines the goal to minimize the maximum makespan of all containers.

Constraints (2)–(7) are constraints related to the QC operation sequence and time. Constraint (2), (3) are constraints for the initial and ending tasks of each QC., ensure that the initial task of each QC is virtual task 0 and the ending task is virtual task f. Constraint (4) represents QC operation sequence, ensure that there is only one container task before and after the container task currently being processed by QC. Constraint (5) represents the uniqueness of each container task handled by one QC, ensures that each container can only be handled by a QC. Constraint (6), (7) represents the time relation between completion of current task by the QC and start of the next task., ensure that on the same QC, the completion time of the previous container is earlier than the starting time of the next container.

$$\sum_{i \in I} X^a_{0i} = 1, \forall a \in A \tag{8}$$

$$\sum_{i \in I} X^a_{if} = 1, \forall a \in A \tag{9}$$

$$\sum_{i' \in I} X^a_{i} - \sum_{i' \in I} X^a_{ii'} = 0, \forall i \in I, \forall a \in A \tag{10}$$

$$\sum_{a \in A} Y^a_i = 1, \forall i \in I \tag{11}$$

$$te^a_i - ts^a_{i'} \le M(1 - X^a_{ii'}), \forall i, i' \in I, \forall a \in A \tag{12}$$

$$te^a_{i'} - ts^a_i \le M \cdot X^a_{ii'}, \forall i, i' \in I, \forall a \in A \tag{13}$$

$$ts^a_i \le te^q_i, \forall i \in N, \forall q, a \in K \tag{14}$$

$$te^a_i \le ts^q_i, \forall i \in L, \forall q, a \in K \tag{15}$$

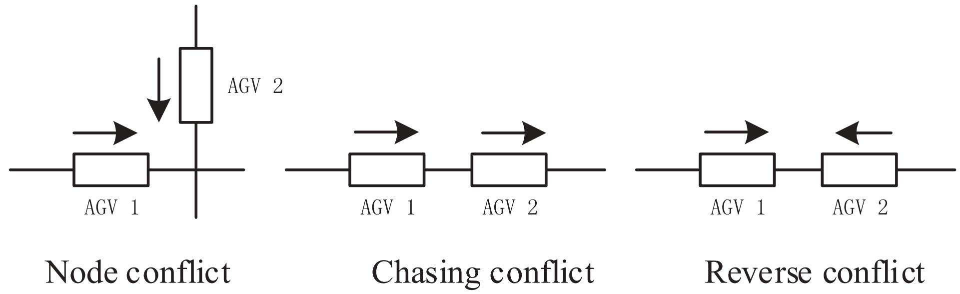

Figure 3. The conflict type of AGV.

$$te_i^a \le ts_i^b, \forall i \in N, \forall a, b \in K \tag{16}$$

$$ts_i^a \le te_i^b, \forall i \in L, \forall a, b \in K \tag{17}$$

Constraints (8)–(17) are constraints related to the AGV operation sequence and time. Constraint (8), (9) are constraints for the initial and ending tasks of each AGV., ensure that the initial task of each AGV is virtual task 0 and the ending task is virtual task f. Constraint (10) represents the AGV operation sequence, ensure that there is only one container task before and after the container task currently being processed by AGV. Constraint (11) represents the uniqueness of each container task handled by AGV, ensures that each container can only be handled by one AGV. Constraint (12), (13) represents the time relation between completion of current task by the AGV and start of the next task, ensure that on the same AGV, the completion time of the previous container is earlier than the starting time of the next container. Constraint (14), (15), (16), (17) are time-related constraints of AGV that arrive at transfer point. This means that AGV needs to reach the transfer point in advance to wait for QC and ASC.

$$\sum_{i \in I} X_{0i}^b = 1, \forall b \in B \tag{18}$$

$$\sum_{i \in I} X_{if}^b = 1, \forall b \in B \tag{19}$$

$$\sum_{i' \in I} X_{i'i}^b - \sum_{i' \in I} X_{ii'}^b = 0, \forall i \in I, \forall b \in B \tag{20}$$

$$\sum_{b \in B} Y_i^b = 1, \forall i \in I \tag{21}$$

$$te_i^b - ts_{i'}^b \le M(1 - X_{ii'}^b), \forall i, i' \in I, \forall b \in B \tag{22}$$

$$te_{i'}^b - ts_i^b \le M \cdot X_{ii'}^b, \forall i, i' \in I, \forall b \in B \tag{23}$$

Constraints (18)–(23) are constraints related to the ASC operation sequence and time. Constraint (18), (19) are constraints for the initial and ending tasks of each ASC., ensure that the initial task of each ASC is virtual task 0 and the ending task is virtual task f. Constraint (20) represents the ASC operation sequence, ensure that there is only one container task before and after the container task currently being processed by ASC. Constraint (21) represents the uniqueness of each container task handled by ASC, ensures that each container can only be handled by one ASC. Constraint (22), (23) represents the time relation between completion of current task by the ASC and start of the next task, ensure that on the same ASC, the completion time of the previous container is earlier than the starting time of the next container.

Lower model: AGV path planning model

The purpose of path planning is to minimize the travel distance of the AGV. Due to the AGV speed is constant, the shortest travel distance means the minimum transportation time.

$$\min \sum_{(s,m) \in T} Z_{sm} \cdot d_{sm} \tag{24}$$

Subject to:

$$\sum_{m:(s,m) \in G} Z_{sm} - \sum_{m:(s,m) \in G} Z_{ms} = \begin{cases} 1, i' = s', \\ -1, i' = m', \\ 0, i' \neq s', m', \end{cases} \tag{25}$$

$$t_{e,sm}^a + d_{sm}/v \le t_{o,sm}^a, \forall a \in A, \forall s, m \in J \tag{26}$$

$$t_{e,sm}^a \le t_{e,sm}^{a'}, \forall a, a' \in A, \forall s, m \in J \tag{27}$$

$$t_{o,sm}^a < t_{o,sm}^{a'}, \forall a, a' \in A, \forall s, m \in J \tag{28}$$

Constraint (25)–(28) are related constraints for AGV running. Constraint (25) is the route choice constraint. Constraint (26) represents the time relationship of AGV a enter and leave the path, means AGV a should leave the path no earlier than the sum of the time of AGV a entering the path and the running time on the path. Constraint (27) represents the time relationship between two consecutive AGVs enter the same path. Constraint (28) represents the time relationship between two consecutive AGVs leave the same path, ensures two consecutive AGVs cannot leave the same path at the same time.

