

## **A hybrid two-stage algorithm for solving synchronous loading operation of quay cranes and straddle carriers**

---

Xuehao Feng, Cong Jin and Jiadong Zhu

Ocean College,  
Zhejiang University,  
Zhejiang, China  
Email: fengxuehao@zju.edu.cn  
Email: jincong151@163.com  
Email: 294176994@qq.com

Yanjie Zhou\*

School of Management,  
Zhengzhou University,  
Zhengzhou 450001, China  
Email: icyjzhou@zzu.edu.cn  
\*Corresponding author

**Abstract:** Container port operation is essential for the vessel's turnaround time. This paper studies the synchronous loading operation of quay cranes and straddle carriers, considering various practical constraints. A mathematical model is proposed in a flexible container operation sequence by minimising the sum of the quay crane completion time and the straddle carrier waiting time. To solve the problem studied, a two-stage algorithm is designed based on greedy strategy and dynamic adjustment. Various experiments are conducted to verify the performance of the proposed solution method. The experimental results show that the two-stage algorithm has a better solution performance on small and medium-sized problems. Finally, the sensitivity analysis is conducted. [Submitted: 29 October 2024; Accepted: 27 January 2025]

**Keywords:** quay cranes and straddle carriers; optimisation model; genetic algorithm; greedy strategy.

**Reference** to this paper should be made as follows: Feng, X., Jin, C., Zhu, J. and Zhou, Y. (2026) 'A hybrid two-stage algorithm for solving synchronous loading operation of quay cranes and straddle carriers', *European J. Industrial Engineering*, Vol. 21, Nos. 3/4, pp.328–370.

**Biographical notes:** Xuehao Feng received his PhD in Logistics Information Technology at Pusan National University in 2013 and received his BS and MS in Logistics Management from Beijing Jiaotong University in 2007 and 2009, respectively. He is currently a Professor with Ocean College at Zhejiang University. He focuses on smart port.

Cong Jin is currently pursuing her Master's degree at Ocean College of Zhejiang University. She focuses on smart port.

Jiadong Zhu received his MS at Ocean College of Zhejiang University. He focuses on smart port.

Yanjie Zhou received his PhD from the Department of Industrial Engineering at Pusan National University in 2020, and received his BS and MS in Computer Science and Computer Applied Technology from Zhengzhou University in 2012 and 2015, respectively. He is currently an Associate Professor with the School of Management at Zhengzhou University. He focuses on solving real-world optimisation problems by using artificial intelligence techniques.

---

## 1 Introduction

Over the past three decades, with the continuous development of economic integration and the deepening of trade globalisation, the port and logistics industry has developed rapidly worldwide (Zhou and Kim, 2020; Zhou et al., 2024b). As an essential transit facility for sea transport and inland transport, ports play a pivotal role in the global logistics system and have become the critical pivot to support the operation of the global supply chain network and the main hub to serve various countries' import and export trade, and port planning has a far-reaching impact on carbon emissions (Feng et al., 2025). Container transport, as one of the most important modes of maritime transport, is characterised by easy operation, high efficiency, and convenient operation. Therefore, the management of container transport, storage, loading, and unloading in port management has attracted more and more attention from the business and theoretical communities.

Congestion both at the seaside and yardside will significantly increase the vessel's turnaround time. Improving the operation efficiency of port facilities, such as quay cranes (QCs) and yard cranes, will reduce the vessel turnaround time (Feng et al., 2022). The previous studies focused on the QC and AGV/straddle carrier (SCs) assignment for reducing congestion at the seaside in a high level of planning. The synchronous relationship between the QC and SCs is also essential, which has been omitted by previous studies. Without considering the synchronous relationship between the QC and SCs, congestion along the seaside area of the container terminal will also be generated. Hence, this study investigates the problem of simultaneous operation of frontline SCs and QCs at container port terminals to synchronise the SCs and QCs. The contributions of this paper are summarised as follows.

To formulate the synchronous relationship between the QC and SCs, a mixed-integer planning model is constructed. The proposed mathematical model, which is used to analyse the problem theoretically and explore the mechanism of the influence of different parameters on the complexity of the problem, could obtain optimal solutions for the small-scale problem. The studied problem is an online optimisation problem and the proposed mathematical model could be used to solve small-scale instances. As the studied problem is a real-time decision problem, the decision should be made quickly. To handle large-scale size problems, this paper proposed a dynamically adjusted

heuristic algorithm with a greedy strategy that could solve large-scale problems within an acceptable time. To verify the performance of the proposed algorithms, various experiments are conducted. The experimental results verified the correctness of the proposed mathematical and showed the proposed algorithm could solve large-scale instances. Finally, a sensitive analysis is performed to evaluate the proposed algorithm.

The rest of the work is organised as follows. Section 2 summarises and discusses the related literature. The problem is described in Section 3. Section 4 presents a two-layer genetic algorithm based on a dynamically tuned greedy strategy. The experimental results are presented in Section 5. Conclusions are outlined in the last section.

## 2 Literature review

This section presents previous studies related to the problem studied.

### 2.1 *QC related studies*

Fu and Diabat (2015) introduced a Lagrange relaxation and heuristic algorithms to solve the assignment and scheduling of QCs. Shang et al. (2016) considered the setting time of the QC moving along the wharf and established a deterministic model for joint scheduling optimisation of berth and QC. Considering the constraint that the QC cannot cross, Zhang et al. (2017) set up a mixed integer programming model for the QC scheduling problem to minimise the working time of container ships in port. Liang et al. (2018) studied the scheduling problem of QCs from task scheduling and quantity allocation. A coupling model was established by analysing the relationship between the two problems. Sun et al. (2019) introduced a new QC scheduling problem model and developed a method to deal with non-cross constraints more easily by dealing with the structure of workload distribution. Kenan et al. (2022) address the integrated QC assignment and scheduling problem while considering carbon regulations and formulating two mixed integer programming models to analyse the effects of carbon taxation and cap-and-trade. Li et al. (2024a) studied the unidirectional QC scheduling problem-based on the uncertainty of container terminal loading and unloading time and established a two-stage robust optimisation model. Wang et al. (2025) introduce a scheduling problem for automatic yard cranes and propose two heuristic methods based on the greedy randomised adaptive search procedure to address this issue.

Ji et al. (2023) tackle the BACAP with stochastic vessel arrival times to create an effective scheduling method, introducing a scenario generation technique to simulate these arrivals and developing a mixed integer linear programming model aimed at minimising vessels' total stay time in port. Chargui et al. (2023) consider the problem of berth and quayside assignment and scheduling with uncertainty and energy efficiency by using a new robust exact decomposition algorithm. Wang et al. (2024b) analysed the allocation of berths and quaysides under the onshore power distribution of container terminals. Zheng et al. (2024) examine berth allocation and QC assignment with shifting strategies using an adaptive large neighbourhood search algorithm. Computational results show the effectiveness of both the model and the ALNS across various instance sizes. Wang et al. (2024a) are the first to introduce a distributed robust chance constraint (DRCC) optimisation model to solve the problem of integrated berth allocation and QC allocation in container terminals.

Based on dual-cycling, El-Abbasy et al. (2021) considered optimising container handling multi-stage fleet size. Yue et al. (2023) explored the block allocation and loading and unloading equipment scheduling of automated container terminals. Based on dual-cycling, Li et al. (2024b) explored the precise algorithm of multi-equipment integrated scheduling for automated container terminals. Zhou et al. (2024a) consider the problem of continuous loading and unloading of uncertain ships and offshore platforms, and establish a Markov decision process model. The results show that the strategy considering double-cycle operation is superior to other strategies.

## *2.2 Horizontal transportation equipment related studies*

Klerides and Hadjiconstantinou (2011) studied the AGV scheduling problem that can transport multiple containers simultaneously and built a model to minimise the allocation cost. Nossack and Pesch (2013) studied the truck scheduling problem in multimodal container transportation, considered the empty container call, and established a truck-loading task allocation model with a time window to minimise the truck-loading running time and solved it by the two-stage heuristic method. Cai et al. (2014) studied the re-planning strategy when new tasks arrived. Kavakeb et al. (2015) studied the impact of a new intelligent vehicle technology on the performance and cost of a European port. Cao et al. (2023) explored the AGV scheduling and two-way conflict-free path of automated container terminals. Gong et al. (2024) explored the real-time AGV scheduling optimisation method of container terminals based on deep reinforcement learning. Based on the problem of ship loading and unloading, Fereidoonian et al. (2024) build a timely, efficient, and multi-objective truck-sharing integrated scheduling model for container terminals, which has made remarkable progress in reducing emissions.

Wang et al. (2017) combined vehicle scheduling with yard crane scheduling and storage allocation, simultaneously considering loading and unloading tasks. Zhu et al. (2022) mainly discussed the comprehensive optimisation of AGV and double yard crane in an automatic container terminal, aiming at improving the working efficiency of the terminal by coordinating multiple sub-operations. Xu et al. (2024) explored the influence of critical factors such as the speed, acceleration, and quantity of AGV on the horizontal transportation efficiency of automated container terminals.

## *2.3 Summary*

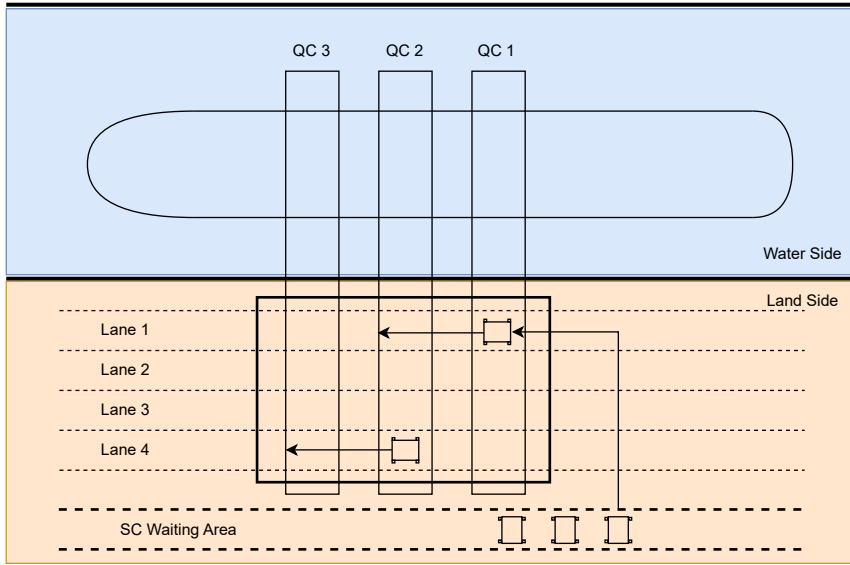
SCs will be affected by the loading and unloading operations of QCs. In the container terminal, SC operation and quayside loading and unloading operation of the same container are two separate and independent operations, which make it possible for the SC and quayside to interact with each other, resulting in additional delay costs. However, previous studies regarded the disembarkation of the QC as a whole buffer zone and considered the capacity of the buffer zone to be a constraint for straddle truck operation. Considering the above practical constraints, this paper will study the optimisation of synchronous operation of the QC and SC to reduce the delay time caused by the conflict between the QC and SC. This paper will provide a theoretical basis for practical operation optimisation.

### 3 Mathematical model

#### 3.1 Problem description

This section discusses the problem of synchronised ship loading operations for QCs and SCs. It includes multiple adjacent QCs and their container loading tasks, divided into two main parts: the unloading operations for SCs and the ship loading operations for QCs. Figure 1 shows an example of the studied problem.

**Figure 1** An example of the studied problem (see online version for colours)



In container terminals where SCs serve as the primary means of horizontal transportation, the simultaneous loading of the QCs and SCs is as follows: upon the arrival of the container carrier at its designated berth, the assigned QC begins its operations. The yard crane grabs the containers to be loaded and drops them at the selected location in the yard, and then the unloaded SC picks up the containers by itself and transports them to the operation area of the QC. There are several lanes under the QC for the SC to travel and drop off containers, but each lane has only one drop-off position under each QC. The drop-off position of the SC refers to the intersection of a vertical area of the QC and a lane, which has the same shape and location as a standard container, i.e., only one container can be placed. After arriving at the front of the terminal, the SC must choose a lane for unloading. Only when the target drop-off location is unobstructed, and no QC is operating in the adjacent lane, can the SC proceed. The SC's travel route must be clear of any obstructions, enabling it to reach the drop-off location and initiate the unloading operation. Furthermore, when the SC exits the lane, it must ensure that its departure path is also free from any obstacles.

The QC loads containers on the lane. For large container carriers, due to the large number of containers to be loaded and unloaded, several QCs are often required to carry out loading and unloading operations to ensure that the ship can complete the

loading and unloading tasks within a certain period. The QC first grabs a container in the operation area according to the operation plan, unloads it to the designated position on the ship, then leaves and continues carrying out the next container's loading and unloading task. If there is a SC operating in the path of the QC with containers, the QC must wait for the SC to complete its operation and leave before continuing with the loading operation. When all containers are loaded, the simultaneous loading of the QC and SC is complete.

It can be found that although the QC operation and the SC are not connected in terms of hardware equipment, there is a strong mutual influence relationship between their operations. The aim of this paper is to synchronise the loading operation of QC and SC, considering various constraints while minimising the QC and SC operation time.

### 3.2 Mixed integer programming model

The assumptions used in this paper are summarised as follows:

- 1 the arrival time of all containers and their SCs are known, and the SC can be parked in the temporary waiting area, which is shown at the bottom of Figure 1
- 2 the travel time of the SC at the lane can be ignored
- 3 the time required to release one container for each SC is fixed
- 4 the time required to pick up one container for each spreader of the QC is fixed.

