

Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals

Luo, J. and Wu, Y.

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:

Luo, J. and 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*, volume 79 : 49–64

<http://dx.doi.org/10.1016/j.tre.2015.03.006>

DOI 10.1016/j.tre.2015.03.006

ISSN 1366-5545

Publisher: Elsevier

NOTICE: this is the author's version of a work that was accepted for publication in Transportation Research Part E: Logistics and Transportation Review. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Transportation Research Part E: Logistics and Transportation Review, [79, (2016)] DOI: 10.1016/j.tre.2015.03.006

© 2016, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals

Abstract

This study proposes a new approach to determine the dispatching rules of AGVs and container storage locations, considering both unloading and loading processes simultaneously. We formulate this problem as a mixed integer programming model, aiming to minimise the ship's berth time. Optimal solutions can be obtained in small sizes, however, large-sized problems are hard to solve optimally in a reasonable time. Therefore, a heuristic method, i.e. genetic algorithm is designed to solve the problem in large sizes. A series of numerical experiments are carried out to evaluate the effectiveness of the integration approach and algorithm.

Keywords container terminal operations, automated container terminal, scheduling, container storage, dual-cycle

1 Introduction

Introduced about half a century ago, containers are large steel boxes of standard dimensions designed for easy and fast handling of cargos. Containers are eight feet wide, and come in three standard lengths - 20 feet, 40 feet or 45 feet. They have a height of either 8.5 feet or 9.5 feet. A 20-foot container carries up to about 28 tonnes of cargo with a volume up to 1000 cubic feet (Christiansen *et al.*, 2007). Before the introduction of containerisation, cargo was transported piece by piece, which made the transportation of goods very expensive and inefficient. One of the most significant benefits resulting from the introduction of containers is the resulting decrease in the potential risk of damage to goods, and the reduction in the need for re-packing between different transportation modes. Today, over 60% of the world's deep-sea general cargo is transported in containers; on some routes, particularly those between economically strong and stable countries, containerisation is up to 100% (Steenken *et al.*, 2004)

With the development of material handling and information technology, a number of terminals, such as Europe Combined Terminal (ECT) in Rotterdam, the Container Terminal Altenwerder (CTA) in Hamburg, the Thames Port in the UK, the Pasir Panjang Terminal (PPT) in Singapore, the Patrick Container Terminal in Brisbane and the Pusan Eastern Container Terminal, have started to employ automated container-handling equipment so as to satisfy the customers' growing demands and lower the labour costs (Luo, 2013). Among quite a few types of automated vehicles, automated guided vehicles (AGVs) are the most representative (Bae *et al.*, 2011). AGVs are robotics that are able to drive on a road-type network which incorporates electric wires or transponders in the ground to control the position of the AGVs. The recent development of automated container terminals has led to an increasing interest in studying the scheduling problems in such terminals. This is because the automated container terminal represents the on-going trend of the traditional terminal, in which all or some of the container-handling equipment is automated. Since, in the case of automation, there is a lack of physical human input, efficient scheduling and coordination of different resources (handling equipment, yard locations, etc.) are crucial to improve the overall performance of the automated container terminal.

In a typical automated container terminal, there are three main container handling equipment involved: quay cranes (QCs), automated guided vehicles (AGVs) and yard cranes (YCs). QCs are engaged for unloading/loading containers from/onto the ship at the quayside; AGVs are built with advanced technology to transport containers between the quayside and

the storage yard on the pre-defined paths; the storage yard is for temporarily storing of containers before they are further transferred by trucks/trains. During the unloading process, a QC discharges a container onto an AGV, which will deliver the container to the storage yard; then a YC collects the container from the AGV and stack it in the assigned slot. The loading process is the reverse of unloading process. Figure 1 shows the handling processes in container terminals. In this work, we consider the unloading and loading operations simultaneously, which is called the dual-cycle strategy.

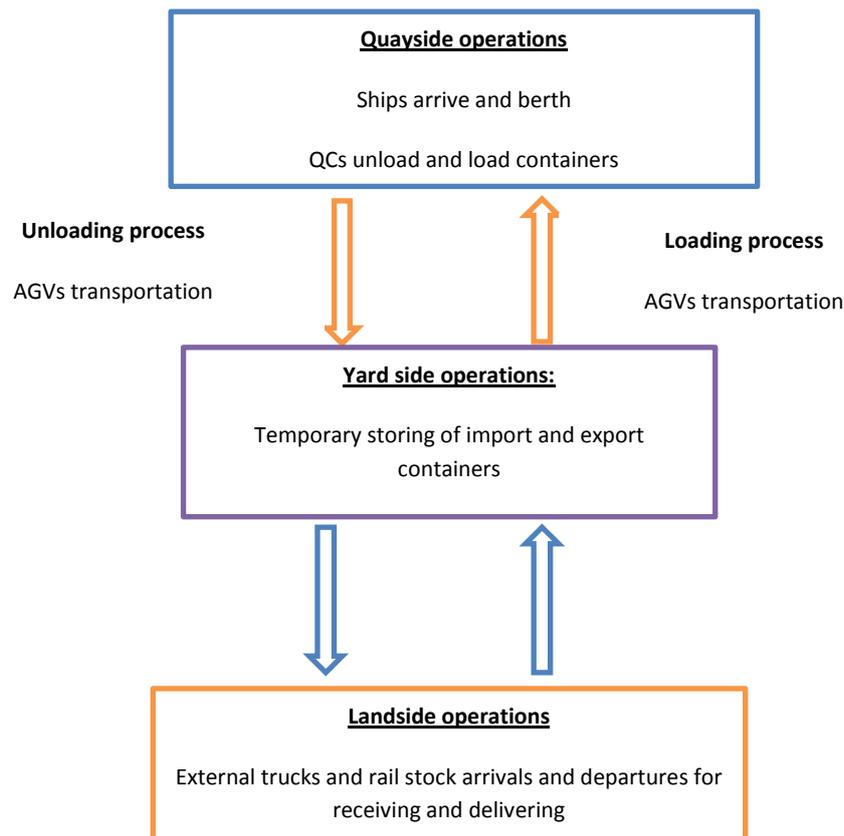


Figure 1: unloading and loading processes in an automated container terminal

Most of the existing literature considers the scheduling of one single type of equipment and the storage allocation problem of containers separately (Kim & Bae, 2004; Ng & Mak, 2005; Nishimura *et al.*, 2005; Ng *et al.*, 2007). However, vehicle scheduling and container storage are two highly interrelated decision problems faced by container terminals. AGV plays a role of interface between quayside and storage yard to coordinate the operations of QCs and YCs. On the other hand, container storage locations determine the YCs' handling sequences and routes in the yard. Therefore, this study concerns with the integration of AGV scheduling and container storage problems, in order to achieve the optimal performance of an automated container terminal.

The main contribution of this study is to provide an integrated modelling approach for AGV, YC scheduling problem and container storage allocation in the dual-cycle strategy, which considers the unloading and loading operations at the same time. We have also provided a novel designed solution method to solve the problem in practical sizes.

The paper is organized as follows. After introduction, section 2 gives a brief review of the related literature. Section 3 precisely describes the integrated problem considered in this study and develops a mathematical formulation of it. We design a novel genetic algorithm particularly for this integrated problem in section 4. Section 5 discusses several numerical experiments to evaluate the performance of the integration approach and the efficiency of GS. Section 5 concludes this study and suggests some extensions of this work.

2 Literature review

Researches related to container terminal operations are receiving more and more attentions due to the increasing importance of marine transportation system. Comprehensive classifications and reviews were provided by Steenken *et al.* (2004) and Stahlbock and Voß (2008), and most recently by Carlo *et al.* (2013). Here we provide a brief review of existing studies related to AGV scheduling and container storage problems in container terminals.