$$Y_i^q \cdot te_i^q < t_{o,s}^a, \forall i \in N, \forall a, q \in K, \forall s \in J \tag{29}$$

$$Y_i^q \cdot ts_i^q < t_{o,s}^a, \forall i \in L, \forall a, q \in K, \forall s \in J \tag{30}$$

$$Y_i^b \cdot ts_i^b < t_{o,s}^a, \forall i \in N, \forall a, q \in K, \forall s \in J \tag{31}$$

$$Y_i^b \cdot te_i^b < t_{o,s}^a, \forall i \in L, \forall a, q \in K, \forall s \in J \tag{32}$$

Constraint (29)–(32) are time constraints for the assignment of AGV to QC and ASC container tasks. Constraints (29) and (30) represent the time relationship between the AGV and QC handover container tasks, ensure that the time for the QC and AGV to transfer the container is less than the time that the AGV leaves the node. Constraints (31) and (32) represent the time relationship between the AGV and ASC handover container tasks, ensure that the time for the ASC and AGV to transfer the container is less than the time that the AGV leaves the node.

The above-integrated scheduling problem and path planning problem are NP-hard problems, which are difficult to solve using optimization software. Therefore, in the next section, we design two optimization algorithms to solve this problem.

Algorithm

Genetic algorithm is a common heuristic algorithm for solving various optimization problems. We developed two optimization methods based on conflict resolution strategies for the model established in the previous section: bi-level adaptive genetic algorithm based on conflict resolution strategy (CRS-BAGA) and bi-level genetic algorithm based on conflict resolution strategy (CRS-BGA).

Bi-level adaptive genetic algorithm based on conflict resolution strategy

In the previous section, we established a bi-level programming model. The upper model is the integrated scheduling of QC, AGV, and ASC, and the lower is the path planning of AGV. In order to solve the conflict and congestion problem of AGV, we developed a bi-level adaptive genetic algorithm based on the conflict resolution strategy. The upper algorithm module is an improved adaptive genetic algorithm, and the lower algorithm module is a path optimization based on the conflict resolution strategy. The algorithm flowchart is shown in Figure 4.

Upper algorithm module

Chromosome coding. The coding method of this paper is real coding. We designed a three-level-structure chromosome. The first level represents the container number. The second level represents the container allocation of QC. The third level represents the container allocation of ASC. For example, we assume that there are eight containers. Numbers 1–4 represent import containers. Numbers 5–8 represent export containers. The number of QC, AGV, and ASC are 2, 4, and 2. Then, the chromosome coding is shown in Figure 5.

There is no difference between AGVs. Therefore, the task order of AGV acquires a set of import and export containers according to the number order of AGV. Taking export container 6 as an example, it is handled by the ASC 1, and then transported by AGV 2. Finally, the loading task is completed by QC 1.

Tabu list. The tabu list is an important part of tabu search, and it can record the information of local optimal solution. Therefore, in order to avoid the search loop and fall into local optimum, we introduce a tabu list to store the new individuals which are searched.

Calculation of fitness value. The fitness function is a criterion for judging the virtues or defect degree of individuals in a population. The design of the fitness function can directly affect the performance of the algorithm. In order to evaluate individuals more simply, we use the reciprocal of the objective function as the fitness function.

Genetic operation. **(1) Selection**

This article used the roulette selection method to achieve the selection operation. The probability of each individual which is selected is equal to the proportion of its fitness value in the total fitness value of the entire population.

(2) Crossover

This paper uses an inversed order cross to achieve crossover operation. First, two random gene locations are determined on the

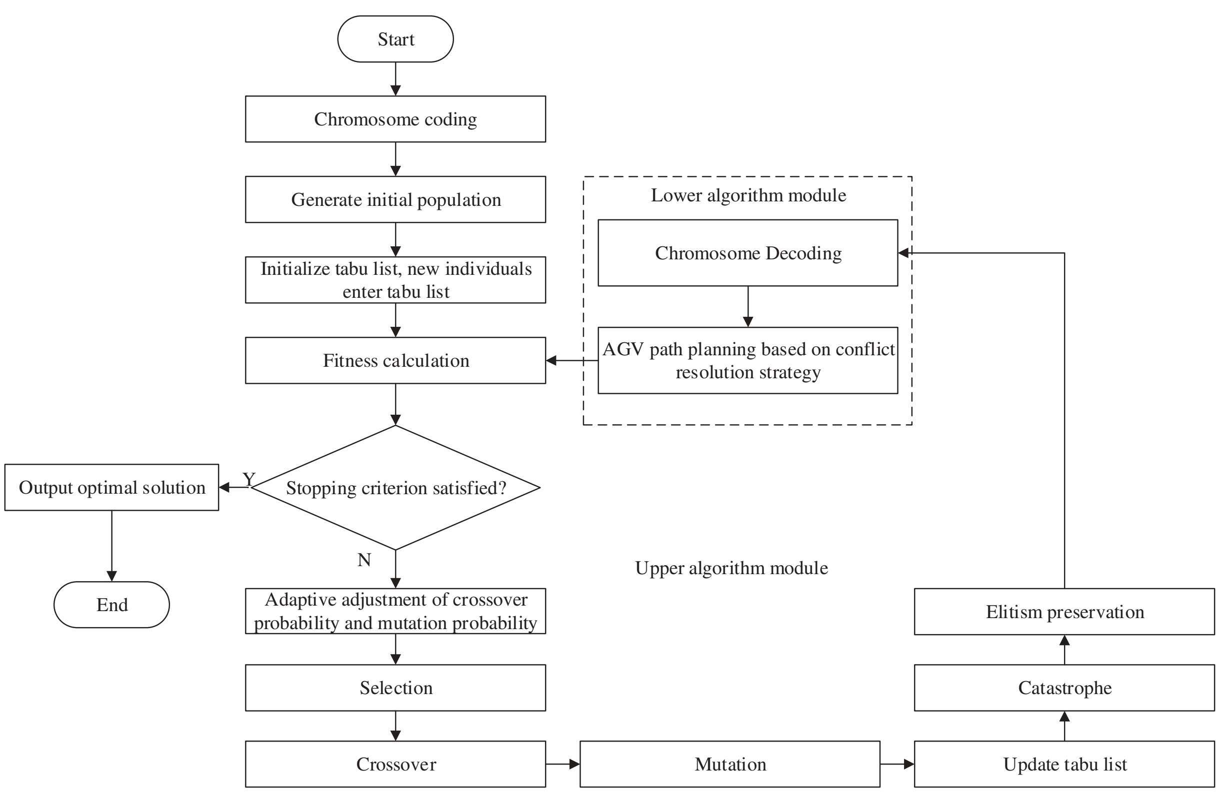

Figure 4. Flowchart of CRS-BAGA.

