Transportation Research Part B 179 (2024) 102871

Contents lists available at ScienceDirect

Transportation Research Part B

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

Integrated scheduling optimization of AGV and double yard cranes in automated container terminals

Xiaoju Zhang [a], Huijuan Li [a], Jiuh-Biing Sheu [b,*]

[a] *Business School, Beijing Technology and Business University, Beijing, China*
[b] *Department of Business Administration, National Taiwan University, No.1, Sec.4, Roosevelt Road, Taipei 10617, Taiwan, ROC*

ARTICLE INFO

Keywords:
Automated container terminal
Double yard cranes
Integrated scheduling optimization
B&B-PSO algorithm

ABSTRACT

Double yard cranes in one yard block refer to two yard cranes that run in different rails and can pass each other. They can both reach to the seaside and landside of the yard blocks. But interference between the double yard cranes occurs during picking up or putting down containers, which has to be solved when assigning tasks to the double yard cranes. AGVs assignment have also to be considered when scheduling yard cranes. This paper focuses on the integrated optimization of AGV and double yard cranes in automated container terminals, aiming to increase terminal efficiency through the coordination of multiple sub-operations. A mixed integer programming model is developed, which considers the interference of the double yard cranes. The objective is to minimize the completion time of all the tasks. To solve the problem, a branch and bound (B&B) based method is designed. Specifically, a heuristic algorithm is used to assign container tasks to AGVs, and B&B method is utilized to assign container tasks to double yard cranes. This paper derives pruning strategies and acceleration strategies based on model analysis to improve the efficiency of the algorithm. The effectiveness and validity of the proposed model and algorithm is verified through small-scale and large-scale experiments. Results are compared with that of commercial solver Gurobi.

1. Introduction

Container terminals are important hubs for global transportation, and terminal operators are sparing no efforts to improve the efficiency of container terminal operations and shorten the time of ships in port. With the development of technology, automated container terminal (ACT) has been built in many countries, such as the ECT terminal in Rotterdam, the CTA terminal in Hamburg, the Yuanhai terminal in Xiamen etc. Studies have shown that ACTs can effectively increase the terminal efficiency, decrease labor force and therefore reduce operational cost (McKinsey andCompany, 2018; Chen et al., 2020).

A typical ACT can be divided into three areas: the quayside, the transfer area and the yard blocks. As shown in Fig. 1, the quayside refers to the area where containers are being loaded/unloaded to/from vessels by quay cranes (QCs). The transfer area is designed for transporting containers between the quayside and the yard blocks, and automated guided vehicles (AGVs) are a type of transport vehicles. The yard blocks are designed for temporary storage of containers, and yard cranes (YCs) are responsible for loading/unloading containers to/from yard blocks. One type of the layout of yard blocks is perpendicular, where the yard blocks are perpendicular to the quayside. AGVs unload/unload containers from the front of the yard blocks, and the other side of the yard blocks

* Corresponding author.
E-mail address: jbsheu@ntu.edu.tw (J.-B. Sheu).

https://doi.org/10.1016/j.trb.2023.102871
Received 10 January 2023; Received in revised form 22 September 2023; Accepted 30 November 2023
Available online 23 December 2023

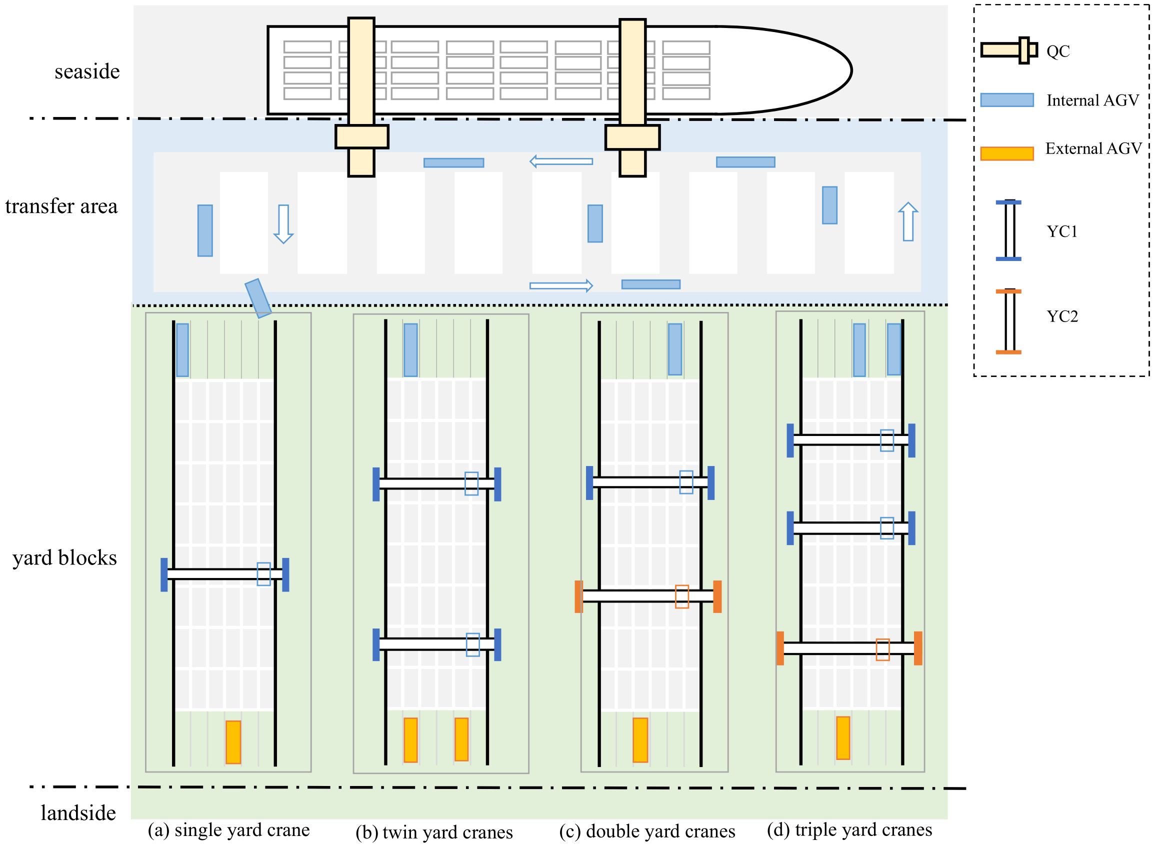


Fig. 1. The layout of an automated container terminal.

are used for external trucks to unload/load containers. In the ACTs, the configuration of YCs is generally divided into four cases (Speer and Fischer, 2017) as shown in Fig. 1: (a) single yard crane, where only one yard crane is responsible for both seaside and landside container retrieval/storage; (b) twin yard cranes, where two non-crossing yard cranes moves on the same rail are employed in one yard block, and one is responsible for the seaside container retrieval/storage and the other is responsible for the landside container retrieval/storage; (c) double yard cranes, where two crossover yard cranes moves on different rails, and both are responsible for both seaside and landside container retrieval/storage; (d) triple yard cranes, where two twin cranes and one larger crane moves on different rails. This paper focuses on the scheduling problem of double yard cranes. Double yard cranes have different heights and sizes, and the smaller yard crane can pass through the larger yard crane, and they cooperate to improve the yard efficiency. However, they may interfere with each other during the operations. For example, the smaller crane cannot pass through the large crane when the larger crane is dropping off or picking up containers. Therefore, the assignment of container tasks for the two cranes addressing the interference is a research challenge.

To minimize the handling time of AGVs at the yard blocks and reduce the total operation time, it is crucial to closely synchronize the arrival time of AGVs with the tasks assigned to the double yard cranes. However, the existing research has optimized the schedules of double yard cranes while neglecting the dependency with AGV operations. This paper studies the integrated scheduling problem of AGV and double yard cranes (ADCSP) in ACTs. We consider both the coordination problem of AGV and yard cranes, and the interference between the double yard cranes. A mixed integer programming model is proposed with the objective of minimizing the completion time, and a branch-and-bound algorithm based on particle swarm optimization (B&B-PSO) is designed to solve the model. The contributions of this paper are:

(1) This paper introduces a new optimization framework that jointly optimizes the task assignment for AGVs and double yard cranes. The proposed model solves the interference between the double yard cranes, which is very critical for double yard cranes scheduling problem. It also considers the impact of AGV arrivals on the yard scheduling. Existing research mainly focus on the double yard cranes while neglecting considering other terminal operations. To model the problem, this paper develops a mixed integer programming model, and the objective is to minimize the completion time of container tasks.
(2) This paper focuses on the assignment for the double yard cranes, which is based on the dual-cycle operations of QCs. AGVs are required to unload an import container after being loaded an export container by the yard cranes. In this way, the assignment of container tasks between the double yard cranes are more crucial than only loading/unloading operations. Based on the

characteristics of dual-cycle operations, we design an exact branch and bound algorithm to effectively solve the assignment for double yard cranes. Despite the probability of interference between the double yard cranes during operations, the utilization of double yard cranes can significantly improve the operation efficiency, especially in the case of dual-cycle operations.

(3) This paper designs several pruning and accelerate strategies based on model analysis to enhance the efficiency. For the double yard crane assignment, an upper bound based on a greedy algorithm is developed, and a lower bound for each node in the branches is proposed based on the routing problem. The upper bound and the lower bound can be used to improve the efficiency of branch and bound method. Additionally, we derive propositions from the proposed model and algorithm, which give suggestions for task assignment between the double yard cranes. Small-scale and large-scale experiments are used to show the effectiveness of the algorithm and accelerate strategies.

The remainder of the paper is as follows. The literature is reviewed in Section 2. A detailed description of the problem and the model are given in Section 3 and 4. The solution algorithm is described in Section 5, and numerical experiments are conducted in Sections 6. Finally, Section 7 concludes the paper and discusses future research directions.

2. Literature review

Many studies have been done to increase the container terminal efficiency. This section mainly reviews the most related research to this paper in equipment scheduling optimization problem and the yard crane scheduling problem.

2.1. Optimization of handling equipment

The handling equipment of container terminals mainly includes quay cranes, AGVs and yard cranes, and a lot of research has been done on the scheduling of these equipment. For example, Jung and Kim (2006) studied the scheduling problem of multiple quay cranes considering the intervention between adjacent quay cranes; Grunow et al. (2006) conducted a simulation study on the scheduling strategy of AGVs; Lee et al. (2010) studied the vehicle scheduling problem in a terminal transfer hub with the objective of minimizing the completion time of the terminal. Wang and Zeng (2022) studied the conflict-free scheduling and path planning problem of AGVs considering congestion. They designed a branch-and-bound method for solving the problem with the objective of minimizing the job completion time, and obtained a conflict-free AGV scheduling scheme. Container operations consist of a continuous multiple operational process, and the integrated optimization of the equipment is studied to improve the operational efficiency of the overall system. Lau and Zhao (2008) assigned different types of handling equipment in an integrated way by considering the constraints associated with the integrated operation between different types of handling equipment. Kaveshgar and Huynh (2015) constructed a mixed integer programming model to optimize the integrated scheduling problem of quay cranes and trucks, and designed a genetic algorithm and a greedy algorithm. Hop et al. (2021) studied the integrated scheduling of quay cranes and trucks and designed an adaptive particle swarm algorithm. Chen et al. (2020) studied the integrated scheduling problem of AGVs and yard cranes, and solved them by designing the Alternating Direction Method of Multipliers (ADMM) method and hybrid heuristic algorithm, respectively. Yang et al. (2018) studied the integrated optimization problem of quay cranes, AGVs and AGV-mates. Rui et al. (2015) developed a two-stage mixed integer programming model to optimize the loading and unloading sequence of export containers, and designed a two-layer genetic algorithm to reduce the operation time of the yard crane; Cao et al. (2010) studied the integrated scheduling problem of trucks and the yard cranes; Luo et al. (2015) minimized the berthing time of container vessels by scheduling the vehicles involved in loading and unloading operations. Jung and Kim (2006) investigated the problem of picking up export containers in a terminal with multiple quay cranes. Genetic algorithms and simulated annealing algorithms were designed to solve the problem considering the interference between adjacent quay cranes. Ahmed et al. (2021) investigated the joint scheduling problem of quay cranes, trucks and yard cranes under single-cycle and double-cycle, respectively. They developed simulation model to solve the problem, and found that double-cycle can reduce vessel turnaround time and cost compared with single-cycle.

2.2. Yard crane scheduling problem

The efficiency of yard blocks is critical to the overall terminal operation efficiency. It affects the turn-around rate of AGVs from the yard blocks to the quay side. Studies have been done to optimize the task sequence and the equipment assignment. The research includes the yard crane optimization in one yard block or among the multiple yard blocks. Froyland et al. (2008) researched the multiple yard cranes in one block. The research developed an integrated programming-based algorithm for minimizing the movement of multiple yard cranes in the terminal track interface. Lee et al. (2007) studied the problem of yard cranes scheduling in different yard blocks. The yard crane can be assigned to a different yard block to handle containers. A simulated annealing-based approach was used to minimize the total loading time for yard cranes. Chu et al. (2019) studied the joint scheduling of three yard cranes in two adjacent storage blocks, constructing a mixed integer planning model and solving it using a heuristic algorithm.

There are several studies optimizing the twin yard cranes in one yard block. Zhou et al. (2009) studied the non-crossing yard crane scheduling problem. The objective is to search for a loading plan that completes all the containers before the due date. If there is not such a schedule, the paper minimizes the maximum tardiness, the competition time or the number of tardy delays. Gharehgozli et al. (2015) abstracted the twin ASC scheduling problem into a multiple asymmetric generalized TSP with priority constraints, and designed an adaptive large neighborhood algorithm to solve it. The interference between the two cranes is considered. Hu et al. (2016) developed three models to optimize the twin automated straddle carriers, and the interference are also modelled by analyzing the

Table 1
Comparisons of the related literature.