This paper aims to optimise the simultaneous operation of QCs and SCs to improve quay loading efficiency. For the ship loading operation of a QC, completing all container loading tasks as soon as possible is one of the goals pursued during port operations. This goal can be converted into minimising the completion time of the last task of the QC. At the same time, reducing their waiting time at the QC front for SCs can improve efficiency and reduce costs. Therefore, this paper constructs a weighted objective function and assigns different weight values to the completion time and total waiting time of SCs. Equation (1) shows the objective function. The notations and decision variables used in this paper are shown in Tables 12 and 13 of Appendix A.

$$\min a \times \sum_{k \in K} \max_{i \in N_k} (F_{ik}^q + e_{ik}) + b \times \sum_{k \in K} \sum_{i \in N_k} W_{ik} \quad (1)$$

subject to

$$S_{ik}^s \geq A_{ik} \quad \forall k \in K, \forall i \in N_k \quad (2)$$

$$W_{ik} = F_{ik}^s - A_{ik} - t_s \quad \forall k \in K, \forall i \in N_k \quad (3)$$

$$\sum_{j \in M_k} D_{ijk} = 1 \quad \forall k \in K, \forall i \in M_k \quad (4)$$

$$\sum_{i \in M_k} D_{ijk} = 1 \quad \forall k \in K, \forall j \in M_k \quad (5)$$

$$R_{ik} - R_{jk} + n_k \times D_{ijk} \leq n_k - 1 \quad \forall k \in K, \forall i, j \in N_k \quad (6)$$

$$R_{jk} > G_{ijk} R_{ik} \quad \forall k \in K, \forall i, j \in N_k \quad (7)$$

$$S_{ik}^s + t_s \leq F_{ik}^s \quad \forall k \in K, \forall i \in N_k \quad (8)$$

$$S_{ik}^q + t_q \leq F_{ik}^q \quad \forall k \in K, \forall i \in N_k \quad (9)$$

$$F_{ik}^s \leq S_{ik}^q \quad \forall k \in K, \forall i \in N_k \quad (10)$$

$$F_{ik}^q + e_{ik} \leq S_{jk}^q + (1 - D_{ijk}) \times M \quad \forall k \in K, \forall i, j \in N_k \quad (11)$$

$$S_{jk}^q - F_{ik}^s < M \times Z_{ijk}^{sq} \quad \forall k \in K, \forall i, j \in N_k \quad (12)$$

$$F_{ik}^s - S_{jk}^q \leq M \times (1 - Z_{ijk}^{sq}) \quad \forall k \in K, \forall i, j \in N_k \quad (13)$$

$$S_{jk}^s - F_{ik}^q < M \times Z_{ijk}^{qs} \quad \forall k \in K, \forall i, j \in N_k \quad (14)$$

$$F_{ik}^q - S_{jk}^s \leq M \times (1 - Z_{ijk}^{qs}) \quad \forall k \in K, \forall i, j \in N_k \quad (15)$$

$$X_{jl} - X_{ik} \leq M \times L_{ijkl} \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (16)$$

$$X_{ik} - X_{jl} < M \times (1 - L_{ijkl}) \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (17)$$

$$F_{jk}^q - F_{ik}^s \geq M \times (L_{ijkk} - Z_{ijk}^{sq} - Z_{jik}^{qs} - 1) \quad \forall k \in K, \forall i, j \in N_k \quad (18)$$

$$2 - X_{ik} + X_{jk} > -M \times U_{ijk} \quad \forall k \in K, \forall i, j \in N_k \quad (19)$$

$$2 - X_{ik} + X_{jk} \leq M \times (1 - U_{ijk}) \quad \forall k \in K, \forall i, j \in N_k \quad (20)$$

$$1 - U_{ijk} - U_{jik} \leq Z_{ijk}^{sq} + Z_{jik}^{qs} \quad \forall k \in K, \forall i, j \in N_k \quad (21)$$

$$X_{jk} - X_{ik} < M \times L_{ijkk} + M \times (Z_{ijk}^{qs} + Z_{jik}^{qs}) \quad \forall k \in K, \forall i, j \in N_k, i \neq j \quad (22)$$

$$S_{jl}^s - S_{ik}^s < M \times Z_{ijkl}^{as} \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (23)$$

$$S_{ik}^s - S_{jl}^s \leq M \times (1 - Z_{ijkl}^{as}) \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (24)$$

$$S_{jl}^s - F_{ik}^q < M \times Z_{ijkl}^{qa} \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (25)$$

$$F_{ik}^q - S_{jl}^s \leq M \times (1 - Z_{ijkl}^{qa}) \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (26)$$

$$S_{jl}^s - F_{ik}^s < M \times Z_{ijkl}^{ds} \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (27)$$

$$F_{ik}^s - S_{jl}^s \leq M \times (1 - Z_{ijkl}^{ds}) \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (28)$$

$$F_{jl}^s - F_{ik}^q < M \times Z_{ijkl}^{qd} \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (29)$$

$$F_{ik}^q - F_{jl}^s \leq M \times (1 - Z_{ijkl}^{qd}) \quad \forall k, l \in K, \forall i \in N_k, \forall j \in N_l \quad (30)$$

$$1 \leq L_{ijkl} + L_{jilk} + Z_{ijkl}^{as} + Z_{jilk}^{qa} \quad \forall k, l \in K, k > l, \forall i \in N_k, \forall j \in N_l \quad (31)$$

$$1 \leq L_{ijkl} + L_{jilk} \quad \forall k, l \in K, k < l, \forall i \in N_k, \forall j \in N_l \quad (32)$$

$$Z_{ijk}^{qs}, Z_{ijk}^{sq}, U_{ijk} \in \{0, 1\} \quad \forall k \in K, \forall i, j \in N_k \quad (33)$$

$$L_{ijkl}, Z_{ijkl}^{as}, Z_{ijkl}^{qa}, Z_{ijkl}^{ds}, Z_{ijkl}^{qd} \in \{0, 1\} \quad \forall k, l \in K, k \neq l, \forall i \in N_k, \forall j \in N_l \quad (34)$$

$$X_{ik} \in \{1, 2, \dots, m\} \quad \forall k \in K, \forall i \in N_k \quad (35)$$

$$R_{ik} \in \{1, 2, \dots, N_k\} \quad \forall k \in K, \forall i \in N_k \quad (36)$$

Constraint (2) means that SC could start its releasing operation only after it arrives at the apron. Constraint (3) calculates the waiting time of SCs for each container. Constraints (4)–(7) ensure that one container can be loaded on the vessel until the container below has been loaded. Among them, (4) and (5) determine the operation sequence of the QC handling containers; constraint (6) ensures that the sub-loop does not appear in the sequence. Finally, the constraint (7) indicates that the container located

below has higher priority in the sequence. Constraint (8) ensures that the finishing time of the SC release operation is larger than the starting time of the same operation plus the required release time. Constraint (9) ensures that the finishing time of the QC pickup operation is larger than the starting time of the same operation, plus the required pickup time by QC. Constraint (10) means that the QC can start its pickup operation only after the completion of the SC releasing operation for the same container. Constraint (11) means that the QC starts its pickup operation after it finishes the entire job for the previous container. Constraints (12) to (17) are definitions of some (0, 1) variables. Constraint (18) ensures that the QC pickup operation for the  $j^{\text{th}}$  container should be delayed until the SC deliver operation for the  $i^{\text{th}}$  container finishes when these two handling times are overlapped with each other and the  $i^{\text{th}}$  container is located closer to the waterside. Constraint (19) and (20) indicate the meaning of  $U^{ijk}$ . Constraint (21) means that the QC pickup operation for one container and the SC release operation for another container cannot be performed in an overlapped time when these two containers are assigned to adjacent lanes. Constraint (22) guarantees that no two containers can be planned to be put down during the overlapped periods at the same lane. Constraints (23) to (30) are definitions of some (0, 1) variables. Constraint (31) ensures that the road is clear when the SC is going to the destination lane. Constraint (32) ensures that the road is clear when the SC leaves the lane. Constraints (33) and (34) limit the range of values for the (0, 1) variable. Constraints (35) limit the lane selection of the container.

### 3.3 Model analysis

From the mathematical model, we can find that the number of continuous decision variables in the model is  $5 \sum_{k \in K} |N_k|$ ; the number of integer decision variables is equal to  $2 \sum_{k \in K} |N_k| + 4 \sum_{k \in K} |N_k|^2 + 5 \sum_{k \in K} \sum_{l \in K} |N_k| |N_l|$ , in which the number of (0, 1) variables is equal to  $4 \sum_{k \in K} |N_k|^2 + 5 \sum_{k \in K} \sum_{l \in K} |N_k| |N_l|$ , and the number of artificial virtual integer variables is equal to  $\sum_{k \in K} |N_k|$ . The number of constraints is equal to  $6 \sum_{k \in K} |N_k| + 2|K| + 15 \sum_{k \in K} |N_k|^2 + 10 \sum_{k \in K} \sum_{l \in K} |N_k| |N_l|$ .

## 4 A two-stage algorithm

Solving the above mathematical model is very time-consuming, especially with the increase in the problem size. The studied problem needs an online decision. Hence, a real-time decision algorithm is required. Meta-heuristics are usually faster than the mathematical modelling approach and have been used to solve many engineering optimisation problems (Shi et al., 2022, 2023; Gao et al., 2017).

### 4.1 Framework

In this paper, the synchronous loading operation of QCs and SCs can be categorised into three parts: the loading sequence of the QC, the order of decision-making of the SC and its container, and the lane allocation problem of the SC. The loading sequence of the QCs is determined flexibly, taking into account the pre-established stowage plan as well as the container storage conditions in the lane. The order of decision-making, that is, the priority of the SC to select lanes, is one of the factors affecting the allocation of the SC.

Varying decision-making sequences lead to different allocation outcomes. The problem of lane allocation is selecting an optimal lane based on the SC's known information and its container.

**Algorithm 1** Pseudocode of double-layer genetic algorithm

---

```

input : max_gen_U, iteration times of upper population; max_gen_L, iteration times
          of lower population.
output: Optimal individual.
1 Generate upper initial population;
2 for  $i \leftarrow 0$  to max_gen_U do
3   Converted into the loading and unloading sequence of the quay crane;
4   Genetic operation (selection, crossover and mutation);
5   Produce a new upper population;
6   Generate lower initial population;
7   for  $j \leftarrow 0$  to max_gen_L do
8     Genetic operation (selection, crossover and mutation);
9     Regeneration of lower population;
10    Straddle lane assignment;
11    Fitness calculation;
12  Output lower-level individuals and their fitness.

```

---

Thus, this paper proposes a two-layer based genetic algorithm (TLGA). The first stage determines the loading sequence of the QC, and the second stage determines the SC's decision order. Finally, we need to determine the lane allocation when calculating the objective function. Algorithm 1 shows the pseudo-code of the two-stage algorithm.

- Step 1 Generate the initial population of the upper layer TLGA. The initial population is randomly generated, and the unqualified chromosomes are repaired so the corresponding QC operation sequence can satisfy the known ship stowage plan.
- Step 2 Generate the initial population of the lower layer TLGA. Based on a chromosome encoding of the upper layer, a lower layer chromosome that meets the requirements is generated; that is, the decision order of the SCs is determined based on the sequence of the QC operation.
- Step 3 Based on the QC operation sequence and the SC decision-making sequence, a heuristic algorithm generates a specific lane allocation scheme and computes the objective function value. The fitness of the chromosome combination is then determined using the fitness function, and the resulting fitness value is recorded.
- Step 4 The lower layer algorithm iterative termination test. Determine whether the maximum number of iterations has been reached. If yes, go to step 6; If not, go to step 5.
- Step 5 Perform the lower layer algorithm genetic operation. Genetic operations such as selection, crossover, and mutation generate a new population of the lower layer algorithm. Go back to step 3.

- Step 6 Output the optimal chromosome and its corresponding fitness value from the lower algorithm, and record it as the combined chromosome and fitness value associated with the upper chromosome.
- Step 7 The upper layer algorithm iterative termination test. Determine whether the maximum number of iterations has been reached. If yes, go to step 9; If not, go to step 8.
- Step 8 Perform the genetic operation of the upper layer algorithm. Genetic operations such as selection, intersection, and mutation generate a new population of the upper algorithm. Go back to step 2.
- Step 9 Output the optimal individual, the QC operation sequence, and the objective function value. Then, record the SC decision sequence, lane allocation plan, and end.

After multiple generations of populations, the individuals in the population tend to be consistent, and the algorithm easily reaches the local convergence state, which may be detrimental to the global optimal solution. Therefore, the restart mechanism is set. When the difference between the optimal fitness and the average fitness within the population is smaller than a certain ratio, the restart mechanism is triggered, and individuals with lower fitness within the population are reinitialised. The restart mechanism can increase the diversity of the population while preserving good. Individuals, avoiding the prematurity of the algorithm and falling into the local optimum. We can get the QC operation sequence and the lane allocation scheme of all SCs and their containers through the above algorithm flow.

## 4.2 Encoding and decoding of a solution

The following contents introduce the encoding and decoding of a solution for the first stage and the second stage, respectively.

### 4.2.1 First stage

#### 4.2.1.1 Encoding for the first stage

In the first stage, a chromosome is encoded as an integer array with length  $n$  ( $n$  is the number of the containers), indicating the loading sequence of the QC. Firstly, we generate a basic chromosome with ascending genes and randomly exchange the positions between genes to generate the initial chromosome of the first stage. 15 containers are going to be loaded by two adjacent QCs, among which containers no. 1 to no. 8 belong to the working range of the first QC. The remaining containers belong to the working range of the second QC. According to the chromosome in Figure 2, the loading sequence of the first QC is  $i_1, i_3, i_2, i_6, i_8, i_4, i_5, i_7$ , and the loading sequence of the second QC is  $i_{10}, i_9, i_{12}, i_{13}, i_{11}, i_{14}, i_{15}$ .

#### 4.2.1.2 Decoding for the first stage

In the decoding process, the gene sequence within the chromosome is employed to represent the operational sequence of the QC when loading containers, with each gene

value corresponding to a specific container number. The relative order of the genes dictates the sequence in which containers are loaded, thereby influencing the efficiency and coordination of the overall loading plan. In the case of cooperative loading involving multiple QCs, chromosome decoding must not only determine the loading order for each container but also allocate these containers to different QCs in a manner that aligns with the loading strategy. This ensures the cooperative optimisation of the QCs operations.