Container nunber	5	1	6	3	8	2	4	7
QC nunber	1	2	1	2	1	2	1	2
ASC nunber	1	2	1	2	1	2	1	2

Figure 5. Chromosome coding.

parent chromosomes X_1 and X_2. Then, retain the gene fragment X'_{11} and X'_{12} between two gene locations, and other gene locations are temporarily replaced with 0. Finally, reverse insert the gene of the parent chromosome X_1 into X'_{12}, and reverse insert the gene of the parent chromosome X_2 into X'_{11}. Therefore, new generations of chromosomes X_{11} and X_{12} are generated. The crossover process is shown in Figure 6.

(3) Mutation

This paper used a two-point crossover to achieve the mutation operation. Two gene locations need to be randomly determined on the parent chromosome, and the gene values at the locations of the two genes are exchanged to generate a new chromosome. The mutation process is shown in Figure 7.

(4) Parameter control

This paper adopts an adaptive adjustment strategy (Srinivas and Patnaik (1994)) (Srinivas and Patnaik 1994) to dynamically adjust the crossover probability P_c and mutation probability P_m. The basic formula is as follows.

X_1	5	1	6	3	8	2	4	7
	1	2	1	2	1	2	1	2
	1	2	1	2	1	2	1	2
X_2	4	5	7	1	6	3	2	8
	2	1	1	1	2	1	2	2
	1	1	2	1	2	2	2	1
X'_{11}	0	1	6	3	8	0	0	0
	0	2	1	2	1	0	0	0
	0	2	1	2	1	0	0	0
X'_{12}	0	0	7	1	0	0	0	0
	0	0	1	1	0	0	0	0
	0	0	2	1	0	0	0	0
X_{11}	2	1	6	3	8	7	5	4
	2	2	1	2	2	1	1	1
	1	2	1	2	1	2	1	2
X_{12}	4	2	7	1	8	3	6	5
	2	1	1	1	2	1	2	2
	2	1	2	1	2	1	2	1

Figure 6. Crossover operation.

Parent	2	1	6	3	8	7	5	4
	2	2	1	2	2	1	1	1
	1	2	1	2	1	2	1	2
Child	2	1	7	3	8	6	5	4
	2	2	1	2	2	1	1	1
	1	2	2	2	1	1	1	2

Figure 7. Mutation operation.

$$P_c = \begin{cases} k_1 \frac{f' - f_{avg}}{f_{\max} - f_{avg}}, f' \geq f_{avg} \\ k_2, f' < f_{avg} \end{cases} \tag{34}$$

$$P_m = \begin{cases} k_3 \frac{f' - f_{avg}}{f_{\max} - f_{avg}}, f \geq f_{avg} \\ k_4, f < f_{avg} \end{cases} \tag{35}$$

f_{max} represents the maximum fitness of the population. f_{avg} represents the average fitness of the population. f' represents the better individual fitness value. f represents the fitness value of the variant individual. k_1, k_2, k_3 and k_4 are constants between 0 and 1. The value of k_1 and k_3 has less influence on the parameters. So according to the general operating experience, the value is 0.5. For k_2, the largest value number should be used to change the individual chromosomes. Therefore, the value is 0.7. Due to the probability of variation is generally smaller, the value of k_4 is 0.1.

Catastrophe. In order to avoid the phenomenon of premature convergence, this paper introduces the catastrophe operator. The catastrophe can destroy the entire population except the optimal individual and produce new individuals. It is possible to increase the diversity of the population while maintaining the same population size. Therefore, our catastrophic implementation strategy is as follows. First, preserve the current best individual and destroy other individuals. Second, randomly generate new individuals and proceed to the next stage of evolution.

Elitism preservation. The elite individual is the optimal individual obtained from the evolution of the entire population to the current level. This paper used an elite retention strategy to preserve the best individuals that have evolved to the current level and avoid genetic operation. Therefore, the optimal individual is not destroyed by genetic operation, and it can also improve the global convergence ability of the algorithm.

Stopping criterion. In this paper, the maximum iteration number is used as the termination condition of the algorithm.

Lower algorithm module

The lower algorithm module is used for collision-free path planning of the AGV. The algorithm flowchart is shown in Figure 8. After the chromosome is decoded, a scheduling scheme of the device is generated. We can get the order of the containers and the corresponding working equipment. After the QC and ASC are determined, the start and end points of the AGV can be determined. According to the start and end points, we can plan the shortest path

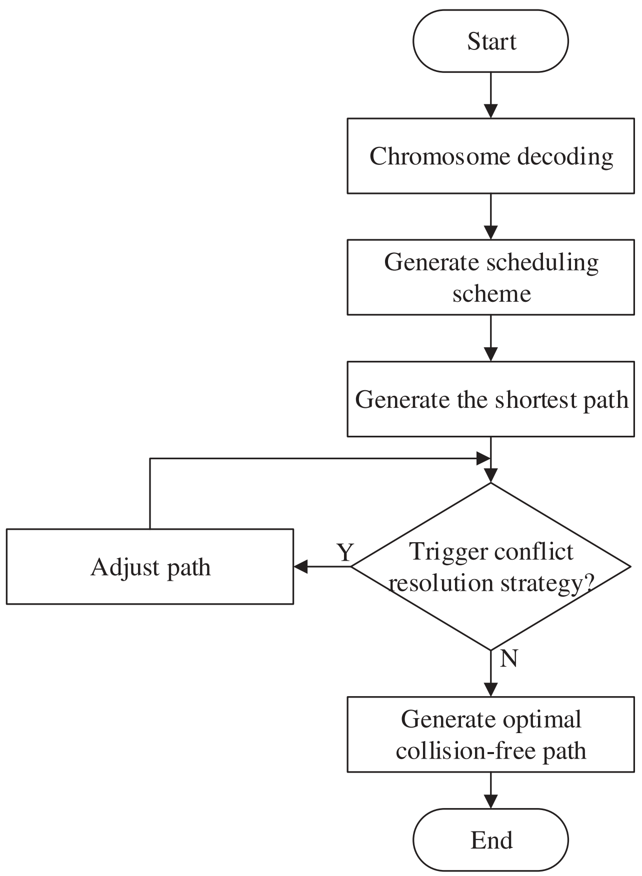

Figure 8. Lower algorithm module.