From the perspective of the scheduling problem in automated container terminals, most studies have focused on the AGV dispatching methods. Here, dispatching can be defined as the assignment of AGVs to deliver containers. For instance, Chen *et al.* (1998) suggested a greedy algorithm for AGVs' scheduling problem with the assumption that all the AGVs are assigned to one single quay crane. Grunow *et al.* (2004) carried out a study on multi-load AGVs' (the AGVs which could carry more than one container at a time) dispatching problem in a seaport container terminal. An alternative approach for scheduling AGVs was performed by Kim and Bae (2004), who proposed a mixed integer programming (MIP) model aiming to minimise both the total travel time of AGVs and the delay in the completion time of QCs. Briskorn *et al.* (2007) presented an alternative formulation of the AGV assignment problem. This formulation was based on a rough analogy to inventory management and is solved using an exact algorithm. Angeloudis and Bell (2010) studied an assignment algorithm for AGVs under uncertain conditions, which is suitable for real-time control of AGVs. The developed algorithm was applied to a simulated port environment where it was found to outperform the well-known heuristics. More recently, Kim *et al.* (2013) proposed a multi-criteria dispatching strategy for efficiently operating AGVS. The objectives include the minimisation of the QCs'

delay in order to achieve the optimal productivity of the terminal, and the minimisation of AGVs' empty travel in order to reduce CO_2 emissions.

There are other studies that are concerned with AGVs, but not specific to container terminals. For example, Bilge and Ulusoy (1995) exploited the interactions between the operations of machines and the scheduling of a material-handling system in a flexible manufacturing system, where material transfer between machines is performed by a number of AGVs. Van der Heijden *et al.* (2002) used several rules and algorithms for scheduling AGVs in an underground cargo transportation system in order to reduce cargo waiting times. Lim *et al.* (2003) introduced another AGV dispatching method by using the bidding concept, which means the decisions were made through the communication between related vehicles and automated machines.

Storage space is a critical resource in container terminals and container storage allocation problem, which determines container storage space and locations, has been extensively studied. Kim and Kim (1999) 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. Genetic algorithm (GA) technique has been used to reduce container handling times and ships' berth time. Factors that influence container terminal's efficiency 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 time become minimal. Chen *et al.* (2003) developed genetic algorithm for the general yard allocation problem (GYAP) aiming to minimise the yard space used. Zhang *et al.* (2003) made the first attempt to formulate storage space allocation problem (SSAP) using a rolling-horizon approach. For each planning horizon, the problem was decomposed into two levels: the first level determines the total number of containers associated with each block in the yard; the second level determines the number of containers associated with each vessel. Murty *et al.* (2005) proposed 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 genetic algorithm (GA) to solve this problem. Nishimura *et al.* (2009) addressed the storage arrangement of transshipment containers on a container yard. An

optimisation model was developed to investigate the flow of containers transfers using intermediate storage at the yard.

The dual-cycle operations have not been explicitly considered in previous studies. Goodchild and Daganzo (2006) presented a method to evaluate the reductions in the number of QC operations and operating time using dual-cycle technique. The research demonstrated that this technique can create significant improvement in the terminal's productivity. Zhang and Kim (2009) extended this work to the sequencing problem for both stacks and hatches. The problem was reformulated by a mixed integer programming model, and solved by a hybrid heuristic approach. Another work considering both unloading and loading processes was on the vehicle dispatching problem in container transshipment hubs (Lee *et al.*, 2010), which aimed to minimise the makespan at the quayside.

To the best of our knowledge, no work has studied the integrate problem of AGV scheduling and storage allocation problem in the dual-cycle operations. However, there are some efforts on integrating yard truck (one type of vehicle) scheduling and storage problem in conventional container terminals. For example, Bish *et al.* (2001) was the first to combine these two problems into a whole. They assumed that for each container, there are a number of potential yard locations assigned to it. A fleet of trucks transferred containers between the ship and yard. A heuristic approach was developed by decomposing the integrated problem into two steps, where location assignments were determined in step one and vehicle schedules were determined in step two. Two extensions were done by Bish (2003) and Bish *et al.* (2005) successively, 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 transshipment hubs, which aimed to minimise the traffic congestions caused by yard trucks. Lee *et al.* (2008) proposed an integrated model for yard truck scheduling and storage allocation during container unloading process in order to minimise the makespan of unloading operations. The two problems were treated into a whole. The problem was modelled as an MIP model and solved by a GA and dedicated heuristic algorithm. Lee *et al.* (2009) proposed a novel approach for that integrates these two problems considering both unloading and loading operations. The objective was to minimise the weighted sum of penalty of total delay and the cost of total travel time.

The integration problems that have been studied in the literature are either the integration of vehicle scheduling and yard crane scheduling, or the integration of vehicle scheduling and container storage. None of the previous studies addressed integrating these three problems together. As a result, the decisions made might be sub-optimal, and the improvement of

efficiency for a container terminal may not be as significant as an integrated scheduling approach, as we do in this work. In addition, there are very few studies on the dual-cycle strategy, which considers the unloading and loading operations of containers at the same time. Therefore, from theoretic point of view, this work provides a modelling technique for a broader integration (comparing with the studies in the literature) with considering vehicle scheduling, yard crane scheduling and container terminal allocation problems all together in dual-cycle strategy. Since all these problems are correlated with each other, this is also very important to practice.

3 Problem description and formulation

It is an accepted convention that the most important objective of day-to-day container terminal operations is the minimisation of a ship's berth time. In this section, we consider a terminal using QCs, YCs and AGVs for handling containers. During the loading process, a container picked up by a YC is put onto an AGV that delivers the container to the quayside. At the quayside, a QC picks up the container from the AGV and locates it in its designated location on the ship. The unloading process performs in the reverse order of the loading process. Figure 2 shows a typical layout of an automated container terminal. Here the dual-cycle operation means that unloading and loading operations are considered simultaneously; it is not necessary that there are both unloading and loading operations taking place in the same cycle.

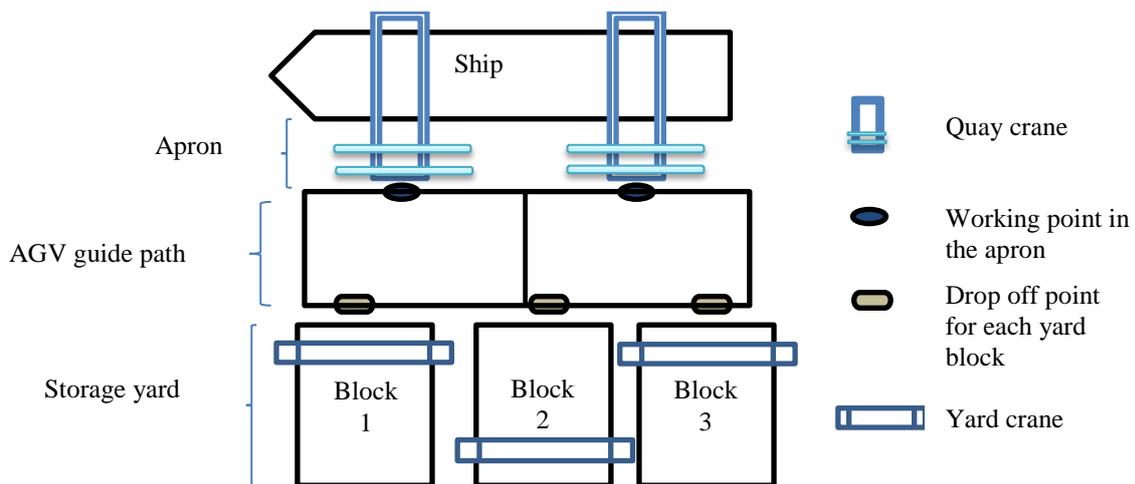


Figure 2: The layout of an automated container terminal

In practice, there are a number of ships berthed along the quayside at any one time, and each ship is served by a set of QCs. A few hours before the arrival of the ship, the terminal receives detailed information about the containers to be discharged from and loaded onto this

ship. This information allows the terminal operators to generate the QC schedule which specifies the handling sequence of containers by each QC and the estimated serving time of each container. Thus, at any time point in the operations, the QC operator has the information on which container to work on next, i.e. the handling sequence of each QC is known. The time required to handle each container is assumed to be deterministic (Bish *et al.*, 2001), and because each container has a specific location on the ship, we assume that the QC's handling times are different among containers (include both import and export containers) according to their ship locations. Because import containers have already been placed on the ship, their ship locations are known. For export containers considered in this problem, we assume their ship locations have been allocated before loading operation starts, which is a common assumption. Table 1 shows an example of the QC's sequence list and handling times. The handle time for a loading container means the time interval from picking it up from an AGV to locating it onto the ship; and the handle time for an unloading container means the time interval from picking it up from the ship to locating it onto an AGV. The first task for QC 1 is a loading container (L), i.e. export container, which will take 153 seconds of handling time by QC 1 to locate it to its ship location - bay 05 row 04 tier 04; the second task of QC 1 is also a loading operation, and it takes 125 seconds of handling time to load the container to the location of bay 04 row 06 tier 02 on the ship.