**Figure 2** Encoding for the first stage (see online version for colours)

QC $k$ :	$k_1$	$k_1$	$k_1$	$k_1$	$k_1$	$k_1$	$k_1$	$k_2$	$k_1$	$k_2$	$k_2$	$k_2$	$k_2$	$k_2$	$k_2$
Container $i$ :	$i_1$	$i_3$	$i_2$	$i_6$	$i_4$	$i_5$	$i_{10}$	$i_7$	$i_9$	$i_{12}$	$i_{13}$	$i_{11}$	$i_{14}$	$i_{15}$	

In this way, the decoding not only realises the transformation of chromosomes into the actual loading sequence and the allocation scheme of QCs but also ensures that high-quality solutions can be generated in multi-QCs collaborative optimisation. The effectiveness of the decoding strategy is directly related to the overall efficiency of the loading task of single-QC or multi-QCs and the convergence speed of the algorithm. Therefore, designing the container sequence and QC allocation in the decoding process is very important.

### 4.2.2 Second stage

#### 4.2.2.1 Encoding

The second stage’s chromosome represents the SC’s decision order and the container’s decision order. As shown in Figure 3, the first gene 1 of the chromosome indicates that the algorithm prioritises the SC to which the first container belongs, and the second gene 3 indicates that the third container is arranged according to the known relevant time information after the first container is allocated. Based on the chromosome order, a lane allocation scheme is generated. The initial lane assignments are made according to the first-come, first-served principle, thereby constructing an initial feasible solution.

**Figure 3** Container lane assignment (see online version for colours)

Container:	$i_1$	$i_3$	$i_2$	$i_6$	$i_4$	$i_8$	$i_5$	$i_{10}$	$i_7$	$i_9$	$i_{12}$	$i_{13}$	$i_{11}$	$i_{14}$	$i_{15}$
Lane:	$m_1$	$m_2$	$m_3$	$m_4$	$m_1$	$m_2$	$m_3$	$m_4$	$m_1$	$m_2$	$m_3$	$m_4$	$m_1$	$m_2$	$m_3$

#### 4.2.2.2 Decoding

Figure 4 shows the structure of variables, where  $i$  stands for container,  $k$  stands for QC, and the value of  $X_{ik}$  stands for lane number for the  $i^{\text{th}}$  container loaded by the  $k^{\text{th}}$  QC. In the decoding process, the gene sequence in the chromosome is used to represent the order of loading containers by SCs, in which the value of each gene represents the lane currently allocated by SCs, that is, the value of variable  $X_{ik}$ . The gene sequence not only determines the sequence of loading tasks performed by SCs but also directly

reflects the lane assignments used by SCs at each time node. Through the decoding process, the system can convert the abstract genetic information in the chromosome into the actual SC scheduling plan, including the order of transporting containers by SCs and their lane assignment during the operation, thus ensuring that SCs can make rational use of available lane resources in different periods and avoiding the conflict and waste of lane resources.

**Figure 4** Structure of variable  $X_{ik}$  (see online version for colours)

Variable:	$X_{ik}$															
QC:	$k_1$	$k_1$	$k_1$	$k_1$	$k_1$	$k_1$	$k_1$	$k_2$	$k_1$	$k_2$	$k_2$	$k_2$	$k_2$	$k_2$	$k_2$	$k_2$
Container:	$i_1$	$i_3$	$i_2$	$i_6$	$i_4$	$i_8$	$i_5$	$i_{10}$	$i_7$	$i_9$	$i_{12}$	$i_{13}$	$i_{11}$	$i_{14}$	$i_{15}$	
Lane:	$m_1$	$m_2$	$m_3$	$m_4$	$m_1$	$m_2$	$m_3$	$m_4$	$m_1$	$m_2$	$m_3$	$m_4$	$m_1$	$m_2$	$m_3$	

### 4.3 Greedy-based local search

This paper proposes a heuristic algorithm based on a greedy strategy to solve the lane allocation problem. The specific process is as follows:

- Step 1 Obtain the upper and lower TLGA's chromosome combinations and solve the lane allocation problem.
- Step 2 Arrange the next SC and its container according to the decision order, which is decided by the chromosome of the lower layer TLGA.
- Step 3 Put the lanes in which the operation time can be estimated into the list P1 (i.e., the empty time of the target lane is known, including the emptying times of positions under all the QCs). If there is no such lane, then a tremendous objective function value OB is obtained directly, going to step 9.
- Step 4 Consider the remaining lanes in the list P1 and calculate the estimated operation time after the SC selects the lane. The time when the SC enters the target position is the maximum time of the time when the SC arrives at the apron and the emptying time of the target position and the road to the position; the time when the SC leaves the lane is the maximum time of the completion time for the SC releases the container and the emptying time of the road the SC passes by when it leaves.
- Step 5 Through the dynamic adjustment algorithm, conflicts between the QC and the SC are effectively addressed. The adjusted operation times for the current QC and its associated SCs, as well as the emptying times of the relevant lanes, are recorded. Additionally, the value of the objective function is calculated and updated accordingly.
- Step 6 Move this lane from the list P1 and put it in the list P2. If there are other lanes in the list P1, return to step 4; otherwise, go to step 7.
- Step 7 Find one lane in the list P2 with the most minor added objective function value, and assign this lane to the SC and its container. Update the operation

time of the SCs, the QC, and the relevant lane's emptying time according to the dynamic adjustment algorithm data.

Step 8 If the container is not the last in the chromosome code, return to step 2: if all the SCs have been assigned, calculate the total objective function value OB and go to step 9.

Step 9 Return the value of the objective function, end.

**Algorithm 2** The dynamic adjustment algorithm (see online version for colours)

---

```

1 if SC operation interrupting the current QC then
2   OB1  $\leftarrow$  0;
3   for conflicting QC operation to all known QC operations afterward do
4     Temporarily delay this QC operation;
5     OB1  $\leftarrow$  OB1 + Increased objective value for QC delays;
6     if this QC operation conflicts with known operations of other QCs then
7       OB1  $\leftarrow$  M; // M is a great number.
8       Break // Jump out of the cycle.
9     else if this QC operation conflicts with other SC operations at the same QC
10    then
11      ob1  $\leftarrow$  ob2  $\leftarrow$  0;
12      if there is still a conflict after delaying the QC operation again then
13        | ob1  $\leftarrow$  M;
14      else if it conflicts with the known operations of other QCs after delaying
15      then
16        | ob1  $\leftarrow$  M;
17      else
18        | ob1  $\leftarrow$  Increased objective value for QC delays again ;
19      if it conflicts with known operations of other QCs after delaying other SCs
20      then
21        | ob2  $\leftarrow$  M;
22      else
23        | ob2  $\leftarrow$  Increased objective value for delaying other SCs
24      if ob1 < ob2 then
25        | OB1  $\leftarrow$  OB1+ob1;
26        | QC operation is delayed again;
27      else
28        | OB1  $\leftarrow$  OB1+ob2;
29        | Other SC operations are delayed;
30    if OB1  $\geq$  M then
31      | Break
32  OB2  $\leftarrow$  Increased objective value for delaying the current SC;
33  if OB1 < OB2 then
34    | Adjust the operation time of QCs and SCs according to the cycle scheme.
35  else
36    | Cancel the adjustment in the cycle, and then delay the SC operation

```

---

The dynamic adjustment algorithm dynamically adjusts the operation time of the current QC and its SCs to reduce the waiting time of the QC and SC as much as possible while avoiding affecting the scheduled work of other QCs. Algorithm 2 shows the specific pseudo code of the algorithm.

Since the QC continuously loads the container according to a given sequence of operations, delaying a specific QC operation will significantly impact its subsequent operations. Therefore, the algorithm adjusts all affected QC operations through a cycle from code 3 to code 34. Among them, we avoid conflicts with the scheduled operations of other QCs by adding a large objective value to those adjustment schemes. The time required for the SC to release the container is much less than the time when the QC spreader leaves the apron to ship to load the container and return to the apron. Therefore, codes 18 to 22 calculate the increased objective value for delaying SC without considering the impact on other operations of the current QC.

The emptying time of the lane includes the emptying time of all the positions under the QC of this lane; it is the time when the last container at the position leaves the apron. If another container preceding one container in the QC operation sequence has not been released by SC, the emptying time of the lane where the container is located is unknown. At the same time, the emptying time of the position in which the current SC passes at the same lane is when the SC passes; that is, the subsequent decision should prioritise ensuring that the current SC can reach the target position. The estimated operation time of the current SC is calculated according to the emptying time of the same lane, so the delay of this SC will not affect the scheduled operations of other QCs. So, we can delay the current SC without considering the impact on the operations of other QCs.

We obtain the scheduling plan and operation time of QCs and SCs using the heuristic algorithm under the current chromosome combination and get the objective function value.

#### *4.4 Genetic operation*

##### *4.4.1 Population initialisation*

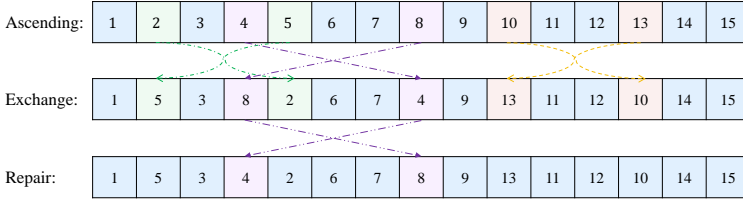
###### *4.4.1.1 First stage*

The upper chromosome represents the specific sequence of loading containers on the QC, and its population initialisation process is to generate a basic chromosome with ascending genes and, on this basis, randomly exchange the positions between genes to generate a completely random chromosome. This random method generates an initial population of chromosomes, enhancing population diversity to prevent premature convergence to local optima and improve global search capability. However, because chromosomes are randomly generated, they do not necessarily meet the requirements of ship stowage, so it is necessary to repair illegal chromosomes in the initial population. Figure 5 shows the initialisation of the upper chromosome. First, the first chromosome arranged in ascending order is generated, and then some genes are randomly exchanged to get the second chromosome. In the figure, gene 2 and gene 5, gene 4 and gene 8, gene 10 and gene 13 are exchanged respectively. And Figure 6 shows the stowage plan.

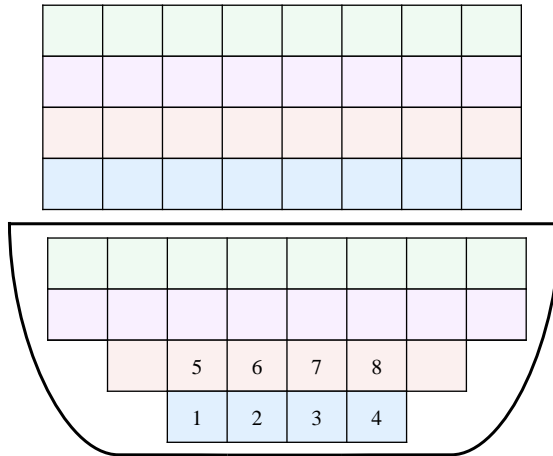
The repair algorithm is as follows: each gene in the chromosome is sequentially checked from the first gene of the chromosome. If the first container is directly below the second container in the stowage plan, and the gene represents the first container

is behind the gene, represents the second container, the positions of the two genes are exchanged. Taking Figures 5 and 6 as an example, when the repair algorithm detects the fourth gene of the chromosome, the gene represents the fourth container located after the gene represents the eighth container, and the positions of the two genes need to be exchanged. After the exchange, the repair algorithm will continue to check from the fourth gene.

**Figure 5** Initial generation and repair of the upper chromosome (see online version for colours)



**Figure 6** Stowage plan (see online version for colours)



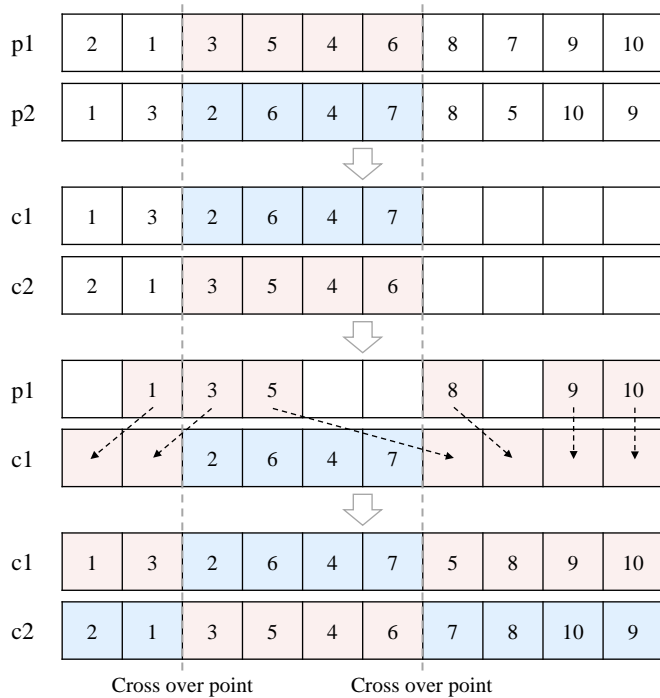
After the chromosomes of the upper algorithm are genetically manipulated or restarted, the repair algorithm will also be applied to obtain legal chromosomes.

4.4.1.2 Second stage

The lower population's chromosome represents the lane assignment scheme for SCs, generated based on the loading order of the upper chromosome. Thus, the design of the lower chromosome must align with the upper chromosome's loading order while ensuring that assigned lanes meet the SC's operational requirements and constraints. To achieve system coordination and effectiveness, the initial feasible chromosome is generated using the first-come, first-served (FCFS) principle. In this method, SCs are assigned available lanes sequentially according to their arrival order. The algorithm prioritises the nearest free lane upon each SC's arrival, ensuring optimal lane resource utilisation and minimising delays due to waiting for lane availability.

The FCFS principle in lane assignment quickly generates an initial feasible chromosome, ensuring it meets the problem’s basic constraints. While it does not guarantee a globally optimal solution, it provides a reasonable starting point for the genetic algorithm’s optimisation process. This reduces the complexity of subsequent optimisation and facilitates crossover and mutation operations. By generating a feasible lower population early on, this approach accelerates algorithm convergence.

