# An Integrated Programming Model For Straddle Carrier Scheduling And Container Storage Problems In Dual-Cycle Operations At Container Terminals

Jiabin Luo[1] and Yue Wu[2]

1. Aston Business School, Aston University, Birmingham, B4 7ET.
   Email: j.luo2@aston.ac.uk
2. Southampton Business School, University of Southampton, SO17 1BJ.
   Email: y.wu@soton.ac.uk

**Abstract:**
The efficiency of transporting and storing a large number of containers to and from the quayside is critical to any container terminal. We investigated the integration of straddle carrier scheduling and container storage problems in the dual-cycle operations. We formulated this integrated problem into a mixed-integer programming model and the objective is to minimise the ship's berth time. Optimal solutions can be obtained for small-sized problems. However, due to computational difficulty, a genetic algorithm (GA) is developed to provide approximately optimal solutions for large-sized problems. The experimental results show the effectiveness of the proposed modelling approach and GA.

**Keywords:** vehicle scheduling, container storage, container terminal operations, genetic algorithm.

Word count: 5478

# 1 Introduction

Container terminals serve as an interface between marine and land transportation systems. Since the introduction of containerisation in 1960s, the number of containers handled worldwide has dramatically grown every year. With the increasing containerisation, nowadays container terminals are working at maximum capacity. Therefore, it is critical to improve the operational efficiency of container terminals in order to deal with the ever-increasing number of container flows.

In a typical container terminal, quay cranes (QCs) are equipped at the seaside for unloading/loading containers from/onto ships; yard cranes (YCs) are used for stacking and storing containers in the yard; and vehicles are travelling in between for transporting containers. In this work, we consider another type of container terminal, named straddle carrier (SC) system, in which only QCs and SCs are used in handling containers. SC is one type of vehicle and it has the ability to lift containers without additional aid from other equipment. Therefore, SCs are able to substitute YCs for stacking containers in the yard. The SC system is mainly adopted in Europe, such as DP world Southampton, Le Havre Port France and Wilhelmshaven Germany (Luo, 2013). Figure 1 shows a straddle carrier working at the terminal. The layout of a SC system is shown in figure 2.

**Figure 1: straddle carrier is used for transporting and stacking containers**

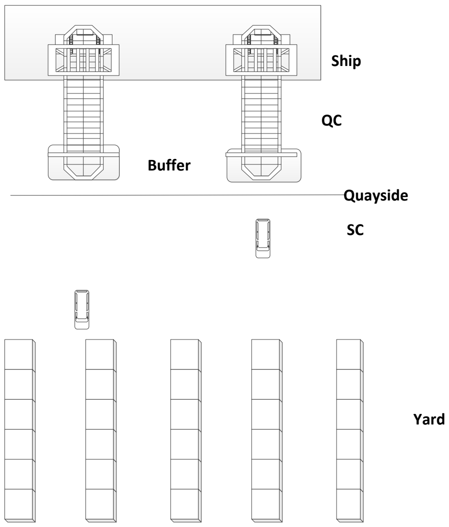


**Figure 2: the layout of a straddle carrier system**

There are two processes in container terminal's operation, which are unloading process and loading process. In a SC system, during the unloading process, a QC picks up a container from the ship, and puts it in the buffer area under this QC. The buffer area functions as a temporary storage site for SCs to come and collect containers, because QCs cannot put containers directly onto SCs. Then a SC picks up a container and delivers it to the yard; there are wide aisles between each yard bay giving enough space for the SCs to pass through; after storing this container in the yard, the SC returns to the ship to process the next container. The loading process is in the reverse direction of the unloading process. Our work considers the unloading and loading processes simultaneously, which is called the dual-cycle operation.

Effectively scheduling of vehicles and allocating storage locations to containers are important to improve the container terminal's performance (Steenken et al., 2004). Therefore, these two main operational issues have been investigated in this study: vehicle scheduling and container storage problems. Specifically, vehicle scheduling is one of the major planning problems in container terminal operations. It is usually formulated as transportation-type assignment problems, which determine how to dispatch vehicles to tasks (Das and Spasovic, 2003). Inefficient vehicle schedules will cause delays in container-handling processes and thus affect the productivity of container terminals; On the other hand, container storage problem is another important issue due to the very limited space for accommodating the increasing volume of containers, which makes the yard storage space becoming a critical resource. Many previous studies only discussed these two problems separately (for example, Kim and Bae (2004), Ng and Mak (2005), Ng et al. (2007) and Nishimura et al. (2005)), however, practically SC scheduling and container storage problems are interdependant on each other. The schedule of a SC decides the storage locations of all the containers it carries, and in turn, container storage locations determine the travelling times of this SC from/to the quayside. If these two problems are considered seperately, a container can be assigned far away from the quayside, thus prolong the travelling times of SCs, which delays the overall performance of the unloading and loading operations.

Our main contribution is that we have provided the integrated modelling approach for such practical problem, considering the SC scheduling and container storage problems simultaneously in the dual-cycle operations. The model determines the time/sequence to deliver each container by SCs and the locations to store each container. We aim to minimise the ship's berth time, i.e. the time all the containers have been unloaded/loaded from/onto the ship, which is one of the most important measures for a container terminal's efficiency and it usually represents the amount of time that a ship spends at the terminal. Another contribution is that we have also developed an efficient genetic algorithm (GA) to provide near-optimal solutions for this integrated problem in large sizes.

The remaining part of this paper is organized as follows: Section 2 reviews the straddle carrier scheduling and container storage allocation problems in the literature. Section 3 provides detailed problem description and associated mathematical formulation of the problem shown in the appendix. Section 4 designs a GA particularly adapted to the nature of our MIP model for deriving approximately optimal solutions, which would be able to handle large-sized problems. The experimental results are presented and discussed in section 5. The paper concludes with a summary of key findings and gives future research directions in section 6.

## 2 Literature review

Researches related to container terminal operations have gained extensive attention recently due to the development of marine transportation system. Stahlbock and Voß (2008) and Carlo et al. (2014b) reviewed the literature on container terminal operations, following the extensive review work by Steenken et al. (2004). In this section, we give a brief review of existing research on straddle carrier scheduling and container storage allocation problem in container terminals. Previous studies on using the GA to solve relevant optimization problems will be presented as well.