Publication	Crane type	Number of blocks	Type of operations	Other operations	objectives	Safety distance	interference	Solution method
Froyland et al. (2008)	MYC	one	ULS	No	Minimize the movements of cranes	Yes	No	Integer programming-based heuristic
Chu et al. (2019)	MYC	two	ULS	No	Minimize the total flow time	No	Yes	Heuristic algorithm
Lee et al. (2007)	TYC	two	LS	No	Minimize the makespan	No	No	Simulated annealing
Zhou et al. (2009)	TYC	one	LS	No	Tardiness time	Yes	No	
Gharehgozli et al. (2015)	TYC	one	ULS	No	Minimize the makespan	Yes	No	Adaptive large neighborhood search heuristic
Hu et al. (2016)	TYC	one	ULS	No	Minimize the makspan	Yes	No	Genetic algorithm
Kress et al. (2019)	TYC	one	ULS	No	Minimize the berthing time of vessels	Yes	No	Heuristic algorithm
Stahlbock and Voß (2010)	DYC	one	ULS	No	Minimize the weighted earliness and lateness and empty travels	No	Yes	Extended simulated annealing
Saanen and Valkengoed (2005)	TYC, DYC	one	ULS	No	Comparison	Yes	Yes	Simulation
Cao et al. (2008)	DYC	one	LS	No	Minimize the makespan	No	Yes	Heuristic algorithm
Vis and Carlo (2010)	DYC	one	ULS	No	Minimize the makespan	No	Yes	Simulated annealing
Nossack et al. (2018)	DYC	one	ULS	No	Minimize the makespan	No	Yes	Branch and cut
Li et al. (2022)	TYC	one	ULS	No	Minimize the makespan	No	Yes	Heuristic algorithm
Dorndorf and Schneider (2010)	triple	one	ULS	No	Maximize the productivity	No	Yes	Simulation
Xu et al. (2021)	DYC	one	ULS	Multiple operations	Minimize the makespan	No	Yes	Heuristic algorithm
This paper	DYC	one	ULS	AGV	Minimize the makespan	No	Yes	Branch and bound based algorithm

TYC-twin yard cranes, DYC-double yard cranes, MYC-multiple yard cranes, US-unloading ships, LS-loading ships, ULS-loading and unloading ships.

minimal temporal intervals between the consecutive tasks. The objective is to minimize the makespan, and an exact algorithm and a heuristic algorithm are designed to solve the model. Practical insights are concluded from the model and experiment results, which can be used to real-world automated container terminals. Kress et al. (2019) studied the scheduling problem with two yard cranes on the same rails in a yard block, and proposed a related beam search heuristic algorithm combined with dynamic programming (DP) with the objective of minimizing the berthing time of vessels. The effectiveness of the algorithm is compared with results of CPLEX.

To answer which type of yard crane is better, Saanen and Valkengoed (2005) compared three different configurations of yard cranes: single crane, double yard cranes and twin yard cranes. The authors select several performance indicators, such as operational cost, efficiency, investment cost et al., and use a simulation method to obtain the results. Cao et al. (2008) proposed an integer programming to solve the double crane problem in the yard blocks. They developed a greedy algorithm, a simulated annealing algorithm and a combined heuristic algorithm to solve the problem. Small-scale and large-scale experiments are conducted to show the effectiveness of the algorithms. Vis et al. (2010) researched the double yard crane scheduling problem, and the interference between the yard cranes is considered. They use a B&B algorithm to obtain an exact solution. The double yard cranes scheduling problem is regarded as a single yard crane scheduling problem to obtain a lower bound. For large scale problem, they used a simulated-annealing based heuristic to solve it. Stahlbock and Voß (2010) proposed a priority-based heuristic and simulated annealing algorithms to solve double yard cranes scheduling problem. The objective is to minimize the total weighted earliness and lateness of the tasks and the empty distance of cranes. The real data from CTA is used to conduct an experiment analysis. Nossack et al. (2018) researched on the container dispatching and conflict-free yard crane routing problem. The objective is to assign container tasks to yard crane and avoid interferences during operations. A branch and cut algorithm is designed to solve the proposed model. Li et al. (2022) investigated the joint scheduling problem of twin yard cranes in a storge block based on dynamic cut-off Time, and proposed a PSO and a local rescheduling strategy (LRPSO) to solve it. Dorndorf and Schneider (2010) focused on triple yard cranes scheduling problem in a yard block. The transfer zones are set at both ends of the block. The solution method dynamically creates crane schedules when container tasks are completed or new container tasks are requested. Xu et al. (2021) studied the scheduling problem of dual trolley quay cranes, AGVs and dual cantilever rail cranes under U-shaped in automated container terminals. The comparisons of the related literature are shown in Table 1.

We can find the overview of problems addressed and advancements made in the related literature from Table 1. This paper focuses on the optimal assignment of container tasks for double yard crane, taking into consideration of AGV operations and dual-cycle operations. Upon analyzing the characteristics of the model, we design a branch and bound based algorithm to solve the problem. The

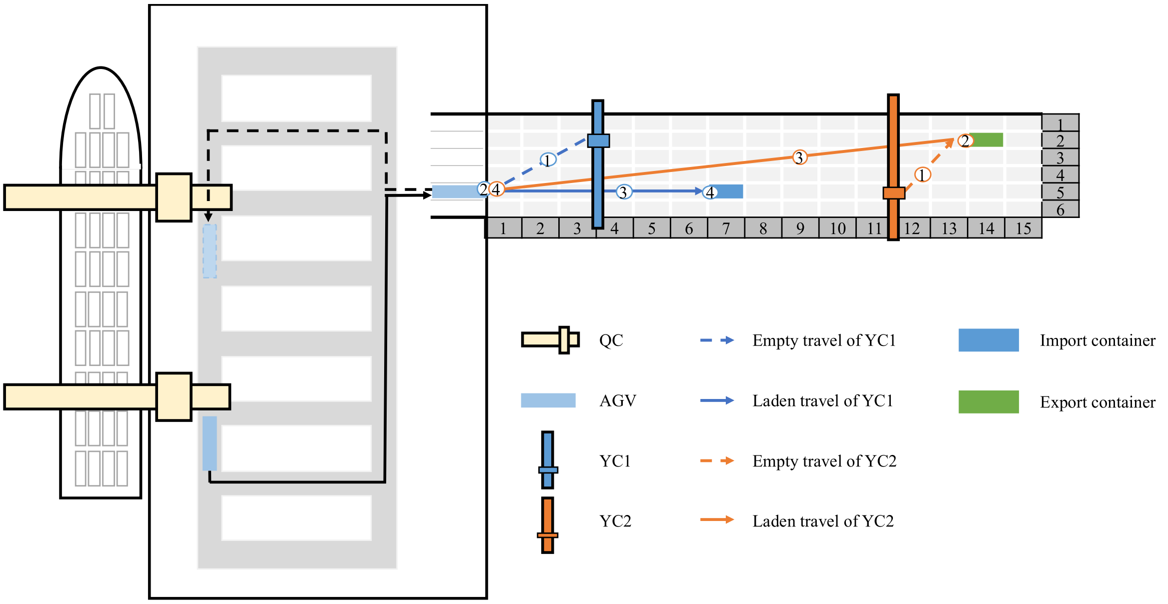


Fig. 2. The operation in an automated container terminal with double yard cranes.

characteristics of the research problem addressed in this paper are: (1) Our research considers the dynamics of AGV operations, which is crucial when optimizing the task assignment of double yard crane. The aim of this paper is to minimize the overall completion time of all container tasks, thereby improving the operation efficiency of container terminals. (2) The problem considers the dual-cycle operations, which involve unloading import containers and loading export containers at the yard block. The utilization of double yard crane can further improve the efficiency of AGVs and yard cranes, as being proved in the subsequent sections of this paper. (3) The algorithm is designed based on the characteristics of the model and problem. We employ a branch and bound algorithm to assign container tasks for double yard cranes, while incorporating accelerate strategies to the enhance the solving efficiency.

3. Problem description

3.1. The operation flow of containers in ACT

In this paper, the ADCSP is studied without considering the specific operations of the quay cranes. Import containers and export containers are stored in the same block (Zhang et al., 2016), and the AGV travels in a dual-cycle operation (Goodchild and Daganzo, 2007; Zhang et al., 2015; Ahmed.et al., 2021), where an AGV carries an export container from the same yard blocks after unloading an import container. The operations of containers at the terminal include three stages: handled at the quayside by quay cranes, transported by AGVs between the quayside and yard blocks, and handled by yard cranes. An AGV request consists of three stages (as shown in Fig. 2):

Stage 1: from the quayside to the yard block. The AGV picks up the import container and transports it to the storage yard, and the operation time includes the picking up time by the quay crane and the horizontal transportation time by the AGV.

Stage 2: at the yard blocks. The yard crane unloads the import container from the AGV and load an export container from the yard onto the AGV. We regard the import container and export container carried by the same AGV as a group. Therefore, for double yard cranes, there are four allocation schedules for each group of containers (in this paper, Crane 1 and Crane 2 refer to the smaller yard crane and larger yard crane respectively).

Schedule 1: Crane 1 handles both the import containers the export containers.

Schedule 2: Crane 1 handles the import container and Crane 2 handles the export container.

Schedule 3: Crane 2 handles import container and Crane 1 handles export container.

Schedule 4: Crane 2 handles both the import containers the export containers.

Each type of container request defines an origin and destination. The import container request starts at the front point of the yard block and ends at the target bay position. On the contrary, the export container request starts from the original bay position and ends at the front point of the yard block. When the yard crane finishes the container request, it remains at the position until it receives the next request. In consecutive handling operations of a yard crane, a cycle consists of: (1) empty move from the current position to the original position of the request; (2) pick up time; (3) transport from the origin to the destination; (4) drop off time. Taking Schedule 3 as an example, the operations for the double yard cranes are:

Crane 1: empty move from the current position to the front of the yard block, picking up the container, horizontal transport from the front of yard block to the designated position and dropping off the container.

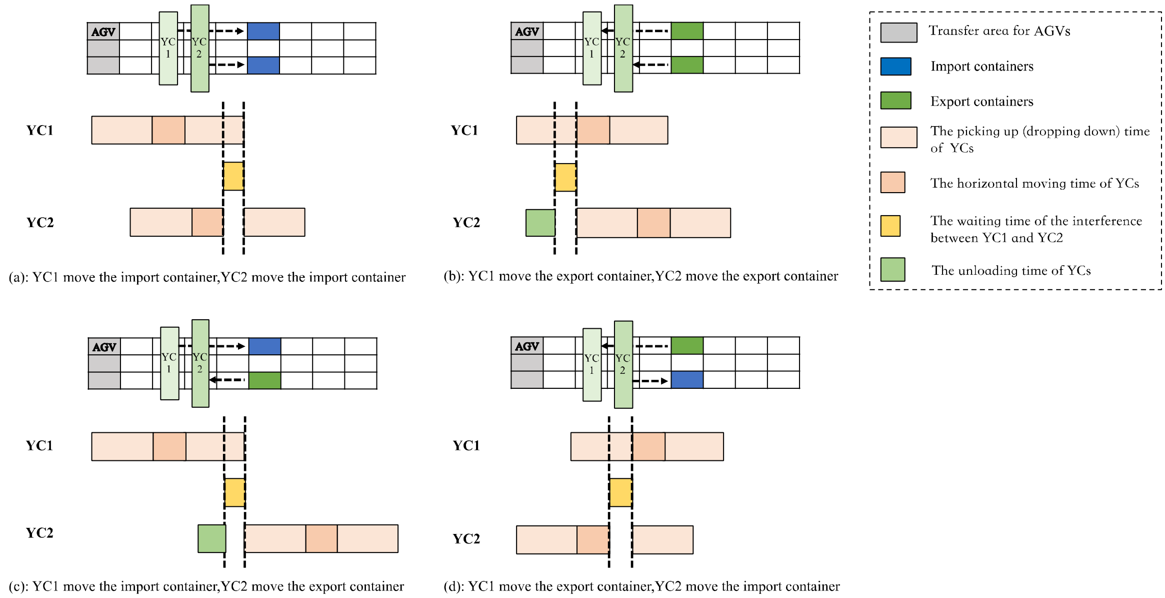


Fig. 3. The four interference situations between yard cranes.

Crane 2: empty move from the current position to the designated position of the export container, picking up the container, horizontal transport from the designated position of the export container to the front of the yard block and dropping off the container.

As the operation time of Crane 1 and Crane 2 may overlap, the completion time of Stage 2 is the completion time of the export container.

Stage 3: from the yard block to the quayside. The AGV transports the export container to the quayside, and the operation time includes the time of horizontal transportation by the AGV and the unloading time by the quay crane.

Since this paper does not consider the specific loading and unloading time by the quay cranes, Stage 1 and Stage 3 will be integrated together for convenience. Then the container operation process is divided into two stages: (1) AGV transportation process, which refers to the operation that the AGV carries an export container leaving the yard block until the AGV carries an import container back to the yard block;(2) the yard crane operation process refers to the double yard cranes cooperating with each other to complete an import container unloading operation and an export container loading operation. The objective of this problem is to minimize the completion time of all the containers. The assignment of containers to AGVs and yard cranes are determined in this paper.

3.2. The interference between the double yard cranes

When assigning containers to double yard cranes, they may interfere each other. The double yard cranes are of different sizes and move on two tracks, therefore there are four situations:

Situation 1: When both Crane 1 and Crane 2 are moving horizontally, they can pass through each other.
Situation 2: Crane 2 can pass Crane 1 when Crane 1 is loading or unloading a container.
Situation 3: Crane 1 may not cross Crane 2 when Crane 2 is dropping off or picking up containers.
Situation 4: Crane 2 or Crane 1 cannot drop off or pick up containers in the same bay position at the same time.

Interference occurs in the situation 3 and situation 4. Since the waiting time of the yard crane in Situation3 is short and can be ignored, this paper only considers the interference in situation4 (Vis and Carlo, 2010), which can be subdivided into four types, as shown in Fig. 3. If two cranes are both handling import container task at the same time, there is an overlap in dropping off containers at the same bay position (Fig. 3(a)); If two cranes are both handling export container tasks at the same time, there is an overlap in picking up containers at the same bay position (Fig. 3(b)); If one crane is handling an import container and the other is handling an export container, there is a time overlap when one crane is dropping off a container while the other crane is picking up a container at the same bay position (Fig. 3(c) and (Fig. 3(d)).