Table 1: An example of a QC's sequence list (L-loading; D-discharging)

Task list	Type	Ship location Bay-Row-Tier	Handle time (sec)
QC 1	L	05-04-04	153
	L	04-06-02	125
	D	07-03-03	190
QC 2	L	04-05-03	112
	D	06-02-02	155
	L	09-05-02	191

The location of a container in the storage block determines the time that the YC needs to handle it. Hence, container storage locations are very important to decide the YC schedule and thus have a great impact on the terminal's overall performance due to the high

operational cost of YCs. We assume there are some potential slots for accommodating the incoming containers (import containers), and the number of these slots is always greater than the number of import containers. This indicates that the storage yard always has the ability to locate the import containers. For export containers, when the loading operations begin, all the export containers must have been located in the yard; therefore the locations of export containers are usually known. Moreover, the YC's travelling time between any two locations (slots) in the yard are known, which can be easily calculated according to the location of containers and the yard layout; for example, the YC's travelling times between blocks are determined by the layout (distance) of the blocks.

After describing the QC and YC operations, we now look at the AGV transportation phase. In this study, all the handling equipment - QCs, YCs and AGVs - can carry only one container at a time. We assume every AGV can serve any QC, i.e. pooling strategy is applied for AGV dispatching. The traffic congestion is not considered here because the study of traffic control requires a more sophisticated engineering mechanism and thus is beyond the scope of our study. In the dual-cycle operations, there are four transportation conditions for any two consecutive containers delivered by the same AGV:

Condition 1: the AGV transfers an import container after another import container.

Condition 2: the AGV delivers an import container after an export container.

Condition 3: the AGV transports an export container after another export container.

Condition 4: the AGV moves an export container after an import container.

Generally, import containers and export containers are not mixed within the same block. We do not consider transshipment containers in our work. Specifically, export containers are usually placed near the quayside, where they can easily be loaded onto the ship. Import containers are usually stored near the gatehouse, close to rail/road exchange area, where they can be quickly picked up by trucks/trains. Therefore, here we assume each YC can work for only one type of containers, i.e. either export containers or import containers, to reduce the unproductive moves between the blocks for import containers and the blocks for export containers. In this case, for YC's movements, there are only two conditions that need to be considered for any two consecutive containers handled by the same YC, i.e. a YC moves an import container after another import container and a YC moves an export container after another export container.

Summarising the descriptions above, this study considers a novel integrated optimisation problem in the dual-cycle mode in which, given the QC's sequence lists and handling times for each container, AGV's schedules and the assignment of yard locations to import

containers then can be determined. In the meanwhile, YC's schedules can also be established.

The following assumptions are made throughout this study:

- (1) The container handling sequences of the QCs are known, which means that containers must be handled by the order exactly appeared in the QC's handling sequence list. Container locations on the ship are known and thus the handling time of each container by the QCs is known.
- (2) Number of containers, number of AGVs, QCs and YCs are all known.
- (3) AGVs, QCs and YCs can only carry one container at a time.
- (4) Travelling times of AGVs/YCs between any two processing locations are known. For example, the AGV's travelling time between each block to each QC, and the YC's travelling time between any two yard locations are known.
- (5) Traffic congestion of AGVs on the path is not considered here.
- (6) The storage yard has the ability to accommodate all the incoming containers.
- (7) The time needs for picking up/dropping off containers from/onto AGVs by QCs and YCs are negligible.
- (8) Interferences among QCs and YCs are not considered.
- (9) Import and export containers are not mixed within one block and each YC can travel among blocks either for import containers or export containers.

Now we introduce the notations related to the integrated problem. Each unloading/loading container action is referred to as a job.

Sets and parameters

D	set of import containers (jobs)
L	set of export containers (jobs)
N	set of all the containers i.e. $N = D \cup L$
P	set of yard locations
B	set of blocks
K	set of QCs
V	set of AGVs
C	set of YCs

k, l	index for QCs
b, a	index for blocks
$(i, k), (j, l)$	index for containers (jobs), job (i, k) means the i th container handled by QC k
N_k	the total number of containers handled by QC k
v	Total number of AGVs
c	Total number of YCs
(n, b)	index for yard locations, $n \in N^+$ positive integer, location (n, b) is slot n in block b
$h_{(i,k)}$	QC's handling time of container (i, k)
$w_{(i,k)}$	YC's handling time for export container (i, k) from the location of container (i, k) to the transfer point in front of the block that container (i, k) is placed. This is known because yard locations of export containers are known.
$\tau_{(i,k)}$	AGV's travelling time for each export container (i, k) from its located block to its assigned QC k
t_k^b	AGV's travelling time between QC k and block b
$\pi_{(k,l)}$	AGV's travelling time from QC k to QC l
$v_{(a,b)}$	YC's travelling time between block a and block b
$v1_{(i,k)}^{(j,l)}$	YC's travelling time from the transfer point of the block which stores export container (i, k) to the yard location of export container (j, l)
$\varphi_{(n,b)}$	YC's travelling time from the transfer point in front of block b to the location (n, b)
$\rho_{(i,k)}^b$	AGV's travelling time from the transfer point of the block that an export container (i, k) locates to transfer point of block b

M	a very large number
(S, I)	dummy starting job
(F, I)	dummy ending job
O_S	The job set which contains all the jobs including the dummy starting job, $O_S = N \cup (S, I)$
O_F	The job set which contains all the jobs including the dummy ending job, $O_F = N \cup (F, I)$
O	The job set which contains all the jobs including dummy starting and ending jobs, $O = \{(S, I), (F, I)\} \cup N$

Decision variables

$u_{(i,k)}$	the time QC k starts handling container (i, k) : for import container, it represents the time a QC picks it up from the ship; for export container, it means the time a QC picks it up from an AGV
$d_{(i,k)}$	the time a YC starts handling container (i, k) : for import container, it represents the time a YC picks it up from an AGV at the transfer point in front of the block; for export container, it means the time a YC picks up this container from its yard location

Note that we assume each QC's handling sequence is known, i.e. it is fixed in advance that which QC needs to handle which container. Now we introduce another four decision variables for AGV delivery sequences, import container yard location assignment and YC handling sequences:

$$x_{(i,k)}^{(j,l)} = \begin{cases} 1, & \text{if an AGV, which is scheduled to deliver container } (j, l), \text{ has just delivered container } (i, k) \\ 0, & \text{otherwise, } \forall (i, k) \in O_S, \forall (j, l) \in O_F \end{cases}$$

$$z_{(i,k)}^{(n,b)} = \begin{cases} 1, & \text{if import container } (i, k) \text{ will be stored in location } (n, b) \\ 0, & \text{otherwise, } \forall (i, k) \in D, \forall (n, b) \in P \end{cases}$$

