AN INTEGRATED SCHEDULING PROBLEM OF CONTAINER HANDLING EQUIPMENT IN THE LOADING OPERATION AT AUTOMATED CONTAINER TERMINALS

Jiabin Luo, Yue Wu
School of Management, University of Southampton
Southampton, SO17 1BJ, UK
j.luo@soton.ac.uk, y.wu@soton.ac.uk

ABSTRACT
In automated container terminals, containers are transported between ships and storage yard by different types of vehicles, for example, guided vehicle (AGV), yard crane (YC) and quay crane (QC). To improve the productivity of the automated container terminals, it is important to harmoniously synchronize operations of different types of container handling equipment. This paper examines the scheduling problem of all handling equipment in an integrated way by utilizing information about yard storage locations for the loading operation in an automated container terminal. A mixed integer programming (MIP) model is developed. As the optimization software can only be used to solve small size problems, we develop a genetic algorithm (GA) to obtain approximate optimal solutions for large size problems. Computational results demonstrate the benefits of integrated model and the effectiveness of the proposed algorithm.

KEY WORDS
Container handling; automated container terminal; integrated scheduling; genetic algorithm

1 Introduction
Containers are large steel boxes of standard dimensions which were first used in the maritime transportation industry in 1950s. The benefits of introducing containers include reducing delivery time for goods, reducing the potential risk of damaged goods, avoiding repacking operations during handling processes, and increasing security of the goods being transported. The proportion of cargos handled with containers has been increasing considerably over the last decades. Nowadays, most deep-sea cargos are transported in containers. Therefore, it is important for port managers to develop effective methods in order to improve the port efficiency and enhance its competitiveness. In all logistical activities, keeping the goods moving towards their destinations as quickly as possible with the least cost is one of the most important tasks for container terminals.

Container terminals perform as an interface between marine and land transportation systems. In the yard area, yard cranes (YCs) are used for stacking containers within the blocks. Vehicles are used for delivering containers from the yard side to the quayside, where quay cranes (QCs) are used for loading containers from the quayside to ships. The efficiency of the container terminal is marked by different factors. The time that the ship spends at the terminal is one of the most important factors for terminal managers to consider. Reducing ship’s berth time is the main objective for both researchers and practitioners. Because the investment in such an automated system is very high, it is of great importance to optimise the container handling operations as a whole in order to reduce the cost and thus increase the productivity of the container terminals.

In this paper, we propose a scheduling approach which integrates the scheduling of QC, AGV and YC during the loading operation to minimise the total makespan, which is defined as the total time the ship spends at the quayside. One contribution of this study is that we provide an integrated approach for scheduling QCs, AGVs and YCs. Another contribution is that we design a genetic algorithm (GA) to solve the problem in large size problems. Furthermore, since automated container terminal represents the next generation of the container terminals, research on the automated systems is important for both researchers and container terminals.

The remaining part of this paper is organized as follows. Section 2 is the literature review. Section 3 gives the description of the problem and an MIP model. Section 4 presents the computational result. Section 5 concludes this paper and gives future research directions.

2 Literature Review
There are an ever increasing number of research papers devoted to optimisation problems in container terminals in the literature. Comprehensive classifications and reviews were provided by Steenken, et al. [36] and Stahlbock and Voß [35]. In the perspective of scheduling problem in automated container terminals, most of studies have focused on the AGV dispatching methods, which can be defined as the assignment of AGVs to deliver containers. For instance, Chen, et al. [11] suggested a greedy algorithm for AGVs’ scheduling problem, which assumes

DOI: 10.2316/P.2013.793-047
that all the AGVs were assigned to one single quay crane. Grunow, et al. [14] carried out a study on multi-load AGVs dispatching problem, which means AGVs can carry more than one container at a time. Kim and Bae [22] developed an alternative approach for scheduling AGVs and who proposed a mixed integer programming model to minimise both the total travel time of AGVs and the delay in the completion time of QCs. Briskorn, et al. [7] presented an alternative formulation of the AGV assignment problem, which was based on a rough analogy to inventory management. Angeloudis and Bell [1] studied an assignment algorithm for AGVs under uncertain condition, which was suitable for real-time control of AGVs. They developed an algorithm, which was applied to a simulated port environment. The results show that the proposed algorithm outperforms the well-known heuristics approaches. There exist some other related works, which are concerned with the AGVs, but not specific to container terminals. For example, Bilge and Ulusoy [5] exploited the interactions between the operations of machines and scheduling of material handling system in a flexible manufacturing system, where material transfer between machines was performed by a number of AGVs. Van der Heijden, et al. [38] used several rules and algorithms for scheduling AGVs in an underground cargo transportation system in order to reduce cargo waiting times. Lim, et al. [30] introduced another AGV dispatching method by using the bidding concept, which means the decisions were made through the communication among related vehicle and machines.

