

A Dynamic Truck Dispatching Problem in Marine Container Terminal

Jianjun Chen^{*†}, Ruibin Bai^{*‡} (IEEE senior member), Haibo Dong^{*§}, Rong Qu^{||} and Graham Kendall^{**}

^{*} Division of Computer Science, University of Nottingham Ningbo, China

[†] Email: jianjun.chen1@nottingham.edu.cn

[‡] Email: ruibin.bai@nottingham.edu.cn

[§] Email: Nick.Dong@nottingham.edu.cn

^{||} School of Computer Science, University of Nottingham

Email: rxq@cs.nott.ac.uk

^{**} School of Computer Science, University of Nottingham, Nottingham, UK and Semenyih, Malaysia

Email: graham.kendall@nottingham.ac.uk

Abstract—In this paper, a dynamic truck dispatching problem of a marine container terminal is described and discussed. In this problem, a few containers, encoded as work instructions, need to be transferred between yard blocks and vessels by a fleet of trucks. Both the yard blocks and the quay are equipped with cranes to support loading/unloading operations. In order to service more vessels, any unnecessary idle time between quay crane (QC) operations need to be minimised to speed up the container transfer process. Due to the unpredictable port situations that can affect routing plans and the short calculation time allowed to generate one, static solution methods are not suitable for this problem. In this paper, we introduce a new mathematical model that minimises both the QC makespan and the truck travelling time. Three dynamic heuristics are proposed and a genetic algorithm hyperheuristic (GAHH) under development is also described. Experiment results show promising capabilities the GAHH may offer.

I. INTRODUCTION

This paper discusses a truck dispatching problem that involves coordinating a fleet of trucks inside a marine container terminal to satisfy container transshipment requests. The container terminal is divided into a quay area and a yard area. In each day, the terminal is visited by several vessels with many containers to be loaded and/or unloaded. In order to support container transfer operations, a total of 20 quay cranes (QCs) are installed across the quay area. In the problem concerned in this paper, these QCs are deployed over a rail, allowing each of them to relocate to a different bay location once its currently assigned jobs are completed. The yard area consists of many yard blocks equipped with yard cranes (YCs), which loads and unloads containers between container stows and trucks. The map for the container terminal is presented in Figure 1. The QCs are located at the top of the map and the coloured squares represent yard blocks.

Upon the arrival of a vessel, the bays that need to be operated are revealed to the coordinators of the company, who then assign several QCs to do these tasks. The vessel will leave the dock once all associated transshipment tasks are finished. Each transshipment task is digitally encoded as a *work instruction (WI)*, which contains the task identifier,

the source crane, the secondary source crane, the destination crane, the container size and the container weight. The crane operation time refers to the time it takes to move a container from a container stow to a truck or the time it takes to move a container from a truck to a stow. A WI with a secondary source crane involves 2 containers that need to be picked up from 2 different yard blocks and delivered to one vessel. When a truck driver receives such a WI, he is allowed to visit these two yards blocks in any order, although the source crane of the WI is usually visited first. In any other cases, the secondary source crane field is empty. In general, the process of completing a WI is described as follows:

- The truck first heads to the source crane.
- At the source crane, the container specified in the WI will be lifted onto the truck by the crane operator.
- If a secondary source crane exists, the truck will head to the secondary source crane and load the container.
- The container(s) are carried to the destination crane and unloaded.

The WIs are grouped as *work instruction lists (WLists)*, which serve as virtual task lists for QCs. The sequence of the WIs inside a WList is determined based on the adjacency of container bays and the locations of containers in each bay. Each QC can do one WList at a time. All WIs should be dispatched in the same sequence as the WList. The WI sequence must be enforced even when containers are being transferred to vessels. This is to make sure that container operations at successive ports of unload/load are convenient. After a QC finishes all WIs in a list, it is either assigned with a new list or wait for other arrangements from coordinators. The reassignment decisions are based on the strategy adopted by the coordinators. Table I gives an example of WList with both containers loaded from and unloaded to a vessel. In the table, the instruction source and destination names beginning with “CR” are QCs.