of the AGV. Taking the import container 3 in Figure 5 as an example. The imported container 3 is unloaded from the ship by QC 2. Then transported by AGV 2. Finally, enter the stacking area by ASC 2. We assume that the transfer points for QC 2 and ASC 2 are node 8 and node 15. Then, the start and end points of the AGV are node 8 and node 15. Therefore, the shortest path of AGV 3 is 8-9-10-11-12-13-14-15.

We used the waiting strategy as a conflict resolution strategy. The flow chart of the conflict resolution strategy is shown in Figure 9. Each time the AGV arrives at a node, it is determined whether the front node is occupied by another AGV. If the front node is occupied by another AGV, the AGV stops at the current node. If the front node is not occupied by another AGV, the AGV continues to move. We assume that when AGV 3 is running to node 9, node 10 is occupied by other AGVs. AGV 3 stops at node 9 and adjusts the path to 8-9-9-10-11-12-13-14-15.

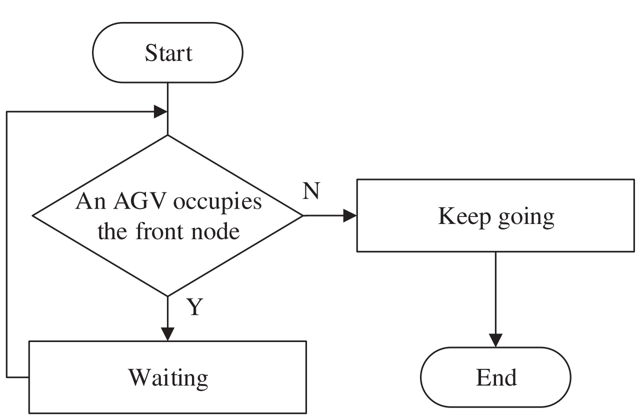

Figure 9. Conflict resolution strategy.

Bi-level genetic algorithm based on conflict resolution strategy

We proposed a second optimization method to solve the comprehensive scheduling problem: bi-level genetic algorithm based on conflict resolution strategy. The upper algorithm module of CRS-BGA is an improved genetic algorithm. This is the difference between CRS-BGA and CRS-BAGA. We used the same chromosome coding, fitness value calculation, and genetic operation as CRS-BAGA. Similarly, the lower algorithm modules are the same. The algorithm flowchart is shown in Figure 10.

Numerical experiments

In this section, we tested the effectiveness of CRS-BAGA and CRS-BGA for small-size problems and large-size problems. Two algorithms were implemented in maltabR2016a, and all experiments were performed on a computer with Intel Core TM i5-8300 H@2.3 GHz and 8 GB RAM under a Windows 10 operating system. Due to the heuristic algorithm is random, it has an impact on the experimental results. Therefore, in order to reduce the deviation caused by the randomness of algorithm, we referred to (Lau and Zhao 2008, Lau and Zhao 2008, He et al 2015, He et al. 2015, Yang, Zhong and Dessouky 2018, Yang, Zhong, and Dessouky 2018) for the processing method of experimental results of combinatorial optimization problems. Considered the processing method from Truong et al. (2015) on determining the number of experimental runs. We synthetically considered each size problems were run 20 times and counted average result.

Parameter settings

(1) The number of containers ranges from 8 to 300, where 8–30 were considered for small-size problems and 50–300 for large-size problems. We also considered the number of QCs and ASCs in the range 2–5, and the number of AGVs ranges from 4 to 10.

(2) The processing time of the QC followed uniform distribution of U (30, 80) seconds, and the processing time of the ASC followed uniform distribution of U (60, 100) seconds.

(3) The traveling network of the AGV is shown in Figure 2. It is a unidirectional rectangular network with a length of 200 m and a width of 150 m. And the horizontal speed of the AGV is 5 m/s.

(4) The parameters of the algorithm are set as follows, crossover rate $P_c = 0.85$, mutation rate $P_m = 0.01$, population size is 100 and the maximum generation is 200.

Results for small-sized problems

Twelve small-sized experiments were examined in this section, where the number of containers varied from 8 to 30. Table 1 shows the comparison of results between CRS-BAGA and CRS-BGA. It can be seen from the experimental results that CRS-BGA has better performance in terms of calculation speed. The computation time of CRS-BAGA ranged from 113.058 s to 409.988 s and that of CRS-BGA ranged from 104.209 s to 391.03 s. In addition, we can find that CRS-BAGA has better solving ability and can get better solutions. In the small-size problems, the average gap rate of OFV was 2.83%.

Figure 11 shows the convergence of the two algorithms for the case with 16 containers, 2 QCs, 5 AGVs, and 2 ASCs. It can be seen that CRS-BGA converges around 20 generations, while CRS-BAGA