In addition to the scheduling problem of AGVs, there exist a few of works regarding scheduling of QCs [27, 31, 32, 34, 37] and YCs [10, 16, 18, 29, 33]. Most of research, however, has focused on the scheduling of one type of equipment. The main contribution of this paper is to schedule AGVs, QCs and YCs simultaneously in order to determine the sequence of loading operations.

Since scheduling of QCs, AGVs and YCs are highly interrelated decision problems, optimising one of them is not sufficient to guarantee the optimisation of the whole container handling system and it is therefore necessary to synchronize the scheduling of all handling equipment. From examining the literature, issues on integrated decision problems have been addressed in recent years as more and more researchers and practitioners have realized the importance of developing integrated methods for solving problems in container terminal operations. Chen, et al. [9] presented an integrated model to schedule different types of equipment in order to minimise the makespan at the ship. The model was formulated as a hybrid flow shop scheduling problem with precedence and block constraints and solved by using a tabu search algorithm. Lee, et al. [26] proposed an MIP model to consider the scheduling of QC and yard truck simultaneously to minimise the makespan for the unloading operation. They proposed a genetic algorithm to solve the problem. Lau and Zhao [24] presented an integrated scheduling model for different types of handling equipment at automated container terminals to minimise the total travel time of YCs, AGVs and the delays of QC operations. A multi-layer genetic algorithm was generated to obtain a near-optimal solution for the integrated problem. Cao, et al. [8] addressed an integrated problem for yard truck and YC scheduling for loading operations and formulated an MIP model. Two efficient solution methods, based on Benders’ decomposition, were also developed in their paper. Wong and Kozan [39] studied the relationship between QCs, yard machines (vehicles with the ability to handle containers independently) and storage location. List scheduling and tabu search algorithms were developed to solve the problem. Lee, et al. [28] considered a transhipment hub, where the loading and discharging activities have to be handled simultaneously. Moreover, they used genetic algorithm to solve the problem.

Our work is different from most of the existing work, as we investigate a broader integration, in which the scheduling of QCs, YCs and AGVs, as well as the information on container yard locations is considered simultaneously. Because the time for loading is highly dependent on the loading sequence of containers while the time for unloading is proportional to the number of containers, it is more important to efficiently schedule the loading operation in order to minimise the operational time [23]. Therefore, in this paper we only focus on the integrated problem during the loading operations.

3 Problem Description and Formulation

In this section, the integrated scheduling problem of QCs, AGVs and YCs is described, and a mixed integer programming (MIP) model is presented.

3.1 Problem Description

In practical operations, before a container ship arrives, a berth will be allocated to the ship according to the tonnage, estimated arrive time and berth time [25]. Then a set of QCs are assigned to serve the ship. During the loading process, a container picked up by an YC is put on 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 the designated location on the ship.

Figure 1 illustrates a typical layout of an automated container terminal, which is used in the numerical experiment of this study. In this figure, transfer points for YCs to drop off containers are located in front of each block. Under each QC, there is one working point for this QC to collect containers from AGVs. The AGVs are travelling along the guided paths for delivering containers in an anti-clockwise direction.