The dispatching orders are sent to truck drivers via handheld devices. After WLists have been created, the WIs will be

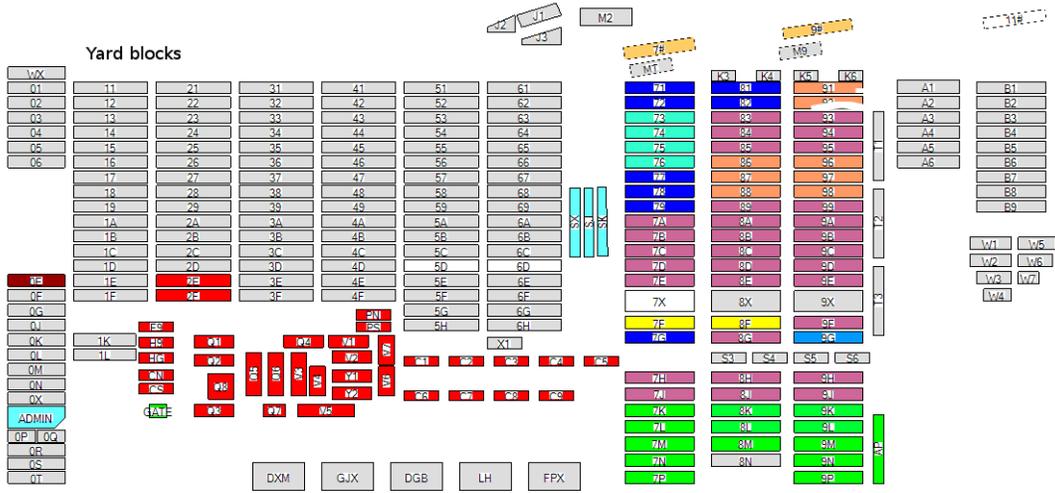


Fig. 1. Map of the container terminal.

id	source	2 nd source	destination	TEUs ¹	weight (ton)
0	7E	1F	CR1	2	26
1	2D	none	CR1	1	17
2	CR1	none	12	1	15
3	CR1	none	12	1	18

TABLE I

AN EXAMPLE WORK INSTRUCTION LIST WITH 4 WORK INSTRUCTIONS.

sent to idle truck drivers via a central message server. A truck driver will only receive a new WI when he finishes the previously assigned WI. The dispatching decisions are made at the moment when the truck driver requests a WI.

The dispatching algorithm currently adopted by the company is simple and straight forward. Once WLists have been decided and assigned to QCs, the coordinators will then estimate the number of trucks required to ensure continuous operations for each QC. The number is based on the current traffic condition of the port as well as the crane operators' performance. However, the truck assignment is fixed and there is no flexible truck dispatching mechanism that allows a truck to service multiple QCs.

In order to be competitive in the market, the company is focusing on improving the efficiency of QCs and trucks. The efficiency of a QC is measured by the amount of time the QC takes to finish all assigned WIs, which is also referred as the *QC makespan*. Thus, the sum of all QC makespan is referred as the *total QC makespan*. The total QC makespan is affected by the proficiency of crane operators and the number of trucks available. A proficient crane operator can finish more container operations within the same amount of time. While sufficient supply of trucks ensures that less waiting time will occur between each two of container operations.

One can infer that, if a QC has a longer list of WIs than the

other QCs, its operation speed will heavily affect the vessel departure time. While this is true in a static environment where WLists remain unchanged until all WIs are completed, in a real life environment, it is not wise to focus on the completion speed of a single WList based on its length because these lists are not fixed throughout its lifespan. Given the fact that long WLists usually involve more container bays than other WLists, the coordinator may split the workload at some point so that free QCs can help finish these WIs. It is also possible that a QC is interrupted by the coordinator to do a different WList in the middle of time. These re-assignment are not necessarily apparent beforehand as the decisions are made according to high level strategies and cannot be described as fixed patterns. As a result, it is important to minimise the total QC makespan even when one or more of these QCs are assigned with a long WList.

The efficiency of trucks is measured by the travelling time needed to finish all WIs. Although shorter travelling time is beneficial to the company's income and the environment, it sometimes conflicts with the goal of minimising QC makespan. As a result, it is considered as the secondary objective in this problem.

II. LITERATURE REVIEW

There are plenty of researches revolving around marine container terminal operations in the literature. These research efforts generally contribute to three main parts, which are the optimisation of QCs, YCs and trucks. For details of these problems and beyond, one can go through the reviews by [1]–[7].

The optimisation of QC operations leads to a higher work rate at the quay side of the port. The first research in this area was done by [8], which investigated various QC scheduling principles and their impact on the port throughput. In the work

of [9], the QC scheduling was solved as an m-parallel machine scheduling problem that tries to minimise the makespan of QC jobs. The study of [10] focused on optimising the QC operations on a single vessel so that the vessel turnaround time is minimised. [11] studied a similar problem but the interference between QCs was also considered. [12] attempted to solve a QCs scheduling problem that has special constraints to avoid traffic congestion in yard. The relationship between QC operations and yard storage locations was researched by [13]. [14] focused on the relationship between QC scheduling and berth allocation and the authors provided a comprehensive review on this problem.