In general, the amount of research on SC (which is also called the lifting vehicle (LV)) scheduling problem is relatively small. For the problem of scheduling SCs, Steenken et al. (1993) tested different methods for the routing of SCs in order to minimise no-load ways. Kim and Kim (1999b) formulated a MIP model for routing a single SC during the loading operation of export containers. They aimed to minimise the total travel time of a SC. Das and Spasovic (2003) presented the procedure for scheduling of SCs by using a simulation method. The objective was to minimise the empty-loaded travel of SCs and the delay of trucks. In both of these studies, containers are handled by a combination of yard trucks and SCs. However, as mentioned above, our study is carried out in a container terminal based on the straddle-carrier system, where SCs could travel between the quayside and storage yard, and also function as flexible moving YCs for stacking containers. Moussi et al. (2011) discussed how to schedule LVs in loading/unloading operations by using information about pickup and delivery locations. The aim was to minimise the total travel time of all LVs. Boysen et al. (2013) studied the SC scheduling problem during the process of moving containers from storage yard to the trains in hinterland. Cai et al. (2013) presented a multi-objective (i.e. minimising SCs travelling time, SC waiting time, and finishing time of high-priority container-transferring jobs) optimisation model for the automated SCs scheduling problem. It was formulated as a binary integer programming model and solved by an exact algorithm based on the Branch and Bound (B&B) method with column generation. Zehendner et al. (2015) addressed an optimisation simulation method for the scheduling of SCs, aiming to minimise the overall delays. Different service strategies in a real-world terminal have been discussed. An agent-based simulation model is proposed by Garro et al. (2015) to determine the SC schedules in a transshipment terminal. The objective was to minimise the total empty-loaded travel of SCs, taking into account other performance measures, such as the total completion time and the idle time.

Storage space is a critical resource in container terminals, and the container storage allocation problem which determines container storage space and locations has been extensively studied. Carlo et al. (2014a) presented an overview of storage yard operations and highlighted current trends and developments. Kim and Kim (1999a) studied how to allocate storage space for import containers by analysing cases when the arrival rate of containers is constant, cyclic and dynamic. For each arriving vessel, spaces have been allocated to minimise the expected total number of rehandles. Kozan and Preston (1999) considered an optimisation problem of container transfer schedules and storage policies. Factors that influence the efficiency of a container terminal were analysed at a container terminal with different types of handling equipment, storage capacities and alternative layouts. Preston and Kozan (2001) modelled the seaport system with the objective of determining the optimal storage strategy for various container-handling schedules, such that the setup times and transport times become minimal. Zhang et al. (2003) made the first attempt to formulate the storage space allocation problem (SSAP) using a rolling-horizon approach. For each planning horizon, the problem was decomposed into two levels: the first level determined the total number of containers associated with each block in the yard, and the second level determined the number of containers associated with each vessel. Murty et al. (2005) proposed a method to incorporate dynamic load attributes into space allocation decisions. Bazzazi et al. (2009) further extended the work of Zhang *et al.* (2003) by considering refer and empty containers, and developed a GA to solve this problem. Nishimura et al. (2009) addressed the storage arrangement of trans-shipment containers on a container yard. An optimisation model was developed to investigate the flow of container transfers using intermediate storage at the yard.

There are some studies on the integration problems, but not specified in straddle carrier system. For example, Meersmans and Wagelmans (2001) presented a branch & bound (B&B) algorithm to solve the integrated scheduling problem of various types of handling equipment at an automated container terminal. Bish et al. (2001) was the first to study the integrated problem of storage allocation and vehicle scheduling. The authors assumed that each container had a number of potential locations in the yard. Containers were delivered from the ship to the yard by trucks. Bish (2003) extended this work to a combined problem of determining every container's storage location, each vehicle's scheduling, and the scheduling of each quay crane. Bish et al. (2005) further extended this study in developing algorithms for solving large sizes problems; however, the integrated problem was still solved separately without considering the interactions between the two sub-problems. Han et al. (2008) provided another way of integrating yard truck and storage allocation problems in trans-shipment hubs, which aimed to minimise the traffic congestions caused by yard trucks. Lee et al. (2008a) addressed a MIP model for integrating yard truck operations and the container storage allocation problem for import containers aiming to minimise the completion time of operations. The authors also provided a GA and another heuristic algorithm to solve the formulated MIP problem. Wong and Kozan (2010) investigated the vehicle scheduling problems in a multi-berth and multi-ship environment. The authors also provided a solution based on list scheduling and tabu search algorithms to tackle the computational complexity issue of the problem. However, in their work, container locations are pre-determined, which is different with our work. Luo and Wu (2015) proposed a new approach to determine the dispatching rules of automated guided vehicles and container storage locations, considering the dual-cycle operations. Niu et al. (2015) focused on the yard truck scheduling and storage allocation problems in the unloading process, which was solved by swam intelligence technique. Dkhil et al. (2018) proposed an integration problem considering the SC scheduling, but only focused in handling import containers, and the objective was to minimise the operating cost involved.

There exist many studies which have developed a GA for applications in container terminal operations. For example, Hartmann (2005) proposed a general model of various scheduling problems for SCs, automate guided vehicles, stacking cranes and workers that handle reefer containers; the computational results suggested that the GA is suitable to use in practice. In the area of quay crane scheduling problems, Tavakkoli-Moghaddam et al. (2009) presented a novel MIP model for the QC scheduling and assignment problem in a container terminal. They proposed a GA to solve this problem in the large-sized real-world situations. The efficiency of the GA was compared against LINGO software in terms of objective function values and computational times. From the perspective of scheduling vehicles, Lee et al. (2008b) formulated the QC scheduling and yard truck scheduling as a MIP model and solved it with a GA because the optimal solutions for large-sized problems cannot be obtained by exact algorithm in a reasonable time duration. Choi et al. (2011) proposed a GA for the efficient dispatching of container trucks to minimise the transportation cost of trucks. In the field of yard operations, Nishimura et al. (2005) addressed the dynamic assignment rules for yard trailers to QCs and the optimisation of yard trailer routing. The GA procedure was employed to obtain the near optimal solution to the problem in order to save yard operation time and costs. An efficient GA had also been applied to solve the storage space allocation problem (SSAP) in a container terminal. More recently, Homayouni et al. (2013) presented a GA to solve the integrated scheduling of quay cranes, automated guided vehicles and storage problems. A new GA operator was developed to enable to performance of the GA more accurate and precise. Kaveshgar and Huynh (2015) applied a GA for solving the integrated quay crane and yard truck scheduling problem. GA has been adapted with simulation method, as in Al-Dhaheri et al. (2016), in order to tackle the QC scheduling problem where the objective was to minimise the vessel handling time.

From examining the above literature, to the best of our knowledge, our work is the first to study the integration of SC scheduling and container storage allocation problem considering both unloading and loading processes. We will also design a novel GA as the solution method for large sizes problems.

## 3 Problem description

A ship’s turn-around time is one of the most important values used to measure the terminal’s efficiency and it consists of the times needed for unloading and loading containers. As explained earlier, in the container terminal with the straddle-carrier system, two main types of equipment are involved in the container unloading and loading processes: QCs and SCs.

Firstly we introduce the special characteristics of the straddle-carrier system and the dual-cycle operations in the container terminal. There is a special characteristic in the layout of the terminal with the straddle-carrier system: a buffer area under each QC for accommodating containers must be considered. In fact, during the unloading process, QCs must put containers down on the ground for SCs to pick up, which is similar with the loading process. However, the capacity of this buffer area is limited due to container re-handling and safety issues. In this section, we assume that, at most, one container is allowed at the buffer area for each QC during the handling process. The same assumption has been made in the work of Wong and Kozan (2010).