In this study, we adopt the single-cycle operational strategy for AGVs where the discharging operation must be performed before the loading operation. This strategy has been widely used at most of container terminals worldwide. In this case, after delivering an export container to a QC, the AGV returns to the storage yard with empty load. When more than two QCs are working for a ship, either non-pooling or pooling policy for AGVs can be used. Although non-pooling is easy to implement because each AGV serves only one designated QC, here we adopt the pooling policy by assuming each AGV can serve any QC, in order to balance the work load during the loading operations. Congestions among AGVs on the path are not considered because the study about the interference of vehicles involves more complex scheduling and control of detailed movements of vehicles, which is another important issue for the AGV system in automated container terminal [12]. The inside of a container ship is constructed in the form of compartment. Containers on the ship are stacked up to 6 levels high due to the limit in weight that the bottom container can withstand from those on top [39]. Each container has a specified location on the ship, and the time required to load an export container by the QC depends on the location where it is stored on the ship. Each vessel is divided into bays; each bay accommodates a row of container stacks. In addition, in the vessel, ship-bays are partitioned, each of which will be covered by one QC [26]. Interferences among QCs are not considered in this study, because in the real case, any two adjacent QCs must be set apart from each other at least one ship-bay to avoid interference. The locations of QCs by which containers are loaded, i.e. the destination of containers, can be considered as known data. The containers can be handled by QCs in any order. The same assumption has been made in the work of Cao, et al. [8]. The detailed QC schedule will then be determined based on the information of container locations on the ship and the synchronization with other handling equipment.

The container storage yard is used to temporarily store containers until they are further picked up by vehicles/trains or loaded onto the ship. A yard may be divided into a number of blocks, where containers are stacked side by side and one on top of each other. Interferences among YCs are not considered here because it requires access to the AGV control mechanisms, which is out of the scope of this study. The width of a block is usually divided into 6 rows of containers. In front of each block are the transfer points for AGVs to interact with YCs. Retrieving and stacking containers within blocks are performed by YCs, which can move among blocks. The export containers usually arrive at the terminal through a period of more than a week before the scheduled loading time, and are assigned to particular slots in the storage yard [20]. Therefore, the locations of export containers in this study are assumed to be pre-determined.

The problem being investigated is minimising the loading time element of berth time of the ship, which is the time duration when all containers have been transferred and placed onto the ship. This loading time duration consists of (1) the retrieving time for containers by YCs at the storage yard; (2) the travelling time of containers from the storage yard to the ship by AGVs; and (3) the handling time for all containers to be placed on the ship by QCs. Because the transportation between the yard side and quay side plays a crucial role in determining the productivity of the terminal [28], special attention is paid to the AGV dispatching problem, in which the sequence and time of containers handled by the AGVs will be determined. Additionally, the problem will also determine the container handling sequences of the YCs during the loading operations, which is very important in improving the overall efficiency of the system [9, 29]. An integrated model is developed in order to address the problems at the same time.

The objective of this paper is to analyse the automated container terminal system and determine a detailed schedule for QCs, AGVs and YCs during container loading process in order to improve the terminal’s operational efficiency.

From the descriptions above, we summarise the assumptions regarded in this study:

1. Only loading operations are studied in this paper.
2. The yard storage locations for export containers are given.
3. The locations of QCs by which containers will be loaded (destinations of containers) are known.
4. Number of export containers, QCs, YCs and AGVs are all fixed.
5. QCs, YCs and AGVs can only take one container at a time.
6. Pooling policy is applied to AGVs, which means AGVs are shared among all the QCs.
7. Travelling times between any two processing locations are known.
8. Traffic congestion of the AGVs on the path is not considered.
9. The interference among YCs and the interference among QCs are also not considered.

3.2 Problem Formulation

In this section, a mixed integer programming (MIP) model is developed for the integrated scheduling problem of container handling equipment during the loading process. As discussed in section 3.1, the objective is to minimise the berth time of the ship. The main operational decisions are to determine the schedules of AGVs, YCs and QCs.

Sets and indices:
- $N$ the set of containers
- $K$ the set of AGVs
- $W$ the set of YCs
- $i, j$ index of containers
- $k$ index of AGVs
- $w$ index of YCs
- $h_i$ handling time of container $i$ by QC
- $t_i$ AGV travel time for container $i$
- $p_i$ serving time of container $i$ by YC
- $s_{ij}$ empty-loaded travel time of AGV between any two containers
- $w_{ij}$ empty-loaded performing time of YC between two container locations
- $M$ a very large number
- $S$ dummy starting job
- $F$ dummy ending job

Decision variables:
- $u_i$: the time QC picks up container $i$ from AGV
- $d_i$: the time YC releases container $i$ onto AGV