The YC deployment problems attempt to optimise yard crane assignment as well as yard crane movement between yard blocks. In the problem tackled by [15], the routing of a single yard crane was solved. A mixed integer programming model was proposed to maximise the efficiency of loading operations. This was later extended by the work of [16], which considered a straddle crane. Another research involving single yard crane scheduling with different job ready time was done by [17]. The objective is to minimise total job waiting time. Later, [18] attempted to address a multi-YC version of this problem with a two-phase heuristic. The first stage breaks the problem into multiple independent YC scheduling problems based on the yard zone. The second stage applies a reassignment procedure to improve the schedule. [19] considered a problem that has two YCs serving one QC to minimise the total transshipment time. [20] proposed a Lagrangian relaxation method to solve a dynamic rubber-tired gantry cranes deployment problem, which aims to minimise the work delay. Other researches considering multiple YCs can be found in the work by [21], [22].

The last major part involves optimising the dispatching of trucks. [23] researched a problem that involves determining the storage locations of unloaded containers and the scheduling of QC loading and unloading times. The discharging vessel is serviced by a single QC and a fleet of vehicles. This research was later extended by [24], which considered multiple QCs with a shared truck pool. [25] investigated a look-ahead dispatching method for automated guided vehicles in an automated port. In this problem, the QC waiting time and vehicle travelling time need to be minimised. A mixed integer model was proposed for the static problem and a heuristic was developed for the dynamic problem. [26] addressed a truck fleet scheduling problem that has sequence-dependent transshipment jobs with different ready times. A genetic algorithm was developed to minimise the makespan. [27] applied a heuristic algorithm to a dispatching problem for automated lifting vehicles. In this problem, each vehicle is capable of lifting one container. Container jobs involve transshipments between yards and ships in both directions. Apart from the above studies, some researchers tried to integrate the scheduling of trucks with QCs (see [28]) or YCs (see [29]) or all cranes (see [30]).

Compared to the previous studies, our research has the following contributions:

- A mathematical model that not only considers the min-

imisation of the QC makespan, but also tries to reduce the total travelling time of trucks is proposed.

- We also developed three dynamic heuristics that are able to respond very quickly to the dispatching requests. The solution quality is shown to be better than the company's method.
- This is the first time a learning based method has been considered to solve dynamic truck dispatching problem in a marine container terminal.

III. PROBLEM FORMULATION

This section introduces the mathematical model, whose main objective is to maximise the efficiency of QCs and trucks. Consider a graph $G = (N, A)$ where N refers to the set of cranes in the terminal and $A = \{(i, j) | i, j \in N\}$ is the set of arcs connecting these cranes. L is the WList set with $|L|$ lists. The l^{th} list contains $|l|$ WIs and the w^{th} WI in the l^{th} list is represented by (l, w) . The cranes associated with WI (l, w) are represented by the following symbols:

- $\alpha(l, w)$ refers to the source crane.
- $\beta(l, w)$ refers to the secondary source crane. Binary variable $m_{l,w} = 1$ indicates that WI (l, w) has a secondary source crane. Otherwise its value will be 0.
- $\gamma(l, w)$ refers to the destination crane.
- $\theta(l, w)$ refers to the QC of the WI, which can either be a source crane or a destination crane.

The travelling time between two cranes is indicated by $t[i, j]$, where $i, j \in N$. For example, $t[\alpha(l, w), \gamma(l, w)]$ is the travelling time from the first source crane of (l, w) to its destination crane. The crane operation durations at source crane, secondary source crane, destination crane and QC of (l, w) are represented by $d_{l,w}^\alpha, d_{l,w}^\beta, d_{l,w}^\gamma$ and $d_{l,w}^\theta$. Similarly, the crane operation begin times are $s_{l,w}^\alpha, s_{l,w}^\beta, s_{l,w}^\gamma$ and $s_{l,w}^\theta$. Note that these crane operation begin times are not considered as decision variables because these operations are done in first-come-first-serve fashion. There is no reason and it is not really possible in real world to deliberately postpone the crane operation begin times in a controlled way. The truck fleet is represented by set V with a cardinality of $|V|$. The initial position of truck v is b_v . The binary decision variable $\delta_{l,w}^{l',w'}$ is 1 if (l, w) is assigned to the truck that just finished the (l', w') and is 0 otherwise. The objective function, shown in Equation 1, consists of two sub-goals: minimising total QC makespan and minimising total travelling time. Note that the two sub-goals may sometimes conflict with each other. As a result, we introduce two weight variables W_1 and W_2 , which be adjusted according to the requirements of the container terminal. In this problem, the value of W_1 should be significantly larger than the value of W_2 because the minimisation of QC makespan is the most important objective.