**Figure 7** Schematic diagram of the crossover operator (see online version for colours)



#### 4.4.2 Chromosome evaluation and selection operator

When the two layers of chromosomes are determined, a heuristic algorithm can be run to obtain a plan of synchronisation operations of the QCs and SCs. Let the objective function value of a chromosome combination be OB. Since the model aims to minimise the waiting and completion times, equation (37) is used as the fitness function.

$$fitness = 1/OB \tag{37}$$

The ‘champion selection algorithm’ is used as the selection operator. This method is easy to implement, but at the same time, the method does not quickly converge to the local optimal. The selection method strategy is very intuitive. A certain number of individuals are randomly selected from the whole population, and individuals with high fitness values are chosen to enter the temporary population. If multiple individuals have the highest fitness, one is randomly selected to enter the temporary population. After multiple screenings, until the number of temporary populations reaches the

specified population, the individuals in the temporary population are subjected to genetic operations such as crossover and mutation to generate a new generation of genetic algorithms.

#### 4.4.3 Crossover operator

In practical applications, the crossover operation must employ appropriate strategies to ensure that the offspring individuals generated not only exhibit diversity but also adhere to the problem’s constraints and the legitimacy of the chromosomes. This paper uses the two-point crossover method as the crossover operator. Take Figure 7 as an example. The specific steps are as follows:

- Step 1 According to the crossover probability, select the chromosomes that need to be crossed, taking the two chromosome parents  $p1$  and  $p2$  as examples.
- Step 2 Randomly select the gene segments that need to cross, as the dark part shown in the figure, and copy the dark gene segments in  $p2$  to the same position in  $c1$ .
- Step 3 Knock out the gene in  $p1$  that is duplicated with the dark gene segment in  $p2$ .
- Step 4 Copy the remaining genes of  $p1$  to  $c1$  except for the dark areas.

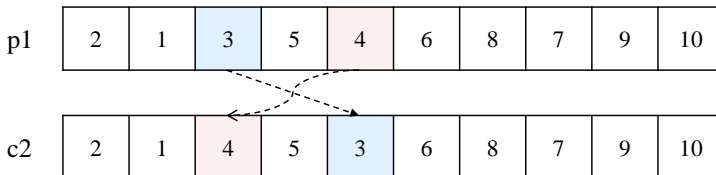
According to the same method, we can generate  $c2$ .

#### 4.4.4 Mutation operator

In this paper, exchange mutation is used as the mutation operator. Figure 8 shows the specific operation:

- Step 1 Screen the chromosomes that need to be mutated based on the set mutation probability.
- Step 2 Randomly select two genes in the parent chromosome, exchange the positions of the two genes, and generate the offspring.

**Figure 8** Schematic diagram of the mutation operator (see online version for colours)



## 5 Experiments

### 5.1 Experimental environment

#### 5.1.1 Computation environment

This paper uses Java programming language to program and realise the double-layer genetic algorithm in the previous chapter to verify its effectiveness in solving the problem of synchronous loading of QCs and SCs in container terminals. It compares the solution with that obtained by the optimisation software CPLEX (12.8.0). All experiments are run on a computer with an Inter(R) Core(TM) i7-12650H CPU @2.30 GHz and 16.0 GB of RAM. Because the model of the problem is complex, CPLEX software is prone to memory overflow after running large-scale problems or operations for more than a certain period, so this paper sets an upper limit of 2 hours of running time.

#### 5.1.2 Benchmark instance environment

##### 5.1.2.1 Single QC

We generate multiple sets of examples, ranging from 5 containers to 20 containers, to test the algorithm's performance at different scales. The time when the SC and its container arrive at the terminal's apron is randomly generated based on the ship's stowage plan. Among them, the operation scale of some examples is the same as other examples. Still, the SCs arrive in a concentrated manner, and the density of the SCs is higher in a certain period. The number of lanes under the QC is 4, and the operation time of releasing a container by SC and picking up a container by spreader are 30 and 30, respectively. The operation time of loading a container on the ship by QC after grabbing the container is a random number between [100, 140]. Based on the past industry experience, considering the influence of the setting of the weighting factors on the overall optimisation goal of the system, the weighting factors  $a$  and  $b$  are set to 0.6 and 0.4 respectively, which reflects a balance, that is, giving priority to shortening the completion time without excessively ignoring the influence of the waiting time. The specific size is shown in Table 1. And each instance is set to run 10 times in the experiment.

##### 5.1.2.2 Multiple QCs

Set multiple sets of different scale examples to test the algorithm. The specific size is shown in Table 2. The other parameters are as follows: the number of lanes under the QC is 4, and the operation time of releasing a container by SC and picking up a container by spreader are 30 and 30, respectively. The operation time of loading a container on the ship by QC after grabbing the container is a random number between [100 and 140]. The weighting factors  $a$  and  $b$  are 0.6 and 0.4, respectively. And each instance is set to run 10 times in the experiment.

**Table 1** Instance parameters of single QC

<i>Instance</i>	<i>QC number</i>	<i>Task number</i>	<i>Arrival density of SCs</i>
S-1-5-d	1	5	dispersive
S-1-5-c	1	5	concentrated
S-1-7-d	1	7	dispersive
S-1-7-c	1	7	concentrated
S-1-10-d	1	10	dispersive
S-1-10-c	1	10	concentrated
S-1-15-d	1	15	dispersive
S-1-15-c	1	15	concentrated
S-1-20-d	1	20	dispersive

**Table 2** Instance parameters of multiple QCs

<i>Instance</i>	<i>QC number</i>	<i>Task number</i>	<i>Arrival density of SCs</i>
M-2-5-5-d	2	5+5	dispersive
M-2-5-5-c		5+5	concentrated
M-2-7-7-d		7+7	dispersive
M-2-7-7-c		7+7	concentrated
M-2-10-8-d		10+8	dispersive
M-2-10-8-c		10+8	concentrated
M-3-5-5-5-d	3	5+5+5	dispersive
M-3-5-5-5-c		5+5+5	concentrated
M-3-7-7-7-d		7+7+7	dispersive
M-3-7-7-7-c		7+7+7	concentrated

## 5.2 Parameter tuning

The Taguchi method aims to select the parameter combination that makes the experimental results stable and fluctuate slightly and has been adopted by many previous studies (Zhou and Lee, 2020; Xin et al., 2024). The orthogonal table is used to choose experimental parameters and arrange experiments. To apply the Taguchi method, seven parameters of our interests are chosen to be tuned. They are upper population size (P1), upper maximum generation (P2), lower population size (P3), lower maximum generation (P4), crossover probability (P5), mutation probability (P6) and reinitialisation probability (P7). Four different levels are chosen as potential optimal levels for parameters. Those levels are summarised in terms of parameters in Table 14 of Appendix B.

The L32.4.7 orthogonal array of the Taguchi method is adopted to design the experimentation. The L32.4.7 orthogonal array with corresponding parameter settings is shown in Table 15 of Appendix B.

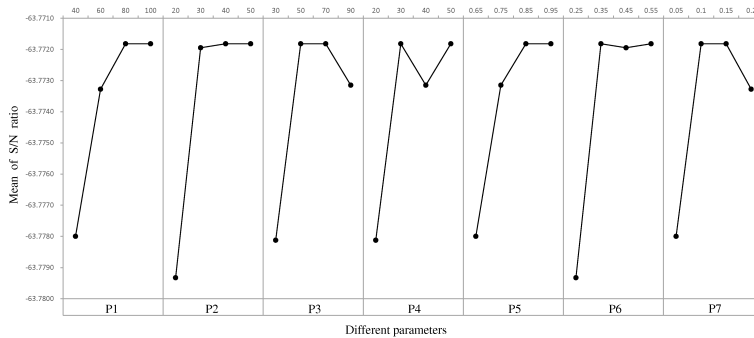
Ten experiments were carried out for each orthogonal number to ensure the experiment's effectiveness. To determine the optimal levels of the parameters, the signal-to-noise (S/N) ratio with the bigger-the-better rule is adopted for the Taguchi method. The S/N ratio is defined in the following:

$$S/N = -10 \log_{10} \left( \frac{\sum_{i=1}^n f_i^2}{n} \right) \tag{38}$$

where  $n$  denotes the number of experimental runs for each orthogonal number and  $f_i^2$  denotes the objective function value at  $i^{\text{th}}$  experimental run. And we take 15 containers of three quaysides as an example to experiment. The experimental data and S/N ratio are shown in Table 3.

Figure 9 shows the S/N ratio plot for different levels of the parameters. When the same parameter has multiple maximum S/N ratios, we choose the low-level parameter value as the optimal parameter value due to the experimental running time. Therefore, the optimal parameter values of the seven parameters are 80, 40, 50, 30, 0.85, 0.35 and 0.1, respectively.

**Figure 9** The S/N ratio plot for different levels of the parameters



### 5.3 Comparison between mathematical model and proposed algorithm

#### 5.3.1 Single QC synchronisation operation

For small-scale problems, both CPLEX and genetic algorithms can obtain optimal solutions quickly, as shown in S-1-5-d through S-1-10-d in Table 4, where the CPLEX solving time is shorter; still, the running time difference between them is within half a minute. As the scale and complexity of the instance continue to expand, the advantages of the TLGA begin to manifest. The TLGA is much more efficient than CPLEX. From S-1-10-c to S-1-15-c, the solution time of TLGA is between half a minute and one and a half minutes, while the solution obtained by CPLEX after two hours of operation is not better than that obtained by TLGA. It can be seen that the TLGA is far superior to the CPLEX solver in terms of solution quality and solution time in the single QC examples, especially when the scale of the problem is large.

For S-1-5-c, S-1-7-c, S-1-10-c, and S-1-15-c, the arrival time distribution density of the SC is higher, and the complexity of the problem is correspondingly increased. For small-scale instance, the increase in complexity has a lower impact on the computational time of CPLEX. As the scale of the instance rises, the complexity of the problem will significantly increase the computation time of CPLEX, such as in S-1-10-d and S-1-10-c. For the TLGA, the increasing of the complexity of the problem has little effect on the algorithm's efficiency. The TLGA can still obtain a satisfactory solution in a very short time, indicating that the algorithm has better performance.



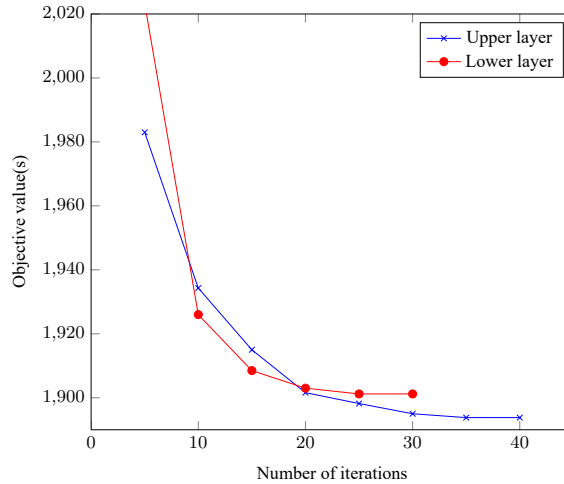
**Table 4** Comparison between Cplex and TLGA for single QC instances

Instance	CPLEX				TLGA				GAP		
	Objective value/s		time/s	GAP	Objective value/s		time/s	Avg	Min	Max	Avg
	Lower bound/s	Upper bound/s			Min	Max	Min	Max	Min	Max	
S-1-5-d	458.4	458.4	0.14	0.00%	458.4	458.4	13.77	14.21	13.94	0.00%	0.00%
S-1-5-c	527.4	527.4	0.15	0.00%	527.4	527.4	14.40	15.14	14.78	0.00%	0.00%
S-1-7-d	693.6	693.6	0.48	0.00%	693.6	693.6	20.76	22.44	21.14	0.00%	0.00%
S-1-7-c	718.2	718.2	0.87	0.00%	718.2	718.2	21.13	23.54	21.55	0.00%	0.00%
S-1-10-d	912.6	912.6	3.69	0.00%	912.6	912.6	31.93	35.90	33.43	0.00%	0.00%
S-1-10-c	1,017.2	1,015.2	7,200.00	0.20%	1,017.2	1,017.2	33.80	35.10	34.19	0.20%	0.20%
S-1-15-d	1,372.8	1,333.2	7,200.00	2.97%	1,372.8	1,372.8	53.82	56.41	54.83	2.97%	2.97%
S-1-15-c	1,938.6	909.6	7,200.00	113.13%	1,890.6	1,896.2	55.48	57.45	56.52	107.85%	108.02%
S-1-20-d	1,797.0	1,654.8	7,200.00	8.59%	1,797.0	1,797.0	77.65	79.83	78.79	8.59%	8.59%

**Table 5** Comparison between Cplex and TLGA for the multiple QCs instances

Instance	CPLEX				TLGA						
	Objective value/s		Time/s	GAP	Objective value/s		time/s	GAP			
	Lower bound/s	Upper bound/s			Min	Max	Min	Max	Avg		
M-2-5-5-d	961.8	961.8	0.77	0.00%	961.8	961.8	37.44	42.64	39.31	0.00%	0.00%
M-2-5-5-c	991.8	991.8	10.38	0.00%	991.8	991.8	37.89	40.62	39.07	0.00%	0.00%
M-2-7-7-d	1,380.0	1,380.0	3.23	0.00%	1,380.0	1,380.0	53.05	55.53	54.18	0.00%	0.00%
M-2-7-7-c	1,441.2	1,441.2	378.05	0.00%	1,441.2	1,441.2	56.05	59.74	57.48	0.00%	0.00%
M-2-10-8-d	1,690.2	1,690.2	4,274.22	0.00%	1,690.2	1,690.2	68.07	72.66	69.90	0.00%	0.00%
M-2-10-8-c	1,692.6	1,692.6	5,962.69	0.00%	1,692.6	1,692.6	69.47	73.65	71.47	0.00%	0.00%
M-3-5-5-5-d	1,423.8	1,423.8	4.97	0.00%	1,423.8	1,423.8	58.09	61.37	60.38	0.00%	0.00%
M-3-5-5-5-c	1,543.8	1,543.8	145.29	0.00%	1,543.8	1,543.8	58.78	63.55	61.34	0.00%	0.00%
M-3-7-7-7-d	2,037.6	2,037.6	3,889.65	0.00%	2,037.6	2,044.4	86.73	90.94	89.07	0.00%	0.12%
M-3-7-7-7-c	2,108.4	2,084.6	7,200.00	1.14%	2,115.2	2,121.2	88.22	90.57	89.06	1.47%	1.60%

**Figure 10** Upper layer and lower layer algorithm objective value convergence graph (see online version for colours)



For S-1-15-c, the complexity of the problem is high, and CPLEX software has not found the optimal solution within 2 hours. Due to the congestion of SCs, the waiting time is also high, which makes the objective function value much larger than the problem of the same scale. The solution obtained by the TLGA is better than the best set of solutions obtained by CPLEX. The convergence of the TLGA is shown in Figure 10, which is the convergence of the upper-layer and lower-layer TLGA of S-1-15-c, which indicates that the TLGA can converge well when the problem is complicated.