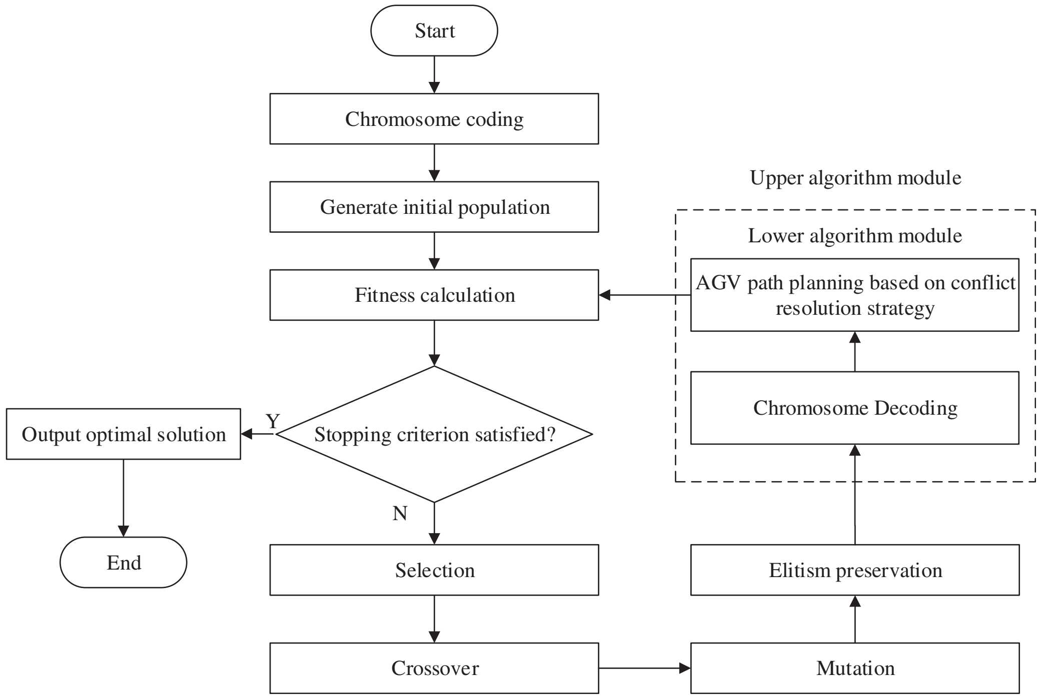

Figure 10. Flowchart of CRS-BGA.

Table 1. Results of computational experiments in small sizes.

			CRS-BAGA		CRS-BGA		
NO.	Containers	QCs-AGVs-ASCs	Computation time (s)	OFV(s)	Computation time (s)	OFV(s)	OFV gap rate (%)
1	8	2-4-2	113.058	3055	104.209	3152	3.18
2	10	2-5-2	190.447	3882	159.633	3923	1.06
3	10	2-6-2	211.347	3714	178.553	3811	2.61
4	16	2-4-2	250.38	6758	234.805	6917	2.35
5	16	2-5-2	264.016	6429	253.662	6746	4.93
6	16	2-6-2	278.443	6315	266.28	6449	2.12
7	20	2-4-3	289.811	7929	273.209	8121	2.42
8	20	2-5-3	296.41	7440	287.049	7839	5.36
9	20	2-6-3	309.933	7142	295.471	7459	4.44
10	30	2-4-3	352.401	12599	340.276	12618	0.15
11	30	2-5-3	368.202	12453	357.447	12637	1.48
12	30	2-6-3	409.988	12027	391.03	12486	3.82

converges around 180 generations. This shows that CRS-BGA has better convergence. And the convergence speed is high.

Results for large-sized problems

Eighteen large-sized experiments were examined in this section, where the number of containers varied from 50 to 300. Table 2 shows the comparison of results between CRS-BAGA and CRS-BGA. From the experimental results, we got the same conclusions as the small-size problem. CRS-BAGA is able to provide a solution that is closer to the optimal solution, but requires longer computation time. CRS-BGA is able to complete calculations faster, but the search ability is poor. On the other hand, we found that increasing the number of containers in the same equipment configuration resulted in an increase in OFV (such as experiments 15 and 17). In the case of the same number of containers, increasing the number of equipment will result in a decrease in OFV (such as experiments 16–18, experiments 19–21, etc.). This means completing the job faster. This shows that a reasonable number of equipment is critical in the actual operation. As the number of devices increases, it takes longer to get a solution.

Figure 12 shows the convergence of the two algorithms for the case with 150 containers, 3 QCs, 9 AGVs, and 5 ASCs. We can see that CRS-BGA has a faster convergence speed, and CRS-BAGA has better search ability.

Through the above experiments, the effectiveness of the proposed two algorithms is verified. In terms of algorithm performance, CRS-BAGA has better optimization ability, but the calculation speed and convergence speed are lower. The CRS-BGA has faster calculation speed and convergence, but the search ability is poor.

In the path planning of AGV, we adopted the parking waiting strategy as a conflict resolution strategy. We drew the routing result of AGVs, which start with the starting task of each AGV. Figure 13 shows the AGV routing result for the case with 16 containers, 2 QCs, 5 AGVs, and 2 ASCs, which without considering the conflict resolution strategy. It can be seen that the conflict problem appeared for AGV 1 and AGV 3 at 155 s in node 13, and then the congestion problem appeared for AGV 1 and AGV 3 within 155 s-165 s. And AGV 4 conflicts with AGV 1 and AGV 3 at 162 s in node 14. In addition, we have also discovered other conflicting congestion issues. For example, the conflict problem

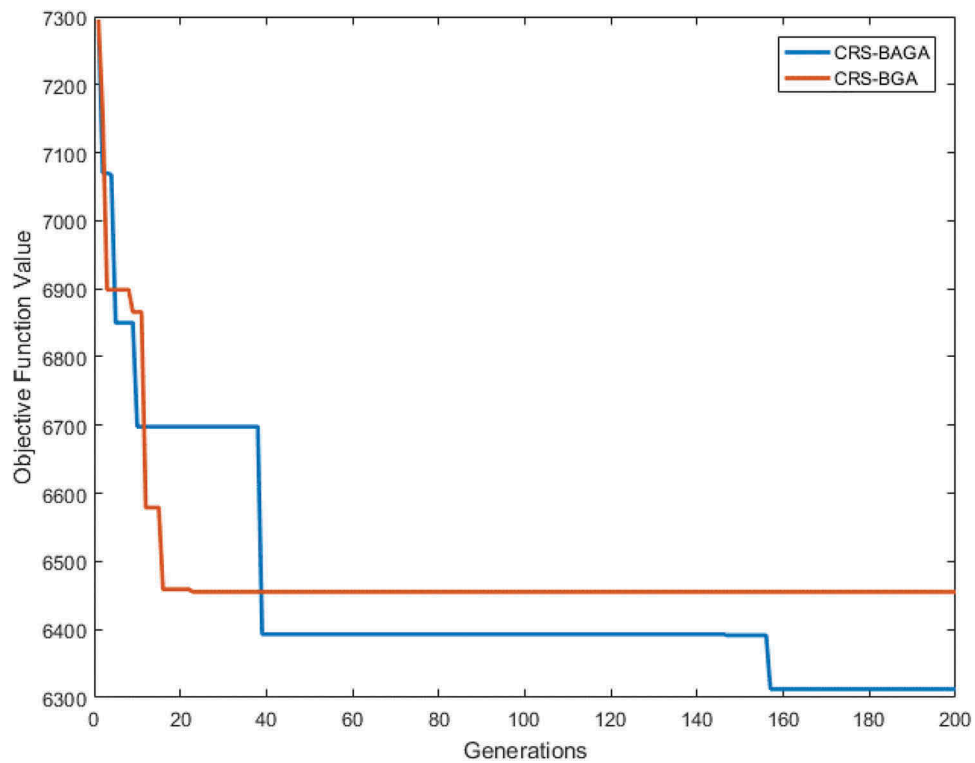

Figure 11. Typical convergence of AGA and GA for the case with 16 containers, 2 QCs, 5 AGVs, and 2 ASCs.

