Scheduling cranes at an indented berth
Beens, Marie-Anne ; Ursavas, Evrim

Published in:
European Journal of Operational Research

DOI:
10.1016/j.ejor.2016.02.038

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2016

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 13-09-2023
Discrete Optimization

Scheduling cranes at an indented berth

Marie-Anne Beens, Evrim Ursavas*

Department of Operations, Faculty of Economics and Business, University of Groningen, Nettelbosje 2, 9747 AE Groningen, The Netherlands

A R T I C L E   I N F O

Article history:
Received 24 July 2015
Accepted 23 February 2016
Available online 2 March 2016

Keywords:
Maritime logistics
Crane sequencing
Crane scheduling
Container terminal operations
Indented berth

A B S T R A C T

Container terminals are facing great challenges in order to meet the shipping industry's requirements. An important fact within the industry is the increasing vessel sizes. Actually, within the last decade the ship size in the Asia–Europe trade has effectively doubled. However, port productivity has not doubled along with the larger vessel sizes. This has led to increased vessel turn around times at ports which indeed is a severe problem. In order to meet the industry targets a game-changer in container handling is required. Indented berth structure is one important opportunity to handle this issue. This novel berth structure requires new models and solution techniques for scheduling the quay cranes serving the indented berth. Accordingly, in this paper we approach the quay crane scheduling problem at an indented berth structure. We focus on the challenges and constraints related to the novel architecture. We model the quay crane scheduling problem under the special structure and develop a solution technique based on branch-and-price. Extensive experiments are conducted to validate the efficiency of the proposed algorithm.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Approximately 90% of the world’s trade is conducted and processed by the international shipping industry, making international shipping the leading carrier of international trade (Tierney, 2014). A key aspect of the shipping industry is container shipping. Containers are designed for easy and rapid shipment of cargo. An important player in container shipping is the container terminals. A container terminal is a facility where cargo containers are transported from container ships and land vehicles for onward transportation, and containers are temporarily stored for further transportation. Globalization and an ever-increasing demand for goods have led to an increase in container shipping. Along with high competition and lower margins, what can directly be noticed in the sector is the increasing vessel sizes to benefit from economies of scale. In fact, since 2007 the ship size in the Asia–Europe trade has effectively doubled. However, port productivity has not been able to keep up with this increase. According to Maersk Lines, on an Asia–Europe round-trip the time spent in port (one vessel on average) has increased from 12 days (2007) to 18 days (Maersk Line Report, 2014). These facts have underlined the need for more efficient terminals in order for shipping companies to maintain a competitive position. The main objective in an efficient terminal is to be able to service the vessels within the time limits agreed upon with the shipping companies. Reducing the agreed limits requires the minimization of time spent in port by container ships: the berthing time. In order to minimize the berthing time it is necessary to handle all loading and discharging processes fluently and to minimize the total container traveling distance in the yard. Adding more resources and most critically the handling equipment that is quay cranes is one important solution to the problem. However, there is above all a physical limit (length of a vessel and safety distances between adjacent cranes) as to the number of cranes a vessel can be assigned to. The issue is that the length of vessels has not increased linearly with their TEU intake, as they have got wider, deeper and stacked higher instead. Therefore, in order to meet the industry standards and decrease berthing time a game-changer in container handling is required as Maersk Line CEO Soren Skou has put it: ‘We continue to build ships that are bigger and bigger, and if we can’t get the containers off faster, the whole thing will come to a grinding halt’ (Tirschwell, 2015).

Indented berths are recently brought as a solution to this severe problem. Upon arrival at port a container ship is assigned a berth with quay cranes, which load and unload containers to and from the ship. Indented berths enable the ship to be unloaded from two sides simultaneously with two rows of quay cranes on opposing sides, whereas traditional berths only allow loading and unloading from one side. Hence, indented berth increases the ship-to-shore interface length with the vessel so the vessel can be simultaneously handled from both sides. This new architecture requires new models and solution techniques to be built as the setting has its own characteristics, challenges and restrictions. We will briefly summarize these as follows:

* Corresponding author. Tel.: +31 50 363 7086.
E-mail address: e.ursavas@rug.nl (E. Ursavas).
http://dx.doi.org/10.1016/j.ejor.2016.02.038
0377-2217/© 2016 Elsevier B.V. All rights reserved.
Firstly, at an indented berth quay cranes are designed in a way that the arm can ascend when it is to pass the quay cranes on the opposite side. Due to this design feature the quay cranes on opposing sides of the indented berth are free of non-crossing constraints. Cranes on the same side and on the opposing sides should be handled differently together with specific restrictions such as the avoidance of working simultaneously on the same bay on the opposing sides.