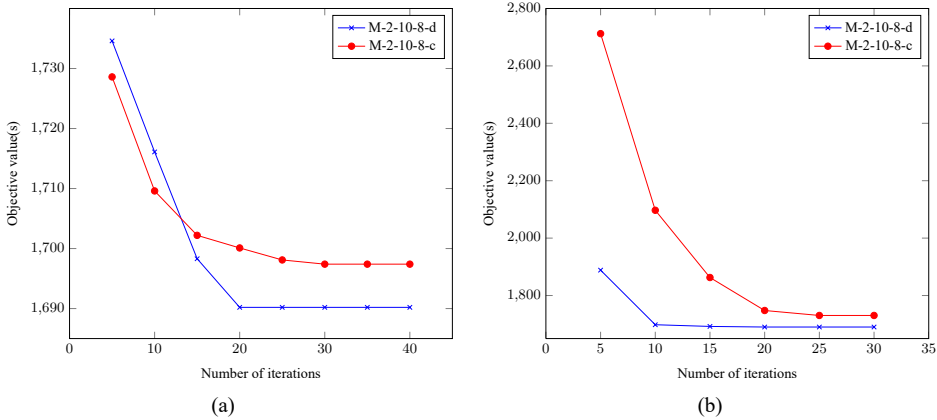
### 5.3.2 Multiple QCs synchronisation operation

Both CPLEX and TLGA can get the optimal solution quickly for small-scale synchronous loading operations under the interference of multi-shore cranes. With the increase of the scale of the problem, the time required for CPLEX operation increases rapidly. The TLGA can get the optimal solution in a reasonable time, such as M-2-10-8-d and M-2-10-8-c. In addition, it is worth noting that the running time of the CPLEX solver varies significantly in different instances. In contrast, the running time of the TLGA algorithm is positively correlated with the complexity of the example, and the growth rate is small, which has a certain robustness.

In addition, it is worth noting that the complexity of the same scale problem significantly affects the solution time of CPLEX, and the running time of the CPLEX solver varies greatly in different examples. In contrast, the population number and iteration times mainly affect the TLGA operation time. The running time of the TLGA algorithm is positively correlated with the complexity of the example and has a small growth rate, which has a certain robustness. Although the solution obtained by TLGA in M-3-7-7-7-c is slightly worse than that obtained by CPLEX, the running time difference between them is 80 times. The GAP value difference is very small. Considering the quality of the solutions and time, TLGA still has significant advantages over the CPLEX solver and can solve complex problems quickly.

The convergence of the objective value when the TLGA solves the multiple QCs problem is shown in Figure 11(a) and 11(b). M-2-10-8-d and M-2-10-8-c are problems with different complexity at the same scale. The information in the figure indicates that the complexity of the problem will slightly affect the convergence of the genetic algorithm. Therefore, for the more complex problem, the population size and the maximum number of iterations can be appropriately increased to obtain a better approximate optimal solution.

**Figure 11** Convergence analysis of the TLGA for M-2-10-8-d and M-2-10-8-c, (a) upper layer (b) lower layer (see online version for colours)



## 5.4 Algorithm analysis

### 5.4.1 Encoding and decoding

Different encoding and decoding methods affect offspring generation, significantly influencing solution quality and efficiency. The encoding mode defines the solution’s representation, while decoding affects the transformation from code to concrete solution. These methods impact population diversity, search space exploration, convergence speed, and global optimisation efficiency.

We compare and analyse two coding and decoding methods to discuss these differences and their influence on the algorithm’s performance. The first method is an encoding and decoding method that generates natural number permutation and realises mutation through random exchange. The second is the random key coding decoding method, which encodes the solution by floating point number and generates permutation using sorting rules.

Figure 12 shows random keys’ encoding and decoding process. Taking a chromosome with a length of 10 as an example, when encoding, ten random numbers from 0 to 1 are randomly generated. Each random number represents a gene, and the sequence numbers of these genes are in ascending order from 1 to 10. Then, the genes are arranged in ascending order according to the size of random numbers. Finally, the corresponding serial numbers are matched. The newly generated serial number sequence is the decoded chromosome.

**Table 6** Comparison of running results under different coding methods

Instance	Ordinary coding												TLGA											
	Objective value/s				time/s				GAP				Objective value/s				time/s				GAP			
	Min	Max	Avg		Min	Max	Avg		Min	Max	Avg		Min	Max	Avg		Min	Max	Avg		Min	Max	Avg	
S-1-5-d	458.4	458.4	458.4	13.77	14.21	13.94	0.00%	0.00%	0.00%	458.4	458.4	458.4	15.77	16.64	16.12	0.00%	0.00%	0.00%	0.00%					
S-1-5-c	527.4	527.4	527.4	14.40	15.14	14.78	0.00%	0.00%	0.00%	527.4	527.4	527.4	16.44	17.19	16.83	0.00%	0.00%	0.00%	0.00%					
S-1-7-d	693.6	693.6	693.6	20.76	22.44	21.14	0.00%	0.00%	0.00%	693.6	693.6	693.6	23.80	25.36	24.30	0.00%	0.00%	0.00%	0.00%					
S-1-7-c	718.2	718.2	718.2	21.13	23.54	21.55	0.00%	0.00%	0.00%	718.2	718.2	718.2	24.40	27.24	25.62	0.00%	0.00%	0.00%	0.00%					
S-1-10-d	912.6	912.6	912.6	31.93	35.90	33.43	0.00%	0.00%	0.00%	912.6	912.6	912.6	37.82	40.94	38.85	0.00%	0.00%	0.00%	0.00%					
S-1-10-c	1,017.2	1,017.2	1,017.2	33.80	35.10	34.19	0.20%	0.20%	0.20%	1,017.2	1,017.2	1,017.2	37.82	45.31	38.90	0.20%	0.20%	0.20%	0.20%					
S-1-15-d	1,372.8	1,372.8	1,372.8	53.82	56.41	54.83	2.97%	2.97%	2.97%	1,372.8	1,452.6	1,419.4	63.75	77.51	69.47	2.97%	8.96%	6.47%	6.47%					
S-1-15-c	1,890.6	1,896.2	1,892.2	55.48	57.45	56.52	107.85%	108.47%	108.02%	1,890.6	1,895.4	1,891.5	64.15	75.62	68.14	107.85%	108.38%	107.95%	107.95%					
S-1-20-d	1,797.0	1,797.0	1,797.0	77.65	79.83	78.79	8.59%	8.59%	8.59%	1,797.0	1,876.8	1,858.6	90.59	95.08	92.64	8.59%	13.42%	12.31%	12.31%					
M-2-5-5-d	961.8	961.8	961.8	37.44	42.64	39.31	0.00%	0.00%	0.00%	961.8	961.8	961.8	39.78	43.51	41.59	0.00%	0.00%	0.00%	0.00%					
M-2-5-5-c	991.8	991.8	991.8	37.89	40.62	39.07	0.00%	0.00%	0.00%	997.4	997.4	997.4	39.77	44.00	42.10	0.56%	0.56%	0.56%	0.56%					
M-2-7-7-d	1,380.0	1,380.0	1,380.0	53.05	55.53	54.18	0.00%	0.00%	0.00%	1,380.0	1,415.4	1,385.8	57.74	62.05	60.17	0.00%	2.57%	0.42%	0.42%					
M-2-7-7-c	1,441.2	1,441.2	1,441.2	56.05	59.74	57.48	0.00%	0.00%	0.00%	1,453.2	1,456.0	1,455.4	56.85	66.50	60.11	0.83%	1.03%	0.99%	0.99%					
M-2-10-8-d	1,690.2	1,690.2	1,690.2	68.07	72.66	69.90	0.00%	0.00%	0.00%	1,690.2	1,772.4	1,712.3	75.54	81.11	77.91	0.00%	4.86%	1.31%	1.31%					
M-2-10-8-c	1,692.6	1,692.6	1,692.6	69.47	73.65	71.47	0.00%	0.00%	0.00%	1,704.4	1,905.6	1,732.5	74.07	81.90	76.86	0.70%	12.58%	2.36%	2.36%					
M-3-5-5-5-d	1,423.8	1,423.8	1,423.8	58.09	61.37	60.38	0.00%	0.00%	0.00%	1,430.6	1,430.6	1,430.6	64.56	69.59	67.08	0.48%	0.48%	0.48%	0.48%					
M-3-5-5-5-c	1,543.8	1,543.8	1,543.8	58.78	63.55	61.34	0.00%	0.00%	0.00%	1,578.2	1,578.2	1,578.2	64.28	70.41	66.33	2.23%	2.23%	2.23%	2.23%					
M-3-7-7-7-d	2,037.6	2,044.4	2,040.1	86.73	90.94	89.07	0.00%	0.33%	0.12%	2,086.6	2,480.6	2,294.2	93.48	100.15	96.60	2.40%	21.74%	12.59%	12.59%					
M-3-7-7-7-c	2,115.2	2,121.2	2,117.9	88.22	90.57	89.06	1.47%	1.76%	1.60%	2,378.8	2,756.8	2,543.5	95.12	102.51	98.22	14.11%	32.25%	22.01%	22.01%					

**Table 7** Comparison of running results of different crossover operations

Instance	Two-point crossover						Single-point crossover											
	Objective value/s			Time/s			GAP			Objective value/s			Time/s			GAP		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
S-1-5-d	458.4	458.4	458.4	13.77	14.21	13.94	0.00%	0.00%	0.00%	458.4	458.4	458.4	13.71	14.08	13.93	0.00%	0.00%	0.00%
S-1-5-c	527.4	527.4	527.4	14.40	15.14	14.78	0.00%	0.00%	0.00%	527.4	527.4	527.4	14.32	14.77	14.61	0.00%	0.00%	0.00%
S-1-7-d	693.6	693.6	693.6	20.76	22.44	21.14	0.00%	0.00%	0.00%	693.6	693.6	693.6	20.70	21.61	21.03	0.00%	0.00%	0.00%
S-1-7-c	718.2	718.2	718.2	21.13	23.54	21.55	0.00%	0.00%	0.00%	718.2	718.2	718.2	21.24	22.73	21.87	0.00%	0.00%	0.00%
S-1-10-d	912.6	912.6	912.6	31.93	35.90	33.43	0.00%	0.00%	0.00%	912.6	912.6	912.6	32.23	36.33	33.56	0.00%	0.00%	0.00%
S-1-10-c	1,017.2	1,017.2	1,017.2	33.80	35.10	34.19	0.20%	0.20%	0.20%	1,017.2	1,017.2	1,017.2	32.83	35.44	34.08	0.20%	0.20%	0.20%
S-1-15-d	1,372.8	1,372.8	1,372.8	53.82	56.41	54.83	2.97%	2.97%	2.97%	1,372.8	1,372.8	1,372.8	52.47	56.98	54.11	2.97%	2.97%	2.97%
S-1-15-c	1,890.6	1,896.2	1,892.2	55.48	57.45	56.52	107.85%	108.47%	108.02%	1,890.6	1,898.6	1,893.3	53.65	57.26	55.30	107.85%	108.73%	108.14%
S-1-20-d	1,797.0	1,797.0	1,797.0	77.65	79.83	78.79	8.59%	8.59%	8.59%	1,797.0	1,825.8	1,802.7	77.00	83.60	79.06	8.59%	10.33%	8.94%
M-2-5-5-d	961.8	961.8	961.8	37.44	42.64	39.31	0.00%	0.00%	0.00%	961.8	961.8	961.8	35.58	43.27	39.36	0.00%	0.00%	0.00%
M-2-5-5-c	991.8	991.8	991.8	37.89	40.62	39.07	0.00%	0.00%	0.00%	991.8	991.8	991.8	35.77	42.92	38.63	0.00%	0.00%	0.00%
M-2-7-7-d	1,380.0	1,380.0	1,380.0	53.05	55.53	54.18	0.00%	0.00%	0.00%	1,380.0	1,380.0	1,380.0	51.19	57.11	53.26	0.00%	0.00%	0.00%
M-2-7-7-c	1,441.2	1,441.2	1,441.2	56.05	59.74	57.48	0.00%	0.00%	0.00%	1,441.2	1,441.2	1,441.2	55.44	61.83	57.85	0.00%	0.00%	0.00%
M-2-10-8-d	1,690.2	1,690.2	1,690.2	68.07	72.66	69.90	0.00%	0.00%	0.00%	1,690.2	1,690.2	1,690.2	68.38	77.00	71.60	0.00%	0.00%	0.00%
M-2-10-8-c	1,692.6	1,692.6	1,692.6	69.47	73.65	71.47	0.00%	0.00%	0.00%	1,692.6	1,692.6	1,692.6	68.26	75.54	70.75	0.00%	0.39%	0.19%
M-3-5-5-5-d	1,423.8	1,423.8	1,423.8	58.09	61.37	60.38	0.00%	0.00%	0.00%	1,423.8	1,423.8	1,423.8	58.18	61.32	59.89	0.00%	0.00%	0.00%
M-3-5-5-5-c	1,543.8	1,543.8	1,543.8	58.78	63.55	61.34	0.00%	0.00%	0.00%	1,543.8	1,543.8	1,543.8	59.14	62.19	60.08	0.00%	0.00%	0.00%
M-3-7-7-7-d	2,037.6	2,044.4	2,040.1	86.73	90.94	89.07	0.00%	0.33%	0.12%	2,037.6	2,044.4	2,042.4	84.90	88.12	86.65	0.00%	0.33%	0.23%
M-3-7-7-7-c	2,115.2	2,121.2	2,117.9	88.22	90.57	89.06	1.47%	1.76%	1.60%	2,115.6	2,123.2	2,119.1	85.72	90.50	88.17	1.49%	1.85%	1.65%