Secondly, we describe the dual-cycle operation considered in our study. We consider unloading and loading processes simultaneously which is defined as the dual-cycle operation in our problem. Currently most of the terminals adopt the single-cycle operation strategy, in which the loading process starts after the unloading process has been finished. However, dual-cycle operations could greatly improve the performance of the container terminal because single-cycle operations generate more empty moves of both QCs and SCs. Figure 3 compares the differences between these two operational strategies. A solid line means the loaded moves of QCs and SCs while a dashed line refers to empty-loaded moves of QCs and SCs. For the unloading process in the single-cycle method, a QC unloads an import container, and places it down in the buffer area under this QC before this container has been collected by a SC. This QC then moves on to handle the next import container which is to be discharged from the ship. For the transportation process of a SC, it takes a container to the yard and stacks it in its location before returning to the quayside without carrying any other containers. The loading process from the storage yard to the quayside in the single-cycle operation is in the reverse order of the unloading process (see figure 3(1)). However, the dual-cycle operations allow both QCs and SCs to perform the unloading of containers in the same cycle as the loading of containers (see figure 3(2)). Therefore, dual-cycle operations can increase the efficiency of both SCs' transportation and QCs' operation and observably reduce the empty-loaded moves of both types of handling equipment.

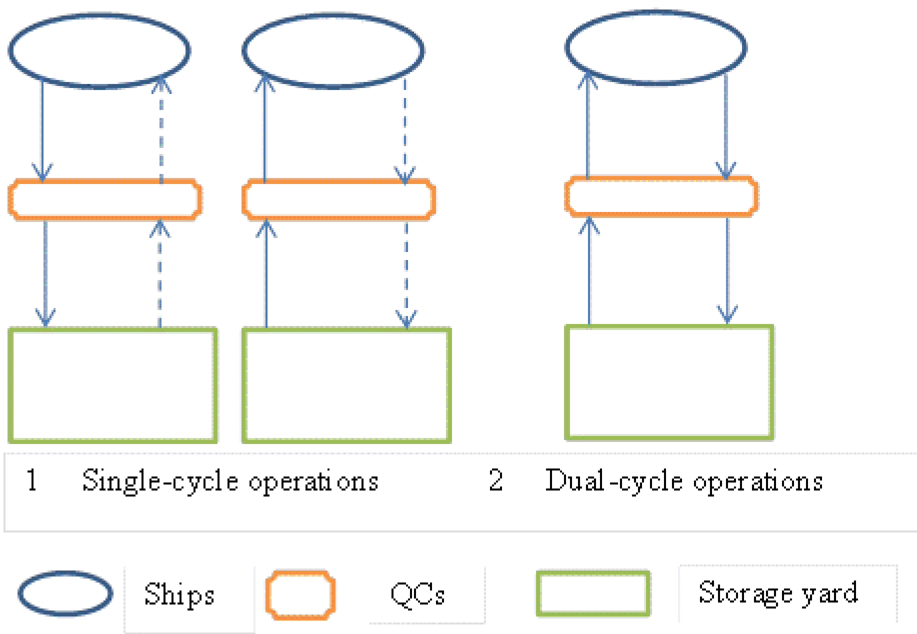


**Figure 3: Single-cycle and dual-cycle operational strategies in container terminals**

In such a situation that considers both unloading and loading operations, there are four transportation conditions for any two consecutive containers delivered by the same SC:
Condition 1: the SC transfers an import container after another import container.
Condition 2: the SC delivers an import container after an export container.
Condition 3: the SC transports an export container after another export container.
Condition 4: the SC moves an export container after an import container.

Thirdly, we introduce the environment of the storage yard. The storage yard consists of blocks, which are represented by bays, rows and tiers. Figure 4 shows the structure of a container yard in the straddle-carrier system. In this study, the reshuffle problem is not considered: we assume all the export containers are located on the top of the stacks and all the import containers will be placed on the top of the stacks.

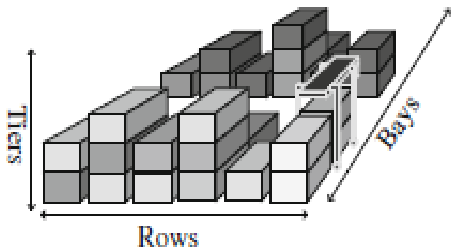


**Figure 4: Yard block structure under the straddle-carrier system (source: Wiese et al. (2010))**

The problems considered in this section are: (1) the SC scheduling problem. Specifically, there are a set of containers and a set of SCs. Each container must be processed by a SC once and each SC can handle one container at a time. The solution to this problem, which is the SC's schedule, defines the sequences and the start/finish times for every container handled by the set of SCs; and (2) the storage allocation problem for import containers. We do not consider the storage problems for export containers. This is because the information of export containers in the yard is provided by the yard map, which shows the locations for containers as well as the information on the type, destination, weight and size of each export container. However, the locations for import containers are unknown and taken as a decision variable. The decision is to determine where to locate these import containers in the yard. Therefore, the main contribution of this study is that it is the first to formulate the integration of SC scheduling and container storage problems by considering both unloading and loading processes, and to provide solution methods to solve it. The detailed model is presented in appendix.

# 4 Genetic algorithm for the proposed problem

Genetic algorithms (GAs) have been used extensively in solving sequencing and scheduling problems. GA is a well-known heuristic approach inspired by the natural evolution of the living organisms that works on a population of the solutions simultaneously. It was first introduced by Holland (1975) and further developed by Goldberg (1989). It combines the concept of survival of the fittest with structured, yet randomised, information exchange to undertake robust exploration of the solution space. GA starts with a set of random solutions called a population. In the natural world, each individual named chromosome is assigned with a fitness value. The exploration processes are performed by genetic operators of crossover and mutation. The new generation is selected based on the Darwinian theory of evolution in which individuals with better performances will have more probability of being chosen, and it is controlled by the parent selection and offspring acceptance strategies.

The reason why we choose GA in this study is that, firstly due to the complexity of the problem under study, the exact algorithm, i.e. CPLEX, is not able to provide solutions for large sizes of over 100 containers; secondly, GA is a well-known heuristic approach that its efficiency is verified by many problems in the literature in order to solve the large-sized problems with approximately optimal solutions (Bazzazi et al. (2009), Han et al. (2010), Lee et al. (2010) and Tavakkoli-Moghaddam et al. (2009)), and thirdly we need a population-based approach such as GA to better explore the solution space. By this means, we want to obtain the best delivery sequences of SCs and the best assigned locations for import containers such that the berth time is minimised.

**Chromosome representation and initialisation:** Considering the decision variables $x_{(i,k)}^{(j,l)}$ and $y_{(i,k)}^{m}$, i.e. the travelling sequences of SCs and the assigned locations for import containers, we construct our initial solutions as follows: Let us denote $|D|$ as the total number of import containers, $|N|$ as the total number of containers, $|M|$ as the total number of available locations and $c$ as the total number of SCs.

(1) SC assignment: randomly choose a SC from 1 to $c$ (constraints (3) and (4) and assign this number to one container; repeat this $|N|$ times until a string of length of $|N|$ is generated. So that constraints (1) and (2) are met.