In this study, the problems are to decide the sequences of containers for QCs, YCs and AGVs to perform. Let $m$ and $n$ be the indices of sequences, where $m, n \in N^+$ (positive integer). Then we introduce another three decision variables:

- $x_{ink} = \begin{cases} 1, & \text{if } i \text{ is the } m\text{th} \text{ container delivered by AGV } k \\ 0, & \text{otherwise} \end{cases}$, $\forall i \in N, \forall m \in N^+, \forall k \in K$
- $y_{inw} = \begin{cases} 1, & \text{if } i \text{ is the } n\text{th} \text{ container handled by YC } w \\ 0, & \text{otherwise} \end{cases}$, $\forall i \in N, \forall n \in N^+, \forall w \in W$
- $z_{ij} = \begin{cases} 1, & \text{if } i, j \text{ are successive containers by same QC} \\ 0, & \text{otherwise} \end{cases}$, $\forall i, j \in N$

Objective: Minimise berth time $\max_i (u_i + h_i)$

\[
\sum_{i \in N} \sum_{k \in K} x_{ink} = 1, \forall i \in N \quad (1)
\]
\[
\sum_{i \in N} x_{ink} \leq 1, \forall m \in N^+, \forall k \in K 
\]
\[
\sum_{i \in N} x_{ink} \geq \sum_{i \in N} x_{im+1k}, \forall m \in N^+, \forall k \in K 
\]
\[
\sum_{i \in N} y_{inw} = 1, \forall i \in N 
\]
\[
\sum_{i \in N} y_{inw} \leq 1, \forall m \in N^+, \forall w \in W 
\]
\[
\sum_{i \in N} y_{inw} \geq \sum_{i \in N} y_{in+1w}, \forall m \in N^+, \forall w \in W 
\]

\[
\sum_{i \in N} z_{ij} = 1, \forall j \in N 
\]
\[
\sum_{j \in N} z_{ij} = 1, \forall i \in N 
\]
\[
d_i + M(2 - x_{ink} - x_{jm+1k}) \geq u_i + s_{ij}, \forall i \in N \cup S, j \in N \cup F, \forall m \in N^+, \forall k \in K 
\]
\[
d_i + M(2 - y_{imw} - y_{jm+1w}) \geq d_i + w_{ij} + p_i, \forall i \in N \cup S, j \in N \cup F, \forall n \in N^+, \forall w \in W 
\]
\[
u_i + M(1 - z_{ij}) \geq u_i + h_i, \forall i, j \in N 
\]
\[
u_i \geq d_i + t_i, \forall i \in N 
\]
\[
x_{ink}, y_{inw}, z_{ij} \in \{0,1\}, \forall i, j \in N, \forall m, n \in N^+, \forall k \in K, \forall w \in W 
\]

As the objective function is to minimise the berth time, Constraints (1)-(8) are resource constraints. Among these constraints, Constraints (1) – (3) are for AGVs which imply that every container can be delivered by exactly one AGV. Constraints (4) – (6) are for YCs, which force that every container has one predecessor and one successor and served by only one YC. Constraints (7) - (8) are for QCs, which represent that every container has one predecessor and one successor handled by the same QC. Constraint (9) is the time constraint, which force that two containers delivered by the same AGV must be set apart at least a certain travelling time. Constrain (10) represents that the times of two containers handled by the same YC must be set apart at least a certain moving time and handling time. Constrain (11) represents two jobs served consecutively by the QC must be set apart at least the handling time. Constraint (12) gives the relationship between the time a container has been put on an AGV by a YC at yard side and the time this container has been picked up by a QC at the quay side. Constraints (13)-(14) state the restrictions of the decision variables are binary and non-negative.

4 Computational Evaluation

In this section, we will evaluate the proposed MIP model by a number of experiments. We aim to obtain the optimal/near optimal solutions within reasonable time duration. As an NP-hard problem, the computation time for solving such a problem increases exponentially as the problem size grows. We use the following two approaches for the evaluation: (1) Branch and bound (B&B) algorithm: it is the benchmark method and has been applied to solve a variety of problems: quay cranes scheduling problem at container terminal [32], vehicle routing problem [3, 21], problem of general cutting planes [2] and scheduling of rail crane and container deliveries [19], etc. It is well known that CPLEX is capable of finding good solutions very quickly. We implement this algorithm in AIMMS 3.11 with solver CPLEX 12.2 [6]. (2) Genetic algorithm (GA): Genetic algorithm (GA) is a well-known meta-heuristic approach for finding solutions to optimisation problems [13, 17]. Its efficiency is verified by many
problems in various literatures, particularly in container terminal operations [4, 15, 28, 37]. It has been proven that GA-based approaches can solve large-size practical problems with approximately optimal solutions.