Constraints 2 and 3 ensure the crane visit precedence of WIs that have the secondary source crane. Constraint 4 ensures the crane visit precedence when no secondary source crane is involved. The precedence of QC operations within a WList is enforced by constraints 5 and 6. Constraint 5 indicates that the

$$\text{Minimise} \quad W_1 \left(\sum_{l \in L} (s_{l,|l|}^\gamma + d_{l,|l|}^\gamma) \right) + W_2 \left(\sum_{(l,w) \in I} \sum_{(l',w') \in I} \delta_{l,w}^{l',w'} t[\gamma(l,w), \alpha(l',w')] \right) + \sum_{v \in V} \sum_{(l,w) \in I} \delta_{b_v}^{l,w} t[b_v, \alpha(l,w)] \quad (1)$$

Subject to

$$m_{l,m} (s_{l,w}^\alpha + d_{l,w}^\alpha + t[\alpha(l,w), \beta(l,w)]) < s_{l,w}^\beta \quad \forall (l,w) \in I. \quad (2)$$

$$m_{l,m} (s_{l,w}^\beta + d_{l,w}^\beta + t[\beta(l,w), \gamma(l,w)]) < s_{l,w}^\gamma \quad \forall (l,w) \in I. \quad (3)$$

$$s_{l,w}^\alpha + d_{l,w}^\alpha + m_{l,m} (t[\alpha(l,w), \beta(l,w)] + t[\beta(l,w), \gamma(l,w)]) + (m_{l,m} - 1) t[\alpha(l,w), \gamma(l,w)] \leq s_{l,w}^\gamma \quad \forall (l,w) \in I. \quad (4)$$

$$s_{l,w}^\theta > d_{l,w-1}^\theta + s_{l,w-1}^\theta \quad \forall (l,w) \in L, w \neq 0. \quad (5)$$

$$s_{l,0}^\alpha > t[b_v, \alpha(l,0)] \delta_{l,w}^{b_v} \quad \forall l < |L|, v \in V. \quad (6)$$

$$\sum_{(l',w') \in L} \delta_{l,w}^{l',w'} = 1 \quad \forall (l,w) \in L. \quad (7)$$

$$\sum_{(l',w') \in L} \delta_{l,w}^{l',w'} = 1 \quad \forall (l,w) \in L. \quad (8)$$

$$\sum_{v \in V} \sum_{l \in L} \delta_{l,0}^{b_v} \geq 1 \quad (9)$$

$$\sum_{v \in V} \sum_{l \in L} \delta_{l,0}^{b_v} \leq |V| \quad (10)$$

order of QC operations for WIs is the same as the order of WLists, while constraint 6 is to give $s_{l,0}^\alpha$ a precise definition so that the WIs assigned to a truck are done in sequence. The incoming flow and outgoing flow are balanced by constraints 7 and 8. They also ensure that each WI is assigned to exactly one truck. Finally, constraints 9 and 10 enforce that the number of trucks used is at least one and is not more than the fleet size.

IV. METHODOLOGIES

This section describes the design of our heuristics as well as the hyperheuristic being developed. We investigated both dynamic and static methodologies for this problem. While generating a static dispatching plan that assigns a series of working instructions to each truck is possible, the traffic condition, QC operation speed and the QC task assignment may change, which will force the plan to be adjusted frequently and cause numerous management problems to the company. The detailed list of limitations of a static solution approach is shown below:

- The WLists may change at some point. Some WIs may be postponed while at the same time other WIs may be prioritised.
- The traffic condition changes unpredictably because some trucks from outside may enter and leave the terminal. It is likely that a routing plan will have to be changed in the middle of the process. However, the computational time allowed is very short (in seconds).
- The crane operation duration of a certain container is hard to determine precisely beforehand because it depends not only on the proficiency of operators, but also on any

additional operations required. Sometimes container shuffling operations are required before the crane can directly operate on the desired container. From the historical data obtained from the company, a crane operation may take from 30 seconds to 3 minutes.

- It is also common that a truck driver is asked to do another WI in the middle of deadheading², if such decision is found to be necessary by the coordinator. As a result, the whole routing plan may have to be changed dramatically in order to ensure the completion order of WIs.
- After some discussions with the company, we found that the dispatching order should be generated within 2 seconds. Due to the short running time available and the fact that it already takes more than 1 second to obtain required container terminal data from the coordination system, the dispatching decisions have to be made in less than 1 second.