Table 2. Results of computational experiments in large sizes.

			SSC-BAGA		SSC-BGA		
NO.	Containers	QCs-AGVs-ASCs	Computation time (s)	OFV(s)	Computation time (s)	OFV(s)	OFV gap rate (%)
13	50	3-4-3	421.066	18886	403.79	20003	5.91
14	50	3-6-3	437.144	18549	430.228	19279	3.94
15	50	4-7-3	448.76	20249	442.116	20322	0.36
16	80	4-6-3	722.017	35848	698.442	36059	0.59
17	80	4-7-3	743.243	35458	730.066	35896	1.24
18	80	4-8-3	761.442	34754	750.328	35139	1.11
19	100	3-8-4	804.98	45361	791.343	46212	1.88
20	100	3-9-4	832.02	44162	804.554	45233	2.43
21	100	3-10-4	846.201	43634	819.957	45041	3.22
22	150	3-8-5	976.441	73735	930.266	74825	1.48
23	150	3-9-5	1012.078	72229	969.211	74618	3.31
24	150	4-8-4	1179.42	84399	1141.022	85758	1.61
25	200	4-9-4	1593.668	114927	1567.313	116882	1.7
26	200	4-10-4	1689.02	112348	1670.366	114643	2.04
027	200	4-9-5	1797.606	111197	1775.329	111534	0.3
28	300	4-10-5	1982.477	166271	1967.05	167927	0.99
29	300	5-9-5	2083.996	160063	2058.431	163,025	1.85
30	300	5-10-5	2292.211	158349	2257.072	161815	2.19

appeared for AGV 1 and AGV 4 at 288 s in node 6. And the congestion problem appeared for AGV 1 and AGV 3 within 385 s-485 s.

Figure 14 shows the AGV routing result for the case with 16 containers, 2 QCs, 5 AGVs, and 2 ASCs, which considering the conflict resolution strategy. It can be seen that there is no conflict or congestion in the AGV routing result. This shows that the AGV path planning method based on the conflict resolution strategy is effective.

Conclusions

In this paper, we consider the integrated scheduling problem of QC, AGV, and ASC and the conflict-free path planning problem of AGV. We studied the more general automated container terminal layout and operation process, and based on its characteristics, established a new bi-level programming model. The upper model is an integrated scheduling of QC, AGV, and ASC with the aim of minimizing the maximum makespan of all containers. The lower model is the path planning of the AGV with the aim of minimizing the travel distance of the AGV. To solve the model, we have designed two optimization algorithms: bi-level adaptive genetic algorithm based on conflict resolution strategy (CRS-BAGA) and bi-level genetic algorithm based on conflict resolution strategy (CRS-BGA). CRS-BGA introduced elitism preservation strategy to ensure global convergence of the algorithm. CRS-BAGA introduced adaptive genetic algorithm to improve the convergence accuracy and speed of the algorithm. While maintaining the diversity of the

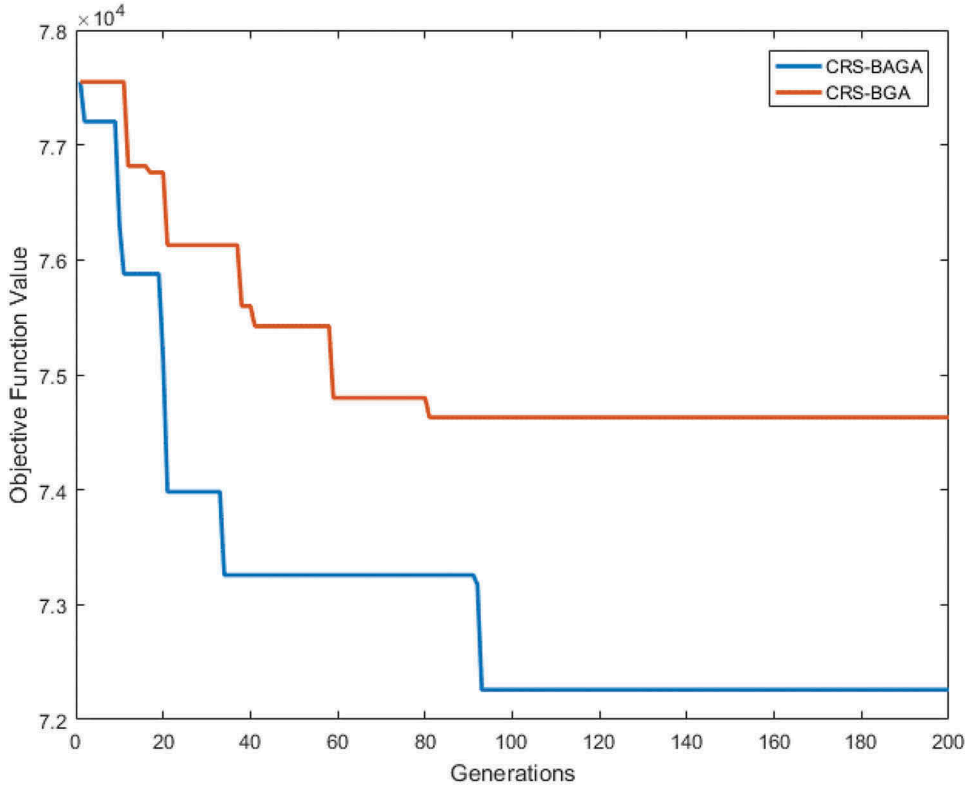

Figure 12. Typical convergence of AGA and GA for the case with 150 containers, 3 QCs, 9 AGVs, and 5 ASCs.