4. Mathematical model

We develop a mixed integer programming to model the ADCSP. To facilitate this model, the assumptions, sets, parameters, decisions variables and the model are defined as follows.

4.1. Assumptions

To avoid losing generality, the following assumptions are made.

(1) Only standard containers are considered, and the numbers of import and export containers are the same.
(2) The number of container tasks, the number of AGVs, the number of QCs and the number of YCs are known.
(3) The AGV, QC and YC can only handle one container once.
(4) The speed of AGVs and YCs is known and the speed of the AGV, QC and YC is the same for an empty and full load.
(5) The storage locations and operation sequence of the import and export containers are given.
(6) Only one yard block is considered, and the capacity of the yard block is known.
(7) The AGV congestion and quay crane operation are not considered.
(8) Container relocating operations are not considered.

4.2. Model parameters

(1): The sets and parameters are as follows:
C: The set of container tasks, including C_I (the set of import container tasks) and C_E (the set of export container tasks)
A: The set of AGVs
B: The set of bay numbers
Y: The set of yard cranes
O: The virtual starting container tasks
F: The virtual terminating container tasks
v_y: The moving speed of the YC, $v_y = 1m/s$
t_{ij}: The time it takes for the AGV to complete a journey from the yard block with the i th export container to the quayside, and then return to the yard block with the j-th import container, where $i \in C_E, j \in C_I$
$\widehat{t}_{ij}$: The time it takes for the AGV from the yard block after carrying the i th import container until returning to the yard block to with the j-th import container, where $i \in C_I, j \in C_I$
B_i^S: The initial bay number of container task i, where $i \in C$
B_i^F: The target bay number of container task i, where $i \in C$
δ: The waiting time of interference between yard cranes
M: A very large number
(2) The decision variables are:
C_{min}: Makespan (we use the latest departure time of AGVs as the makespan)
T_{mi}^A: The arrival time at which the m-th AGV arrive at the yard block with the i th import container, where $i \in C_I$
T_{iy}^S: The time at which the y-th yard crane starts handling the i th container task, where $i \in C$
T_{mi}^L: The departure time at which the m-th AGV leaves the yard block carrying the i th export container, where $i \in C_E$
T_{iy}^F: The time at which the y-th yard crane completes the i th container task, where $i \in C$
$t_{i\alpha y}$: The operation time of the α-th action when yard crane y handles the i th container task, where $i \in C$, $\alpha = 0, 1, 2, 3$. α means the stage of a yard crane to complete a container task.$\alpha = 0$ means the horizontal movement of the yard crane from the current position to the starting position of container task i,$\alpha = 1$ means the pick-up operation of the yard crane, $\alpha = 2$ means the horizontal movement of the yard crane from the starting position to the target position of container task i, $\alpha = 3$ means the drop-off operation of the yard crane.
$P_{i\alpha y}^S$: The bay number of the α -th stage when the yard crane y starts handling the i th container task, where $i \in C$, $\alpha = 0, 1, 2, 3$
$P_{i\alpha y}^F$: The bay number of the α-th stage when the yard crane y finishes handling the i th container task, where $i \in C$, $\alpha = 0, 1, 2, 3$
$T_{i\alpha y}^F$: The completion time point of the α-th stage for the yard crane y handling the i th container task, where $i \in C$
$X_{ijy} = 1$ if the i th container taskis handled immediately before the j-th container task, and both are handled by yard crane y, where $i \in C$, $j \in C$; $X_{ijy} = 0$ otherwise
$X_{iy} = 1$ if the i th container task is handled by yard crane y, where $i \in C$, $y \in Y$; $X_{iy} = 0$ otherwise
$Y_{ij}^m = 1$ if the i th import container task is handled immediately before the j-th import container task, and both are handled by AGV m, where $i \in C_I, j \in C_I$; $Y_{ij}^m = 0$ otherwise. This is a variable to decide the sequence of import containers, and actually there is an export container between the i th import container and the j-th import container.
$Z_{ij}^m = 1$ if the i th import container task is handled immediately before the j-th export container task and both are handled by AGV m, where $i \in C_I, j \in C_E$; $Z_{ij}^m = 0$ otherwise
$Y_i^m = 1$ if the i th container task is handled by AGV m, where $i \in C$, $m \in A$; $Y_i^m = 0$ otherwise

4.3. Mathematical model

The integrated optimization model is shown in [M0]. Our objective is to minimize the completion time of all the container tasks, and we use the latest departure time of AGV as the completion time.

$$[\text{M0}] \quad \min C_{min}$$

$$\text{S.t.} \quad C_{min} \geq \max_{m.i}\left\{T^L_{mi}\right\}, \forall m \in A, i \in C_E \tag{1}$$

The constraints of the model can be divided into five groups, namely the task allocation constraints, the time constraints, the location constraints for yard crane operations, the interference constraints for yard crane operations, and the variable range constraints.

(1) The task allocation constraints:

$$\sum_{m \in A} Y^m_i = 1, \forall i \in C \tag{2}$$

$$\sum_{i \in C_I \cup O} Y^m_{ij} = \sum_{k \in C_I \cup F} Y^m_{jk} = Y^m_j, \forall m \in A, j \in C_I \tag{3}$$

$$\sum_{i \in C_I} Z^m_{ij} = Y^m_j, \forall m \in A, j \in C_E \tag{4}$$

$$\sum_{i \in C_E} Z^m_{ji} = Y^m_j, \forall m \in A, j \in C_I \tag{5}$$

$$Y^m_{ij} + Y^m_{ji} \leq 1, \forall i, j \in C_I, i \neq j, m \in A \tag{6}$$

$$\sum_{j \in C_I} Y^m_{Oj} = 1, \forall m \in A \tag{7}$$

$$\sum_{i \in C_E} Y^m_{iF} = 1, \forall m \in A \tag{8}$$

$$\sum_{y \in Y} X_{iy} = 1, \forall i \in C \tag{9}$$

$$\sum_{i \in C \cup O} X_{ijy} = \sum_{k \in C \cup F} X_{jky} = X_{jy}, \forall j \in C, y \in Y \tag{10}$$

$$X_{ijy} + X_{jiy} \leq 1, \forall i, j \in C, i \neq j, y \in Y \tag{11}$$

$$\sum_{j \in C \cup F} X_{Ojy} = 1, \forall y \in Y \tag{12}$$

$$\sum_{i \in C \cup O} X_{iFy} = 1, \forall y \in Y \tag{13}$$

Constraint (2) means that each container task can be handled by only one AGV; Constraints (3)-(6) ensure the task flow balance constraints of AGVs; Constraint (3) ensures that each container task j which is transported by AGV m has only one immediately preceding task and one immediately following task; Constraint (4) and Constraint (5) ensures each import container task which is transported by AGV m is followed by an export container task; Constraint (6) ensures the relationship between two consecutive tasks; Constraint (7) means that each AGV starts with a virtual container task; Constraint (8) means that each AGV ends with a virtual container task; Constraint (9) means that each container task is handled by only one YC; Constraints (10)-(11) are the task flow balancing constraints of YCs, which ensures that any container task that handled by YC y has only one immediately preceding task one immediately following task; Constraint (12) means that each YC starts with a virtual container task; Constraint (13) means that each YC ends with a virtual container task.

(2) The time constraints:

$$T^A_{mj} + M\left(1 - Y^m_{ij}\right) \geq T^A_{mi} + \widehat{t}_{ij}, \forall i \in C_I, j \in C_I, m \in A \tag{14}$$

$$T^A_{mj} \geq t_{0j} - M\left(1 - Y^m_{0j}\right), \forall j \in C_I, m \in A \tag{15}$$

$$T^A_{mj} \geq T^L_{mi} + t_{ij} - M\left(1 - Y^m_{kj}\right) - M\left(1 - Z^m_{ki}\right), \forall i \in C_E, k \in C_I, j \in C_I, m \in A \tag{16}$$

$$T^S_{jy} + M\left(1 - X_{ijy}\right) \geq T^F_{iy}, \forall i \in C \cup O, j \in C, y \in Y \tag{17}$$

$$T_{iy}^{F} \geq T_{iy}^{S} + \sum_{\alpha=0}^{3} t_{i\alpha y} - M\left(1 - X_{iy}\right), \forall i \in C, y \in Y \tag{18}$$

$$T_{iy}^{S} \geq T_{mi}^{A} - M\left(1 - Y_{i}^{m}\right) - M\left(1 - X_{iy}\right), \forall i \in C_{I}, y \in Y, m \in A \tag{19}$$

$$T_{mi}^{L} \geq T_{iy}^{F} - M\left(1 - Y_{i}^{m}\right) - M\left(1 - X_{iy}\right), \forall i \in C_{E}, y \in Y, m \in A \tag{20}$$

$$T_{jy}^{S} + M\left(1 - X_{jy}\right) \geq T_{ik}^{S} - M\left(1 - Z_{ij}^{m}\right) - M(1 - X_{ik}), \forall i \in C_{I}, j \in C_{E}, y \in Y, k \in Y, m \in A \tag{21}$$

$$T_{mi}^{A} + M\left(1 - Z_{ij}^{m}\right) \geq T_{mj}^{A} \geq T_{mi}^{A} - M\left(1 - Z_{ij}^{m}\right), \forall i \in C_{I}, j \in C_{E}, m \in A \tag{22}$$

Constraints (14)-(16) state the time relationship between two consecutive container tasks carried by the same AGV. Constraint (14) and Constraint (15) state that the arrival time at the yard block of the subsequent import container is no earlier than the arrival time of the former import container task transported by the same AGV plus the round trip time; Constraint (16) means that the arrival time at the yard block of the subsequent container is no earlier than the departure time of the previous container from the yard block plus the round trip time; Constraints (17) and (18) state the time relationship of two consecutive container tasks handled by the same YC, constraint (17) means that the start time of the subsequent container task is no earlier than the finishing time of the previous container handled by the same YC; Constraint (18) means that the completion time is no earlier than the beginning time plus the handling time; Constraints (19) and (20) represent the time relationship between AGVs and YCs that handle the same container task. Constraint (19) means that the start time of YC is no earlier than the arrival time of AGV that handles the same container; Constraint (20) means that the departure time of AGV from the yard block is no earlier than the completion time of YC that handle the container task; Constraint (21) represent the start time of the immediate following export container is no earlier than the start time of the immediate previous import container, where the two containers are transported by the same AGV; Constraint (22) states the arrival time of the export container (actually it is a virtual time) is equal to the arrival time of the import container.

(3) The position constraints for yard crane operations:

$$-M\left(1 - X_{ijy}\right) \leq P_{i3y}^{F} - P_{j0y}^{S} \leq M\left(1 - X_{ijy}\right), \forall i, j \in C, y \in Y \tag{23}$$

$$-M\left(1 - X_{Oiy}\right) \leq P_{i0y}^{S} \leq M\left(1 - X_{Oiy}\right), \forall i \in C, y \in Y \tag{24}$$

$$-M\left(1 - X_{iy}\right) \leq P_{i\alpha y}^{S} - P_{i\alpha y}^{F} \leq M\left(1 - X_{iy}\right), \forall i \in C, y \in Y, \alpha = 1, 3 \tag{25}$$

$$-M\left(1 - X_{iy}\right) \leq P_{i2y}^{S} - P_{i1y}^{F} \leq M\left(1 - X_{iy}\right), \forall i \in C, y \in Y \tag{26}$$

$$-M\left(1 - X_{iy}\right) \leq P_{i\alpha y}^{F} - B_{i}^{S} \leq M\left(1 - X_{iy}\right), \forall i \in C, y \in Y, \alpha = 0, 1 \tag{27}$$

$$-M\left(1 - X_{iy}\right) \leq P_{i\alpha y}^{F} - B_{i}^{F} \leq M\left(1 - X_{iy}\right), \forall i \in C, y \in Y, \alpha = 2, 3 \tag{28}$$

$$-M\left(1 - X_{iy}\right) \leq t_{i\alpha y} - \delta \leq M\left(1 - X_{iy}\right), \forall i \in C, y \in Y, \alpha = 1, 3 \tag{29}$$

$$-M\left(1 - X_{iy}\right) \leq t_{i\alpha y} - \frac{\left|P_{i\alpha y}^{F} - P_{i\alpha y}^{S}\right|}{v_{y}} \leq M\left(1 - X_{iy}\right), \forall i \in C, y \in Y, \alpha = 0, 2 \tag{30}$$

Constraints (23–24) indicate the position relationship of two consecutive containers carried by the same YC. Constraint (23) means that the beginning position of the immediately following container task is equal to the end position of the immediate previous container task that are handled by the same YC; Constraint (24) states the starting position of a virtual start container task. Constraints (25)-(28) indicate the position relationship among each stage of YC. Constraint (25) means that the position of YC remains the same during the pick-up/drop-off operations; Constraint (26) means that the position of YC when completes the pick-up operation is equal to the starting position when finishes horizontal movement; Constraint (27) means that the start position and the complete position of pick-up operation are equal to the initial bay position of container task; Constraint (28) means that the completion position of horizontal movement and drop-off operation are equal to the target bay position of container task; Constraints (29)-(30) calculate the YC handling time. Constraint (29) represents the pick-up/drop-off time of YC; Constraint (30) represents the horizontal movement time of YC.

(4) The interference constraints for yard crane operations:

For the fourth interference situation mentioned in Section 3.2, the yard crane that arrives later at the specified location will wait for the other yard crane and the waiting time is set to δ. We define the following sets and variables.