The algorithm is implemented and embedded in a dispatching module, which is invoked whenever new WIs are imported or when a truck completes its WI. Each algorithm run only decides the WI for one truck. The dispatching module consists of the following logical steps:

1. The module first retrieves the data from the terminal management system of the company.
2. The data is then processed by a feature evaluation module that converts the obtained terminal status into the information needed by the algorithms.
3. The algorithm then evaluates the feature object and

²When a truck is heading to a location without carrying any container, it's status is called deadheading

	CR01	CR02	CR03
supply	4	3	3
demand	9	9	8
difference	-5	-6	-5

TABLE II
AN EXAMPLE OF AN EVALUATED TERMINAL STATUS.

returns the best WI found.

Due to the scope of this paper, the first logical step is not explained. We will introduce the second step in Section IV-A and the third step in Sections IV-B and IV-C.

A. Feature Evaluation

Before dispatching decisions are made, the current status of the terminal is evaluated first to provide an estimation on the quantity of trucks needed for each QC to ensure continuous operations. In the evaluation process, the following properties are calculated for each QC:

- The number of truck arrivals within a predefined time window.
- The number of trucks needed for each QC within a predefined time window.

To calculate the number of truck arrivals for each QC, the evaluation module first retrieves the locations of trucks that are currently assigned with WIs. Then, based on the crane locations described in WIs, it estimates the QC arrival time of that truck. If the truck has not yet visited the QC, the count of truck arrivals at the corresponding QC will be increased by one.

The second property can be directly obtained using the remaining number of WIs in a list, which is $|l|$ as described in our formulation in Section III. However, this method fails to reveal any temporal information, which can be critical in deciding the true priority of a certain QC. When two crane operators with different operation speeds have the same $|l|$, it is important to supply trucks more frequently to the operator with a faster speed. By limiting the evaluation to a predefined time window, it is then possible to take the crane speed into consideration because slower QCs will have less truck demands within the same period than faster ones. The calculation method is shown in Equation 11. Due to the difficulty in estimating the crane operation speed precisely, we use the estimated speed obtained from the historical crane operation data. Table II describes an example of the feature with three QCs (WLists) involved. The way features are utilised will be discussed in Sections IV-B and IV-C.

$$\text{trucks needed} = \min(|l|, \text{time window length} \div \text{crane speed}) \quad (11)$$

B. Heuristic Methods

This section describes the three dispatching heuristics implemented. These heuristics share a similar structure shown in Algorithm 1. Each heuristic has a different strategy as listed below:

1. The QC with the smallest supply-demand difference (examples are shown in Table II) is chosen. If two QCs have the same difference value, the WI with the closest distance from the truck's current position to the source crane is chosen.
2. The WI with the closest source crane is chosen.
3. The WI with the shortest distance to destination crane is chosen. Note that the routine covers all intermediate cranes of the WI.

Algorithm 1 Common structure of proposed heuristics.

Require: Truck t , WList L
for the first WI $(l, 0)$ in L **do**
 evaluate the fitness of $(l, 0)$ based on the strategy of heuristic.
 if the current WI is the best found so far. **then**
 $l_{best} = l;$
 end if
end for
return $(l_{best}, 0)$

The first heuristic aims to minimise the QC makespan. By subtracting the number of trucks needed with the number of truck arrivals, it is possible to find out which QC needs most trucks to ensure its continuous operations. For the example in Table II, the first WI³ of the QC named ‘‘CR02’’ will be sent to the truck. Currently, this heuristic is used in the dispatching module as it minimises the QC makespan.

The second and the third heuristics are implemented to minimise the total travelling time. They are designed to be used under the circumstances when the continuous operations of cranes is already ensured, which is likely to happen when there are many trucks available and the QCs have a lot of trucks queueing for loading/unloading containers. Although these two heuristics have not been embedded in the dispatching module yet, they will be included in the framework of the genetic algorithm hyperheuristic (GAHH) in the future and become part of the dispatching module. Through the learning process of GAHH, it is possible to find out the situations where these two heuristics can lead to better results.

C. The Genetic Algorithm Hyperheuristic

We introduce a genetic algorithm hyperheuristic (GAHH) for this problem, which adopts a similar framework proposed in [31]. The underlying idea of the hyperheuristic is to find strategies of invoking/creating lower level heuristics so that their combined performance is good across a set of problems. Our GAHH, whose main procedure is shown in Algorithm 2, is developed to find a set of invocation rules for the three heuristics we proposed based on the feature obtained from the terminal status.

The chromosome in the GAHH consists of a set of blocks. Each block has a feature vector that reflects the status of a

³Since the order of WLists must be followed, choosing a WList is essentially choosing the first WI of this WList.

Algorithm 2 The main structure of the GAHH.