We introduce the following intermediate variable $y_{(i,k)}^b$, where

$$\sum_{n \in N^+} z_{(i,k)}^{(n,b)} = y_{(i,k)}^b, \forall (i, k) \in D, \forall b \in B$$

$$y_{(i,k)}^b = \begin{cases} 1, & \text{if container } (i, k) \text{ will be located in block } b \\ 0, & \text{otherwise, } \forall (i, k) \in D, \forall b \in B \end{cases}$$

$$\sigma_{(i,k)}^{(j,l)} =$$

$$\begin{cases} 1, & \text{if a YC, which is scheduled to handle container } (j, l), \text{ has just handled container } (i, k) \\ 0, & \text{otherwise, } \forall (i, k) \in D \cup (S, I), \forall (j, l) \in D \cup (F, I) \text{ or } \forall (i, k) \in L \cup (S, I), \forall (j, l) \in L \cup (F, I) \end{cases}$$

Objective: Minimise the berth time of the ship

$$\text{Min: } \max_k (u_{(N_k, k)} + h_{(N_k, k)})$$

Constraints:

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

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

$$\sum_{(j,l) \in N} x_{(S,I)}^{(j,l)} = v \quad (3)$$

$$\sum_{(i,k) \in N} x_{(i,k)}^{(F,I)} = v \quad (4)$$

$$\sum_{b \in B} y_{(i,k)}^b = 1, \forall (i, k) \in D \quad (5)$$

$$\sum_{(n,b) \in P} z_{(i,k)}^{(n,b)} = 1, \forall (i,k) \in D \quad (6)$$

$$\sum_{(i,k) \in D} z_{(i,k)}^{(n,b)} \leq 1, \forall (n,b) \in P \quad (7)$$

$$\sum_{n \in N^+} z_{(i,k)}^{(n,b)} = y_{(i,k)}^b, \forall (i,k) \in D, \forall b \in B \quad (8)$$

$$\sum_{(j,l) \in DU(F,I) \text{ or } LU(F,I)} \sigma_{(i,k)}^{(j,l)} = 1, \forall (i,k) \in D \text{ or } L \quad (9)$$

$$\sum_{(i,k) \in DU(S,I) \text{ or } LU(S,I)} \sigma_{(i,k)}^{(j,l)} = 1, \forall (j,l) \in D \text{ or } L \quad (10)$$

$$\sum_{(j,l) \in D \text{ or } L} \sigma_{(S,I)}^{(j,l)} = c \quad (11)$$

$$\sum_{(i,k) \in D \text{ or } L} \sigma_{(i,k)}^{(F,I)} = c \quad (12)$$

$$u_{(i,k)} + h_{(i,k)} + \sum_{b \in B} t_k^b y_{(i,k)}^b \leq d_{(i,k)}, \forall (i,k) \in D \quad (13)$$

$$d_{(i,k)} + w_{(i,k)} + \tau_{(i,k)} \leq u_{(i,k)}, \forall (i,k) \in L \quad (14)$$

$$\begin{aligned} d_{(i,k)} + \sum_{(n,b) \in P} \varphi_{(n,b)} * z_{(i,k)}^{(n,b)} + \sum_{a,b} v_{(a,b)} * y_{(i,k)}^b * y_{(j,l)}^a \\ \leq d_{(j,l)} + M * \left(1 - \sigma_{(i,k)}^{(j,l)}\right), \forall (i,k), (j,l) \in D \end{aligned} \quad (15)$$

$$d_{(i,k)} + w_{(i,k)} + v1_{(i,k)}^{(j,l)} \leq d_{(j,l)} + M * \left(1 - \sigma_{(i,k)}^{(j,l)}\right), \forall (i,k), (j,l) \in L \quad (16)$$

$$u_{(i+1,k)} - u_{(i,k)} \geq h_{(i,k)}, \forall (i+1,k), (i,k) \in N, i = 1, 2, \dots, N_k - 1 \quad (17)$$

$$d_{(i,k)} + \sum_{b \in B} t_l^b y_{(i,k)}^b \leq u_{(j,l)} + h_{(j,l)} + M * \left(1 - x_{(i,k)}^{(j,l)}\right), \forall (i,k), (j,l) \in D \quad (18)$$

$$u_{(i,k)} + \pi_{(k,l)} + \tau_{(j,l)} \leq d_{(j,l)} + w_{(j,l)} + M * \left(1 - x_{(i,k)}^{(j,l)}\right), \forall (i,k), (j,l) \in L \quad (19)$$

$$u_{(i,k)} + \pi_{(k,l)} \leq u_{(j,l)} + h_{(j,l)} + M * \left(1 - x_{(i,k)}^{(j,l)}\right), \forall (i,k) \in L, (j,l) \in D \quad (20)$$

$$d_{(i,k)} + \sum_{b \in B} \rho_{(j,l)}^b y_{(i,k)}^b \leq d_{(j,l)} + w_{(j,l)} + M * (1 - x_{(i,k)}^{(j,l)}), \forall (i,k) \in D, (j,l) \in L \quad (21)$$

$$x_{(i,k)}^{(j,l)}, y_{(i,k)}^b, z_{(i,k)}^{(n,b)}, \sigma_{(i,k)}^{(j,l)} \in \{0,1\}, \forall (i,k), (j,l) \in O, \forall (n,b) \in P, \forall b \in B \quad (22)$$

$$u_{(i,k)}, d_{(i,k)} \geq 0, \forall (i,k) \in N, i = 1, 2, \dots, N_k, \forall k \in K \quad (23)$$

The objective is to minimise the ship's berth time for unloading and loading a set of containers. Constraint (1) implies that for every container $(i,k) \in N$, there is one container $(j,l) \in O_F$ delivered after it by the same AGV.

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

Constraints (3) and (4) guarantee that the number of AGVs employed for delivering these containers is exactly v .

Constraint (5) guarantees that each import container $(i,k) \in D$ will be assigned to one block b after the unloading process.

Constraint (6) ensures that each import container $(i,k) \in D$ will be located into one of the available slots in the yard.

Constraint (7) represents that each available slot (n,b) in the yard can locate, at most, one container.

Constraint (8) means that if a container is assigned to a block b , it can only be assigned to one available slot (n,b) within that block b . This constraint gives the relationship between two decision variables $y_{(i,k)}^b$ and $z_{(i,k)}^{(n,b)}$.

Constraint (9) implies that for every container $(i,k) \in D$ or L , there is one container $(j,l) \in D \cup (F,I)$ or $L \cup (F,I)$ handled after it by the same YC.

Constraint (10) represents that for every container $(j,l) \in D$ or L , there is one container $(i,k) \in D \cup (S,I)$ or $L \cup (S,I)$ handled before it by the same YC.

Constraints (11) and (12) guarantee that the number of YCs employed to handle containers in the storage yard is exactly c .

Constraint (13) means that for each import container (i, k) , the time between a YC picking it up from an AGV and a QC picking it up from the ship must be at least a certain QC's handling time $h_{(i,k)}$ plus the AGV's travelling time between QC k and the assigned block b for container (i, k) .

Constraint (14) means that for each export container (i, k) , the time between a QC picking it up from an AGV and a YC picking it up from its yard location must be at least a certain YC's handling time $w_{(i,k)}$ plus AGV's travelling time $\tau_{(i,k)}$ between the yard and QC.

Constraints (15) and (16) are the time constraints for two successive containers handled by the same YC in the two handling conditions - import-import and export-export, because YCs can only handle one type of container, either import or export container.

Specifically, constraint (15) means that if import container (j, l) is handled immediately after import container (i, k) by the same YC, then the time between this YC picking up container (j, l) and picking up container (i, k) must be set at least by the handling time to move container (i, k) to its assigned location (n, b) plus the YC's travelling time between the assigned blocks of the two containers.

Constraint (16) means that if export container (j, l) is handled immediately after export container (i, k) by the same YC, then the time between this YC picking up container (j, l) and picking up container (i, k) must be set at least by a certain YC's handling time and travelling time.

Constraint (17) states that all the containers are handled according to the QC schedule and the time interval between handling any two successive containers (i, k) and $(i + 1, k)$ must be at least the time this QC takes to handle container (i, k) .

Constraints (18)-(21) are the time constraints for two successive containers delivered by the same AGV in the four transportation conditions: import-import, export-export, export-import and import-export.