Φ_1: The set of container groups, and each group has two import containers with the same target bay position

Φ_2: The set of container groups, and each group has two export container tasks with the same initial bay position

Φ_3: The set of container groups, and each group has an import container task and an export container task that need to be handled at the same bay

Φ_4: The set of container groups, and each group has an export container task and import container task that need to be handled at the same bay

a_{ij}=1 if the completion time handled by YCs of container task i is earlier than the completion time of container task j, where $(i, j) \in \Phi_1$; otherwise a_{ij}=0

b_{ij}=1 if the start time handled by YCs of $(i, j) \in \Phi_3$ container task i is earlier than the start time of container task j, where $(i, j) \in \Phi_2$; otherwise b_{ij}=0

c_{ij}=1 if the completion time handled by YCs of container task i is earlier j than the start time of container task j, where $(i, j) \in \Phi_3$; otherwise c_{ij}=0

d_{ij}=1 if the start time by handled by YCs container task i is earlier than the completion time of container task j, where $(i, j) \in \Phi_4$; otherwise d_{ij}=0

$$a_{ij} + a_{ji} = 1, \forall (i,j) \in \Phi_1 \tag{31}$$

$$b_{ij} + b_{ji} = 1, \forall (i,j) \in \Phi_2 \tag{32}$$

$$c_{ij} + c_{ji} = 1, \forall (i,j) \in \Phi_3 \tag{33}$$

$$d_{ij} + d_{ji} = 1, \forall (i,j) \in \Phi_4 \tag{34}$$

Constraints (31)-(34) define four interference scenarios. The constraints ensures that the start time or completion time of two containers that in the same bay could not be the same. Constraint (31) states the relationship of start time between two import containers; Constraint (32) states the relationship of completion time between two export containers; Constraint (33) states the relationship of start time between an import container and an export container; Constraint (34) states the relationship of start time between an import container and an export container.

$$T^F_{jk} + M\left(1 - X_{jk}\right) \geq T^F_{iy} + \delta + M\left(a_{ij} - 1\right) + M\left(1 - X_{iy}\right), \forall (i,j) \in \Phi_1, y, k \in Y \tag{35}$$

$$T^S_{jy} + M\left(1 - X_{jk}\right) \geq T^S_{iy} + \delta + M\left(b_{ij} - 1\right) + M\left(1 - X_{iy}\right), \forall (i,j) \in \Phi_2, y, k \in Y \tag{36}$$

$$T^S_{jy} + M\left(1 - X_{jk}\right) \geq T^F_{iy} + \delta + M\left(c_{ij} - 1\right) + M\left(1 - X_{iy}\right), \forall (i,j) \in \Phi_3, y, k \in Y \tag{37}$$

$$T^F_{jy} + M\left(1 - X_{jk}\right) \geq T^S_{iy} + \delta + M\left(d_{ij} - 1\right) + M\left(1 - X_{iy}\right), \forall (i,j) \in \Phi_4, y, k \in Y \tag{38}$$

Constraints (35)-(38) state the relationship between the start time and the completion time of container tasks that are handled by two interfere yard cranes at the same bay. Constraint (35) means the relationship between the completion time of import containers if interference occurs; Constraint (36) means the relationship between the completion time of two export containers if interference occurs; Constraint (37) means the relationship between the completion time of import container and the start time of export container if interference occurs; Constraint (38) means the relationship between the start time of export container and the completion time of import container if interference occurs.

(5) The variable value range constraints:

$$T^A_{mi} \geq 0, \forall i \in C_I, m \in A \tag{39}$$

$$T^S_{iy}, T^F_{iy} \geq 0, \forall i \in C, y \in Y \tag{40}$$

$$T^L_{mi} \geq 0, \forall i \in C_E, m \in A \tag{41}$$

$$t_{i\alpha y}, T_{i\alpha y}, P^S_{i\alpha y}, P^F_{i\alpha y} \geq 0, \forall i \in C, y \in Y, \alpha = 0, 1, 2, 3 \tag{42}$$

$$X_{ijy} \in \{0, 1\},\ \ \forall i,j \in C, y \in Y \tag{43}$$

$$X_{iy} \in \{0, 1\}, \forall i \in C, y \in Y \tag{44}$$

$$Y^m_{ij}, Z^m_{ij} \in \{0, 1\}, \forall i,j \in C, m \in A \tag{45}$$

$$Y^m_i \in \{0, 1\}, \forall i \in C, m \in A \tag{46}$$

Constraints (39)-(42) are non-negative constraints on continuous variables, and constraints (43)-(46) are constraints on 0–1 variables.

Section 1 introduces four configurations of the yard blocks, and each configuration has its disadvantages and advantages. The schedule for single yard crane is the easiest, because interference does not exist. But the single yard crane consumes more time to complete all the tasks. Especially when the AGVs carry an export container and import container in a cycle, single yard crane is less

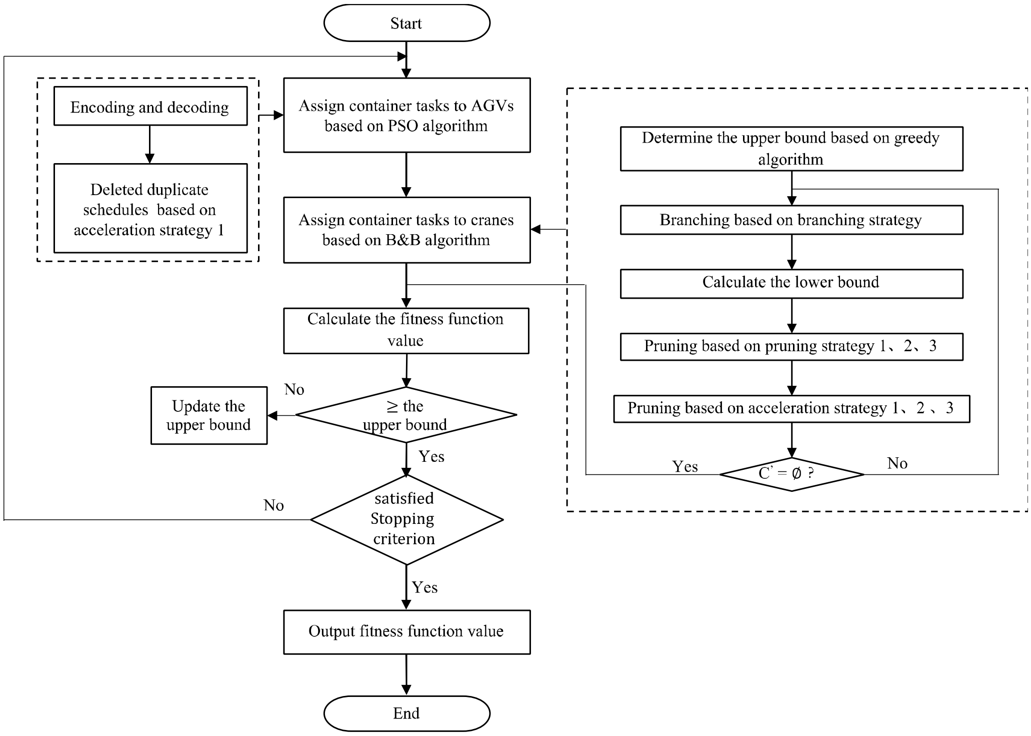


Fig. 4. The outline of B&B-PSO algorithm.

efficient. Double yard cranes can increase the flexibility of the schedules in case one of the cranes has broken down. The efficient of the configurations rises attention from researchers, for example, Saanen and Valkengoed (2005) compared the indicators. But there is no research shows the results of the dual cycle of AGVs.

Proposition 1. ***The double yard cranes configuration is unconditionally greater than that of the single yard crane.***

Proof: In the proposed B&B algorithm in Section 5, each node is a container task, and each branch is an assignment schedule for each group of containers. There are four assignment schedules for each node, (YC1,YC1), (YC1,YC2), (YC2,YC1), (YC2,YC2) respectively, where (YC1,YC2) represents that YC1 handles the import container and YC2 handles the export container for an AGV. Let ω_i=(YC1,YC2) represents a branch for the i th level. Let $\pi_N = \{\omega_1,\omega_2,\omega_3,\ldots,\omega_N\}$ represents a set of branches consisting the branches from the origin to the destination. Let C_{π_N} represents the makespan of the branch set π_N. Let C^* represents the minimum makespan. Then we can obtain that:

$$C_{\pi_N} \geq C^* \tag{47}$$

Let $\omega_1 = \omega_2 = \omega_i = \omega_N = (YC1,\ YC1)$, then the branch set $\pi_N$consists of (YC1,YC1), which means that all the container tasks are handled by YC1. YC2 is idle all the time. Therefore, no matter what the assignment schedule is the best, the branch set for $\omega_1 = \omega_2 = \omega_i = \omega_N = (YC1,\ YC1)$ always has the longer makespan. In other words, the double yard cranes configuration is more efficient than the single yard crane. From the above analysis, terminal operators can choose the double yard cranes configuration to increase the yard efficiency.

5. Solution approach

The mixed integer programming formulation of the ADCSP belongs to an NP-hard problem. We use Gurobi to solve the small size of instances, and a B&B-PSO algorithm to solve the large scale of the problem. The outline of the B&B-PSO algorithm is shown in Fig. 4. The algorithm is divided into two stages: In the first stage, Particle Swarm Optimization (PSO) is used to assign container tasks to AGV, and a greedy algorithm is used to obtain the initial solution. In the second stage, a B&B based heuristic algorithm is used to assign container tasks to yard cranes based on the AGV assignment schedule obtained in the first stage. Branching strategies and acceleration strategies are designed to improve the algorithm.

Table 2
Position vector P of a particle.

Vector coding	Container assignment number											
	1	2	3	4	5	6	7	8	9	10	11	12
P	1	3	2	5	1	4	3	2	5	1	2	4

5.1. *Assigning tasks to AGVs based on PSO*

5.1.1. *Framework of Particle swarm optimization*

The particle swarm optimization is a global search algorithm that mimics the foraging process of birds. In the particle swarm optimization, each feasible solution of the problem can be regarded as a particle, and each particle i has two important parameters, which are the position vector P_i^k and the velocity vector V_i^k(k is used to denote the number of iterations). Generally, $P_{best(i)}^k$ is used to denote the historical optimal position of each particle in the k-th iteration, and g_{best}^k is used to denote the optimal solution in the current population. In addition, the superiority of the solution is evaluated by the value of the fitness function. The position and velocity of the particles during the iteration are updated as Eqs. (48) and (49).

$$V_i^{k+1} = \omega \cdot V_i^k + c_1 r_1 \left(P_{best(i)}^k - P_i^k\right) + c_2 r_2 \left(g_{best}^k - P_i^k\right) \tag{48}$$

$$P_i^{k+1} = P_i^k + V_i^{k+1} \tag{49}$$

Where c_1 and c_2 are acceleration factors, which are usually positive and generally taken as 2; r_1 and r_2 are random numbers between [0, 1]; ω is the inertia weight coefficient, and it is dynamically adjusted by Eq. (50) during the search process.

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times k}{K_{max}} \tag{50}$$

Where ω_{max} denotes the maximum inertia weight, ω_{min} denotes the minimum inertia weight, and K_{max} denotes the maximum number of iterations.

5.1.2. *Coding*

In the particle swarm optimization, each particle represents a feasible solution to the current problem. In this paper, the position vector P is used to represent the allocation schedule of m groups of container tasks and n AGVs. For example, assuming that the number of container tasks is 12 and the number of AGVs is 5, one position vector P of a particle is shown in Table 2.

According to Table 2, the container tasks and sequence operations for AGV are as follows: AGV1 carries the 1-st group container, the 5-th group container and the 10-th group container in sequence. And AGV2 carries the 3-rd group container, the 8-th group container and the 11-th group container in sequence. Other containers are allocated as so on.

5.1.3. *Steps of the particle swarm optimization*

The specific algorithm flow for assigning AGVs to container tasks is summarized as Algotithm 1.

Step1: Initialize the particle swarm, i.e. determine the initial population size of the swarm and initialize the position and velocity of each particle.

Step1.1 The position vector P_i^k of each particle is randomly taken as an integer between 1 and A (A refers to the number of AGV).

Step1.2 The velocity vector V_i^k of each particle is randomly taken as an integer between -(A-1) and (A-1).

Step2: Based on the AGV allocation schedule for the container tasks, the branch-and-bound algorithm in Section 5.2 is used to allocate yard cranes to the container tasks, and the fitness function for each allocation schedule is calculated to determine the $P_{best(i)}^k$ of each particle and the g_{best}^k in the current population.

Step2.1 Obtain the yard crane assignment schedule for each group of containers based on Step1, and apply the method in Section 5.2.

Step2.2 Calculate the fitness values of each particle in the initial particle swarms based on the allocation schedule in Section 5.2, set the particle's historical optimal $P_{best(i)}^k$ as the current position, and the position of the optimal solution in the current population as the current g_{best}^k.

Step3: Update the particle swarm by the particle swarm update strategy (Section 5.1.2), and update the historical optimal solution and the global optimal solution of each particle in the current particle swarm.

Step3.1 Update the particles in the particle swarm and calculate and update the fitness function values of the particles based on the yard crane assignment schedule obtained in Section 5.2.

Step3.2 update the $P_{best(i)}^k$ of each particle. If the current fitness function value of a particle is better than the historical optimum, then the current position vector is recorded as the historical optimum position P_i^k of that particle, otherwise it remains unchanged.

Step3.3 Update g_{best}^k. if the historical optimal fitness function value of a particle is better than the global optimal value, then the

Algorithm 1
Particle swarm optimization.

data: N—Number of containers, P_i^k-the position of particle i on dimension k, V_i^k-the velocity of particle i on dimension k, $P_{best(i)}^k$-the best position for particle i on dimension k, g_{best}^k-the swarm's best solution on dimension k.
result Ω^* -solution
procedure
$k = 0$ # initialize particle swarm
  for each particle i
    initialize P_i^0 and V_i^0
    generate the allocation schedule of yard crane based on Section 5.2.
    calculate $f(P_i^0)$ and update $P_{best(i)}^0$
  end for
  $g_{best}^0 = \min\{P_{best(i)}^0\}$
  while not stop
    $k \in K$
    for i=1 to N
      update P_i^k and V_i^k,
      generate the allocation schedule of yard crane based on Section 5.2.
      calculate $f(P_i^k)$
    if $f(P_i^k) < f(P_{best(i)}^{k-1})$
      $P_{best(i)}^k = P_i^k$
    end if
    if $f(P_{best(i)}^k) < f(g_{best}^{k-1})$
      $g_{best}^k = P_{best(i)}^k$
    end if
  end for
end while
return Ω^*
end procedure

historical optimal of the particle is recorded as the global optimal, otherwise it remains unchanged.