**Figure 12** Coding and decoding process of the random key method for the studied problem (see online version for colours)

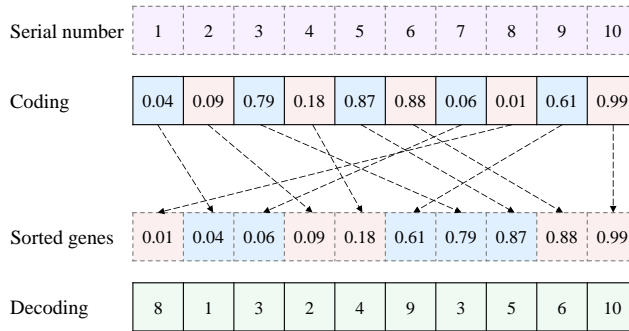


Table 6 shows the running results under different coding methods. It can be found that the quality and running time of the solution in random key coding mode are not better than those in ordinary coding mode for both single-shore and multi-shore crane instances. Compared with ordinary natural number permutation coding, the random key coding decoding method involves processing floating-point numbers and sorting floating-point numbers into the final solution. This process is relatively complicated, which may increase the overall solution time, especially in large-scale problems that require frequent coding and decoding. To sum up, although the random key coding decoding method has high flexibility and adaptability in some aspects, it may be too complicated, leading to a waste of computing resources, thus affecting the efficiency and effectiveness of the algorithm.

#### 5.4.2 Crossover

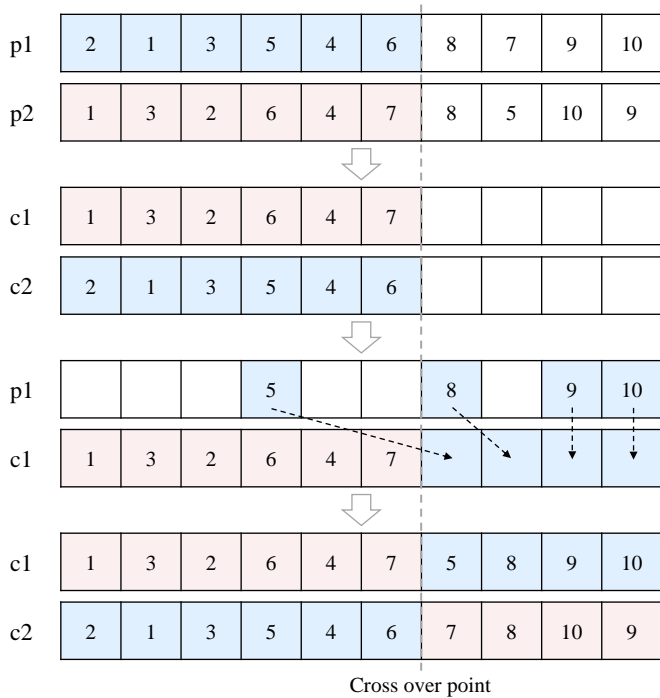
Figure 13 shows the process diagram of single-point crossing. And the experimental results in Table 7 compare different crossover operations and show the difference in solution time and quality between single-point crossover and two-point crossover. Specifically, although there is little difference in the computational time between the two crossover operations. There is almost no difference in the quality of the solution, but the two-point crossover is better than the single-point crossover.

The nature of crossover operation can explain this difference. Single-point crossover may not be able to break some unfavourable genetic structures in chromosomes because it only exchanges genes in one position, thus limiting the quality of understanding. In contrast, the more flexible recombination mode of two-point intersection enables the algorithm to explore more areas in the search space, thus improving the quality of the solution. Therefore, the experimental data in this paper show that although different crossover operations have little influence on computational efficiency, two-point crossover has advantages in solving quality, especially in problems with large and complex solution space. The two-point crossover is helpful to improve the overall performance of the algorithm.

**Table 8** Comparison of running results of different mutation operations

Instance	Exchange mutation						Insertion mutation											
	Objective value/s			Time/s			GAP			Objective value/s			Time/s			GAP		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
S-1-5-d	458.4	458.4	458.4	13.77	14.21	13.94	0.00%	0.00%	0.00%	458.4	458.4	458.4	13.54	13.87	13.70	0.00%	0.00%	0.00%
S-1-5-c	527.4	527.4	527.4	14.40	15.14	14.78	0.00%	0.00%	0.00%	527.4	527.4	527.4	14.11	14.57	14.32	0.00%	0.00%	0.00%
S-1-7-d	693.6	693.6	693.6	20.76	22.44	21.14	0.00%	0.00%	0.00%	693.6	693.6	693.6	20.33	21.77	20.91	0.00%	0.00%	0.00%
S-1-7-c	718.2	718.2	718.2	21.13	23.54	21.55	0.00%	0.00%	0.00%	718.2	718.2	718.2	20.29	22.24	21.13	0.00%	0.00%	0.00%
S-1-10-d	912.6	912.6	912.6	31.93	35.90	33.43	0.00%	0.00%	0.00%	912.6	957.0	917.0	31.48	34.45	32.75	0.00%	4.87%	0.49%
S-1-10-c	1,017.2	1,017.2	1,017.2	33.80	35.10	34.19	0.20%	0.20%	0.20%	1,017.2	1,026.0	1,019.0	31.43	32.97	32.42	0.20%	1.06%	0.38%
S-1-15-d	1,372.8	1,372.8	1,372.8	53.82	56.41	54.83	2.97%	2.97%	2.97%	1,372.8	1,418.4	1,390.9	51.86	54.14	52.60	2.97%	6.39%	4.33%
S-1-15-c	1,890.6	1,896.2	1,892.2	55.48	57.45	56.52	107.85%	108.47%	108.02%	1,895.4	1,928.2	1,912.4	51.45	53.90	52.88	108.38%	111.98%	110.24%
S-1-20-d	1,797.0	1,797.0	1,797.0	77.65	79.83	78.79	8.59%	8.59%	8.59%	1,842.6	1,843.8	1,843.4	73.73	89.24	76.62	11.35%	11.42%	11.40%
M-2-5-5-d	961.8	961.8	961.8	37.44	42.64	39.31	0.00%	0.00%	0.00%	961.8	961.8	961.8	35.75	40.80	38.01	0.00%	0.00%	0.00%
M-2-5-5-c	991.8	991.8	991.8	37.89	40.62	39.07	0.00%	0.00%	0.00%	991.8	991.8	991.8	35.92	39.68	37.32	0.00%	0.00%	0.00%
M-2-7-7-d	1,380.0	1,380.0	1,380.0	53.05	55.53	54.18	0.00%	0.00%	0.00%	1,380.0	1,380.0	1,380.0	51.46	55.43	53.00	0.00%	0.00%	0.00%
M-2-7-7-c	1,441.2	1,441.2	1,441.2	56.05	59.74	57.48	0.00%	0.00%	0.00%	1,441.2	1,448.2	1,442.6	53.12	58.38	55.56	0.00%	0.49%	0.10%
M-2-10-8-d	1,690.2	1,690.2	1,690.2	68.07	72.66	69.90	0.00%	0.00%	0.00%	1,690.2	1,734.6	1,703.5	65.62	72.03	68.98	0.00%	2.63%	0.79%
M-2-10-8-c	1,692.6	1,692.6	1,692.6	69.47	73.65	71.47	0.00%	0.00%	0.00%	1,692.6	1,714.4	1,707.6	67.40	73.28	70.35	0.00%	1.29%	0.89%
M-3-5-5-5-d	1,423.8	1,423.8	1,423.8	58.09	61.37	60.38	0.00%	0.00%	0.00%	1,423.8	1,423.8	1,423.8	57.78	60.90	59.37	0.00%	0.00%	0.00%
M-3-5-5-5-c	1,543.8	1,543.8	1,543.8	58.78	63.55	61.34	0.00%	0.00%	0.00%	1,543.8	1,563.0	1,558.2	56.45	63.39	58.65	0.00%	1.24%	0.93%
M-3-7-7-7-d	2,037.6	2,044.4	2,040.1	86.73	90.94	89.07	0.00%	0.33%	0.12%	2,044.4	2,055.4	2,050.2	85.67	89.51	87.44	0.33%	0.87%	0.62%
M-3-7-7-7-c	2,115.2	2,121.2	2,117.9	88.22	90.57	89.06	1.47%	1.76%	1.60%	2,123.2	2,168.8	2,146.2	84.72	88.84	86.40	1.85%	4.04%	2.96%

**Figure 13** Single-point intersection process diagram (see online version for colours)



### 5.4.3 Mutation

Figure 14 shows a schematic diagram of insertion mutation, defined as inserting the first gene after the last gene to generate a new individual. This mutation operation can significantly increase the diversity of individuals while retaining most of the genetic information.

**Figure 14** The process diagram of insertion mutation (see online version for colours)

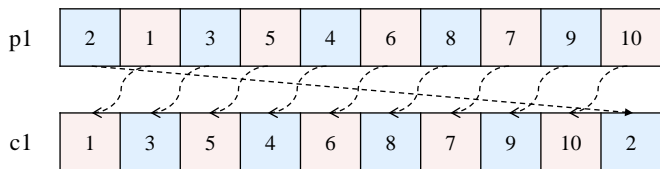
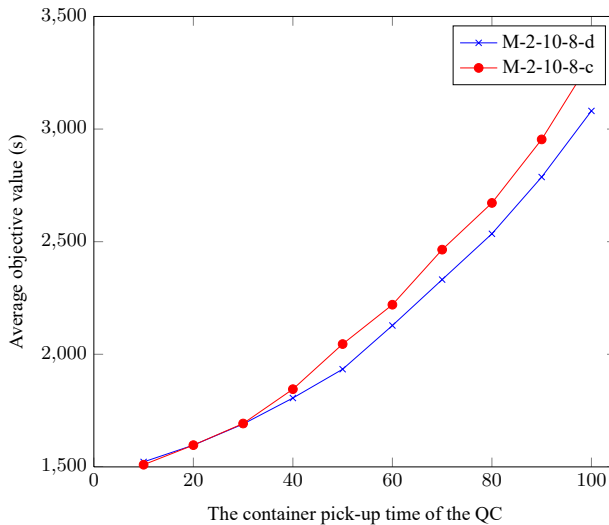


Table 8 compares the running results of different mutation operations. Due to the simple operation, insertion mutation has low complexity and shorter running time. However, the degree of variation is low; most of the parent’s information is retained, and it is easy to fall into the local optimum. The quality of the solution is worse than that of the exchange variation.

### 5.5 Sensitive analysis

Through the sensitivity analysis, the effects of the parameters on the synchronous operation of the QC and the SC are studied, such as the container pick-up time of the QC, the release operation time of the SC, and the number of lanes. We found that the pick-up operation time of the QC and the number of lanes have a more significant impact on the objective value, while the release operation time of the SC has less effect. This is because the SC's container dropping time is shorter than the overall operation time of the QC. At the same time, the collision between the SC and the QC is less after the scheduling of the algorithm, which makes the impact of the SC's container dropping on the QC's operation less. In addition, only the waiting time of the SC is considered in the objective function. The impact of the SC's container dropping time on the target value is much less. However, the time of container grabbing and the number of lanes directly affect the completion time of the QC and the waiting time of the SC, and they have a significant influence on the target value.

**Figure 15** Sensitive analysis for container pick-up times (see online version for colours)

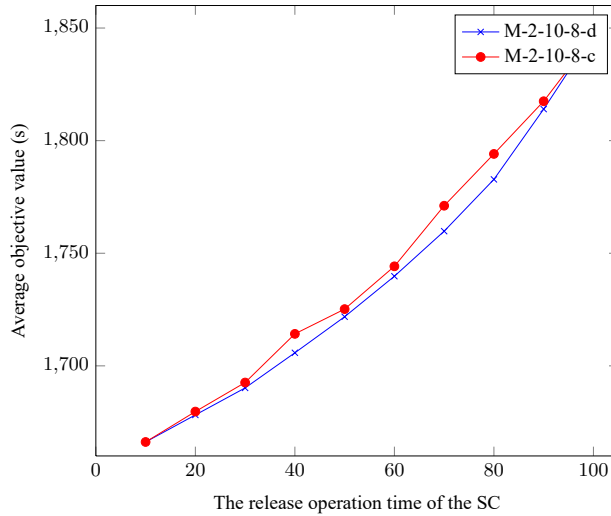


#### 5.5.1 The impact of the pick-up operation time of the QC

In this section, two medium-scale examples (18 container tasks, 2 QCs) are set up, and the distribution density of the SC arrival time of M-2-10-8-c is higher. The impact on the synchronous loading operation efficiency and algorithm performance was tested by changing the pick-up operation time. The experimental results are shown in Table 9. And Figure 15 shows the running results under different examples. As can be seen from the figure, with the increase of container grabbing time of the QC, the objective function values of the two examples show an increasing trend, and the increase rate is gradually increasing. It is worth noting that the increase of the target value of M-2-10-8-c is more significant than that of M-2-10-8-d, which indicates that the rise of the grabbing time of the QC has different effects on different operation scenarios. Specifically, the extension

of the container grabbing time of the QC not only has a significant negative impact on its working efficiency, resulting in a significant decrease in its working efficiency, but it also further aggravates the SC's waiting time. The scheduling of SCs depends on the grab operation of the QC, so the idle waiting time of SCs increases with the extension of the operation time of the QC, which affects the overall operation efficiency. It can be inferred that optimising the operation time of the QC is of great significance for improving the overall efficiency of the system and reducing the waiting time.