Require: Problem instance set Π
 Initialise a population Pop with n chromosomes.
 Solve Π with these initial chromosomes and obtain their average performance based on the objective function.
while Stopping criteria is not met **do**
 $Pop \leftarrow$ Cross-over chromosomes and create $2n$ new chromosomes.
 for Each chromosome Chr in population Pop **do**
 Solve Π and obtain its average performance.
 end for
 Select n best chromosomes from the new chromosomes to replace Pop .
end while
return The best chromosome found so far.

container terminal during different lengths of evaluation time window, and the corresponding dispatching heuristic that will be invoked. Below shows an example of a chromosome with 2 blocks. The letters “a” and “b” refer to two different terminal statuses. Each status is evaluated using three different time windows. Since the GAHH is still under development, the number of time windows is not final and can be changed through experiments.

a(5 min); a(10 min); a(15 min) --> 1
 b(5 min); b(10 min); b(15 min) --> 3

Once the final chromosome is returned by the GAHH and its performance is verified on a separate set of problem instance, it will be used as the algorithm invocation rule in the dispatching module. However, the feature evaluated from real life is still likely to be different with any blocks in the chromosome. This is due to the vast amount of possibilities the terminal status can have. To solve this issue, a distance function is used to find the most promising algorithm. The distance function compares the similarity between the feature vector in each block and the feature vector from the current terminal status. Each comparison is measured by a score and the block with the best score will be used.

In order to make this distance function work, two issues must be addressed. The first issue is the accuracy of the similarity comparison. The second issue is the number of blocks in chromosomes. A small number of blocks may not be sufficient to cover certain possibilities of the port statuses, which causes the invocation of a wrong algorithm.

V. PROBLEM INSTANCES AND EXPERIMENT RESULTS

This section first describes the problem instance set that has been generated. Then the comparative results between the company’s method and the insertion heuristic that minimises the total QC makespan are discussed. Due to the focus of minimising QC makespan, the other two heuristics are not included in the experiments.

To test the algorithms under various situations, a total of 16 instances have been generated. They are named in format:

instance	t.time	improve	op.duration	improve
20-1500;1500-30-1:1	979097	11.84%	632507	10.01%
20-1000;2000-25-1:2	999830	13.65%	1370582	15.95%
20-2000;1000-25-2:1	854125	15.59%	1410946	12.17%
20-2000;1000-50-3:2	1304771	12.05%	1631321	13.85%
10-202;198-10-1:1	101878	9.11%	55255	0.31%
10-300;100-10-1:1	128986	5.86%	59363	0.54%
10-100;300-10-1:1	121206	2.85%	53725	2.34%
10-500;500-10-1:1	294550	15.79%	139958	0.56%
10-751;498-10-1:1	272428	3.33%	142096	-0.22%
10-250;751-10-1:1	276822	7.57%	136618	0.47%
6-51;49-10-1:1	40848	0.00%	14948	1.63%
6-75;25-10-1:1	36878	0.00%	15649	5.71%
6-25;75-10-1:1	32185	0.00%	14660	0.96%
6-149;151-10-1:1	83003	4.22%	39247	1.10%
6-75;225-10-1:1	92451	2.67%	41202	3.49%
6-225;75-10-1:1	108092	4.46%	41620	3.19%
Gap: 4.46%			Gap: 11.84%	

TABLE III
THE RESULTS OBTAINED USING 80 TRUCKS

instance	t.time	improve	op.duration	improve
20-1500;1500-30-1:1	981364	11.74%	588097	9.25%
20-1000;2000-25-1:2	1004051	13.12%	1395393	13.11%
20-2000;1000-25-2:1	858395	15.08%	1405567	11.61%
20-2000;1000-50-3:2	1309938	11.61%	1656131	9.97%
10-202;198-10-1:1	105493	6.83%	53919	0.80%
10-300;100-10-1:1	130378	5.33%	58237	1.34%
10-100;300-10-1:1	121646	2.25%	52818	3.97%
10-500;500-10-1:1	295521	15.41%	135232	1.09%
10-751;498-10-1:1	273484	3.37%	138248	0.30%
10-250;751-10-1:1	276392	7.56%	134947	-0.24%
6-51;49-10-1:1	40848	0.00%	14948	1.63%
6-75;25-10-1:1	36878	0.00%	15649	5.52%
6-25;75-10-1:1	32185	0.00%	14660	0.96%
6-149;151-10-1:1	83135	5.22%	39247	2.21%
6-75;225-10-1:1	93009	1.29%	41202	4.75%
6-225;75-10-1:1	108767	3.97%	41620	3.65%
Gap: 10.97%			Gap: 9.96%	

TABLE IV
THE RESULTS OBTAINED USING 100 TRUCKS

[C]-[L:U]-[Y]-[YL:YU]. Each symbol in the format holds one of the instance generation criteria shown below:

- C is the number of QCs involved in the problem instance.
- L and U indicate the number of container loads (L) versus container unloads (U) at the quay area.
- Y is the number of YCs involved in the problem instance.
- YL and YU is the ratio of the number of YCs dedicated to vessel load versus the number of YCs dedicated to vessel unload.