(2) Location assignment: since the locations are uniformly distributed, we first randomly choose $|D|$ locations from 1 to$|M|$; then assign a series of numbered labels from 1 to$|D|$ to these selected locations; now we can assign these locations to $|D|$ import containers. For each import container, choose a location from 1 to $|D|$.So that constraints (5) and (6) are met.

(3) Chromosomes are generated respectively following steps 1-2 until the population size reaches a given number (say 100) to ensure a large search space.

(4) Evaluate each individual in the initial population by calculating the objective function value (OFV), according to constraint (9)-(21).

Let us use an example where there are eight containers, from which containers (1, 1), (2, 1), (3, 1) and (4, 1) are handled by QC 1 and containers (1, 2), (2, 2), (3, 2) and (4, 2) are handled by QC 2. Particularly, containers (2, 1), (3, 1), (1, 2) and (4, 2) are import containers; the others are export containers. Figures 5 and 6 show a chromosome representation of this example; for example, in figure 5, import container (2, 1) and export container (2, 2) will be handled by SC 1 in sequence; and in figure 6, import container (2, 1) will be assigned to location 2 and container (4, 2) will be assigned to location 4.

| *Containers* | *Dispatched SCs* |
|---|---|
| ***(1, 1)*** | **2** |
| ***(2, 1)*** | **1** |
| ***(3, 1)*** | **3** |
| ***(4, 1)*** | **2** |
| ***(1, 2)*** | **3** |
| ***(2, 2)*** | **1** |
| ***(3, 2)*** | **2** |

| | |
|---|---|
| ***(4, 2)*** | **2** |

**Figure 5: An initial solution of the dispatched SCs for an example with eight containers**

| *Containers* | *Assigned locations* |
|---|---|
| ***(2, 1)*** | 2 |
| ***(3, 1)*** | 3 |
| ***(1, 2)*** | 1 |
| ***(4, 2)*** | 4 |

**Figure 6: An initial solution of the assigned locations for an example with four import containers**

**Genetic operators design:** (1) Crossover: we use uniform crossover for the chromosome of 'dispatched SCs' and use uniform order-based crossover for the chromosome of 'assigned locations' to avoid missing and conflict genes. (2) Mutation: it provides individual diversity during the search process, and it is controlled by a mutation rate. Swap mutation is adopted here in which two randomly chosen genes are exchanged for both two parts of the chromosome.

**Offspring acceptance:** The semi-greedy strategy is used here to accept the offspring generated by the genetic operators: an offspring is accepted for the next generation if its fitness is better than the average fitness of its parent(s). This strategy is able to reduce the computation time of the proposed algorithm and makes a monotonous convergence toward the best solution.

**Parents selection strategy:** The selection mechanism used here is binary tournament selection, in which two randomly chosen individuals are compared in terms of their fitness values, and the one with higher fitness will be selected as one parent for generating offspring in the next generation. In addition, the best individual in each generation is always chosen as parent for next generation.

**Stopping criterion:** When the number of evolving generations reaches the defined maximum number of generations or the standard deviation of the objective function values (OFVs) in the current generation is below a small value, the algorithm stops. These two criteria can decrease the computational time.

## 5 Computational results

To assess the efficiency of the genetic algorithm and the proposed integrated approach, we conduct a series of experiments through a comparison study between the solver CPLEX and GA to determine the best result for different sizes of problems. All the experiments were performed on an Intel® Core™ i3 CPU M370@2.40GHz and 4GB RAM under the Windows 7 operating system. Our proposed GA is implemented using MATLAB 7.14; and for solving the MIP formulations in small sizes, the solver CPLEX 12.5 by AIMMS 3.12 is used. For each example, we examined it in GA by 20 runs using the same set of parameters, and the means of OFVs and computation times are presented.

**Parameters settings**

(1) The number of containers varies from 5 to 300, where 5-20 are considered as small-sized problems and 20-300 are considered as large-sized problems. We also consider the number of SCs varies from 2 to 10, and the number of QCs varies from 2 to 3.
(2) The uniform distribution was assumed for all the operational times. The handling times of each QC on these containers follow uniform distribution U(30, 180)s, and the travelling times of SCs between each QC and each container's available location follow uniform distribution U(40, 300)s.
(3) GA parameters take the following settings based on preliminary tests: Crossover rate $P_c$: 0.8; Mutation rate $P_m$: 0.01; Population size *Pop*: 100; and Maximum number of generations $M_g$: 40.

**Small-sized problems**

We examine 10 problems to compare the performances of the proposed GA with CPLEX in small sizes. Table 1 provides comparison results for CPLEX and GA for the various small-sized problems.

**Table 1: Comparison results of CPLEX and GA in small sizes**

| No | No. of containers | QCs/ SCs | CPLEX | | GA | | OFV Gap rate (%) |
|---|---|---|---|---|---|---|---|
| | | | Computation time (s) | OFV (s) | Computation time (s) | OFV (s) | |
| **1** | 5 | 2/2 | 0.03 | 286 | 0.12 | 286 | 0% |
| **2** | 6 | 2/2 | 0.17 | 502 | 0.21 | 502 | 0% |
| **3** | 7 | 2/2 | 0.03 | 519 | 0.27 | 519 | 0% |
| **4** | 8 | 2/3 | 0.02 | 538 | 0.36 | 544 | 0% |
| **5** | 9 | 2/3 | 1.08 | 620 | 1.74 | 629 | 1.45% |
| **6** | 10 | 2/3 | 1.03 | 742 | 1.08 | 754 | 1.62% |
| **7** | 10 | 2/4 | 1.08 | 715 | 1.55 | 730 | 2.09% |
| **8** | 15 | 2/4 | 2.47 | 1143 | 1.92 | 1162 | 1.66% |
| **9** | 20 | 2/3 | 2.72 | 1647 | 2.64 | 1694 | 2.85% |
| **10** | 20 | 2/4 | 4.77 | 1754 | 2.58 | 1797 | 2.45% |

In order to compare the two algorithms, we consider the computation time and objective function value (OFV) as the measures of efficiency and effectiveness. In each problem, the related values are listed. In order to determine whether there is a significant difference among the performance of the two algorithms, an OFV gap (in percentage for each example) is calculated. These results indicate that generally, for small-sized problems, CPLEX outperforms GA in obtaining better OFV in a shorter computation time. However, the average gap between GA and CPLEX is quite small with the average gap of 1.21%, from which we are convinced about the validity of our proposed GA.