Specifically, constraint (18) implies that if both container (i, k) and container (j, l) are import containers, then the time between picking up container (j, l) from the QC l and transferring container (i, k) to a YC must be at least the AGV's travelling time from the assigned block b of container (i, k) to QC l .

Constraint (19) means that if both container (i, k) and container (j, l) are export containers, then the time between delivering container (i, k) to QC k and picking up container (j, l) from a YC must be at least the AGV's travelling time between QC k and QC l plus the AGV's travelling time from quayside to yard-side for picking up container (j, l) .

Constraint (20) represents that if container (i, k) is an export container and container (j, l) is an import container, then the time between picking up container (j, l) from QC l and transferring container (i, k) to QC k must be at least the AGV's travelling time between QC k and QC l .

Constraint (21) indicates that if container (i, k) is an import container and container (j, l) is an export container, then the time between picking up container (j, l) from the yard and transferring container (i, k) to a YC must be at least the AGV's travelling time between the blocks which locate container (i, k) and container (j, l) .

Constraints (22) and (23) are binary and non-negative constraints.

By introducing the dummy starting and ending jobs, the AGVS and YCs can be formulated by the above constraint (3), (4), (11) and (12) without causing the problem of symmetric solutions. Hence, such a MIP formulation is very compact. Because the proposed model is NP-hard, which is difficult to solve for large sizes, we propose the following genetic algorithm to solve the large-sized problems.

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 proposed by Goldberg (1989). It combines the concept of survival of the fittest with structured, yet randomised, information exchange to undertake robust exploration and exploitation of the solution space. The GA we proposed here consists of the following steps.

Chromosome representation and initialisation: By considering the main decision variables $x_{(i,k)}^{(j,l)}$, $y_{(i,k)}^b$, $z_{(i,k)}^{(n,b)}$ and $\sigma_{(i,k)}^{(j,l)}$, we use a matrix structure to represent the solution of the proposed problem. Here we differentiate between the chromosomes for import containers and those for export containers because the decision variables are not the same for these two types of container:

- For import containers, the decisions indicate the handling sequences of AGVs, YCs and assigned yard locations associated with each import container. More specifically, we have a matrix $\Psi 1$ with three columns, which represent the AGV delivery sequences, YC handling sequences and yard location assignments.
- For export containers, since the locations of export containers are known, the chromosome representation demonstrates two main decision variables: the handling sequences of containers for AGVs and YCs. We have a matrix $\Psi 2$ with two columns.

Figure 3 shows an example of the chromosome representation for the example of 10 containers. Three AGVs can carry both import and export containers. There are four YCs working in the storage yard, with YC 1 and YC 2 handling import containers, and YC 3 and YC 4 handling export containers. Since there are five import containers, we select five available locations and assign them to store these containers (see column 3 of matrix $\Psi 1$).

Import Containers			
Container	<i>Dispatched AGV</i>	<i>Assigned YC</i>	<i>Assigned Location</i>
(1, 1)	1	1	1
(3, 1)	2	2	3
(5, 1)	2	2	4
(2, 2)	3	1	2
(4, 2)	1	1	5

(a) The chromosome representation for import containers-the matrix $\Psi 1$

Export Containers		
Container	<i>Dispatched AGV</i>	<i>Assigned YC</i>
(2, 1)	2	3

(4, 1)	1	3
(1, 2)	3	4
(3, 2)	1	4
(5, 2)	2	3

(b) The chromosome representation for export containers- the matrix of $\Psi 2$

Figure 3: An illustration of chromosome representation for import and export containers

Let us denote $|D|$ as the total number of import containers, v as the total number of AGVs and c as the total number of YCs, in which YCs from 1 to $|C1|$ ($|C1| < c$) are used for handling import containers and YCs from $|C1| + 1$ to c are used for handling export containers. The initial population is constructed by the following steps:

- (1) Comparing travelling times from each QC to each available yard location, including the time AGV travels from QC to block and the YC moves from the transfer point of the block to the assigned location for import containers. We choose the locations with shortest distances for all the import containers, which satisfies constraint (7). Then we assign numbered labels from 1 to $|D|$ for all selected locations.
- (2) Randomly assign these locations to each import container and each location can only be assigned to one container (column 3 of matrix $\Psi 1$) to satisfy constraints (5) and (6).
- (3) Randomly choose an AGV from 1 to v (constraints (3) and (4)), i.e. assign one AGV to deliver a container (column 1 of matrix $\Psi 1$ and matrix $\Psi 2$) so that constraints (1) and (2) are met. This applies to both import and export containers.
- (4) Randomly choose a YC from 1 to $|C1|$, i.e. assign one YC to handle an import container (column 2 of matrix $\Psi 1$) and also choose a YC from $|C1| + 1$ to c (constraints (11) and (12)) to handle an export container (column 2 of matrix $\Psi 2$), so that constraints (9) and (10) are met.
- (5) Chromosomes are generated respectively by steps (1)-(4) until the population size Pop reaches a given number (e.g. 100) to ensure a large search space to start with.
- (6) Evaluate all the matrices $\Psi 1$ and $\Psi 2$ in the initial population by calculating the values of $u_{(i,k)}$ according to constraints (13)-(23). The objective function value is obtained by $\text{Max}(u_{(N_k,k)} + h_{(N_k,k)})$.

Parents selection strategy: We use ‘roulette wheel’ sampling for this problem. The fitness of each solution is obtained by calculating its objective function value (OFV). Here, as

the objective is to minimise the berth time, it is preferable that the individuals with smaller OFVs are chosen as parents for the next generation.

Genetic operators design: we use two-point crossover for the AGV and YC assignment parts of chromosomes and use uniform order-based crossover for the location assignment part of the chromosomes. Two-point crossover is an extended version of single-point crossover in which two selected parents are recombined by two randomly chosen points. This crossover applies to import and export container chromosomes (columns 1 and 2 of matrix $\Psi1$ and the matrix $\Psi2$). Uniform order-based crossover is an extended version of classical uniform crossover in which, after recombination, conflict genes are deleted and missing genes are replaced to prevent generating infeasible solutions, as shown in figure 4. This crossover works for column 3 of import container chromosome matrix $\Psi1$. Again, swap mutation is applied to exploit neighbour solutions, in which the two randomly selected genes are exchanged. The mutation operator works separately for the chromosomes of import containers and export containers.

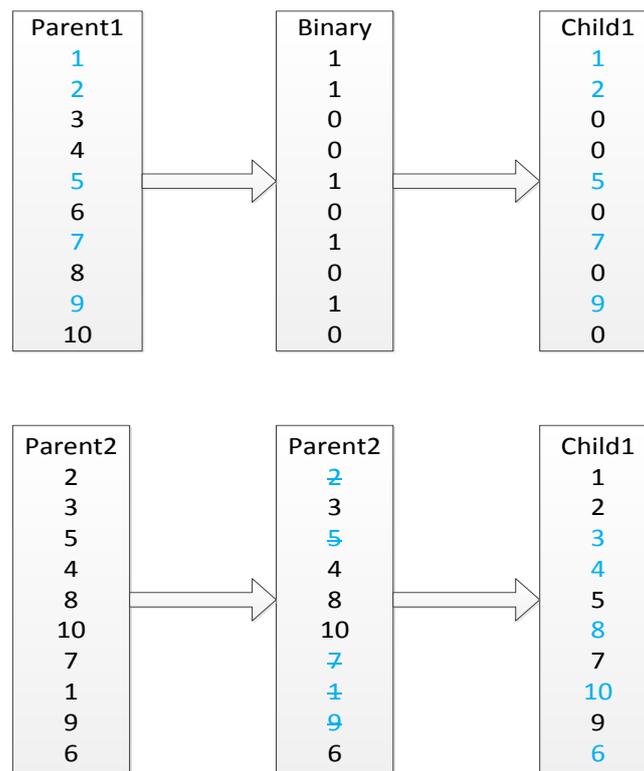


Figure 4: An illustration of uniform order-based crossover for an example of 10 locations

Each operator is performed with a certain probability that is known in advance by GA parameter settings. The crossover rate P_c and mutation rate P_m determine the performance of GA; therefore, proper value settings are needed in order to ensure the convergence of GA to

the global optimal neighbourhood in a reasonable time. The population size is kept unchanged during the crossover and mutation operations.