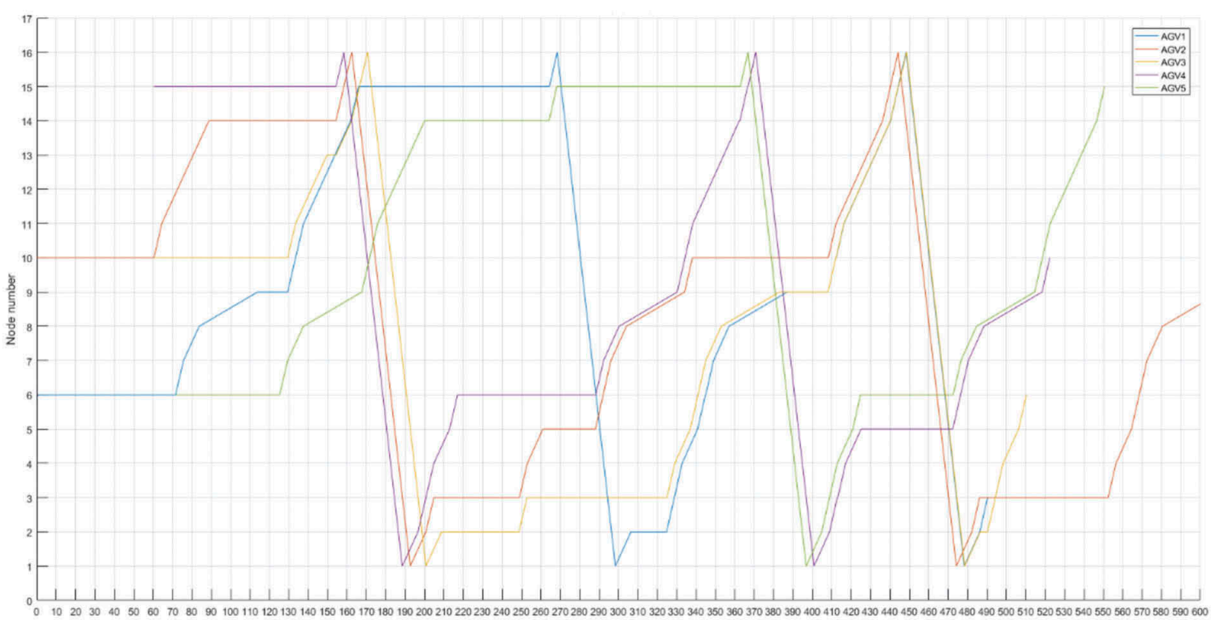

Figure 13. AGV routing result for the case with 16 containers, 2 QCs, 5 AGVs, and 2 ASCs without considering the conflict resolution strategy.

population, it also ensures the convergence of the genetic algorithm. In addition, the taboo list and the catastrophe operator were introduced to prevent the algorithm from appearing cyclic search and premature convergence.

There are many experiments to verify the effectiveness of the model and algorithms. By solving the model, we can get the integrated scheduling scheme and the conflict-free path of the AGV. The experimental results show that the proposed two optimization algorithms can provide a good solution for small-size problems and large-size problems. CRS-BAGA has better search ability to find approximate optimal solution, but it is poor in terms of calculation speed and convergence. CRS-BGA has faster calculation speed and convergence, but has poorer search ability. Both optimization algorithms have their advantages and disadvantages.

From the perspective of academic theory, this paper considers the synchronization loading and unloading mode of automated container terminals, and studies the integrated scheduling problems

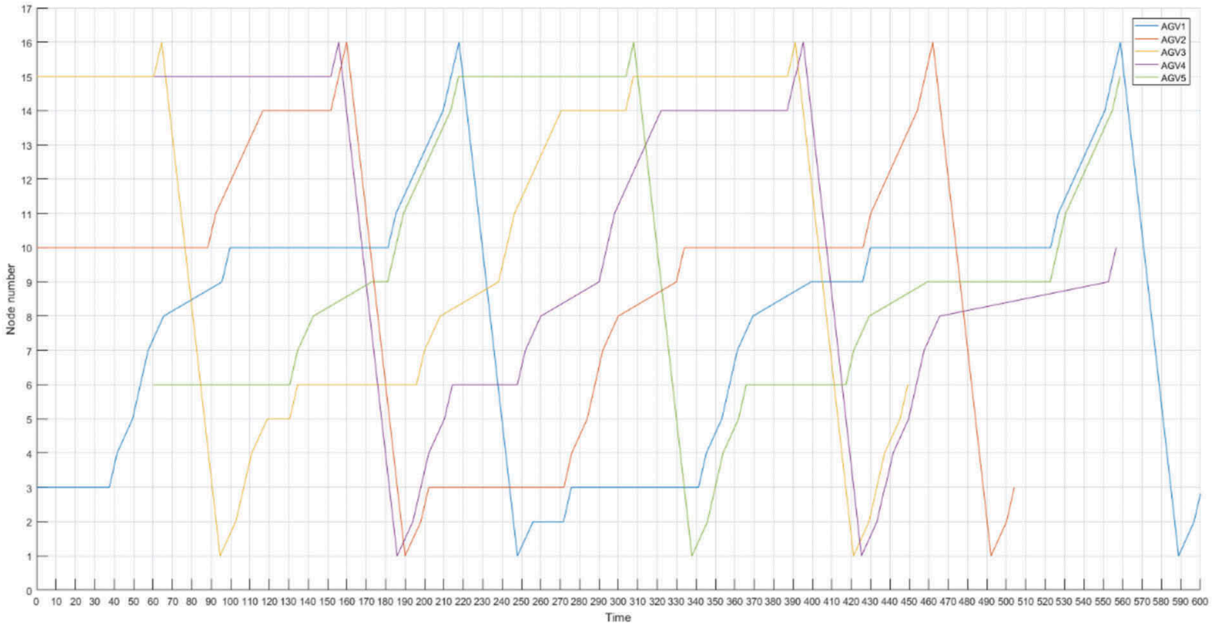

Figure 14. AGV routing result for the case with 16 containers, 2 QCs, 5 AGVs, and 2 ASCs considering the conflict resolution strategy.

of QC, AGV, and ASC. A mathematical model suitable for most automated container terminal layout and operation flow was established. Two optimization algorithm solving models based on conflict resolution strategy were designed for the model characteristics. From the perspective of actual operation, this paper adopts the waiting strategy as conflict resolution strategy to solve the conflict and congestion problem of AGV.