We will first introduce the GA design for this problem, and then evaluate and compare the small-size problems by B&B and GA. Due to the computational difficulty for solving large-size problems by using B&B, GA is adopted for such problems and used for further analyse the effects of different factors on the performance of the model and algorithm.

4.1 Genetic algorithm

**Chromosome representation and initialization:** The initial step of the GA design is the chromosome/solution representation. It is known that the structure of the initial population plays an important role in the performance of the GA [13]. Therefore, for our approach, the chromosome representation demonstrates three main decision variables: the schedules for AGVs, YCs and QCs. Figure 2(a) and (b) show a feasible solution, namely chromosome, of the problem by two strings and one matrix that imply the sequences of the handling equipment. There are 6 containers considered in this table. In Figure 2(a), for example, container 3 will be handled by YC 2 and delivered by AGV 2; as in assumption 3, the destinations of containers are known, we assume that container 1, 3, 4 will be covered by QC 1 and container 2, 5, 6 will be handled by QC 2. Note that the actually order of QCs for handling these containers is the decision variable. Figure 2(b) shows one possible solution of the QC handling sequences. For example, QC 1 will handle container 4 first, then container 3 and container 1 in sequence.

<table>
<thead>
<tr>
<th>Container</th>
<th>AGV</th>
<th>YC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 2: (a) an illustration of chromosome representation for the AGV and YC part

<table>
<thead>
<tr>
<th>QC1</th>
<th>QC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

(b) an illustration of chromosome representation for the QC part

**Parents selection strategy:** Parents selection strategy decides how to choose chromosomes in the current population as parents for creating offspring for the next generation. Generally, it is better to choose the best solutions in the current generation to create offspring. The most common method is the 'roulette wheel' sampling, in which each chromosome is assigned a slice of circular roulette wheel and the size of the slice is proportional to the chromosome’s fitness. The wheel spins population-size times. On each spin, the chromosome under the wheel’s maker is selected to be in the pool of parents for the next generation. Here, as the objective is to minimize the berth time, we choose the ones with smaller objective function values (OFVs) as parents for the next generation.

**Genetic operator design:** A genetic operator helps to improve the solution gradually in the evolving process while maintaining the feasibility of the newly generated offspring for the problem. In this respect, crossover and mutation play an important role in the design of genetic algorithm. Here we introduce the following uniform crossover and swap mutation used in this study: (1) uniform crossover: uniform crossover makes every place a potential crossover point to improve global solution search space. This type of crossover operator generates a template binary string of uniformly distributed “1”s and “0”s in the same length of parents. The template string is then mapped to one of the parents, in which the genes that have the same positions with “1”s in the template string are given to a child, and the remaining empty genes of this child are filled from another parent. This crossover can be directly used for the chromosome of AGV and YC part. For the QC part, we need to delete the duplicate genes from another parent and then filled the remaining genes. (2) swap mutation: mutation is used for maintaining the diversity of the population in successive generations and maximizing the exploitation of the solution space. According to the mutation rate, swap mutation is carried out by choosing two positions of the chromosomes, and then swapping the genes on these positions.

**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 OFVs of its parent(s). This approach enables GA to reduce the computation time and evolves 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 criterions as stopping rules: (1) the maximum number of evolving generations allowed for GA. This is a common criterion adopted by many GA based optimisation problem [4, 18, 24]; and (2) the standard deviation of the fitness values of chromosomes (σT) in the current generation is below a small value [39]. This parameter implies the diversity of the current generation in terms of the OFVs. The decreasing σT is equivalent with the decreasing diversity. If σT decreases below a small arbitrary constant ε, then the algorithm is stopped.
4.2 Initial settings

For small-sized problems, results obtained by B&B algorithm and GA are compared in terms of OFV and computation time. Due to the exponential increasing in the computation time when the problem size getting larger, the problem is infeasible to solve by B&B algorithm. Therefore, GA is adopted for solve large-sized problems with providing approximately optimal solutions within 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. Each problem is solved by GA 20 times and the mean of objective function values and computation times are reported.