Offspring acceptance strategy: We use a semi-greedy strategy to accept the offspring created by the GA operators. In this strategy, an offspring is accepted as the new generation only if its OFV is less than the average of the OFVs of its parent(s). This approach enables GA to reduce the computation time and results in a fast convergence toward an optimal solution.

Stopping criterion: In order to balance the searching computation time as well as evolving an approximate optimal solution, we use two criteria as stopping rules: (1) the maximum number of evolving generations M_g allowed for GA, and (2) the standard deviation of the fitness values of chromosomes (σ_T) in the current generation T is below a small value.

5 Computational results

For small-sized problems, results obtained by the Branch and Bound (B&B) algorithm and GA are compared in terms of OFV and computation time. Small-sized problems are solved by commercial software AIMMS 3.11, which uses CPLEX 11.2 solver with B&B as the solution method. Due to the exponential increase in the computation time by the B&B algorithm as the problem size is getting larger, it is infeasible to try to solve the problem by the B&B algorithm to get exact solution. Therefore, GA is adopted for solving large-sized problems by providing approximately optimal solutions within a reasonable time. GA is implemented in MATLAB 7.11. All the experiments are performed on a computer with Intel® Core™ i3 CPU M370@2.40GHz and 4GB RAM under the Windows 7 operating system. For each problem, we run it by GA for 20 times using the same parameter settings and the means of objective function values and computation times are reported to justify the effectiveness and reliability of the proposed approach.

Parameters settings

- (1) The number of containers varies from 5 to 200, where 5-25 are considered as small-sized problems and 25-200 are considered as large-sized problems in this experiment. We also consider the number of AGVs varies from 2 to 10, the number of YCs varies from 2 to 7 and the number of QCs varies from 2 to 3.
- (2) The uniform distribution was assumed for all the operation times. The processing times of each QC on these containers follows uniform distribution $U(30, 180)$ s, and the handling times of each YC from each container's available location to the transfer

point of the block in which this container locates follows uniform distribution $U(40, 160)$ 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 generations M_g : 60.

Results for small-sized problems

Table 2: Results of computational experiments in small sizes

No	Containers	AGVs/ QCs/ YCs	B&B (MIP)		GA		OFV Gap rate (%)
			Computation time (s)	OFV (s)	Computation time (s)	OFV (s)	
1	5	2/2/2	0.02	355	0.01	355	0%
2	6	2/2/2	0.26	420	0.12	424	0.9%
3	7	2/2/2	0.44	496	0.22	499	0.6%
4	8	2/2/2	0.56	547	1.56	557	2%
5	9	2/2/2	0.62	608	2.06	615	1.2%
6	10	3/2/2	79.54	966	2.38	979	1.4%
7	10	4/2/3	169.82	926	4.62	948	2.4%
8	15	3/2/2	187.01	2029	6.46	2059	1.5%
9	20	3/2/2	1108.60	2563	9.05	2619	2.1%
10	25	5/2/3	/	/	12.14	3731	/

Ten random instances in small sizes are examined in this section. The problem parameters and GA parameters are set as above. Table 2 shows the comparison of results between B&B and GA. As shown in Table 2, the computation time of AIMMS 3.11 grows exponentially as the problem size increases, in which the proposed problem is known as NP-hard. For the problem with over 20 containers, AIMMS took over three hours without providing a solution. Moreover, it is found that the proposed GA can obtain the optimal/near-optimal solutions in

all the small-sized cases in reasonable computation times. Compared with the B&B results from AIMMS 3.11, the average of the relative OFV gap rate is 1.24%, which is a promising outcome. In terms of the computation times, GA outperforms B&B to give solutions at a faster speed. It demonstrates that our proposed GA is able to obtain the near optimal solution fast.

Results for large-sized problems

Table 3: Results of computational experiments in large sizes

No	Containers	AGVs/QCs/ YCs	Computation time (s)	OFV (s)
11	30	3/2/3	14.53	4867
12	30	4/2/3	8.01	4228
13	30	5/2/3	15.90	4123
14	40	4/2/2	42.26	6372
15	40	4/2/3	20.97	5206
16	40	4/2/4	46.43	3169
17	50	4/2/4	60.23	4341
18	50	4/3/4	70.97	3884
19	50	5/3/4	111.36	3382
20	60	4/2/3	187.61	6487
21	60	5/2/3	245.88	5862
22	60	6/2/3	143.81	5128
23	70	6/2/3	127.44	6400
24	80	5/3/4	223.05	7589
25	80	5/3/5	418.58	5024
26	80	6/3/5	211.33	4782
27	90	6/3/5	285.35	7119

28	100	6/3/5	216.68	7200
29	100	8/3/5	171.53	6655
30	100	10/3/5	275.59	5685
31	120	10/3/5	549.71	8677
32	150	8/3/5	701.37	11182
33	150	10/3/5	260.03	9034
34	150	12/3/5	169.59	10822
35	180	10/3/5	264.53	12907
36	200	10/3/5	612.37	13811
37	200	10/3/6	225.62	11917
38	200	10/3/7	203.70	11992

Table 3 shows the average computation times and OFVs in large-sized instances. Firstly, we examine the effect of the total number of containers on the optimal berth time and computation time. A general tendency is that the optimal berth time is increasing with the increasing number of containers.

Secondly, it is also noted that the decrease in the optimal berth time with respect to the number of QCs (such as case 17 and case 18) is more obvious than in other cases with different numbers of AGVS (such as case 12 and case 13) and YCs (such as case 24 and case 25), which means that the optimal berth time is affected most significantly by the number of QCs in container terminals. However, in reality, there are usually two to three QCs serving a medium-sized ship. This is because, when assigning more QCs to work for the ship, these QCs may have to operate together in a narrow berth space which may cause conflict. Also, the effect of the number of YCs is more significant on the optimal berth time than the effect of the number of AGVs. However, the number of cranes and vehicles must take reasonable values to avoid traffic congestions and conflicts. Thirdly, there is no apparent relationship between the problem size and the computation time for particular problems. However, generally, the computation time increases with the problem size when the number of containers increases.

Generally, these experiments indicate that the proposed GA is reliable in solving problems in different sizes and can be used in the real-life context to find the best combination of equipment for handling a set of containers.

From Figure 5, we observe that our proposed GA reached convergence quickly before 60 generations where the OFV gives the best value for an instance of 80 containers, five AGVs, three QCs and five YCs.

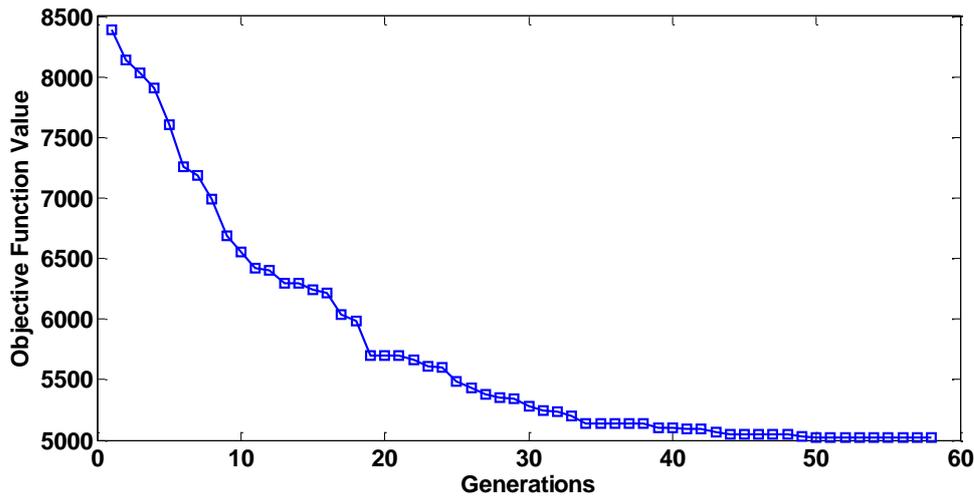


Figure 5: The typical convergence process of GA in a single run with the case of 80 containers, five AGVs, three QCs and five YCs