Step4: Repeat Step2 until the termination condition is satisfied or the maximum number of iterations is reached and the optimal solution is output (Algorithm 1).

5.2. *Assigning tasks to cranes using B&B based heuristic algorithm*

The B&B based heuristic algorithm is used to assign container tasks to yard cranes, but the solution space increases with the number of containers. We use pruning strategies and accelerating strategies to cut the solution space, and solve the problem by container task (from level 1 to n). Each node of B&B is a group of container tasks, and we decide the assignment of yard cranes for two containers handled by one AGV. A solution is contained in each node by solving the conflict, and the UB is obtained by a greedy algorithm. The B&B algorithm is presented in Algorithm 2.

5.2.1. *Branching strategy*

Branching strategy is a method to generate branches from the existing node. An AGV transports an import container to the storage yard, and then carries an export container from the storage yard to the quayside. We use each branch to denote an assignment schedule for yard cranes. According to the four assignment schedules in Section 3.1, each node has four branches, (YC1, YC1), (YC1, YC2), (YC2, YC1) and (YC2, YC2), where (YC1, YC2) represents YC1 handles the import container and YC2 handles the export container. We can estimate the makespan of each node and record this value (as shown in Fig. 5).

5.2.2. *Pruning strategy*

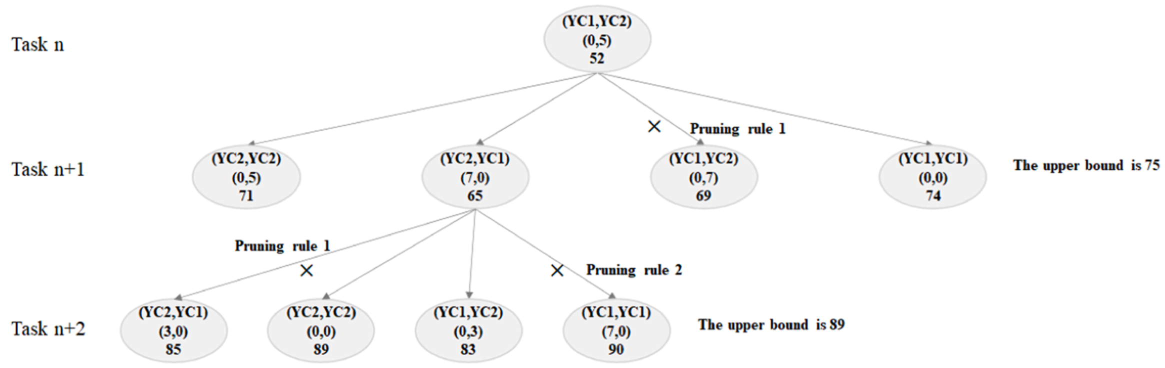
The number of nodes increases exponentially with the number of container tasks. There are 4^N assignment schedules for N groups of containers, which will consume a large amount of time to obtain an optimal solution. We design two pruning strategies to disregard inefficient nodes.

Proposition 2. (*Pruning rule 1*) *Among the branches of the same node, if the locations of the two cranes after completing the current containers are the same, cut the branch which has longer makespan. As shown in Fig. 5, for branches 2 and 3 of level* n+1, *assignment schedules (YC1, YC2) and (YC2, YC1) have the same yard crane positions after containers are completed, so the branch with a longer completion time is cut off.*

Proof. $T_{i\alpha y}^F(n)$ refers to the completion time of the α-th stage for yard crane y handling the i th container task of the n-th group of containers, where $i \in C$, $\alpha = 0, 1, 2, 3$, $y = 1, 2$. $i = 2n - 1$ refers to the import container and $i = 2n$ refers to the export container, $y = 1$ refers to YC1 and $y = 2$ refers to YC2, and α means the stage of a yard crane to complete a container task. $\alpha = 0$ means the horizontal movement of the yard crane from the current position to the starting position of container task i, $\alpha = 1$ means the pick-up operation of

Algorithm 2
B&B algorithm.

Data:data file, I-the set of containers, K-the set of alternative schedule, C-the set of containers been assigned, C'-the set of containers to be assigned, v represent nodes, P_n-the set of nodes in level n, T_n^v-the completion time of node v in level n, $s_n^{v,k}$-the k-th branch of node v in level n, S_n^v-the set of branches of node v in level n, S_n- the set of branches of level n, w-the filter width.
Result:Ω- solution for double yard cranes
procedure
  generate the upper bound T^{UB} based on the algorithm proposed in Section 5.2.2.
  $v = 0$, $P_n \leftarrow \varnothing$, $C \leftarrow \varnothing$
  for $n = 0$ **do** # branch from level 0 to level l
    $P_0 = \{1\}$
    for $k \in K$ **do**
      branch node 0
      $s_1^k = s_0^{1,k}$
      $P_1 = P_1 \cup \{s_1^k\}$
    end for
  end for
  for $n \in N$ **do** # branch from level 1 to level n
    branch according to the branch rules
    calculate the competition time of each node
      if $T_n^v > T^{UB}$ **do**
        continue #disregard the nodes that are larger than upper bound
      end if
    update S_n^v
    Pruning strategy 1 are used to optimize S_n^v
    calculate the lower bound of each node and cut the branches according to Pruning strategy 3
    update S_n^v
    $P_{n+1} = P_{n+1} \cup \{S_n^v\}$
    if $|P_{n-1}| > w$ **do** # w is the filter width
      accelerating strategy 2 are used to disregard repeated nodes
    end if
    $C_n = C_{n-1} \cup \{n\}$, **update** C_n'
    if $C_n' = \varnothing$ **do**
      continue
    end if
  end for
  return Ω
end procedure



Task	n	n+1	n+2
The position of import container	5	7	3
The position of export container	1	4	9

(A,B)
(C,D)
E

A: The YC which operates import containers
B: The YC which operates export containers
C: The position where YC2 finished the n-th container task
D: The position where YC1 finished the n-th container task
E: The time when both YC1 and YC2 finished their container task

Fig. 5. The branching processes.

the yard crane, α = 2 means the horizontal movement of the yard crane from the starting position to the target position of container task i, α = 3 means the drop-off operation of the yard crane.

$t_{i\alpha y}(n)$ refers to the operation time of the α-th action when yard crane y conducting the i th container task of the n-th group of containers, where $i \in C$, $\alpha = 0, 1, 2, 3$, $y = 1, 2$.

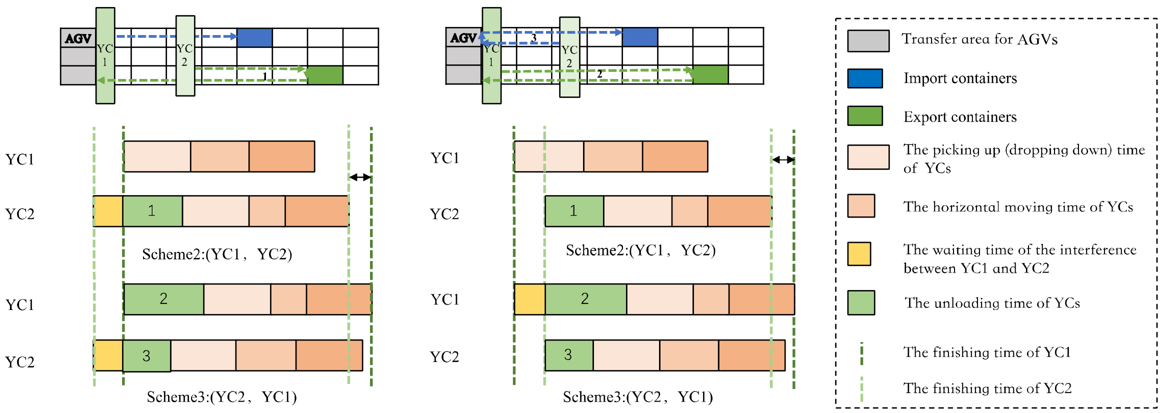


Fig. 6. Illustrations of Pruning rule 1.

$P^F_{i\alpha y}(n)$ refers to the bay number of the α-th stage when the yard crane y finishes handling the i th container task of the n-th group of containers, where $i \in C$, $\alpha = 0, 1, 2, 3$, $y = 1, 2$.

$B^S_i(n)$ refers to the initial bay number of container task i of the n-th group of containers, where $i \in C$.

$B^F_i(n)$ refers to the target bay number of container task i of the n-th group of containers, where $i \in C$.

As shown in Fig. 6, YC1 operates the import container and YC 2 operates the export container of the n-th group of container tasks. For schedule (YC1, YC2) of the *(n+1)*-th container:

$$t_{(2a+1),0,1}(n+1)\big|_{(YC1,\ YC2)} = \frac{\left|P^F_{(2a+1),0,1}(n+1) - P^F_{(2a+1),3,1}(n)\right|}{v_y} = \frac{\left|B^S_{(2a+1)}(n+1) - B^F_{(2a+1)}(n)\right|}{v_y} = \frac{B^F_{(2a+1)}(n)}{v_y}$$

$$t_{(2a+1),1,1}(n+1)\big|_{(YC1,\ YC2)} = t_{(2a+1),3,1}(n+1)\big|_{(YC1,\ YC2)} = 30$$

$$t_{(2a+1),2,1}(n+1)\big|_{(YC1,\ YC2)} = \frac{\left|P^F_{(2a+1),2,1}(n+1) - P^F_{(2a+1),1,1}(n)\right|}{v_y} = \frac{\left|B^F_{(2a+1)}(n+1) - B^S_{(2a+1)}(n+1)\right|}{v_y}$$

$$t_{(2a),0,2}(n+1)\big|_{(YC1,\ YC2)} = \frac{\left|P^F_{(2a),0,2}(n+1) - P^F_{(2a),3,2}(n)\right|}{v_y} = \frac{\left|B^S_{2a}(n+1) - B^F_{2a}(n)\right|}{v_y} = \frac{B^S_{2a}(n+1)}{v_y}$$

$$t_{(2a),1,2}(n+1)\big|_{(YC1,\ YC2)} = t_{(2a),3,2}(n+1)\big|_{(YC1,\ YC2)} = 30$$

$$t_{(2a),1,2}(n+1)\big|_{(YC1,\ YC2)} = \frac{\left|P^F_{(2a+1),2,1}(n+1) - P^F_{(2a+1),1,1}(n)\right|}{v_y} = \frac{\left|B^F_{(2a+1)}(n+1) - B^S_{(2a+1)}(n+1)\right|}{v_y}.$$

We can obtain $t_{i\alpha y}(n+1)|_{(YC2,YC1)}$ similarly. Therefore, $t_{(2a+1),\alpha,1}(n+1)|_{(YC1,YC2)} = t_{(2a+1),\alpha,2}(n+1)|_{(YC2,YC1)}$ and $t_{(2a),\alpha,2}(n+1)|_{(YC1,YC2)} = t_{(2a),\alpha,1}(n+1)|_{(YC2,YC1)}$ where $\alpha = 1, 2, 3$.

(1) If $T^F_{2a+1,3,1}(n)|_{(YC1,\ YC2)} < T^F_{2a,3,2}(n)|_{(YC1,\ YC2)}$, we can obtain that $T^S_{2a+1,0,1}(n+1)|_{(YC1,\ YC2)} = T^S_{2a+1,0,2}(n+1)|_{(YC2,\ YC1)}$ and $T^S_{2a,0,2}(n+1)|_{(YC1,\ YC2)} = T^S_{2a,0,1}(n+1)|_{(YC2,\ YC1)}$, then $T^F_{2a,3,2}(n+1)|_{(YC1,\ YC2)} - T^F_{2a,3,1}(n+1)|_{(YC2,\ YC1)} = \left|t_{(2a),0,2}(n+1) - t_{(2a+1),0,2}(n+1)\right|$. If $T^S_{2a,0,2}(n+1)|_{(YC1,\ YC2)} \geq T^S_{2a,0,1}(n+1)|_{(YC2,\ YC1)}$ and $T^F_{2a,3,2}(n+1)|_{(YC1,\ YC2)} \geq T^F_{2a,3,1}(n+1)|_{(YC2,\ YC1)}$, cut down the schedule (YC1, YC2), otherwise cut down the schedule (YC2, YC1).

(2) If $T^F_{2a+1,3,1}(n)|_{(YC1,\ YC2)} \geq T^F_{2a,3,2}(n)|_{(YC1,\ YC2)}$, compare the completion time of the crane that handles the export container of the two schedules and cut down the longer one similarly. Proof ends.

Pruning rule 2: Calculate the completion time of the cranes for all nodes in each level and cut down the branch whose completion time is longer than the upper bound.

Pruning rule 3: Cutting down the branch whose lower bound is larger than the upper bound. The lower bound of each node is obtained in Section 5.2.4.