Parameters settings

(1) The number of containers varies from 5 to 250, where 5-20 are considered as small-sized problems and 30-250 are considered as large-sized problems. We also consider the number of AGVs varies from 2 to 15, the number of YCs varies from 2 to 8 and 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), and the handling times of each YC from each available location to the transfer point follows uniform distribution U (60, 240).

(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 $P_s$: 100; Maximum generations $M_g$: 30.

4.3 Small-size examples

Table 1 shows the results of B&B using AIMMS software and GA for small-size problem. Generally, our proposed GA can obtain approximate optimal solutions in a faster computation speed. Compared with AIMMS results, the average OFV difference of GA is only 1.5%, which is promising result. Although our proposed GA cannot always return the optimal solution, this small gap is acceptable by container terminal managers due to the shorter computation time (planning time). As expected, when increasing the number of containers, i.e. problem size getting larger, the OFV increases accordingly. We also find that AIMMS could not return results for problem with more than 10 containers in a reasonable time due to the exponential trend of the B&B’s computation time.

Table 1: results of computational experiment in small sizes

<table>
<thead>
<tr>
<th>No</th>
<th>Number of Containers</th>
<th>AGVs/QCs/YCs</th>
<th>B&amp;B (MIP)</th>
<th>GA</th>
<th>OFV</th>
<th>OFV Gap rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Computation time (s)</td>
<td>OFV (s)</td>
<td>Computation time (s)</td>
<td>OFV (s)</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2/2/2</td>
<td>2.82</td>
<td>407</td>
<td>1.97</td>
<td>407 0%</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3/2/2</td>
<td>24.37</td>
<td>541</td>
<td>2.88</td>
<td>543 0.3%</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>3/2/2</td>
<td>172.02</td>
<td>604</td>
<td>3.06</td>
<td>612 1.3%</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>2/2/2</td>
<td>183.69</td>
<td>640</td>
<td>3.44</td>
<td>654 2.2%</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>3/2/3</td>
<td>734.09</td>
<td>692</td>
<td>4.03</td>
<td>708 2.3%</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>2/2/2</td>
<td>1604.91</td>
<td>1867</td>
<td>3.98</td>
<td>1898 1.7%</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>2/3/3</td>
<td>/</td>
<td>/</td>
<td>4.32</td>
<td>2342 /</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>2/3/2</td>
<td>/</td>
<td>/</td>
<td>4.61</td>
<td>2985 /</td>
</tr>
</tbody>
</table>

4.4 Large-size experiments

Because it is difficult to obtain solutions for large size problems by using B&B, we proposed a genetic algorithm, which is able to obtain near-optimal solutions in the reasonable computation time. In order to evaluate the performance of the model and GA, we evaluate a series of large-size problems. The number of containers varies from 30 to 250 with different numbers of AGVs, QCs and YCs. Figure 3 shows the typical convergence of the GA approach for a case of 50 containers, 5 AGVs, 3 QCs and 4 YCs. It has a monotonous convergence towards a fixed value, which demonstrates the capability of our proposed GA in finding optimal solutions in large-sized problems. Table 2 shows the evaluation results of GA approach in large-sized problems. These indicate that the proposed approach is reliable to solve problems in different sizes and can be adapted to the real situations. Both the computation times and the OFVs steadily increase as the problems size increases. In most of the experiments, the GA is able to get solutions within a minute. Even for the case with 250 containers, 15 AGVs, 3 QCs and 8 YCs, it only takes 131.13 second to get the solution.

Now we look at the effect of the total number of QCs, AGVs and YCs on the average berth time. A general trend is found that the berth time is improved with the increasing number of available QCs, AGVs and YCs. We also observe that the effect of the number of QCs and YCs are more significant than that of AGVs, which verifies that QCs and YCs are the bottleneck resources in

...
container terminal operations. For example, as the number of AGVs increases, AGVs may wait at the quay crane/yard crane queues and it causes delay of the container handling operations. Thus all the operations must be synchronized with cranes’ schedules.