Two sets of results, shown in Tables III and IV, are obtained by running the algorithm with different number of trucks. The result set using 80 trucks is more realistic as the fleet size is the same as the company’s. An additional set of experiments using 100 trucks aims to discover what will happen if the company decides to expand its truck fleet. The first column of these two tables shows the instance name. The second column lists the total travelling time of trucks expressed in seconds. Its improvement over company’s method is shown in column three. The total QC makespan obtained by our heuristic (in seconds) and its improvement over company’s method are

shown in columns 4 and 5 respectively. The final row of each table shows the gaps between the company's method and our heuristic, which are calculated as:

$$\frac{\text{company} - \text{algorithm}}{\text{algorithm}} \times 100\%$$

Both result sets indicate increased overall QC performance over the company's dispatching strategy. For the results obtained with 80 trucks, the overall improvement reaches 11.84%. If we inspect individual results of instances, it can be found that the heuristic yields significantly more performance boost when the problem involves more cranes and WIs. The overall improvement is slightly less when 20 more trucks are available for dispatching, which is reasonable because increasing the fleet size helps to reduce the crane idle time.

While our heuristic does not focus on reducing the travelling time, its improvement over the company's strategy is still noticeable. The gap of total travelling time increases to 10.97% when 100 trucks are involved. We believe that the capability of sharing trucks among QCs leads to such improvement. As the fleet size increases, it gets easier to keep QCs working because the truck supply per QC is increased. As a result, the strategy can be changed to focus more on reducing the travelling time.

VI. CONCLUSION AND FUTURE RESEARCH

In this paper, we discussed a dynamic truck dispatching problem in a marine container terminal. A mathematical formulation for the static version of the problem is introduced, which allows flexible tuning of the focus between the minimisation of total QC makespan and the minimisation of truck travelling time. Three heuristics have been developed for the dynamic version of the problem. Experiments are carried out on the heuristic that minimises the total QC makespan. The results show that our method leads to 11.84% improvement over the strategy currently adopted by the company. Since these heuristics will be integrated into the GAHH being developed, the performance of the GAHH seems promising.

Apart from the better solution quality the GAHH may bring, it can also help to answer the following questions:

- How many trucks per QC is enough to ensure continuous crane operations?
- How does the above number change with the travelling time between cranes and the crane operation speed?

If the above questions are answered, the knowledge will be useful for deciding the suitable fleet size for a new container terminal.

Currently, our model assumes that a QC is idle once all its assigned WIs have been completed. In real-life situations, however, the QC constantly move to new locations to serve other container blocks. This is an important feature that worth exploring as it leads to a more automatic terminal management system.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. 71471092) and the Ningbo Science and Technology Bureau (Grant No. 2014A35006).