**Large-sized problems**

To further demonstrate the benefits of our proposed GA, we compare GA with CPLEX in solving the problems in large sizes; we conduct the following experiments and present the results in table 2. It is observed that GA provides much more stable solutions in that it is able to solve problems with more than 100 containers which CPLEX cannot achieve. In addition, the convergence time of GA is less than that of CPLEX, which indicates that the proposed integrated problem is difficult to solve within a limited time by exact algorithms. However, when container number is small, computation time by CPLEX is less than that of GA; it shows an exponential increase in the computation time by CPLEX when the container number increases. As shown in table 2, CPLEX could only obtain the optimal solutions from cases 11 to 25. However, the average of the relative gap between GA and the best solution obtained by CPLEX in terms of OFVs for these 15 cases is about 2.6 %, which is a promising result. It is observed from this table that the OFV is reduced as the number of QCs is increased (for example case 20 and case 21). This trend is the same when the number of SCs is increased (for example case 21 and case 22). As discussed in previous sections, there is a trade-off between the improvement of OFV and the increased number of equipment. Further analysis has been performed to see the evolving process of our proposed GA as shown in figure 7. This figure shows the typical GA convergence performance for an example with 100 containers, two QCs and eight SCs. It can be seen that the evolving process converges fast and achieves the near-optimal solution before 30 generations. These results clearly indicate that our proposed GA can obtain better-quality solutions with much shorter computing time compared with CPLEX.

**Table 2: Comparison results of CPLEX and GA in large sizes**

| No | No. of containers | QCs/ SCs | CPLEX | | GA | | OFV Gap rate (%) |
|---|---|---|---|---|---|---|---|
| | | | Computation time (s) | OFV (s) | Computation time (s) | OFV (s) | |
| **11** | 30 | 2/3 | 8.63 | 2591 | 21.19 | 2629 | 1.46% |
| **12** | 30 | 2/4 | 13.04 | 1963 | 40.71 | 1988 | 1.27% |
| **13** | 30 | 2/5 | 12.03 | 1872 | 20.97 | 1898 | 1.38% |
| **14** | 40 | 2/3 | 9.61 | 3309 | 72.51 | 3374 | 1.96% |
| **15** | 40 | 2/4 | 10.33 | 2891 | 51.32 | 2988 | 3.35% |
| **16** | 40 | 2/5 | 12.42 | 2413 | 112.11 | 2441 | 1.16% |
| **17** | 50 | 2/5 | 19.50 | 3243 | 78.18 | 3316 | 2.25% |
| **18** | 50 | 2/6 | 165.27 | 3027 | 104.56 | 3093 | 2.18% |
| **19** | 50 | 2/7 | 113.15 | 2898 | 135.23 | 2938 | 1.38% |
| **20** | 60 | 2/6 | 124.76 | 3607 | 179.82 | 3721 | 3.16% |
| **21** | 60 | 3/6 | 227.35 | 3013 | 122.90 | 3105 | 3.05% |
| **22** | 60 | 3/7 | 334.78 | 2871 | 98.63 | 2939 | 2.36% |
| **23** | 80 | 2/6 | 1163.52 | 5106 | 245.14 | 5182 | 1.48% |
| **24** | 80 | 2/7 | 1245.81 | 4992 | 192.74 | 5102 | 2.20% |
| **25** | 80 | 3/7 | 2667.93 | 3897 | 272.01 | 3991 | 2.41% |
| **26** | 100 | 2/6 | / | / | 414.68 | 6553 | / |
| **27** | 100 | 2/7 | / | / | 255.04 | 6491 | / |
| **28** | 100 | 2/8 | / | / | 301.70 | 5503 | / |
| **29** | 150 | 2/7 | / | / | 966.14 | 9383 | / |
| **30** | 150 | 3/7 | / | / | 970.27 | 7280 | / |
| **31** | 150 | 3/8 | / | / | 810.10 | 7253 | / |
| **32** | 200 | 3/8 | / | / | 1004.10 | 9925 | / |
| **33** | 200 | 3/9 | / | / | 881.22 | 9501 | / |
| **34** | 200 | 3/10 | / | / | 1574.70 | 9209 | / |
| **35** | 300 | 3/8 | / | / | 2013.73 | 16003 | / |
| **36** | 300 | 3/9 | / | / | 1964.01 | 15852 | / |
| **37** | 300 | 3/10 | / | / | 2316.28 | 15074 | / |

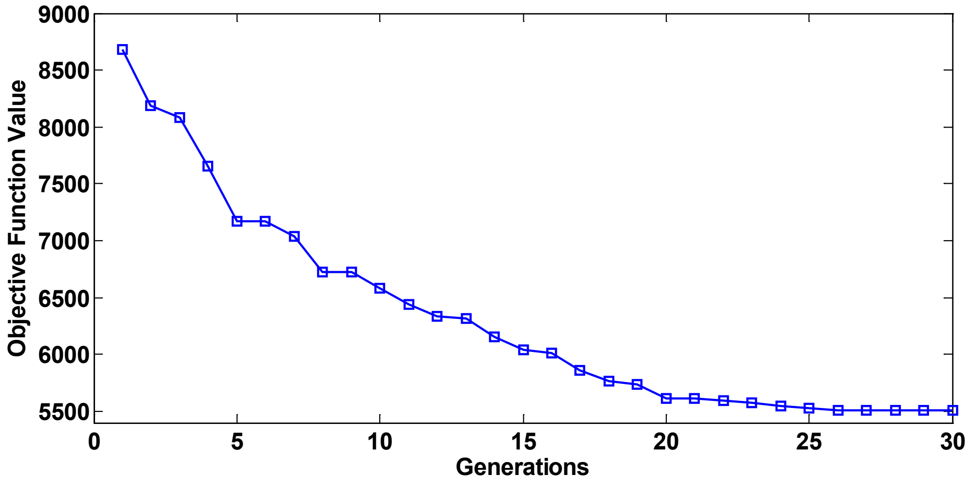

**Figure 7: GA convergence process for an example with 100 containers, two QCs and eight SCs**

# 6 Conclusions

The novelty of this study lies in the formulation of the new model and the heuristic methods of the model for solving the integrated problem. We consider the integration of SC scheduling and container storage allocation problem in the straddle-carrier system. These problems of searching optimal schedules of SCs and optimal locations of containers are very important in practical port logistics. Container unloading and loading processes are considered simultaneously. The problem is formulated as the mixed-integer programming problems aiming to minimise the berth time of the ship. As the NP-hard problem, the computational complexity increases exponentially with the increasing problem size. This makes it difficult to solve within a reasonable time with an exact method (CPLEX for example). This requires the use of heuristics methods (i.e. GA in this paper) to find approximately optimal solutions for large sizes. The adoption of GA provides a strong potential to solve real-world problems. The novelty of GA proposed here lies in the matrix representations of the initial solution, i.e. chromosomes, which are specially tailored to the problem under study. In this paper, its efficiency is proved first by the quality of solutions obtained in small-sized problems compared with the solutions from CPLEX, and the average gap between CPLEX and GA in terms of OFVs for the small-sized problems is very small; then the superiority of the proposed GA can be verified through large-scale problems in providing good solutions in a short computation time; and finally we are looking for the solutions with both efficiency and stability, which are illustrated by GA convergence performance. Thus, it was concluded that GA outperforms CPLEX in the average computational time and the ability of solving large-sized problems.

There still are, however, some possible directions for further research. The proposed models can be extended for uncertain environments. In practice, uncertainty exists in a number of areas, such as traffic congestions, machine breakdown, incorrect information of container, or ship delay, among others. To incorporate such uncertainties into the problem may require other approaches; for example, the stochastic programming approach or rolling horizon approach. Besides, implementing other heuristics or developing hybrid heuristics for these problems in order to find out whether the solutions can be further improved also suggests a promising piece of work. With the improved efficiency of the developed algorithm, the model can be further extended into a more general multi-ship multi-berth case.