5.2.3. The upper bound of B&B

The upper bound is obtained by the greedy algorithm to allocate container tasks for cranes. The operation sequence of the con-

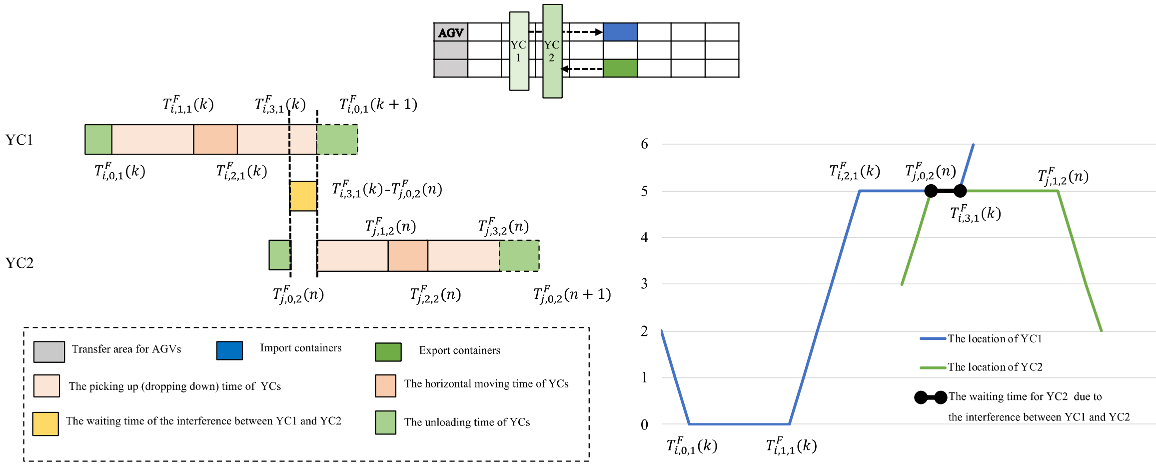


Fig. 7. The interference between the double yard cranes.

tainers is represented by i. Let C represents the assigned containers set, C'represents the unassigned containers set, t_i^{YC} refers to the YC's operation time for the i th group of containers, T_i^A refers to the time when the i th AGV arrives at the yard block, T_i^L refers to the time when the i th AGV leave the yard block. We assume that the beginning time of the yard cranes cannot be earlier than the arrival time of AGVs for the i th group of containers. The steps of the algorithm are as follows:

Step 1: Generate the AGV allocation schedules by the PSO algorithm. Let $C = \varnothing$, $T_0^L = 0$, $i = 1$, update T_i^A, generate C'.
Step 2: For the i th pair of containers, calculate $t_i^{YC}$of the four assignment schedules, $T_i^L = \max(T_i^A, T_{i-1}^L) + t_i^{YC}$, select the schedule with the shortest T_i^L as the best schedule for the current container task, and let $i = i + 1$, update C and C', record T_i^L.
Step 3: Determine whether C' is an empty. If C' is not an empty set, turn to step 2. If $C' = \varnothing$, the algorithm stops. The solution of the container task set of AGV and yard crane and the completion time of the last container (the upper bound of the model) is output.

5.2.4. The lower bound of B&B

A lower bound is used to cut down the branch whose lower bound is larger than the upper bound. A tighter lower bound can speed up the computation time of the B&B algorithm. Let $lb_n(i)$ denotes the lower bound of the i th branch in the n-th level. Let $f_n(i)$ denotes the competition time of the i th branch at the n-th level. I is the number of container groups, C is the set of container groups has been assigned, C' is the set of container groups to be assigned. $\varphi(i)$ is the handling time by the yard crane where $i \in C'$. Then the lower bound of each node can be obtained by:

$$lb_n(i) = f_n(i) + \min\left\{\sum_{j=i+1}^{I} \varphi(j)\right\}, i \in C \tag{51}$$

If the sequence of the container tasks is given and the arrivals of AGVs are ignored, we only need to assign the container tasks to the double yard cranes. This problem can be regarded as a task assignment problem that minimizes the makespan. We need to find a shortest path for each yard cranes. Let (k, k') denotes the container task groups transported by the same AGV in the same cycle, where k is an import container and k' is an export container. Let Ωdenotes a set of aces (m, n) which means container n can be handled after container m by the same yard crane. The aces have two situations: (1) m and n belong to different container group, m and n can be both export containers and import containers; (2) m and n belong to the same container group, m must be the import container and n must be the export container, because import container must be handled before the export container. t_{mn},$(m, n) \in \Omega$ denotes the handling time of a yard crane of the ace(m, n). The details of the algorithm are as follows:

Step 1: Initialization. Let 0 denotes the original task. Assign a container task to the yard cranes, and record the competition time of the YC1 and YC2 by T_1,T_2 respectively. $T_1 = t_{0m}$,$T_2 = t_{0n}$,$(0, m)$, $(0, n) \in \Omega$
Step 2: If $T_1 > T_2$, select an ace with longer handling time from the aces $(n, p) \in \Omega$, $t_{np} = \max\{t_{nk}\}$,$(n, k) \in \Omega$, and select an ace with shorter handling time from the aces $(m, q) \in \Omega$, $t_{mq} = \min\{t_{mk}\}$,$(m, k) \in \Omega$. Otherwise, select an ace with shorter handling time from the aces $(n, p) \in \Omega$, $t_{np} = min\{t_{nk}\}$,$(n, k) \in \Omega$, and select an ace with longer handling time from the aces $(m, q) \in \Omega$, $t_{mq} = \max\{t_{mk}\}$,$(m, k) \in \Omega$. $T_1 = T_1 + t_{mq}$, $T_2 = T_2 + t_{np}$. This step ensures the balance of the workload of the two yard cranes.
Step 3: Repeat Step 2 until all the containers are assigned to yard cranes. The competition time $C_T = \max\{T_1,T_2\}$.

The lower bound of the handling time in the right of Eq. (51) satisfies:

$$\sum_{j=i+1}^{I} \varphi(j) \geq C_T \tag{52}$$

Vis and Carlo (2010) derived a lower bound, where the assignment of double yard cranes is regarded as one yard crane, and the lower bound can be calculated by $C_T = C_s/2$, where C_s is the competition time if only one yard crane is used. The algorithm of obtaining C_s is similar to that of the above steps. We compare the values, and finally $C_T = \min\{\max\{T_1,T_2\},C_s/2\}$.

Therefore the lower bound of each node can be obtained by:

$$lb_n(i) = f_n(i) + C_T, i \in C \tag{53}$$

5.2.5. Algorithm to solve interference between double yard cranes

According to Vis and Carlo (2010), only four interference cases of the same bay are considered. For each group of container tasks, when one crane is picking up /dropping down one container, the crane needs to determine whether another crane also needs to pick up/drop down container at the same bay. If this is the case, the later crane needs to wait until the former crane leaves.

We use an example to illustrate the interference, as shown in Fig. 7. For a group of container tasks, if $B_i^F(k) = B_j^S(n)$ for $\forall i \in C_I, j \in C_E$, compare $T_{i,3,1}^F(k)$ and $T_{j,0,2}^F(n)$. If $|T_{i,3,1}^F(k) - T_{j,0,2}^F(n)| < T_s$, there exists interference between Crane1 and Crane2, where T_s refers to the maximum waiting time for interference. If $T_{i,3,1}^F(k) > T_{j,0,2}^F(n)$, Crane1 need to wait for a period time of $|T_{i,3,1}^F(k) - T_{j,0,2}^F(n)|$.

5.2.6. Acceleration strategies

In the branching process of the B&B-PSO algorithm, although some pruning strategies are designed, the solving speed of the algorithm remains slow when the size of the solved problem increases to a certain degree. Therefore, this paper proposes some acceleration strategies to accelerate pruning based on the original pruning strategy.

Acceleration strategy 1: For the same AGV allocation schedules, cut the repeated schedule of cranes. The two YCs have no differences in operating containers, therefore some duplicate scheduling plans are generated when assigning container tasks. The purpose of acceleration strategy 1 is to screen out the repeated schedule of cranes for the same AGV assignment plan. For the same AGV allocation schedule, different nodes at the same level may have the same allocation schedule of cranes which means the schedule of Crane1 handles the import container and Crane2 handles the export container is the same as the schedule of Crane1 handles the export container and Crane2 handles the import container. For example, AGV 1 handles the container tasks 1, 2 and 3, while the allocation schedule of the crane {(Crane1, Crane2), (Crane2, Crane1), (Crane1, Crane2)} is the same as the allocation schedule {(Crane2, Crane1), (Crane1, Crane2), (Crane2, Crane1)}.

Acceleration strategy 2: Based on the assumption conditions in this paper, since the handling sequence of containers is known, the waiting time of AGVs is relatively uncontrollable compared to the operation time of AGVs and YCs. But the waiting time of AGVs can directly affect the final completion time. Therefore, a threshold value Q of the waiting time of AGV is set in the pruning process. If the waiting time of AGV of a branch exceeds Q, the branch will be pruned. We set Q as the average waiting time of the assigned containers:

$$Q = \frac{\sum_{M \in A} \sum_{i \in C_I} t_{mi}^w}{N} \tag{54}$$

where t_{mi}^w denotes the waiting time from the time point of AGV m carrying import container task i arriving at the yard to the time point of YC begins to handle container task i. And N denotes the number of container task groups.

Acceleration strategy 3: For the B&B-PSO algorithm, each node indicates the assignment of a group of container tasks, and each branch corresponds to an assignment scheme. Due to the limited operation time of AGV and YC, the completion time of some branches is larger than that with the minimum value. Therefore, we set a parameter D to prune the branches whose competition time is larger than the sum of the minimum value and D. Let T_n^v denote the completion time of the v-th branch in the n-th level and V_n^{min} denote the minimum value in the n-th level, then the branch which satisfies$T_n^v > V_n^{min} + D$ is pruned.

5.3. Lower bound of the original model

We drive two methods to obtain the lower bound of the original model. The first method is relaxing interference constraints between double yard cranes. The second method is estimating the lower bound of objective values. The efficiency of the two methods is tested in Section 6.2.

(1) Lower bound 1 by relaxing interference constraints between yard cranes

Our algorithm aims to find a schedule that deals with interference between double yard cranes based on a given AGV task assignment schedule. According to the definition of interference in Section 3, for each feasible task assignment schedule between double yard cranes, the completion time will be longer if any interference occurs. Therefore, the completion time of a schedule that don't solve the interference will not be longer than that solves the interference. Based on this analysis, we relax the interference constraints of the original model to obtain a lower bound model, which is denoted by [M1]. The results of lower bound model are used to compare the efficiency of the original model and proposed algorithm. The objective of the model is:

Table 3
The target bays of containers of case (3, 10).

Task Number	1	2	3	4	5	6	7	8	9	10
import container	16	11	29	11	19	5	2	25	28	3
export container	5	5	14	15	3	23	25	20	24	22

Table 4
The results of case (3, 10).

Result	Gurobi	B&B-PSO
AGV	AGV1:1-20-15-18-19-12	AGV1:3-4-9-10-17-18
	AGV2:5-8-17-14-11-6	AGV2:1-2-7-8-13-14-19-20
	AGV3:9-16-3-2-13-4-7-10	AGV3:5-6-11-12-15-16
YC	YC1:1-16-8-3-2-17-14-10-11	YC1:1-6-3-4-7-12-9-14-18-19
	YC2:9-20-5-15-18-13-4-19-12-7-6	YC2:2-5-8-11-10-13-16-17-20

$$\begin{aligned} & \min C_{min} \\ \text{S.t. Constraints } & (2)-(30),\ (39)-(46) \end{aligned}$$

(1) Lower bound 2 by estimating the lower bound of objective values

The objective of the proposed model is to minimize the makespan. Then the key part of the integrated optimization model is to minimize the waiting time between the AGVs and yard cranes to make the integration more flexible. We can calculate the total operation time C^T by summarizing horizontal transportation time T^S and handling time by the yard crane T^Y. The estimated total operation time C^T is the lower bound of the objective values. When calculating the handling time by the yard crane, we assume a yard crane is always ready to load the export container. Therefore, the handling time by the yard crane can be estimated by summarizing the lifting time of export containers. Let $t^i,\ i \in I \cup E$ be the transportation time of container i between the quay side and yard block. N^a is the number of AGVs. Let $s^i, i \in E$ be the operation time for the yard crane to load an export container onto the AGV. If the interval of arriving time of AGVs at the yard block is shorter than the expected operation of the yard crane $\frac{E(t^i)}{N^a} \le E(s^i)$, AGVs have to queue at the yard block to wait for yard cranes. Then the total operation time can be estimated by $C^T = T^S + T^Y = 2 \cdot E(t^i) + \sum_{i \in E} s^i$. If the interval of arriving time of AGVs at the yard block is longer than the expected operation of the yard crane $\frac{E(t^i)}{N^a} > E(s^i)$, yard crane will wait for the AGVs. The lower bound can be estimated by $C^T = \frac{T^S + T^Y}{N^a} = \frac{\sum_{i \in I \cup E} t^i + \sum_{i \in E} S^i}{N^a}$

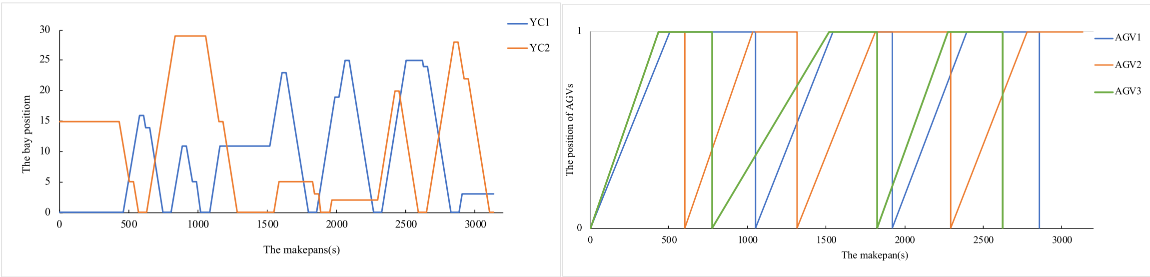
6. Computational experiments

We collect the data from automated container terminals and references to evaluate the efficiency of the model and algorithm proposed in this paper, and we analyze and discuss some insights based on the optimization results. Due to the complexity of this research problem, we use the Gurobi solver to solve the small-scale cases and the B&B-PSO to solve the large-scale cases, and compare the results of small-scale cases. The experiments were run on a PC equipped with Intel Core i5 CPU 1.60 GHz and 8 GB RAM. The B&B-PSO algorithm was programmed in Python 3.9.7, and Gurobi 9.5.2 was chosen as the MIP solver.

6.1. Parameter values

In a yard block, the number of bays is set to 30, and the number of rows in one bay is set to 6. The moving speed of the double yard cranes is the same, and is set to 1 m/s, and the pick-up/drop-off time of a container by the yard crane is 30 s. The initial position of both cranes is 0. The moving speed of AGV is 4 m/s.