**Figure 16** Sensitive analysis for release operation time (see online version for colours)



### 5.5.2 The impact of the release operation time of the SC

A medium-sized task (18 container tasks and 2 QCs) is also set up, and its influence on the algorithm's performance and the efficiency of synchronous operation is tested by changing the time when the SC falls into the container. The experimental results are shown in Table 10, which clearly shows the influence of the change of SC's container dropping time on the target value and running time of two examples. It can be observed that the target value and running time of the two examples show an increasing trend with the increase in the time for the SC to drop the container. Figure 16 further shows the specific situation in which the target values of different examples change with the time the SC falls into the container. As can be seen from the figure, the increase of the target value of the two examples is similar, which indicates that the change in the container-dropping time of SCs has little influence on the efficiency of synchronous operation. That is to say, although the increase in the SC's container-dropping time has a certain impact on the target value and running time, compared with the significant effect of the grab time of the QC on the operation efficiency, the impact of the SC's container-dropping time on the overall synchronous operation efficiency is relatively small. It can be inferred that under medium-scale tasks, optimising the container-dropping time of SC will not significantly improve the overall efficiency of synchronous operation. On the contrary, more remarkable efficiency

improvement may come from optimising the operation time of the QC. This shows that, in practice, the scheduling of SCs should be controlled within a reasonable range, and the improvement of operation efficiency mainly depends on the optimal management of critical links, such as the operation time of QC.

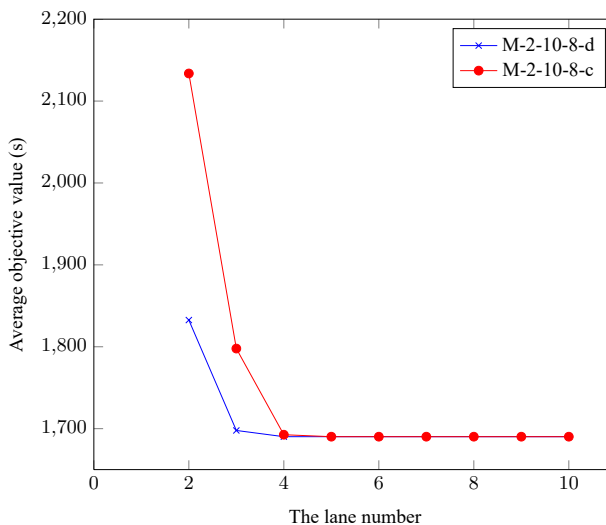
### 5.5.3 The impact of the lane number

The number of lanes, that is, the size of the buffer zone under the QC, is an index to measure the QC's ability to solve special situations (a large number of SCs arrive at the wharf front in a short time). Two sets of SC examples (18 containers and 2 QCs) with different time distribution densities are also designed for comparative experiments to study the influence of the number of lanes on the efficiency of wharf front work. This experiment aims to evaluate the correlation between the number of lanes and the efficiency of synchronous operation.

The experimental results are shown in Table 11, which shows that with the increase in the number of lanes, the efficiency of wharf operation is gradually improved, and the target values of the two groups of examples show a downward trend. Figure 17 shows the specific influence of the number of lanes on the target value. It can be observed that with the increase in the number of lanes, the synchronous operation efficiency is continuously improved, and the objective function value is gradually reduced, reflecting the optimisation effect of the algorithm. Specifically, M-2-10-8-d and M-2-10-8-c achieved their optimal target values when using four and five lanes, respectively, indicating that the number of lanes in these two examples can achieve the best resource scheduling and utilisation effect.

To sum up, rationally allocating the number of lanes in the buffer zone under the QC is crucial for optimising the wharf front's operation efficiency. Especially in a scene with a large time distribution density of SCs, increasing the number of lanes can significantly improve the efficiency of synchronous operation and reduce the target value, thus improving the overall operation performance.

**Figure 17** Sensitive analysis for line number (see online version for colours)



**Table 9** Experimental results under different container pick-up times of the QC

Container pick-up time by QC (s)	M-2-10-8-d						M-2-10-8-c					
	Objective value/s			Time/s			Objective value/s			Time/s		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
10	1,522.2	1,522.2	1,522.2	67.91	72.59	70.54	1,509.6	1,509.6	1,509.6	68.50	70.96	69.80
20	1,596.6	1,596.6	1,596.6	68.33	73.59	70.46	1,596.6	1,596.6	1,596.6	67.37	72.99	70.09
30	1,690.2	1,690.2	1,690.2	68.07	72.66	69.90	1,692.6	1,692.6	1,692.6	69.47	73.65	71.47
40	1,804.2	1,808.2	1,805.8	68.79	73.66	71.32	1,835.0	1,849.4	1,845.1	68.19	73.82	70.66
50	1,922.4	1,945.2	1,933.7	68.94	73.38	71.21	2,039.0	2,057.0	2,045.4	68.91	77.71	73.35
60	2,114.2	2,138.6	2,127.8	68.92	71.67	70.42	2,163.4	2,271.8	2,220.1	69.50	77.05	72.12
70	2,282.6	2,373.4	2,331.6	70.23	75.60	72.68	2,425.0	2,511.4	2,464.6	72.38	76.30	73.83
80	2,365.0	2,604.6	2,535.0	71.18	79.46	73.26	2,570.6	2,719.0	2,672.0	73.71	89.20	76.81
90	2,659.0	2,852.2	2,787.0	72.68	80.04	75.29	2,873.8	3,015.2	2,954.1	73.74	79.41	75.81
100	3,059.6	3,145.0	3,080.6	73.10	79.21	75.41	3,280.2	3,328.2	3,308.4	73.00	77.85	75.28

**Table 10** Experimental results under different release operation time.

Release operation time of the SC (s)	M-2-10-8-d						M-2-10-8-c					
	Objective value/s			Time/s			Objective value/s			Time/s		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
10	1,666.2	1,666.2	1,666.2	69.11	72.43	70.39	1,666.2	1,666.2	1,666.2	68.31	74.10	70.45
20	1,678.2	1,678.2	1,678.2	68.61	72.07	70.72	1,678.2	1,693.2	1,679.7	68.47	72.91	70.04
30	1,690.2	1,690.2	1,690.2	68.07	72.66	69.90	1,692.6	1,692.6	1,692.6	69.47	73.65	71.47
40	1,705.8	1,705.8	1,705.8	68.23	72.73	70.18	1,705.8	1,730.2	1,714.2	68.83	73.96	70.94
50	1,721.8	1,721.8	1,721.8	69.33	74.69	71.24	1,724.4	1,726.0	1,725.2	68.96	73.48	71.15
60	1,739.8	1,739.8	1,739.8	69.05	71.93	70.60	1,740.4	1,749.6	1,744.2	70.40	73.34	71.76
70	1,759.8	1,759.8	1,759.8	69.77	74.26	71.43	1,764.0	1,808.2	1,771.1	69.60	74.39	71.31
80	1,780.8	1,783.6	1,782.8	68.88	76.85	71.65	1,786.0	1,832.2	1,794.1	70.54	75.36	72.74
90	1,808.0	1,814.8	1,814.0	70.71	76.51	73.52	1,810.0	1,836.8	1,817.5	70.42	78.89	74.89
100	1,841.2	1,869.6	1,847.6	71.01	74.50	72.34	1,840.0	1,862.6	1,846.4	70.27	77.99	74.23

**Table 11** Experimental results under different lane numbers

Lane numbers	M-2-10-8-d						M-2-10-8-c					
	Objective value/s			Time/s			Objective value/s			Time/s		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
2	1,832.6	1,832.6	1,832.6	43.51	45.60	44.52	2,048.6	2,490.0	2,133.7	44.75	47.39	45.79
3	1,697.8	1,697.8	1,697.8	56.55	59.02	57.83	1,787.0	1,831.8	1,797.8	56.66	62.03	59.65
4	1,690.2	1,690.2	1,690.2	68.07	72.66	69.90	1,692.6	1,692.6	1,692.6	69.47	73.65	71.47
5	1,690.2	1,690.2	1,690.2	81.32	84.70	82.66	1,690.2	1,690.2	1,690.2	82.14	86.49	84.53
6	1,690.2	1,690.2	1,690.2	96.81	100.80	98.79	1,690.2	1,690.2	1,690.2	98.84	105.67	101.21
7	1,690.2	1,690.2	1,690.2	109.71	120.64	112.15	1,690.2	1,690.2	1,690.2	110.29	114.13	111.98
8	1,690.2	1,690.2	1,690.2	120.68	124.97	122.59	1,690.2	1,690.2	1,690.2	124.49	132.36	126.60
9	1,690.2	1,690.2	1,690.2	132.97	136.69	135.20	1,690.2	1,690.2	1,690.2	136.67	142.26	139.27
10	1,690.2	1,690.2	1,690.2	142.65	148.13	144.34	1,690.2	1,690.2	1,690.2	146.83	151.01	148.67

## 6 Conclusions

The QC, which is the most critical equipment at the container terminal, affects the efficiency of the loading and unloading operation of container vessels. This paper considers studying the QC and SC in cooperative operation. The contributions are summarised as follows.

- 1 An optimisation model for the simultaneous loading of QCs and SCs under flexible loading and unloading sequences is established to minimise QC completion time and SC waiting time. The model takes into account several practical constraints in the operation of QCs and SCs, including QC trolleys cannot cross SCs with boxes, QCs and SCs cannot work in adjacent lanes at the same time, and the constraints of ship stowage. The effect of containers stored on the lower lanes of other QCs on the SC when multiple QCs are loading and unloading at adjacent locations is also considered. An optimal solution to the small-scale problem can be obtained by solving the model using mixed integer programming solvers.
- 2 This paper designed a dynamic adjustment heuristic algorithm combined with a greedy strategy, which can efficiently solve the simultaneous operation scheme of the QC and SC under the given container loading sequence and priority. To optimise the container loading order and priority, a two-layer genetic algorithm is established to solve the subproblem, in which the upper layer of the genetic algorithm solves the operation sequence of the QC, and the lower layer of the genetic algorithm solves the decision-making sequence of the SCs (the priority of the SCs in choosing the lanes). Then, the heuristic and genetic algorithms are integrated to construct a two-layer hybrid genetic algorithm to solve the problem studied in this paper.
- 3 The combination of genetic algorithm parameters that makes the experimental results stable and less volatile is selected through the Taguchi test. The comparison experiments of CPLEX and the two-layer genetic algorithm are carried out. The experimental results show that the two-layer genetic algorithm has a better-solving performance on small and medium-sized problems. This paper also analyses the effects of different coding methods, crossover operations, and mutation operations on the quality of the solution. The results show that the natural number coding method outperforms the random key coding method. The two-point crossover operation yields higher-quality solutions than the single-point crossover operation. Compared with the insertion mutation operation, the exchange mutation operation can increase the population's diversity and the solution's quality.
- 4 Sensitivity analyses show that the terminal crane container-grabbing time. The number of lanes significantly impacts the target value, and the trans-carrier drop-off time has a minor impact on the target value. Among them, the terminal crane container-grabbing time not only affects the completion time of the terminal crane but also increases the waiting time of the trans-carrier because it reduces the operational efficiency of the terminal crane and makes the container task pile up. Therefore, priority needs to be given to ensuring the operational efficiency of QCs under certain circumstances. The transporter's drop time is shorter than the QC's overall operating time. At the same time, after the scheduling algorithm, there are

fewer conflicts between the transporter and the QC, which makes the drop of the transporter have less impact on the operation of the QC. The number of lanes affects the buffer capacity of the QC in the face of the peak period of trans-shipment trucks, and an increase in the number of lanes reduces the waiting time of trans-shipment trucks and improves the efficiency of trans-shipment truck operations. Therefore, in the actual operation, the focus should be on improving the operational efficiency of the QC to grab the container.

In this paper, the time of the trans-carrier's arrival at the quay front is considered known, and the trans-carrier's scheduling problem can be considered in the future studies. In addition, the ship unloading operation can be also taken into account in the problem. It can also be considered in an integrated way with the scheduling problem of the trans-carrier for the optimisation of the simultaneous operation of loading and unloading of the container vessel.

### Acknowledgements

His work was supported by the National Natural Science Foundation of China (No. 72171211).

### References

- Cai, B., Huang, S., Liu, D. and Dissanayake, G. (2014) 'Rescheduling policies for large-scale task allocation of autonomous straddle carriers under uncertainty at automated container terminals', *Robotics and Autonomous Systems*, Vol. 62, No. 4, pp.506–514.
- Cao, Y., Yang, A., Liu, Y., Zeng, Q. and Chen, Q. (2023) 'AGV dispatching and bidirectional conflict-free routing problem in automated container terminal', *Computers & Industrial Engineering*, Vol. 184, p.109611.
- Chargui, K., Zouadi, T., Sreedharan, V.R., El Fallahi, A. and Reghioui, M. (2023) 'A novel robust exact decomposition algorithm for berth and quay crane allocation and scheduling problem considering uncertainty and energy efficiency', *Omega (Oxford)*, Vol. 118, p.102868.
- El-Abbasy, M.S., Ahmed, E., Zayed, T., Alfalah, G. and Alkass, S. (2021) 'Multi-level fleet size optimization for containers handling using double-cycling strategy', *Expert Systems with Applications*, Vol. 170, p.114526.
- Feng, X., He, Y. and Kim, K-H. (2022) 'Space planning considering congestion in container terminal yards', *Transportation Research. Part B: Methodological*, Vol. 158, pp.52–77.
- Feng, X., Liu, M., Zhang, W., Yin, W. and Chao, Y. (2025) 'The impacts of pilotage planning on green maritime logistics', *Regional Studies in Marine Science*, Vol. 81, p.103989.
- Fereidoonian, F., Sadjadi, S.J., Heydari, M. and Al-e-Hashem, S.M.J.M. (2024) 'A timely efficient and emissions-aware multiobjective truck-sharing integrated scheduling model in container terminals', *Proceedings of the Institution of Mechanical Engineers. Part M, Journal of Engineering for the Maritime Environment*, Vol. 238, No. 4, pp.982–1008.
- Fu, Y-M. and Diabat, A. (2015) 'A Lagrangian relaxation approach for solving the integrated quay crane assignment and scheduling problem', *Applied Mathematical Modelling*, Vol. 39, Nos. 3–4, pp.1194–1201.