## References

AL-DHAHERI, N., JEBALI, A. & DIABAT, A. 2016. A simulation-based Genetic Algorithm approach for the quay crane scheduling under uncertainty. *Simulation Modelling Practice and Theory,* 66**,** 122-138.

BAZZAZI, M., SAFAEI, N. & JAVADIAN, N. 2009. A genetic algorithm to solve the storage space allocation problem in a container terminal. *Computers & Industrial Engineering,* 56**,** 44-52.

BISH, E. K. 2003. A multiple-crane-constrained scheduling problem in a container terminal. *European Journal of Operational Research,* 144**,** 83-107.

BISH, E. K., CHEN, F. Y., LEONG, Y. T., NELSON, B. L., NG, J. W. C. & SIMCHI-LEVI, D. 2005. Dispatching vehicles in a mega container terminal. *OR Spectrum,* 27**,** 491-506.

BISH, E. K., LEONG, T. Y., LI, C. L., NG, J. W. C. & SIMCHI LEVI, D. 2001. Analysis of a new vehicle scheduling and location problem. *Naval Research Logistics,* 48**,** 363-385.

BOYSEN, N., EMDE, S. & FLIEDNER, M. 2013. Scheduling train loading with straddle carriers in container yards. *Journal of the Operational Research Society,* 64**,** 1841-1850.

CAI, B., HUANG, S., LIU, D., YUAN, S., DISSANAYAKE, G., LAU, H. & PAGAC, D. 2013. Multiobjective Optimization for Autonomous Straddle Carrier Scheduling at Automated Container Terminals. *Automation Science and Engineering, IEEE Transactions on,* 10**,** 711-725.

CARLO, H. J., VIS, I. F. & ROODBERGEN, K. J. 2014a. Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research,* 235**,** 412-430.

CARLO, H. J., VIS, I. F. & ROODBERGEN, K. J. 2014b. Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research,* 236**,** 1-13.

CHOI, H., PARK, B., LEE, J. & PARK, C. 2011. Dispatching of container trucks using genetic algorithm. Interaction Sciences (ICIS) 4th International Conference, 16-18 Aug 2011 Busan, South Korea. IEEE, 146-151.

DAS, S. K. & SPASOVIC, L. 2003. Scheduling material handling vehicles in a container terminal. *Production Planning & Control,* 14**,** 623-633.

DKHIL, H., YASSINE, A. & CHABCHOUB, H. 2018. Multi-objective optimization of the integrated problem of location assignment and straddle carrier scheduling in maritime container terminal at import. *Journal of the Operational Research Society,* 69**,** 247-269.

GARRO, A., MONACO, M. F., RUSSO, W., SAMMARRA, M. & SORRENTINO, G. 2015. Agent-based simulation for the evaluation of a new dispatching model for the straddle carrier pooling problem. *Simulation,* 91**,** 181-202.

GOLDBERG, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning,* Reading, MA, Addison-Wesley.

HAN, X., LU, Z. & XI, L. 2010. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research,* 207**,** 1327-1340.

HAN, Y., LEE, L. H., CHEW, E. P. & TAN, K. C. 2008. A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub. *OR Spectrum,* 30**,** 697-720.

HARTMANN, S. 2005. *A general framework for scheduling equipment and manpower at container terminals*, Springer.

HOLLAND, J. H. 1975. *Adaptation in natural and artificial systems,* Ann Arbor, MI, University of Michigan press.
HOMAYOUNI, S. M., TANG, S. H. & MOTLAGH, O. 2013. A genetic algorithm for optimization of integrated scheduling of cranes, vehicles, and storage platforms at automated container terminals. *Journal of Computational and Applied Mathematics*.
KAVESHGAR, N. & HUYNH, N. 2015. Integrated quay crane and yard truck scheduling for unloading inbound containers. *International Journal of Production Economics,* 159**,** 168-177.
KIM, K. H. & BAE, J. W. 2004. A look-ahead dispatching method for automated guided vehicles in automated port container terminals. *Transportation science,* 38**,** 224-234.
KIM, K. H. & KIM, H. B. 1999a. Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics,* 59**,** 415-423.
KIM, K. Y. & KIM, K. H. 1999b. A routing algorithm for a single straddle carrier to load export containers onto a containership. *International Journal of Production Economics,* 59**,** 425-433.
KOZAN, E. & PRESTON, P. 1999. Genetic algorithms to schedule container transfers at multimodal terminals. *International Transactions in Operational Research,* 6**,** 311-329.
LEE, D. H., CAO, J. X. & SHI, Q. 2008. Integrated Model for Truck Scheduling and Storage Allocation Problem at Container Terminals. Transportation Research Board 87th Annual Meeting, 13-17 Jan 2008a Washington, D.C, USA. IEEE
LEE, D. H., CAO, J. X. & SHI, Q. 2008. Integrated quay crane and yard truck schedule for inbound containers. IEEE International Conference on Industrial Engineering and Engineering Management, 2008b Singapore. IEEE, 1219-1223.
LEE, L. H., CHEW, E. P., TAN, K. C. & WANG, Y. 2010. Vehicle dispatching algorithms for container transshipment hubs. *OR Spectrum,* 32**,** 663-685.
LUO, J. 2013. *Modelling of quayside logistics problems at container terminals.* PhD University of Southampton.
LUO, J. & WU, Y. 2015. Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals. *Transportation Research Part E: Logistics and Transportation Review,* 79**,** 49-64.
MEERSMANS, P. J. & WAGELMANS, A. P. 2001. Effective algorithms for integrated scheduling of handling equipment at automated container terminals.
MOUSSI, R., YASSINE, A., KANSOU, A. & GALINHO, T. 2011. Scheduling of lifting vehicles with time windows in an automated port container terminal. 4th International Conference on Logistics (LOGISTIQUA), May 31-June 3 2011 Le Havre, France. IEEE, 55-61.
MURTY, K. G., LIU, J., WAN, Y. & LINN, R. 2005. A decision support system for operations in a container terminal. *Decision Support Systems,* 39**,** 309-332.
NG, W. & MAK, K. 2005. Yard crane scheduling in port container terminals. *Applied Mathematical Modelling,* 29**,** 263-276.
NG, W., MAK, K. & ZHANG, Y. 2007. Scheduling trucks in container terminals using a genetic algorithm. *Engineering Optimization,* 39**,** 33-47.
NISHIMURA, E., IMAI, A., JANSSENS, G. K. & PAPADIMITRIOU, S. 2009. Container storage and transshipment marine terminals. *Transportation Research Part E: Logistics and Transportation Review,* 45**,** 771-786.
NISHIMURA, E., IMAI, A. & PAPADIMITRIOU, S. 2005. Yard trailer routing at a maritime container terminal. *Transportation Research Part E: Logistics and Transportation Review,* 41**,** 53-76.