REFERENCES

- [1] M. Caserta, S. Schwarze, and S. Voß, *Container Rehandling at Maritime Container Terminals*. New York, NY: Springer New York, 2011, pp. 247–269.
- [2] D. Steenken, S. Voß, and R. Stahlbock, "Container terminal operation and operations research - A classification and literature review," *OR Spectrum*, vol. 26, no. 1, pp. 3–49, 2004.
- [3] H.-O. Günther and K.-H. Kim, "Container terminals and terminal operations," *OR Spectrum*, vol. 28, no. 4, pp. 437–445, 2006.
- [4] J. Lehnfeld and S. Knust, "Loading, unloading and premarshalling of stacks in storage areas: Survey and classification," *European Journal of Operational Research*, vol. 239, no. 2, pp. 297–312, 2014.
- [5] H. Rashidi and E. P. K. Tsang, "Novel constraints satisfaction models for optimization problems in container terminals," *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 3601–3634, 2013.
- [6] H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen, "Transport operations in container terminals: Literature overview, trends, research directions and classification scheme," *European Journal of Operational Research*, vol. 236, no. 1, pp. 1–13, 2014.
- [7] I. F. A. Vis and R. De Koster, "Transshipment of containers at a container terminal: An overview," *European Journal of Operational Research*, vol. 147, no. 1, pp. 1–16, 2003.
- [8] C. F. Daganzo, "The crane scheduling problem," *Transportation Research Part B*, vol. 23, no. 3, pp. 159–175, 1989.
- [9] K. H. Kim and Y. M. Park, "A crane scheduling method for port container terminals," *European Journal of Operational Research*, vol. 156, no. 3, pp. 752–768, 2004.
- [10] W. C. Ng and K. L. Mak, "Quay crane scheduling in container terminals," *Engineering Optimization*, vol. 38, no. 6, pp. 723–737, 2006.
- [11] D. H. Lee, H. Q. Wang, and L. Miao, "Quay crane scheduling with non-interference constraints in port container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 1, pp. 124–135, 2008.
- [12] S. Choo, D. Klabjan, and D. Simchi-Levi, "Multiship crane sequencing with yard congestion constraints," *Transportation Science*, vol. 44, no. 1, pp. 98–115, 2010.
- [13] A. Wong and E. Kozan, "Optimization of container process at seaport terminals," *Journal of the Operational Research Society*, vol. 61, no. 4, pp. 658–665, 2010.
- [14] C. Bierwirth and F. Meisel, "A follow-up survey of berth allocation and quay crane scheduling problems in container terminals," *European Journal of Operational Research*, vol. 244, no. 3, pp. 675–689, 2015.
- [15] K. H. Kim and K. Y. Kim, "An Optimal Routing Algorithm for a Transfer Crane in Port Container Terminals," *Transportation Science*, vol. 33, no. 1, pp. 17–33, 1999.
- [16] K. Young Kim and K. Hwan Kim, "Routing algorithm for a single straddle carrier to load export containers onto a containership," *International Journal of Production Economics*, vol. 59, no. 1, pp. 425–433, 1999.
- [17] W. C. Ng and K. L. Mak, "Yard crane scheduling in port container terminals," *Applied Mathematical Modelling*, vol. 29, no. 3, pp. 263–276, 2005.
- [18] W. C. Ng, "Crane scheduling in container yards with inter-crane interference," *European Journal of Operational Research*, vol. 164, no. 1, pp. 64–78, 2005.
- [19] D. H. Lee, Z. Cao, and Q. Meng, "Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm," *International Journal of Production Economics*, vol. 107, no. 1, pp. 115–124, 2007.
- [20] C. Zhang, Y. wah Wan, J. Liu, and R. J. Linn, "Dynamic crane deployment in container storage yards," *Transportation Research Part B: Methodological*, vol. 36, no. 6, pp. 537–555, 2002.
- [21] S. H. Jung and K. H. Kim, "Load scheduling for multiple quay cranes in port container terminals," *Journal of Intelligent Manufacturing*, vol. 17, no. 4, pp. 479–492, 2006.

- [22] M. E. H. Petering and K. G. Murty, "Effect of block length and yard crane deployment systems on overall performance at a seaport container transshipment terminal," *Computers and Operations Research*, vol. 36, no. 5, pp. 1711–1725, 2009.
- [23] E. K. Bish, T.-Y. Leong, C.-L. Li, J. W. C. Ng, and D. Simchi-Levi, "Analysis of a new vehicle scheduling and location problem," *Naval Research Logistics*, vol. 48, no. 5, pp. 363–385, 2001.
- [24] E. K. Bish, "A multiple-crane-constrained scheduling problem in a container terminal," *European Journal of Operational Research*, vol. 144, no. 1, pp. 83–107, 2003.
- [25] K. H. Kim and J. Bae, "A Look-Ahead Dispatching Method for Automated Guided Vehicles in Automated Port Container Terminals," *Transportation Science*, vol. 38, no. 2, pp. 224–234, may 2004.
- [26] W. C. Ng, K. L. Mak, and Y. X. Zhang, "Scheduling trucks in container terminals using a genetic algorithm," *Engineering Optimization*, vol. 39, no. 1, pp. 33–47, 2007.
- [27] V. D. Nguyen and K. H. Kim, "A dispatching method for automated lifting vehicles in automated port container terminals," *Computers and Industrial Engineering*, vol. 56, no. 3, pp. 1002–1020, 2009.
- [28] J. Cao, Q. Shi, and D.-H. Lee, "Integrated Quay Crane and Yard Truck Schedule Problem in Container Terminals," *Tsinghua Science & Technology*, vol. 15, no. 4, pp. 467–474, 2010.
- [29] J. X. Cao, D. H. Lee, J. H. Chen, and Q. Shi, "The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 3, pp. 344–353, 2010.
- [30] L. Chen, A. Langevin, and Z. Lu, "Integrated scheduling of crane handling and truck transportation in a maritime container terminal," *European Journal of Operational Research*, vol. 225, no. 1, pp. 142–152, 2013.
- [31] P. Ross, J. G. Marín-Blázquez, S. Schulenburg, and E. Hart, "Learning a procedure that can solve hard bin-packing problems: A new ga-based approach to hyper-heuristics," in *Genetic and Evolutionary Computation*. Springer, 2003, pp. 1295–1306.