However, as the volume of automated container terminals increases, more efficient scheduling decisions and algorithms will be required. In future research, the specific allocation problem of the container can be further considered. In addition, considering the uncertain factors into the integrated scheduling problem is also a future research direction. And it is also worth studying to develop other precise algorithms and heuristic algorithms for the model designed in this paper.

Disclosure Statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Key Research and Development Project under [grant number 2017YFE0134600].

References

Cao, J. X., D. H. Lee, J. H. Chen, and Q. Shi. 2010. "The Integrated Yard Truck and Yard Crane Scheduling Problem: Benders' Decomposition-based Methods." *Transportation Research Part E* 46 (3): 0–353. doi:10.1016/j.tre.2009.08.012.

Chang, Y., X. Zhu, B. Yan, and L. Wang. 2017. "Integrated Scheduling of Handling Operations in Railway Container Terminals." *Transportation Letters* 11 (7):402–412.

Chen, L., and A. Langevin, LUZQ. 2013. "Integrated Scheduling of Crane Handling and Truck Transportation in a Maritime Container Terminal. Eur J Oper Res". *European Journal of Operational Research* 225 (1): 142–152. doi:10.1016/j.ejor.2012.09.019.

He, J., Y. Huang, W. Yan, and S. Wang. 2015. "Integrated Internal Truck, Yard Crane and Quay Crane Scheduling in a Container Terminal considering Energy Consumption." *Expert Systems with Applications* 42 (5): 2464–2487. doi:10.1016/j.eswa.2014.11.016.

Jun-Jun, L., X. Bo-Wei, P. Octavian, Y. Yong-Sheng, and W. Hua-Feng. 2018. "Impact Analysis of Travel Time Uncertainty on Agv Catch-up Conflict and the Associated Dynamic Adjustment." *Mathematical Problems in Engineering* 2018: 1–11.

Kaveshgar, N., and N. Huynh. 2015. "Integrated Quay Crane and Yard Truck Scheduling for Unloading Inbound Containers." *International Journal of Production Economics* 159: 168–177. doi:10.1016/j.ijpe.2014.09.028.

Lau, H. Y. K., and Y. Zhao. 2008. "Integrated Scheduling of Handling Equipment at Automated Container Terminals." *Annals of Operations Research* 159 (1): 373–394. doi:10.1007/s10479-007-0262-5.

Lu, Y., and M. Le. 2014. "The Integrated Optimization of Container Terminal Scheduling with Uncertain Factors." *Computers & Industrial Engineering* 75: 209–216. doi:10.1016/j.cie.2014.06.018.

Luo, J., Y. Wu, and A. B. Mendes. 2016. "Modelling of Integrated Vehicle Scheduling and Container Storage Problems in Unloading Process at an Automated Container Terminal." *Computers & Industrial Engineering* 94: 32–44.

Małopolski., W. 2018. "A Sustainable and Conflict-free Operation of AGVs in A Square Topology." *Computers & Industrial Engineering* 126: 472–481. doi:10.1016/j.cie.2018.10.002.

Miyamoto, T., and K. Inoue. 2016. "Local and Random Searches for Dispatch and Conflict-free Routing Problem of Capacitated Agv Systems." *Computers & Industrial Engineering* 91: 1–9. doi:10.1016/j.cie.2015.10.017.

Moorthy, R. L., W. H. Guan, N. W. Cheong, and T. C. Piaw. 2003. "Cyclic Deadlock Prediction and Avoidance for Zone-controlled Agv System." *International Journal of Production Economics* 83 (3): 309–324. doi:10.1016/S0925-5273(02)00370-5.

Nishi, T., M. Ando, and M. Konishi. 2006. "Experimental Studies on a Local Rescheduling Procedure for Dynamic Routing of Autonomous Decentralized Agv Systems." *Robotics and Computer-integrated Manufacturing* 22 (2): 154–165. doi:10.1016/j.rcim.2005.02.010.

Nishi, T., Y. Hiranaka, and I. E. Grossmann. 2011. "A Bilevel Decomposition Algorithm for Simultaneous Production Scheduling and Conflict-free Routing for Automated Guided Vehicles." *Computers and Operations Research* 38 (5): 876–888. doi:10.1016/j.cor.2010.08.012.

Roy, D., A. Gupta, and R. B. M. De Koster. 2015. "A Non-linear Traffic Flow-based Queuing Model to Estimate Container Terminal Throughput with Agvs." *International Journal of Production Research* 54 (2): 472–493.

Srinivas, M., and L. M. Patnaik. 1994. "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms." *IEEE Transactions on Systems, Man and Cybernetics* 24 (4): 656–667. doi:10.1109/21.286385.

Tang, L., J. Zhao, and J. Liu. 2014. "Modeling and Solution of the Joint Quay Crane and Truck Scheduling Problem." *European Journal of Operational Research* 236 (3): 978–990. doi:10.1016/j.ejor.2013.08.050.

Truong, L. T., M. Sarvi, G. Currie, and T. M. Garoni. (2015). "How Many Simulation Runs are Required to Achieve Statistically Confident Results: A Case Study of Simulation-Based Surrogate Safety Measures." 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Spain, 274–278.

Xin, J., R. R. Negenborn, and G. Lodewijks. 2014a. "Energy-aware Control for Automated Container Terminals Using Integrated Flow Shop Scheduling and Optimal Control." *Transportation Research Part C: Emerging Technologies* 44: 214–230. doi:10.1016/j.trc.2014.03.014.

Xin, J., R. R. Negenborn, and G. Lodewijks. 2014b. "Trajectory Planning for Agvs in Automated Container Terminals Using Avoidance Constraints: A Case Study." *IFAC Proceedings Volumes* 47 (3): 9828–9833. doi:10.3182/20140824-6-ZA-1003.01306.

Yang, Y., M. Zhong, and Y. Dessouky. 2018. "An Integrated Scheduling Method for AGV Routing in Automated Container Terminals." *Computers & Industrial Engineering* 126: 482–493. doi:10.1016/j.cie.2018.10.007.

Zhen, L., S. Yu, S. Wang, and Z. Sun. 2016. "Scheduling Quay Cranes and Yard Trucks for Unloading Operations in Container Ports." *Annals of Operations Research* 273: 455–478.