NIU, B., XIE, T., TAN, L., BI, Y. & WANG, Z. 2015. Swarm intelligence algorithms for Yard Truck Scheduling and Storage Allocation Problems. *Neurocomputing*.

PRESTON, P. & KOZAN, E. 2001. An approach to determine storage locations of containers at seaport terminals. *Computers & Operations Research,* 28**,** 983-995.

STAHLBOCK, R. & VOß, S. 2008. Operations research at container terminals: a literature update. *Or Spectrum,* 30**,** 1-52.

STEENKEN, D., HENNING, A., FREIGANG, S. & VOß, S. 1993. Routing of straddle carriers at a container terminal with the special aspect of internal moves. *Operations-Research-Spektrum,* 15**,** 167-172.

STEENKEN, D., VOß, S. & STAHLBOCK, R. 2004. Container terminal operation and operations research-a classification and literature review. *OR Spectrum,* 26**,** 3-49.

TAVAKKOLI-MOGHADDAM, R., MAKUI, A., SALAHI, S., BAZZAZI, M. & TAHERI, F. 2009. An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering,* 56**,** 241-248.

WIESE, J., SUHL, L. & KLIEWER, N. 2010. Mathematical models and solution methods for optimal container terminal yard layouts. *OR Spectrum,* 32**,** 427-452.

WONG, A. & KOZAN, E. 2010. Optimization of container process at seaport terminals. *Journal of the Operational Research Society,* 61**,** 658-665.

ZEHENDNER, E., RODRIGUEZ-VERJAN, G., ABSI, N., DAUZÈRE-PÉRÈS, S. & FEILLET, D. 2015. Optimized allocation of straddle carriers to reduce overall delays at multimodal container terminals. *Flexible Services and Manufacturing Journal,* 27**,** 300-330.

ZHANG, C., LIU, J., WAN, Y., MURTY, K. G. & LINN, R. J. 2003. Storage space allocation in container terminals. *Transportation Research Part B: Methodological,* 37**,** 883-903.

## Appendix: Mathematical formulation

We summarise the assumptions made in this study:

- Both loading and unloading processes are considered simultaneously in the single vessel environment.
- Travelling time between any two QCs are known, assuming QCs are not moving along the quayside during the unloading and loading processes.
- There is at most one container at the buffer area at any time during the operations.
- The sequences for each QC to handle both import and export containers are known, which means that the dual-cycle operations are prescribed by the fixed container sequences of the QCs.
- Number of containers, number of SCs and number of QCs are all known.
- A SC can only deliver one container at a time, and a QC can only handle one container at a time.
- SCs are shared among all the QCs. In other words, SCs can deliver containers to any of the QCs.
- The travelling times between export containers' yard locations to any of the QCs are known. This is because the locations of export containers and the locations of QCs are known.
- Yard locations for all the export containers are known; therefore the SCs' travelling times between any two export containers in the yard are known.
- Container reshuffles in the yard are not included in this study. All the containers to be processed are either on the top or will be placed on the top of the stacks.
- The pickup/drop-off times of containers by QCs and SCs are assumed to be included in the travel times.
- Traffic congestion on the road of SCs is not considered.

In the mathematical model, the following sets and parameters are defined:

| | |
|---|---|
| K | set of QCs |
| C | set of SCs |
| N | set of containers |
| L | set of export containers |
| D | set of import containers |
| $P$ | a very large number |
| M | set of locations for import containers |
| $(S, I)$ | the dummy starting job (container) |
| $(F, I)$ | the dummy ending job (container) |
| $O_S$ | The job set which contains all the jobs including the dummy starting job, i.e. |

$$O_S = N \cup (S, I)$$

$O_F$ — The job set which contains all the jobs including the dummy ending job, i.e.

$$O_F = N \cup (F, I)$$

$O$ — The job set which contains all the jobs including dummy starting and ending jobs, i.e.

$$O = \{(S, I), (F, I)\} \cup N$$

$k, l$ — index for QCs

$(i, k), (j, l)$ — index for containers, container *(i, k)* means the *i*th container handled by QC *k*

$N_k$ — The total number of containers handled by QC *k*

$m$ — index for yard locations, m is a positive integer

c — The total number of SCs

$h_{(i,k)}$ — QC *k*'s handling time of container $(i, k)$

$t_{(i,k)}$ — The SC travel time for each export container *(i, k)* from its yard location to its QC *k*

$\tau_k^m$ — The SC travel time between any QC *k* to any available yard location *m*

$\theta_{(i,k)}^l$ — The SC travel time for each export container *(i, k)* from its yard location to any QC *l*

$\sigma_{(i,k)}^m$ — The SC travel time for each export container $(i, k)$between its yard location and any of the available location *m*

$\pi_k^l$ — The SC travel time between any two QCs, i.e. QC *k* and QC *l*

The decision variables are as follows:

$u_{(i,k)}$ — the time QC *k* starts handling container (*i*, *k*); for import containers, this corresponds to the time QC *k* picks it up from the ship; for export containers, it corresponds to the time QC *k* picks it up from the buffer

$w_{(i,k)}$ — the time container $(i, k)$ is at the buffer. For import containers, it corresponds to the time a SC picks up container *(i, k)* from the buffer; for export containers, it corresponds to the time a SC sets container *(i, k)* down on the buffer

$v_{(i,k)}$ — the time container $(i, k)$ is in the yard. For import containers, it corresponds to the time container *(i, k)* has been placed into its yard location by a SC; for export containers, it corresponds to the time container *(i, k)* has been picked up from its yard location by a SC

The decisions of SC schedules and storage allocation can be represented using the following two decision variables:

$$x_{(i,k)}^{(j,l)} = \begin{cases} 1, \text{if container } (j,l) \text{ is delivered immediatly after container (i, k) by} \\ \text{the same SC} \\ 0, \text{otherwise } \forall (i,k) \in O_S, \forall (j,l) \in O_F \end{cases}$$

$$y_{(i,k)}^{m} = \begin{cases} 1, \text{if import container } (i,k) \text{will be located in location } m \\ 0, \text{otherwise } \forall (i,k) \in D, \forall m \in M \end{cases}$$