- Gao, X., Zhou, Y., Amir, M.I.H., Rosyidah, F. and Lee, G. (2017) 'A hybrid genetic algorithm for multi-emergency medical service center location-allocation problem in disaster response', *International Journal of Industrial Engineering: Theory, Applications and Practice*, Vol. 24, No. 6, pp.663–679.
- Gong, L., Huang, Z., Xiang, X. and Liu, X. (2024) 'Real-time AGV scheduling optimisation method with deep reinforcement learning for energy-efficiency in the container terminal yard', *International Journal of Production Research*, Vol. 62, No. 21, pp.7722–7742.
- Ji, B., Huang, H. and Yu, S.S. (2023) 'An enhanced NSGA-II for solving berth allocation and quay crane assignment problem with stochastic arrival times', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 1, pp.459–473.
- Kavakeb, S., Nguyen, T.T., McGinley, K., Yang, Z., Jenkinson, I. and Murray, R. (2015) 'Green vehicle technology to enhance the performance of a European port: a simulation model with a cost-benefit approach', *Transportation Research. Part C, Emerging Technologies*, Vol. 60, pp.169–188.
- Kenan, N., Jebali, A. and Diabat, A. (2022) 'The integrated quay crane assignment and scheduling problems with carbon emissions considerations', *Computers & Industrial Engineering*, Vol. 165, p.107734.
- Klerides, E. and Hadjiconstantinou, E. (2011) 'Modelling and solution approaches to the multi-load AGV dispatching problem in container terminals', *Maritime Economics & Logistics*, Vol. 13, No. 4, pp.371–386.
- Li, Y., Li, X., Zhang, C. and Wu, T. (2024a) 'Decomposition algorithms for the robust unidirectional quay crane scheduling problems', *Computers & Operations Research*, Vol. 167, p.106670.
- Li, Y., Sun, Z. and Hong, S. (2024b) 'An exact algorithm for multiple-equipment integrated scheduling in an automated container terminal using a double-cycling strategy', *Transportation Research. Part E, Logistics and Transportation Review*, Vol. 186, p.103565.
- Liang, C., Fan, L., Xu, D., Ding, Y. and Gen, M. (2018) 'Research on coupling scheduling of quay crane dispatch and configuration in the container terminal', *Computers & Industrial Engineering*, Vol. 125, pp.649–657.
- Nossack, J. and Pesch, E. (2013) 'A truck scheduling problem arising in intermodal container transportation', *European Journal of Operational Research*, Vol. 230, No. 3, pp.666–680.
- Shang, X.T., Cao, J.X. and Ren, J. (2016) 'A robust optimization approach to the integrated berth allocation and quay crane assignment problem', *Transportation Research. Part E, Logistics and Transportation Review*, Vol. 94, pp.44–65.
- Shi, Y., Liu, W. and Zhou, Y. (2023) 'An adaptive large neighborhood search based approach for the vehicle routing problem with zone-based pricing', *Engineering Applications of Artificial Intelligence*, Vol. 124, p.106506.
- Shi, Y., Zhou, Y., Boudouh, T. and Grunder, O. (2022) 'A memetic algorithm for a relocation-routing problem in green production of gas considering uncertainties', *Swarm and Evolutionary Computation*, Vol. 74, p.101129.
- Sun, D., Tang, L. and Baldacci, R. (2019) 'A benders decomposition-based framework for solving quay crane scheduling problems', *European Journal of Operational Research*, Vol. 273, No. 2, pp.504–515.
- Wang, C., Liu, K., Zhang, C. and Miao, L. (2024a) 'Distributionally robust chance-constrained optimization for the integrated berth allocation and quay crane assignment problem', *Transportation Research. Part B: Methodological*, Vol. 182, p.102923.
- Wang, Z., Hu, H. and Zhen, L. (2024b) 'Berth and quay cranes allocation problem with on-shore power supply assignment in container terminals', *Computers & Industrial Engineering*, Vol. 188, p.109910.

- Wang, H., Villa, F., Vallada, E. and Ruiz, R. (2025) 'Solving the yard crane scheduling problem with dynamic assignment of input/output points', *Computers & Operations Research*, Vol. 173, p.106853.
- Wang, Y., Jiang, X., Lee, L.H., Chew, E.P. and Tan, K.C. (2017) 'Tree based searching approaches for integrated vehicle dispatching and container allocation in a transshipment hub', *Expert Systems with Applications*, Vol. 74, pp.139–150.
- Xin, J., Yuan, Q., D'Ariano, A., Guo, G., Liu, Y. and Zhou, Y. (2024) 'Dynamic unbalanced task allocation of warehouse AGVs using integrated adaptive large neighborhood search and Kuhn-Munkres algorithm', *Computers & Industrial Engineering*, Vol. 195, p.110410.
- Xu, B., Fan, H. and Li, J. (2024) 'Energy-efficiency and safety-driven multidimensional evolutionary game for AGVs transportation at automated container terminals', *Computers & Industrial Engineering*, Vol. 192, p.110192.
- Yue, L.-J., Fan, H.-M. and Fan, H. (2023) 'Blocks allocation and handling equipment scheduling in automatic container terminals', *Transportation Research. Part C, Emerging Technologies*, Vol. 153, p.104228.
- Zhang, A., Zhang, W., Chen, Y., Chen, G. and Chen, X. (2017) 'Approximate the scheduling of quay cranes with non-crossing constraints', *European Journal of Operational Research*, Vol. 258, No. 3, pp.820–828.
- Zheng, H., Wang, Z. and Fan, X. (2024) 'Integrated rescheduling optimization of berth allocation and quay crane allocation with shifting strategies', *Ocean Engineering*, Vol. 301, p.117473.
- Zhou, C., Yuan, M., Zhang, J. and Zhang, W. (2024a) 'A tree search algorithm for uncertainty-considered consecutive discharging and loading operations between ship and offshore platform', *European Journal of Operational Research*, Vol. 315, No. 2, pp.729–749.
- Zhou, Y., Luo, L., Zhao, Q.Q., Haonan, C., Qian, Z. and Leng, S. (2024b) 'Influence index analysis of inland waterway ports along the Yangtze River', *European Journal of Industrial Engineering*, No. 2.
- Zhou, Y. and Kim, K.H. (2020) 'A game theoretic model and a coevolutionary solution procedure to determine the terminal handling charges for container terminals', *Computers & Industrial Engineering*, Vol. 144, p.106466.
- Zhou, Y. and Lee, G.M. (2020) 'A bi-objective medical relief shelter location problem considering coverage ratios', *International Journal of Industrial Engineering: Theory, Applications and Practice*, Vol. 27, No. 4, pp.971–988.
- Zhu, S., Tan, Z., Yang, Z. and Cai, L. (2022) 'Quay crane and yard truck dual-cycle scheduling with mixed storage strategy', *Advanced Engineering Informatics*, Vol. 54, p.101722.

**Appendix A**

*Notations and variables*

**Table 12** Notations

<i>Set</i>	
$K$	Set of all QCs assigned for a vessel
$N_k$	Set of containers planned to be loaded by the $k^{\text{th}}$ QC
$C$	A virtual container task
$M_k$	The union of $N_k$ and $C$
<i>Parameters</i>	
$a$	Weighting factor of the objective function
$b$	Weighting factor of the objective function
$M$	A sufficiently large positive number
$m$	Number of lanes under the QC.
$n_k$	number of containers needed to be loaded by the $k^{\text{th}}$ QC.
$t_q$	The time required for the spreader of the QC to pick up a container (s).
$t_s$	The time required for the SC to release a container (s).
$e_{ik}$	The operation time of the $k^{\text{th}}$ QC for the $i^{\text{th}}$ container from the lane to the ship, which is the time required for the spreader to move from the apron to the assigned slot, put the container into the slot and return from the slot to the apron (s).
$A_{ik}$	The arrival time of the SC with the $i^{\text{th}}$ container loaded by the $k^{\text{th}}$ QC.
$G_{ijk}$	1, if the $j^{\text{th}}$ container is loaded on the $i^{\text{th}}$ container by the $k^{\text{th}}$ QC; 0, otherwise.

**Table 13** Decision variables

<i>Decision variables</i>	
$X_{ik}$	Lane number for the $i^{\text{th}}$ container loaded by the $k^{\text{th}}$ QC.
$S_{ik}^s$	Release starting time for the $i^{\text{th}}$ container by an SC loaded by the $k^{\text{th}}$ QC.
<i>Derived decision variables</i>	
$D_{ijk}$	1, if the $k^{\text{th}}$ QC handles $j^{\text{th}}$ container after $i^{\text{th}}$ container immediately; 0, otherwise.
$R_{ik}$	A variable to make sure there are no sub-loop.
$F_{ik}^s$	release finishing time for the $i^{\text{th}}$ container by an SC.
$S_{ik}^q$	pickup starting time for the $i^{\text{th}}$ container by QC.
$F_{ik}^q$	pickup finishing time for the $i^{\text{th}}$ container by QC.
$L_{ijkl}$	1, if the $i^{\text{th}}$ container loaded by the $k^{\text{th}}$ QC is positioned in a lane closer to the waterside than the $j^{\text{th}}$ container loaded by the $l^{\text{th}}$ QC; 0, otherwise. The lane is numbered in the increasing sequence from the waterside to the landside.
$Z_{ijk}^{qs}$	1, if the QC pickup operation for the $i^{\text{th}}$ container is finished before the SC release operation for the $j^{\text{th}}$ container starts; 0, otherwise.
$Z_{ijk}^{sq}$	1, if the SC release operation for the $i^{\text{th}}$ container is finished before the QC pickup operation for the $j^{\text{th}}$ container starts; 0, otherwise.
$Z_{ijkl}^{as}$	1, if the SC release operation for the $i^{\text{th}}$ container loaded by the $k^{\text{th}}$ QC is started before the SC release operation for the $j^{\text{th}}$ container loaded by the $l^{\text{th}}$ QC starts; 0, otherwise.

**Table 13** Decision variables (continued)

<i>Derived decision variables</i>	
$Z_{ijkl}^{qa}$	1, if the $k^{\text{th}}$ QC pickup operation for the $i^{\text{th}}$ container is finished before the SC release operation for the $j^{\text{th}}$ container loaded by the $l^{\text{th}}$ QC starts; 0, otherwise.
$Z_{ijkl}^{ds}$	1, if the SC release operation for the $i^{\text{th}}$ container loaded by the $k^{\text{th}}$ QC is finished before the SC release operation for the $j^{\text{th}}$ container loaded by the $l^{\text{th}}$ QC starts; 0, otherwise.
$Z_{ijkl}^{qd}$	1, if the $k^{\text{th}}$ QC pickup operation for the $i^{\text{th}}$ container is finished before the SC release operation for the $j^{\text{th}}$ container loaded by the $l^{\text{th}}$ QC finish; 0, otherwise.
$U_{ijk}$	1, if the $i^{\text{th}}$ container is handled at the lane which is larger than the lane of the $j^{\text{th}}$ container by at least two lanes; 0, otherwise.
$W_{ik}$	the waiting time of one SC when operating the $i^{\text{th}}$ container.

## Appendix B

### Parameter levels

**Table 14** Parameter levels

<i>Level</i>	<i>Parameters</i>						
	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>	<i>P7</i>
1	40	20	30	20	0.65	0.25	0.05
2	60	30	50	30	0.75	0.35	0.10
3	80	40	70	40	0.85	0.45	0.15

**Table 15** The L32.4.7 orthogonal array of the Taguchi method

<i>Orthogonal number</i>	<i>Parameters</i>						
	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>	<i>P7</i>
1	40	20	30	20	0.65	0.25	0.05
2	40	20	50	30	0.95	0.55	0.15
3	40	30	70	50	0.65	0.35	0.15
4	40	30	90	40	0.95	0.45	0.05
5	40	40	30	40	0.75	0.55	0.10
6	40	40	50	50	0.85	0.25	0.20
7	40	50	70	30	0.75	0.45	0.20
8	40	50	90	20	0.85	0.35	0.10
9	60	20	70	50	0.85	0.55	0.10
10	60	20	90	40	0.75	0.25	0.20
11	60	30	30	20	0.85	0.45	0.20
12	60	30	50	30	0.75	0.35	0.10
13	60	40	70	30	0.95	0.25	0.05
14	60	40	90	20	0.65	0.55	0.15
15	60	50	30	40	0.95	0.35	0.15

**Table 15** The L32.4.7 orthogonal array of the Taguchi method (continued)

<i>Orthogonal number</i>	<i>Parameters</i>						
	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>	<i>P7</i>
16	60	50	50	50	0.65	0.45	0.05
17	80	20	70	20	0.95	0.35	0.20
18	80	20	90	30	0.65	0.45	0.10
19	80	30	30	50	0.95	0.25	0.10
20	80	30	50	40	0.65	0.55	0.20
21	80	40	70	40	0.85	0.45	0.15
22	80	40	90	50	0.75	0.35	0.05
23	80	50	30	30	0.85	0.55	0.05
24	80	50	50	20	0.75	0.25	0.15
25	100	20	30	50	0.75	0.45	0.15
26	100	20	50	40	0.85	0.35	0.05
27	100	30	70	20	0.75	0.55	0.05
28	100	30	90	30	0.85	0.25	0.15
29	100	40	30	30	0.65	0.35	0.10
30	100	40	50	20	0.95	0.45	0.10
31	100	50	70	40	0.65	0.25	0.10
32	100	50	90	50	0.95	0.55	0.20