GA parameter sweep experiments

GA searches with different parameter settings are investigated for a particular case in order to select an effective parameter setting combination for this problem. Here, we consider the instance of 60 containers, six AGVs, two QCs and four YCs. GA parameter settings take the following values:

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

$$\text{Mutation rate } P_M = \{0.01, 0.02, 0.1, 0.2\};$$

$$\text{Population size } Pop = \{30, 50, 100, 150\};$$

$$M_g = 50.$$

Figures 6-8 show the performance comparisons among different parameter settings. Figure 6 shows the convergence curves on different crossover rates for this particular case with mutation rate of 0.01 and population size of 100. The results in figure 6 show that the OFVs with the best solutions after 25 generations are close to each other with the crossover rates of 0.8 and 0.9, but the curve with crossover rate of 0.9 terminates earlier; figure 6 shows the convergence curves on different mutation rates for this particular case with crossover rate of

0.9 and population size of 100. The results in figure 7 show that the curve with mutation rate of 0.01 converges to a smaller OFV; figure 8 shows the convergence curves on different population sizes for this particular case with crossover rate of 0.9 and mutation rate of 0.01. The results in figure 8 show that the OFVs with best solutions are close to each other after 20 generations with population size of 100 and 150, but the curve with population size of 100 terminates earlier. Therefore, according to these convergence curves, the crossover rate = 0.9, mutation rate = 0.01, and population size = 100 is the best set for this typical problem. These figures also show that for all the experiments, OFVs are not improved after 45 generations due to the fast convergence of our proposed GA. So, the maximum generations of 50 will be sufficient to acquire the near-optimal solutions. Besides, in figures 6 and 7, the OFVs of the best convergence solutions are close to each other with different settings of crossover rates and mutation rates; however, in figure 8, the OFVs of the best convergence solutions vary with different population sizes, and the curve with larger population size gives a smaller OFV. This further proves that the scope of the initial search space is important to the performance of the GA.

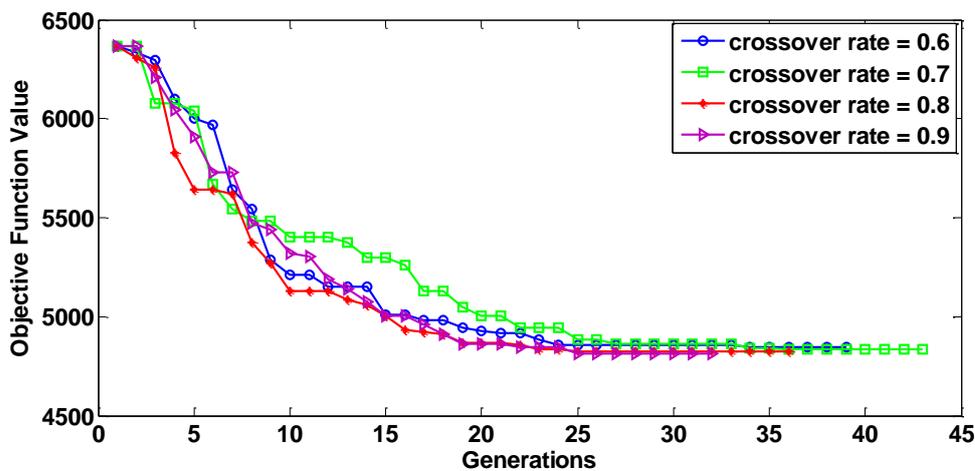


Figure 6: Performance comparison of different crossover rates for an example under the $Pop = 100$ and $P_m = 0.01$

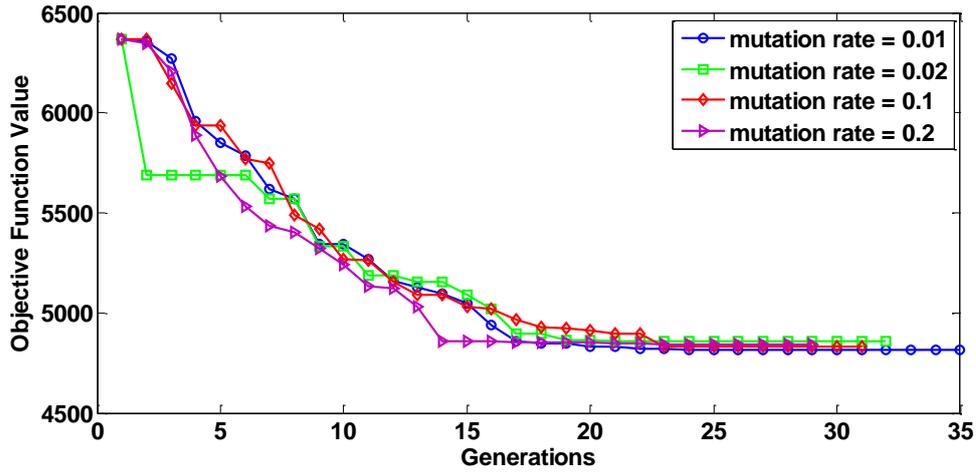


Figure 7: Performance comparison of different mutation rates for an example under the $Pop = 100$ and $P_c = 0.9$

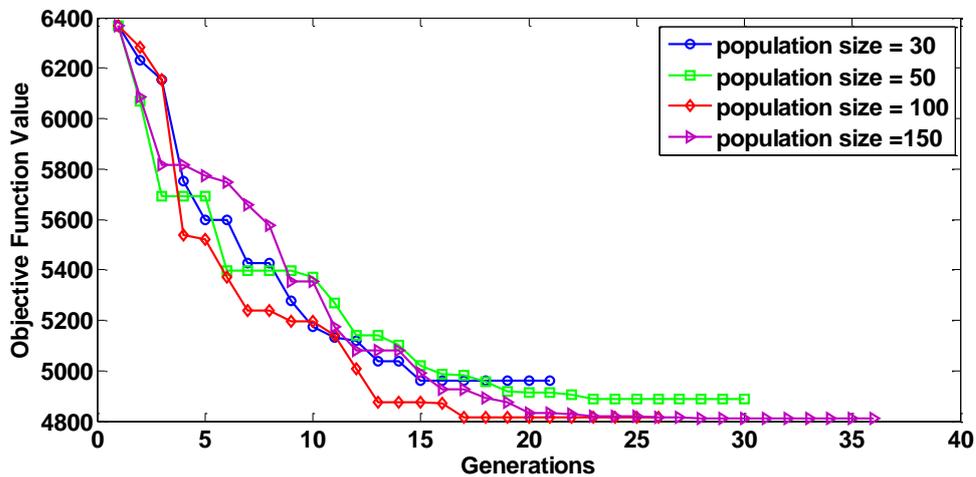


Figure 8: Performance comparison of different population sizes for an example under the $P_c = 0.9$ and $P_m = 0.01$

Lastly, we apply our proposed GA to the problem with 60 containers, six AGVs, two QCs and four YCs with the above optimal GA parameter settings, and the algorithm was run 10 times. The results are illustrated following the pattern shown by the box plot in figure 9. Each box represents the OFVs of the 10 runs in one generation. The central mark is the median of the OFVs, the edges of the box are the 25th and 75th percentiles and the whiskers are the most extreme data points. We can find that from the same starting point, each of the experiments improves the initial solution gradually, i.e. our proposed GA performs a stable manner in all the experiments; the deviation of OFVs in each generation is becoming stable during the evolving progress, and all the OFVs converge to a small neighbourhood of optimal solution at about 32 generations.