According to Section 3.1, since the specific operation of the quay crane and AGV from the sea side to the yard block is not considered, we divide the operation time of this problem into the AGV operation time and the yard crane operation time. Our aim is to obtain the AGV operation time. Without considering container relocation, the time for QCs loading and unloading containers is mainly affected by the initial ship positions of containers. According to Wang and Zeng (2022), the operation time for QC to handle an export container or an import container is assumed to follow the uniform distribution U (120, 210)s. According to the actual data of AGV horizontal transportation in Qingdao port and the transportation time in Roy et al. (2020), the average horizontal transportation time of AGV is set to 300 s (300 × 100). Assuming that there are three quay cranes and one yard block. Therefore, the operating time of the AGV operation process is an integer value that follows the uniform distribution U (420, 510)s. The maximum iteration number is set to 1000 and the algorithm parameter D is set to 60.



(a) Interference between the double yard cranes (b) The trails of AGVs

Fig. 8. The operation time of YCs and AGVs.

Table 5
The result of small-size experiments.

Instance	Scale (AGVs, tasks)	Gurobi Obj (s)	Gurobi CPU time (s)	LB1 Obj (s)	LB1 CPU time (s)	LB2 Obj (s)	B&B-PSO Obj(s)	B&B-PSO CPU time (s)	Gap1 (%)	Gap2 (%)	Gap3 (%)	Gap4 (%)
1	(2,4)	1392	22.3	1380	18.4	1257	1392.0	10.5	0.00	0.86	9.70	9.70
2	(2,5)	1693	1459.3	1652	508.8	1523	1693.0	12.1	0.00	2.42	10.04	10.04
3	(2,6)	2018	3600.0	2001	3600.0	1858	2018.0	14.1	0.00	0.84	7.93	7.93
4	(3,6)	1665	3600.0	1572	3600.0	1475	1702.0	17.3	2.22	5.59	11.41	13.34
5	(2,7)	2456	3600.0	2397	3600.0	2200	2536.0	18.2	3.26	2.40	10.42	13.25
6	(3,7)	2304	3600.0	1957	3600.0	1750	2098.0	20.5	−8.94	15.06	24.05	16.59
7	(2,8)	2887	3600.0	3022	3000.0	2625	3102.0	26.7	7.45	−4.68	9.08	15.38
8	(3,8)	2905	3600.0	2583	3600.0	2250	2639.3	29.6	−9.15	11.08	22.55	14.75
9	(4,8)	2572	3600.0	2235	3600.0	2065	2442.0	25.2	−5.05	13.10	19.71	15.44
10	(2,9)	3864	3600.0	3358	3600.0	3078	3560.0	31.5	−7.87	13.10	20.34	13.54
11	(3,9)	3431	3600.0	2891	3600.0	2398	2890.0	31.3	−15.77	15.74	30.11	17.02
12	(4,9)	2954	3600.0	2384	3600.0	2185	2650.0	27.0	−10.29	19.30	26.03	17.55
13	(2,10)	4376	3600.0	3817	3600.0	3410	4015.0	29.8	−8.25	12.77	22.07	15.07
14	(3,10)	3200	3600.0	3386	3600.0	2707	3244.7	31.6	1.40	−5.81	15.41	16.57
15	(4,10)	2981	3600.0	2891	3600.0	2470	2940.0	34.5	−1.38	3.02	17.14	15.99

Notes: Gap 1= (B&B-PSO- Gurobi)/Gurobi, Gap 2= (Gurobi-LB1)/Gurobi, Gap 3=(Gurobi -LB2)/Gurobi, Gap 4=(B&B-PSO -LB2)/B&B-PSO.

6.2. *Small-scale experiments*

6.2.1. *Solution schedule*

We take a small-scale example (3, 10) to show the solution schedules. Assume there are 3 AGVs and 10 groups of container tasks, and the initial positions and the target positions of the container tasks are shown in Table 3, where container 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 are import containers and container 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 are export containers. The target bays of containers are set as Table 3. The experiment is solved by Gurobi and B&B-PSO algorithm respectively, and it is computed 10 times. We take the results that are the most similar to average values as the final results, whose assignment results are shown in Table 4.

The operation time of YC and AGV for case (3,10) are shown in Fig. 8. Fig. 8(a) shows the trails of the double yard cranes, and the y-axis represents the bay number and x-axis represents the time horizon. The initial bay position of YC1 is 0 and the initial bay position of YC2 is 15. The completion time of 10 groups of container tasks is 3132 s. Interferences between the double yard cranes are solved according to Fig. 8(a), and we can find that double yard cranes can cooperate with the container tasks. Fig. 8(b) shows the trails of three AGVs. “0” in the y-axis represents the quay side, “1” in the y-axis represents the yard, and other values in the y-axis represents the positions on the road from the quay side to yard blocks. In Section 3.1, we combine the time from yard to quay side with the time from the quay side to yard, where we use the oblique line to denote the combination time.

6.2.2. *Results for small-scale problems*

The cases where the number of containers is less than 20 are designed as small-scale cases. To evaluate the effectiveness of B&B-PSO algorithm, we designed 15 small-scale experiments and solved them by Gurobi and B&B-PSO algorithm respectively. We also solved the lower bound model [M1] by Gurobi. The upper bound of computation time for Gurobi is set to 3600 s and the value of parameter D in B&B-PSO algorithm is taken as 60. Each case is computed 10 times, and we take the average values as the results. The results are shown in Table 5. Gurobi can only solve the small-scale cases, and the computation time of Gurobi increases exponentially as the problem size gradually increases. For the first and second case, Gurobi can obtain an optimal schedule. Moreover, if the number of containers is 6 groups (12 containers), the CPU time of Gurobi has reached 3600 s while the CPU time of B&B-PSO algorithm is much shorter. Gurobi cannot solve the problem within the time limit (3600 s) when the problem size increases to a certain level. Gap 1 shows

Table 6
The result of large-size experiments.

Instance	(AGVs, tasks)	SA-PSO		B&B-PSO				Gap1 (%)	Gap2 (%)
		Obj(s)	CPU time(s)	Average(s)	Maximum(s)	Minimum(s)	CPU time(s)		
1	(3,15)	4908.2	55.2	4803.2	4945	4786	65.6	3.32	−2.19
2	(4,20)	5791.5	68.5	5688.2	5833	5565	91.1	4.82	−1.82
3	(5,20)	5658.1	72.1	5555.6	5720	5452	90.2	4.92	−1.84
4	(5,30)	8609.4	85.6	8187.4	8417	8047	167.1	4.60	−5.15
5	(6,30)	8852.8	89.2	7974.6	8251	7756	177.4	6.38	−11.01
6	(6,40)	11,368.5	145.7	10,786.0	11,047	10,486	298.2	5.35	−5.40
7	(7,40)	11,531.7	189.8	10,853.4	11,103	10,471	256.4	6.04	−6.25
8	(7,50)	12,979.0	295.3	13,494.2	13,735	13,140	463.9	4.53	3.82
9	(8,50)	14,023.2	376.9	12,805.8	13,205	12,614	515.7	4.69	−9.51
10	(8,60)	17,056.8	487.6	15,592.6	16,170	15,316	737.3	5.58	−9.39
11	(9,60)	16,905.4	573.4	15,688.8	16,043	15,333	658.0	4.63	−7.75
12	(9,80)	22,378.5	796.5	21,258.4	21,577	20,752	1665.6	3.98	−5.27
13	(10,80)	22,983.7	912.7	20,882.0	21,253	20,449	2342.1	3.93	−10.06
14	(10,100)	25,032.8	1238.1	25,631.0	25,848	25,513	3600.0	1.31	2.33
15	(12,100)	28,036.5	1610.5	26,217.4	26,939	25,454	3385.6	5.83	−6.94

Notes: Gap 1= Obj (Maximum-Minimum)/ Minimum; Gap 2= Obj (Average-SA)/ Average.

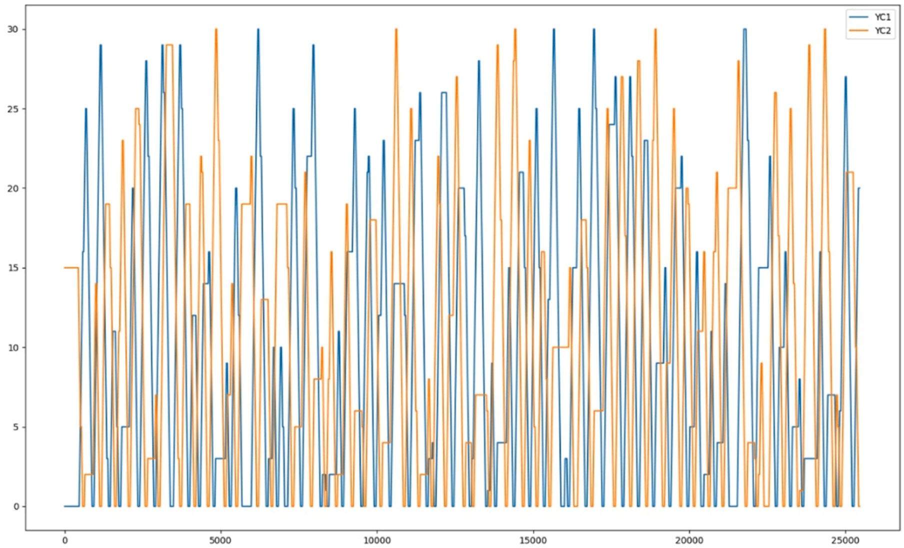


Fig. 9. Moving routes of the double yard cranes.

that the B&B-PSO can obtain an optimal schedule, and the results of Gurobi are a little better than B&B-PSO if the number of containers is small, while the results of B&B-PSO are better as the number of containers increases. Gap 2 shows the difference between the results of lower bound model [M1] and the original model [M0]. The schedules of lower bound model [M1] ignore the interference between the double yard cranes. From the first and second instance, we can find that [M0] is a tight lower bound. However it can't be solved in 1 h as the variables and constraints increases. Gap 3 shows that the difference between the results of lower bound 2 and the original model. Gap 4 shows the difference between the results of lower bound 2 and B&B-PSO. From Gap 4 we can find that the difference is under 20 %. While obtaining a perfect lower-bound quality in the integrated optimization model proves challenging, it remains permissible to utilize lower bound 2 as a means of demonstrating the stability of the proposed algorithm.

6.3. *Large-scale experiments*

We obtain 15 large-scale experiments by increasing the number of AGVs and container tasks. Large-scale experiments are solved by lower bound and B&B-PSO algorithm. The algorithm parameter D is set to 60. Each case is computed 10 times, and the minimum, maximum, average values and average computation time of B&B-PSO algorithm are recorded respectively. Results are shown in Table 6. The gap (Gap 1) between the results of SA and B&B-PSO shows that the fluctuation among the objective values for each case is under 7 %.

Furthermore, we compare the results of B&B-PSO with those obtained by Vis and Carlo (2010), who addressed the task assignment between double yard cranes. Their research problem is similar to that in this paper. However, this paper considers AGV operations and

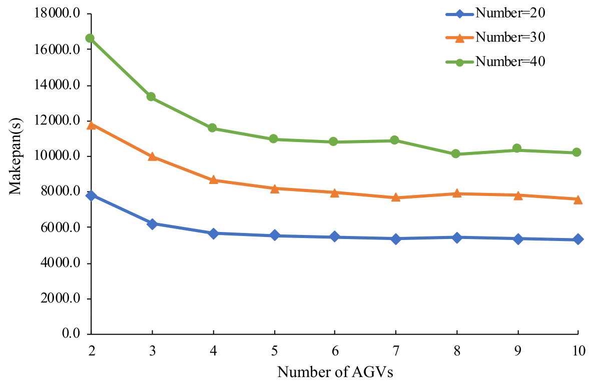


Fig. 10. The results of different numbers of AGVs.

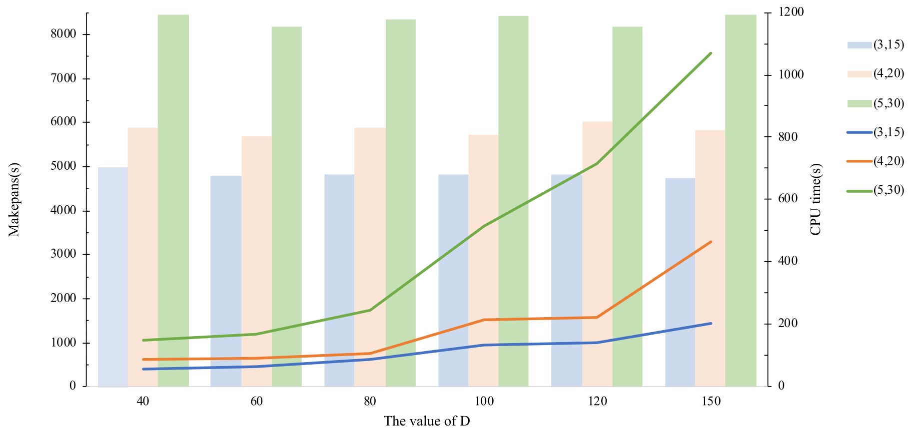


Fig. 11. The impact analysis of the value of parameter D.

the schedule is based on dual-cycle operations. We refer to the algorithm details in Vis and Carlo (2010), and apply the SA algorithm designed by Vis and Carlo (2010) to solve the double yard cranes scheduling problem in our paper (replace the B&B based algorithm). The schedule is determined by SA, and then solve the interference between the double yard cranes. The AGV task assignment is still solved by PSO designed in Section 5. Each case is computed 10 times and we record the average computation time and objective values of each case. The computation time of SA is faster than that of B&B-PSO algorithm in extremely large instances. In practical operation scenarios, terminal operators frequently create schedules for shorter time intervals and subsequently develop subsequent schedules for consecutive time periods. It is worthy to note that SA is a random searching algorithm, and it can't guarantee the quality of the solutions obtained. Conversely, the B&B based algorithm in this paper is the result of a comprehensive analysis of the operation characteristics and model, and the acceleration strategies and pruning rules are designed based on the branch and bound method. In the task assignment of the double yard cranes, the B&B based algorithm can obtain a better solution than that of SA. After integrating with PSO to solve the assignment of AGVs, the final solution of B&B-PSO is better than SA-PSO in most cases. It is worth noting, however, that there is some variability in solution performance, as indicated by Gap 2.