![Figure 3: the typical convergence of GA in a single run](image)

Table 2: results of computational experiments in large sizes by GA

<table>
<thead>
<tr>
<th>No</th>
<th>Number of Containers</th>
<th>AGVs/QCs/ YCs</th>
<th>Computation time (s)</th>
<th>OFV (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>30</td>
<td>3/3/2</td>
<td>5.95</td>
<td>4773</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>4/3/2</td>
<td>7.72</td>
<td>4377</td>
</tr>
<tr>
<td>11</td>
<td>30</td>
<td>5/3/2</td>
<td>12.17</td>
<td>4444</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
<td>4/3/2</td>
<td>8.28</td>
<td>5327</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>4/3/3</td>
<td>10.19</td>
<td>4515</td>
</tr>
<tr>
<td>14</td>
<td>40</td>
<td>4/3/4</td>
<td>15.14</td>
<td>3573</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>5/2/4</td>
<td>7.91</td>
<td>4136</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>5/2/5</td>
<td>19.38</td>
<td>3423</td>
</tr>
<tr>
<td>17</td>
<td>50</td>
<td>5/3/5</td>
<td>14.70</td>
<td>3138</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>6/3/6</td>
<td>27.33</td>
<td>6352</td>
</tr>
<tr>
<td>19</td>
<td>100</td>
<td>8/3/6</td>
<td>41.06</td>
<td>6099</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>10/3/6</td>
<td>45.05</td>
<td>6028</td>
</tr>
<tr>
<td>21</td>
<td>150</td>
<td>10/3/5</td>
<td>74.92</td>
<td>9794</td>
</tr>
<tr>
<td>22</td>
<td>150</td>
<td>10/3/6</td>
<td>54.05</td>
<td>8822</td>
</tr>
<tr>
<td>23</td>
<td>150</td>
<td>10/3/7</td>
<td>48.17</td>
<td>8338</td>
</tr>
<tr>
<td>24</td>
<td>200</td>
<td>10/3/6</td>
<td>89.81</td>
<td>11768</td>
</tr>
<tr>
<td>25</td>
<td>200</td>
<td>15/3/6</td>
<td>66.23</td>
<td>10196</td>
</tr>
<tr>
<td>26</td>
<td>200</td>
<td>10/3/8</td>
<td>60.73</td>
<td>9716</td>
</tr>
<tr>
<td>27</td>
<td>250</td>
<td>10/3/4</td>
<td>134.06</td>
<td>21402</td>
</tr>
<tr>
<td>28</td>
<td>250</td>
<td>10/3/6</td>
<td>107.56</td>
<td>15508</td>
</tr>
<tr>
<td>29</td>
<td>250</td>
<td>10/3/8</td>
<td>109.05</td>
<td>12284</td>
</tr>
<tr>
<td>30</td>
<td>250</td>
<td>15/3/8</td>
<td>131.13</td>
<td>11553</td>
</tr>
</tbody>
</table>

5 Conclusion

This paper addresses an integrated decision problem in the automated container terminal, determining the schedules of different handling equipment simultaneously. This problem is formulated as a mixed integer programming (MIP) model to describe the relationship between the schedules of all the equipment. The objective of the problem is to minimize the berth time of the ship. Due to the computational difficulty, existing optimisation software cannot obtain optimal solutions. A genetic algorithm is designed to solve the problem. The main contributions of this study are to provide the integrated approach for the problem and to develop an efficient genetic algorithm (GA) to find solutions.

From the computational experiments, we observe that the proposed integrated approach can highly improve the operational efficiency of container terminal. Compared with the results from AIMMS software, the proposed GA can obtain near optimal solutions for small-size problems, with the average differences accuracy of 1.5%. In addition, the proposed GA converges fast. For the large-size problems, it can provide approximate solutions within minutes.

Future work can consider a broader integration, i.e. consider the unloading and loading operations simultaneously. Because there exist inherent interrelationship between the schedules of AGVs, QCs and YCs, it is valuable to find the optimal number of AGVs to synchronize the cranes’ schedule. Furthermore, investigating other meta-heuristic techniques might improve the results obtained from this paper.

References