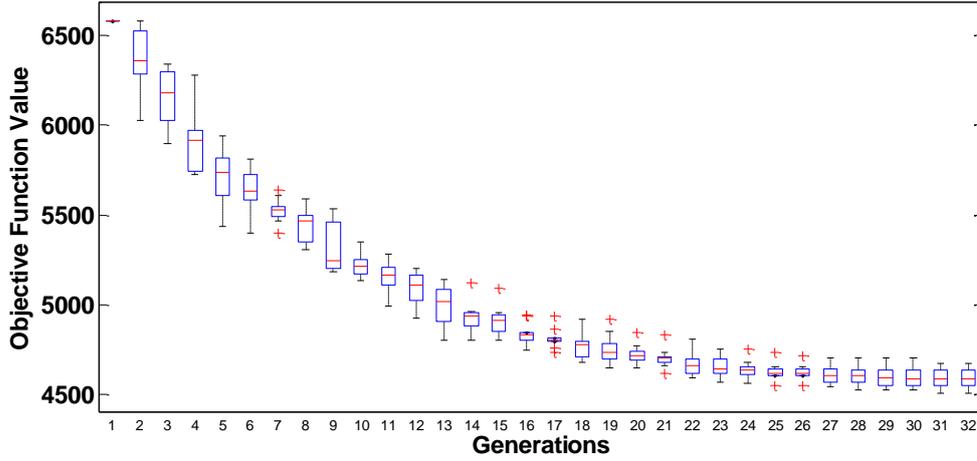


Figure 9: GA performance in 10 runs with $P_c = 0.9$, $P_m = 0.01$ and $Pop = 100$ for the case with 60 containers, six AGVs, two QCs and four YCs

6 Conclusions

The efficiency of container terminals depends extensively on the effectiveness of allocation of the terminal resources (vehicles/cranes/locations). In this respect, we considered the integrated vehicle scheduling and container storage problems in automated container terminals when both unloading and loading processes take place simultaneously. From the container-handling point of view, the automated container terminal has its own specific characteristic, i.e. automated vehicles/equipment, which aims to reduce the cost of labour and thus requires more advanced planning of this complex environment. The motivation for this study comes from the fact that vehicle/crane scheduling and container storage allocation problems are linked, a factor which is not considered in previous studies. The objective is to increase the productivity of automated container terminals by reducing the berth time of ships.

The technique used is to formulate this integrated problem as mixed-integer programming (MIP) model. The model can easily be solved using available optimisation software AIMMS 3.11 in small-sized problems. However, the computation time increases exponentially as the problem size gets larger. Therefore, to solve the complicated NP-hard problem, we suggested an evolutionary heuristics, GA, for the proposed model.

Sufficient computational experiments are carried out to test the performance of the models and the proposed solution methods. By solving the model, detailed schedules of AGVs and YCs as well as the container yard storage locations (for import containers) are determined. It has also been shown that compared with the B&B algorithm embedded in AIMMS 3.11, our proposed GA for the model performs in a stable manner in solving the problem of different

sizes. The average gap between B&B and GA in terms of the objective function values for small-sized problems is very small.

The contributions of the paper to the literature are: from an academic standpoint, this study is among the first in the field to investigate this integrated problem, and to provide the heuristic methods and suggest solutions. From a practical standpoint, this model incorporates the real-life operational issues, such as how to dispatch vehicles/cranes, where to locate containers, and the findings will benefit terminal managers in their day-to-day operations.

However, as the sizes of containerships are continuously increasing and the capacity of containerships will by implication also increase, more efficient operational research techniques and decision-making algorithms are needed for solving problems in real-time situations. Developing a mathematical formulation (MIP model) for the integrated problem considering both vehicle/crane scheduling and storage is one of the contributions of this paper, since the MIP model can provide a sound base for developing exact algorithms and other heuristic approaches for further research. Specifically, adapting other efficient heuristic algorithms to the problem and comparing the performances with the GAs proposed here are a needed direction of future research, to see if these techniques are able to achieve better results more efficiently than GA and thus improve the solutions to dispatch AGVs and cranes, and assign container locations. In addition, more practical considerations may be included. For example, precedence relationships because of the physical locations among tasks on the ship can be added in the model, and stochastic factors on the container information may also influence the implementation of the dual-cycle strategy.

References

- Angeloudis, P. & Bell, M.G.H. (2010) An uncertainty-aware AGV assignment algorithm for automated container terminals. *Transportation Research Part E: Logistics and Transportation Review*, **46**, 354-366.
- Bae, H.Y., Choe, R., Park, T. & Ryu, K.R. (2011) Comparison of operations of AGVs and ALVs in an automated container terminal. *Journal of Intelligent Manufacturing*, 1-14.
- 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.
- Bilge, Ü. & Ulusoy, G. (1995) A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research*, **43**, 1058-1070.
- 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.
- Briskorn, D., Drexl, A. & Hartmann, S. (2007) Inventory-based dispatching of automated guided vehicles on container terminals. In Kim, K., Guenther, H.O. (eds) *Container Terminals and Cargo Systems*. Springer Berlin Heidelberg, pp. 195-214.
- Carlo, H.J., Vis, I.F. & Roodbergen, K.J. (2013) Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research*, **236**,1-13.
- Chen, P., Fu, Z., Lim, A. & Rodrigues, B. (2003) The general yard allocation problem *Genetic and Evolutionary Computation-GECCO 2003*. Springer Berlin Heidelberg, pp. 209-209.
- Chen, Y., Leong, Y., Ng, J., Demir, E., Nelson, B. & Simchi-Levi, D. (1998) Dispatching automated guided vehicles in a mega container terminal *Paper presented at INFORMS, May 1998*, Montreal.
- Christiansen, M., Fagerholt, K., Nygreen, B. & Ronen, D. (2007) Maritime transportation. In Barnhart C, L. (ed) *Transportation. Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, pp. 189-284.
- Goldberg, D.E. (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA.
- Goodchild, A.V. & Daganzo, C.F. (2006) Double-cycling strategies for container ships and their effect on ship loading and unloading operations. *Transportation Science*, **40**, 473-483.
- Grunow, M., Günther, H.O. & Lehmann, M. (2004) Dispatching multi-load AGVs in highly automated seaport container terminals. *OR Spectrum*, **26**, 211-235.
- 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.
- Kim, K.H. & Kim, H.B. (1999) Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics*, **59**, 415-423.
- Kim, J., Choe, R. & Ryu, K.R. (2013) Multi-objective optimization of dispatching strategies for situation-adaptive AGV operation in an automated container terminal. *Proceedings of the 2013 Research in Adaptive and Convergent Systems*. ACM, City. p. 1-6.
- 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.

- 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*. IEEE City.
- Lee, D.H., Cao, J.X., Shi, Q. & Chen, J.H. (2009) A heuristic algorithm for yard truck scheduling and storage allocation problems. *Transportation Research Part E: Logistics and Transportation Review*, **45**, 810-820.
- Lee, L.H., Chew, E.P., Tan, K.C. & Wang, Y. (2010) Vehicle dispatching algorithms for container transshipment hubs. *OR Spectrum*, **32**, 663-685.
- Lim, J.K., Kim, K.H., Yoshimoto, K., Lee, J.H. & Takahashi, T. (2003) A dispatching method for automated guided vehicles by using a bidding concept. *OR Spectrum*, **25**, 25-44.
- Luo, J. (2013) Modelling of quayside logistics problems at container terminals. Ph.D thesis, Southampton Business School, University of Southampton, UK.
- 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.
- 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., Voß, S. & Stahlbock, R. (2004) Container terminal operation and operations research-a classification and literature review. *OR Spectrum*, **26**, 3-49.
- Van der Heijden, M., Ebben, M., Gademann, N. & van Harten, A. (2002) Scheduling vehicles in automated transportation systems Algorithms and case study. *OR Spectrum*, **24**, 31-58.

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.

Zhang, H. & Kim, K.H. (2009) Maximizing the number of dual-cycle operations of quay cranes in container terminals. *Computers & Industrial Engineering*, **56**, 979-992.