We take the case of (12,100) as example to obtain the schedules for double yard cranes. The solution is obtained by B&B-PSO algorithm. The parameter D is set to 60. We compute 10 times and take the results that the most near to the average results as the final moving routes of double yard cranes, which is shown in Fig. 9. From the results we can find that the interference is solved, and the double yard cranes can cooperate with the container tasks.

6.4. Analysis and discussions

6.4.1. The sensitivity analysis of the number of AGVs

For the same container tasks, different numbers of AGVs have an impact on the results of container task assignment. We calculate the results with different AGV numbers when the number of container groups is 20, 30 and 40, respectively. The value of algorithm parameter D is set to 60. Each case is computed 10 times, and we take the average value as the results. Fig. 10 shows that for the same amount of container tasks, an increase in the number of AGVs can reduce the completion time. However, it no longer has an impact on the results when the number of AGVs continue increasing. Because increasing the number of AGVs will not significantly reduce the

Table 7
The impact of different accelerating strategies.

Instance	scale	B&B-PSO		B&B-PSO without strategy1		B&B-PSO without strategy2		B&B-PSO without strategy3		Gap 1(%)	Gap 2(%)	Gap 3(%)
	(AGVs, Containers)	Obj(s)	CPU time(s)	Obj(s)	CPU time(s)	Obj(s)	CPU time(s)	Obj(s)	CPU time(s)			
1	(2,10)	4015.0	29.8	4361.7	151.2	4133.3	106.0	4016.3	403.6	8.64	2.95	0.03
2	(3,10)	3244.7	31.6	3529.7	183.9	3372.3	112.2	3246.3	695.3	8.78	3.93	0.05
3	(3,15)	4803.2	65.6	4955.0	239.1	4884.0	208.4	4724.3	1359.6	3.16	1.68	−1.64
4	(4,15)	4244.0	62.6	4654.7	236.6	4447.7	191.9	4246.0	2056.8	9.68	4.80	0.05
5	(5,15)	4188.7	58.0	4427.7	199.6	4335.7	171.0	4192.3	2422.9	5.71	3.51	0.09
6	(4,20)	5688.2	91.1	6120.7	386.9	5884.0	355.3	5810.7	3600.0	7.60	3.44	2.15
7	(5,20)	5555.6	90.2	5644.3	371.4	5653.7	304.6	5608.3	3600.0	1.60	1.77	0.95
8	(6,20)	5486.7	90.0	5779.3	293.4	5658.0	265.3	5681.7	3600.0	5.33	3.12	3.55
9	(5,30)	8187.4	167.1	8438.3	1168.4	8402.3	985.6	8370.3	3600.0	3.06	2.62	2.23
10	(6,30)	7974.6	177.4	8440.0	980.9	8190.3	917.0	8322.7	3600.0	5.84	2.70	4.37
11	(6,40)	10,786.0	298.2	11,094.3	1413.9	11,078.3	1044.0	11,165.0	3600.0	2.86	2.71	3.51
12	(7,60)	15,843.3	874.8	17,303.0	3537.9	16,812.0	3567.1	16,644.0	3600.0	9.21	6.11	5.05
13	(8,60)	15,592.6	737.3	16,698.0	2443.5	16,158.3	1840.8	16,609.7	3600.0	7.09	3.63	6.52
14	(8,80)	20,195.0	2535.3	21,864.3	3600.0	21,717.7	3600.0	21,344.0	3600.0	8.27	7.54	5.69
15	(10,100)	25,631.0	3600.0	27,711.0	3600.0	27,629.3	3600.0	27,012.7	3600.0	8.12	7.80	5.39

Notes: Gap 1= Obj(B&B-PSO without strategy1-B&B-PSO)/ B&B-PSO; Gap 2= Obj (B&B-PSO without strategy2- B&B-PSO)/ B&B-PSO; Gap 3= Obj (B&B-PSO without strategy3- B&B-PSO)/ B&B-PSO.

total operation time, and the completion time will be bounded the handling rate of other stages.

6.4.2. The sensitivity analysis of algorithm parameter D

Accelerate strategy 2 gives the value of D to cut down branches, therefore different value of D can result in different computation time. We compared the effects of the value of D on the makepans and the CPU time with cases (3, 15), (4, 20) and (5, 30). The value of D is set to 40, 60, 80, 100, 120 and 150. Each case is computed 10 times, and we take the average value as the results which are shown in Fig. 11. With the increase of D, the computation time increases especially for the case (5,30). This is because a smaller value of D will cut down more branches in the B&B method. But the completion time of different values of D is almost the same, which shows the effectiveness of the designed algorithm. If a faster solution is needed, a smaller value of D can be set, and this will not significantly impact the quality of the solution.

6.4.3. Impact of acceleration strategy on results

In Section 5.2.6, we propose several acceleration strategies to improve the solving speed of the B&B-PSO algorithm. Acceleration strategy 1 eliminates the repeated allocation schemes of AGVs, and acceleration strategies 2 and 3 remove the allocation schemes that have poor performance from the perspective of AGVs' waiting time and the completion time of container tasks respectively. In this section, the effects of different acceleration strategies on the algorithm's solution results and speed are studied. Each case is computed 10 times, and we take the average values as the results. It can be seen from Table 7 that any acceleration strategy can speed up the B&B-PSO algorithm, among which acceleration strategy 3 had the most efficient effect. By applying strategy 3, the algorithm can compute faster and the objective values are almost the same. Removing acceleration strategy 1 had a certain effect on the solution results, while removing acceleration strategies 2 and 3 had little effect on the solution results.

7. Conclusions

The integration of AGV and yard cranes plays an important role in the automated container terminal operations. In this paper, we studied a novel problem for integrated scheduling of AGV and double yard cranes in one yard block. A big crane and a small crane cooperate in a yard block to complete the unloading and loading tasks. But interference occurs between the double yard cranes. The constraints of the interference between the double yard cranes are considered when developing mathematical programming models. Dual-cycle operations for AGVs are used in this paper, in which an AGV carries an export container from the yard block after carrying an import container to the yard block. The aim of the proposed model is to obtain an optimized schedule for AGVs and double yard cranes respectively. We design a B&B based heurism algorithm to solve the task assignment of double yard cranes and a PSO algorithm is used to assign tasks for AGVs. Upper bound and lower bound are used to prune branches and improve the solving efficiency. Besides acceleration strategies designed to accelerate the algorithm. The small-scale experiments are compared with those solved by commercial solver Gurobi. Results show that our algorithm can obtain a schedule in a shorter computation time. For small scale cases, the gap between the objective obtained by Gurobi and B&B-PSO is small. For large scale cases, the proposed algorithm can also solve experiments efficiently. We also compare the results of our experiments with lower bound of the original model and other heuristic algorithms. The acceleration strategies are tested in terms of efficient and solution quality, and results show that they can obtain a solution without influencing the solution quality.

As stated in Section 3, there are two types of interferences between the double yard cranes. We have focused on one of the interferences, and the scenario that the small one cannot traverse the large one when the large one is picking up or dropping off is ignored. This scenario is neglected because the waiting time of yard cranes is short and can be ignored. In the future studies, these constraints can be considered when developing double yard crane scheduling models that incorporate these interferences. Our current model is based on a single yard block which contain double yard cranes. Because our aim is to eliminate the interferences between the double yard cranes. In the future studies, application of this research can be expanded by considering multiple yard blocks.

CRediT authorship contribution statement

Xiaoju Zhang: Conceptualization, Methodology, Validation, Writing – original draft. **Huijuan Li:** Methodology, Software, Writing – original draft. **Jiuh-Biing Sheu:** Conceptualization, Supervision, Writing – review & editing.

Data availability

No data was used for the research described in the article.

Acknowledgments

This paper is funded by National Natural Science Foundation of China (Grant Number: 71901005) and Social Science Program of Beijing Municipal Education Commission (Grant number: SM202010011008). This paper is also supported by the grants NSC 112-2410-H-002-046-MY3, MOST-111-2410-H-002-089-MY3 and MOST 109-2410-H-002-076-MY3 from the National Science Council of Taiwan, R.O.C. The authors also wish to thank the handling editors and referees for their helpful comments and suggestions. Any errors or omissions remain the sole responsibility of the authors.

References

Ahmed, E., EI-Abbasy, M.S., Zayed, T., et al., 2021. Synchronized scheduling model for container terminals using simulated double-cycling strategy. Comput. Ind. Eng. 154 (1), 107118.

Chen, X., He, S., Zhang, Y., et al., 2020. Yard crane and AGV scheduling in automated container terminal: a multi-robot task allocation framework. Transport. Res. Part C: Emerg. Technol. 114, 241–271.

Cao, J., Lee, D., Chen, J., et al., 2010. The integrated yard truck and yard crane scheduling problem: benders' decomposition-based methods. Transport. Res. Part E 46 (3), 344–353.

Cao, Z., Lee, D., Meng, Q., 2008. Deployment strategies of double-rail mounted gantry crane systems for loading outbound containers in container terminals. Int. J. Prod. Econ. 115 (1), 221–228.

Chu, F., He, J., Zheng, F., et al., 2019. Scheduling multiple yard cranes in two adjacent container blocks with position-dependent processing times. Comput. Ind. Eng. 136, 355–365.

Dorndorf, U., Schneider, F., 2010. Scheduling automated triple crossr-over stacking cranes in a container yard. OR Spectrum 32 (3), 617–632.

Froyland, G., Koch, T., Megow, N., et al., 2008. Optimizing the landside operation of a container terminal. OR Spectrum 30 (1), 53–75.

Grunow, M., Günther, H.O., Lehmann, M., 2006. Strategies for dispatching AGVs at automated seaport container terminals. OR Spectrum 28 (4), 587–610.

Gharehgozli, A.H., Laporte, G., Yu, Y., et al., 2015. Scheduling twin yard cranes in a container block. Transport. Sci. 49 (3), 686–705.

Goodchild, A.V., Daganzo, C.F, 2007. Crane double cycling in container ports: planning methods and evaluation. Transport. Res. Part B 41 (8), 875–891.

Hop, D.C., Van Hop, N., Anh, T.T.M, 2021. Adaptive particle swarm optimization for integrated quay crane and yard truck scheduling problem. Comput. Ind. Eng. 153, 107075.

Hu, Z., Sheu, J., Luo, J.X., 2016. Sequencing twin automated stacking cranes in a block at automated container terminal. Transport. Res. Part C: Emerg. Technol. 69, 208–227.

Jung, S.H., Kim, K.H, 2006. Load scheduling for multiple quay cranes in port container terminals. J. Intell. Manuf. 17 (4), 479–492.

Kaveshgar, N., Huynh, N., 2015. Integrated quay crane and yard truck scheduling for unloading inbound containers. Int. J. Product. Econ. 159, 168–177.

Kress, D., Dornseifer, J., Jaehn, F., 2019. An exact solution approach for scheduling cooperative gantry cranes. Eur. J. Oper. Res. 273 (1), 82–101.

Lau, H.Y.K., Zhao, Y., 2008. Integrated scheduling of handling equipment at automated container terminals. Ann. Oper. Res. 159 (1), 373–394.

Lee, L.H., Chew, E.P., Tan, K.C., et al., 2010. Vehicle dispatching algorithms for container transshipment hubs. OR Spectrum 32 (3), 663–685.

Lee, D.H., Cao, Z., Meng, Q., 2007. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. Int. J. Prod. Econ 107 (1), 115–124.

Luo, J., Wu, Y., 2015. Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals. Transport. Res. Part E: Logist. Transport. Rev. 79, 49–64.

Li, J.J., Yang, J.Y., Xu, B.W., et al., 2022. A flexible scheduling for twin yard cranes at container terminals considering dynamic cut-off time. J. Mar. Sci. Eng. 10 (5), 675.

McKinsey & Company. The future of automated ports, 2018, <https://www.mckinsey.com/industries/travel-transport-and-logistics/our-insights/the-future-of-automatedports>.

Nossack, J., Briskorn, D., Pesch, E., 2018. Container dispatching and conflict-free yard crane routing in an automated container terminal. Transport. Sci. 52 (5), 1059–1076.

Rui, Z., Jin, Z., Yu, M., et al., 2015. Optimization for two-stage double-cycle operations in container terminals. Comput. Ind. Eng. 83, 316–326.

Roy, D., De Koster, R., Bekker, R., 2020. Modeling and design of container terminal operations. Oper. Res. 68 (3), 686–715.

Speer, U., Fischer, K., 2017. Scheduling of different automated yard crane systems at container terminals. Transport. Sci. 51, 305–324.

Saanen, Y.A., Valkengoed, M.V., 2005. Comparison of three automated stacking alternatives by means of simulation. In: Proceedings-Winter Simulation Conference, 1574425, pp. 1567–1576.

Stahlbock, R., Voß, S., 2010. Efficiency considerations for sequencing and scheduling of double-rail-mounted gantry cranes at maritime container terminals. Int. J. Shipping Transport Logist. 2, 95–123.

Vis, I.F.A., Carlo, H.J., 2010. Sequencing two cooperating automated stacking cranes in a container terminal. Transport. Sci. 44 (2), 169–182.

Wang, Z.H., Zeng, Q.C., 2022. A branch-and-bound approach for AGV dispatching and routing problems in automated container terminals. Comput. Ind. Eng. 166, 107968.

Xu, B.W., Jie, D.P., Li, J.J., et al., 2021. Integrated scheduling optimization of U-shaped automated container terminal under loading and unloading mode. Comput. Ind. Eng. 162, 107695.

Yang, Y., Zhong, M., Dessouky, Y., et al., 2018. An integrated scheduling method for AGV routing in automated container terminals. Comput. Ind. Eng. 126, 482–493.

Zhou, W., Wu, X., 2009. An efficient optimal solution of a two-crane scheduling problem. Asia-Pacific J. Oper. Res. 26 (1), 31–58.

Zhang, X., Zeng, Q., Yang, Z., 2016. Modeling the mixed storage strategy for quay crane double cycling in container terminals. Transport. Res. Part E: Logist. Transport. Rev. 94, 171–187.

Zhang, R., Jin, Z., Ma, Y., et al., 2015. Optimization for two-stage double-cycle operations in container terminals. Comput. Ind. Eng. 83 (5), 316–326.