**Objective**: Minimise berth time $\max_k ( u_{(N_k,k)} + h_{(N_k,k)})$

**Subject to**:

$$\sum_{(j,l) \in O_F} x_{(i,k)}^{(j,l)} = 1, \forall (i,k) \in N \tag{1}$$

$$\sum_{(i,k) \in O_S} x_{(i,k)}^{(j,l)} = 1, \forall (j,l) \in N \tag{2}$$

$$\sum_{(j,l) \in N} x_{(S,I)}^{(j,l)} \leq c \tag{3}$$

$$\sum_{(i,k) \in N} x_{(i,k)}^{(F,I)} \leq c \tag{4}$$

$$\sum_{m \in M} y_{(i,k)}^{m} = 1, \forall (i,k) \in D \tag{5}$$

$$\sum_{(i,k) \in D} y_{(i,k)}^{m} \leq 1, \forall m \in M \tag{6}$$

$$v_{(i,k)} + t_{(i,k)} \leq w_{(i,k)}, \forall (i,k) \in L \tag{7}$$

$$w_{(i,k)} \leq u_{(i,k)}, \forall (i,k) \in L \tag{8}$$

$$w_{(i,k)} + \sum_{m \in M} \tau_k^m \, y_{(i,k)}^{m} = v_{(i,k)}, \forall (i,k) \in D, \forall k \in K \tag{9}$$

$$u_{(i,k)} + h_{(i,k)} \leq w_{(i,k)}, \forall (i,k) \in D \tag{10}$$

$$u_{(i+1,k)} - u_{(i,k)} \geq h_{(i,k)}, \forall (i+1,k), (i,k) \in N, i = 1,2, \ldots, N_k - 1, \forall k \in K \tag{11}$$

$$w_{(i,k)} + \theta_{(j,l)}^{k} \leq v_{(j,l)} + P(1 - x_{(i,k)}^{(j,l)}), \forall (i,k) \in L \cup (S,I), \forall (j,l) \in L \cup (F,I), \forall k \in K \tag{12}$$

$$v_{(i,k)} + \sum_{m} \tau_l^m y_{(i,k)}^{m} \leq w_{(j,l)} + P\left(1 - x_{(i,k)}^{(j,l)}\right), \forall (i,k) \in D \cup (S,I), \tag{13}$$

$$\forall (j,l) \in D \cup (F,I), \forall l \in K$$

$$w_{(i,k)} + \pi_k^l \leq w_{(j,l)} + P\left(1 - x_{(i,k)}^{(j,l)}\right), \forall (i,k) \in L \cup (S,I), \forall (j,l) \in D \cup (F,I), \forall k,l \in K \tag{14}$$

$$v_{(i,k)} + \sum_{m \in M} \sigma_{(j,l)}^m y_{(i,k)}^m \leq v_{(j,l)} + P\left(1 - x_{(i,k)}^{(j,l)}\right), \forall (i,k) \in D \cup (S,I), \forall (j,l) \in L \cup (F,I) \tag{15}$$

$$u_{(i,k)} \leq w_{(i+1,k)}, \forall (i+1,k), (i,k) \in L, i = 1,2,\dots,N_k - 1, \forall k \in K \tag{16}$$

$$w_{(i,k)} \leq u_{(i+1,k)} + h_{(i+1,k)}, \forall (i+1,k), (i,k) \in D, i = 1,2,\dots,N_k - 1, \forall k \in K \tag{17}$$

$$u_{(i,k)} + h_{(i,k)} \leq u_{(i+1,k)}, \forall (i+1,k) \in D, (i,k) \in L, i = 1,2,\dots,N_k - 1, \forall k \in K \tag{18}$$

$$w_{(i,k)} \leq w_{(i+1,k)}, \forall (i+1,k) \in L, (i,k) \in D, i = 1,2,\dots,N_k - 1, \forall k \in K \tag{19}$$

$$x_{(i,k)}^{(j,l)}, y_{(i,k)}^m \in \{0,1\}, \forall (i,k), (j,l) \in O, \forall m \in M \tag{20}$$

$$u_{(i,k)}, w_{(i,k)}, v_{(i,k)} \geq 0, \forall (i,k) \in N \tag{21}$$

The objective is to minimise the makespan of unloading and loading containers at the quayside, i.e. ship's berth time. It provides a way to calculate the completion time for all the QCs.

Constraints (1)-(6) are resource constraints.

Constraint (1) implies that for every container $(i,k) \in N$, there is one container $(j,l) \in N$ handled after it by the same SC.

Constraint (2) represents that for every container $(j,l) \in O_F$, there is one container $(i,k) \in O_S$ delivered before it by the same SC.

Constraint (3) and (4) guarantee that the total number of SCs that are employed for unloading and loading containers is *c*.

Constraint (5) ensures that every import container $(i,k) \in D$ will be located in exactly one location $m$ after the unloading process.

Constraint (6) means that every available location $m \in M$ can only accommodate, at most, one container.

Constraint (7) states that the time every export container has been placed in the buffer and the time it has been picked up from the yard by a SC must be set at least by a certain travel time between the yard and the quayside.

Constraint (8) guarantees that each export container will be loaded by a QC after it has been placed onto the buffer.

Constraint (9) means that the time of QC $k$ between picking up any import containers by a SC at the buffer and the time SC placing it onto the container's assigned location must be set at least by a certain travelling time between QC $k$ and location $m$.

Constraint (10) means that an import container can only be picked up from the buffer after the QC has released it to the buffer.

Constraint (11) ensures that the times for two containers *(i+1, k)* and *(i, k)* handled by the same QC $k$ must be set at least by a certain handling time of container *(i, k)*.

Constraints (12)- (15) give the time relationships for delivering two successive containers by the same SC in the four transportation conditions.

Specifically, if container $(j, l)$ is delivered immediately after container $(i, k)$ by the same SC, constraint (12) means that if both $(j, l)$ and $(i, k)$ are export containers, then the time between picking up container $(j, l)$in the yard and putting container$(i, k)$ in the buffer must be set at least a certain travelling time of SC from QC $k$ to the yard location of $(j, l)$.

Constraint (13) means that if both containers $(j, l)$and $(i, k)$are import containers, then the time between releasing container $(i, k)$ in the storage yard and picking up container $(j, l)$ in the buffer must be set by at least a certain travelling time from container$(i, k)$'s assigned location to QC $l$.

Constraint (14) means that if container $(i, k)$ is an export container and container $(j, l)$ is an import container, then time between taking container $(i, k)$ down to the buffer and picking up container $(j, l)$ must be set at least by a certain travelling time from QC $k$ to QC $l$.

Constraint (15) means that if container $(i, k)$ is an import container and container $(j, l)$ is an export container, then the time between placing container $(i, k)$ into its assigned location and picking up container $(j, l)$ must be set at least by a certain travelling time from the assigned location $m$ of container $(i, k)$ to the yard location of $(j, l)$.

Constraints (16)- (19) are for the time relationship for handling two successive containers by the same QC for the four conditions to ensure there is, at most, one container at the buffer at any time. Constraint (16) implies that if container $(i + 1, k)$ and container $(i, k)$ are both export containers, then container $(i + 1, k)$ can be placed on the buffer by a SC after container $(i, k)$ has been picked up by QC $k$.

Constraint (17) implies that if container $(i + 1, k)$ and container $(i, k)$ are both import containers, then container $(i + 1, k)$ can be placed on the buffer after container $(i, k)$ has been picked up by a SC.

Constraint (18) implies that if container $(i + 1, k)$ is an import container and container $(i, k)$ is an export container, then the starting time to handle container $(i + 1, k)$ must be after container $(i, k)$ has been place in its ship location.

Constraint (19) implies that if container $(i + 1, k)$ is an export container and container $(i, k)$ is an import container, then the time in which container $(i + 1, k)$ can be placed on the buffer must be set after the time that container $(i, k)$ has been picked up by a SC.

Constraint (20) and (21) are binary and non-negative constraints.