Secondly, an important characteristic of indented berth operations is the importance of vessel stability which if not treated appropriately could lead to disasters. A ship contains a selection of add-on active stability systems, allowing it to adjust to the effects of waves and wind gusts. However, these do not increase the stability of the ship in calm seas. Since in indented berths quay cranes are able to handle the ship simultaneously from two sides, there is increased activity as compared to traditional berths. This increases the risk of an unbalanced ship. Balance constraints ensure that the difference between the number of containers on the left and the right sides of the ship is within a certain safe range. This requires the container load to be distributed equally along the width of the ship during the loading and unloading process. The balance constraints are forced by a certain latitudinal-balance threshold and restrict the number of containers on one side of the ship being significantly larger than the other side, preventing to have an incline and eventually, to tilt.

Another difference in an indented berth layout stems from the fact that the importance of minimizing the distance a container has to travel between vessel and yard block becomes more significant in the planning of the cranes. For an indented berth the distance of a task depends on the quay crane it is assigned to, whereas in a traditional berth the distance of a task is approximately similar for all quay cranes. This characteristic of indented berths enables to choose a quay crane to unload or load a container with the smallest distance from ship to yard block in order to minimize the total distance and provides opportunities for better solutions. This indeed requires the consideration of distances while planning the operations.

Finally, simultaneously operating quay cranes is indeed useful in shortening the berthing time for a ship. However, the suggestion that boosting the speed or number of quay cranes operating on a ship in order to increase the speed and productivity of the loading and unloading process is not necessarily the optimal solution, since more or faster operating quay cranes increase the risk of congestion on the quay crane rail tracks and in the terminal. Containers that are being discharged from a ship are transshipped to a yard block and containers that are being loaded onto a ship come from a yard block. A major issue is the avoidance of congestion at yard blocks. In comparison to a traditional berth layout, a system focused on simultaneously operating quay cranes and decreasing congestion is essential to consider in an indented berth setting.

Previous literature has not yet attended the indented berth layout problem considering the characteristics and restrictions stated above. Consequently, this paper contributes to the literature by creating a novel model for the quay crane scheduling problem considering the mentioned specific features of a terminal with an indented berth layout. We aim to minimize the distance traveled within the storage yard while adhering to the time limit contracted with the shipping company and the restrictions imposed with the novel setting. We have further developed a solution technique based on branch-and-price to solve the described problem.

Thereafter, numerical experiments on both the compact model and the branch-and-price algorithm are performed in order to compare results and performance. We further discuss concepts important to the industry for implementing the new layout. In the last section, we conclude this paper.

1.1. Port operations at an indented berth layout

Before arrival at the terminal, the shipping company provides the stowage plan. This stowage plan contains information about the location and destination of tasks. The expected vessel turn around time, that is the time that the vessel can sail to other destinations is agreed upon by contracts. Given these, upon arrival of the vessel at the terminal, the ship is berthed where loading and unloading tasks can be performed by quay cranes positioned on two sides of the ship. Quay cranes are mounted on a rail track at the quay and can move parallel to the ship. It is not possible for a quay crane to switch sides or cross the path of a quay crane mounted on the same side. As mentioned earlier, quay cranes are designed in a way that the arm can ascend when it is to pass quay cranes on the opposite side of the ship. This feature allows opposing quay cranes to be free from non-crossing constraints. However, simultaneous operations on the same bay from opposing sides should be avoided. Once containers are unloaded from the ship, the quay cranes place containers onto a vehicle for transportation to a determined yard block. A yard block consists of two parts: one part where containers that are unloaded from the ship are stacked and one part where containers that will be loaded onto the ship are stacked. A container terminal with an indented berth structure is illustrated in Fig. 1.

Transportation between ship and yard block is performed by automated vehicles or trucks. The distance between a yard block and the ship is affected by the quay crane handling a container. If a container is handled by a quay crane on the opposite side of the yard block, the distance is larger than if the container is handled by a quay crane on the same side as the yard block. In an indented berth there are two sizes of yard blocks. An indented berth is considered to have four accompanying yard blocks: two adjacent to the ship and two located in the yard at the front of the ship. Yard blocks that are adjacent to the ship are smaller than yard blocks located in the yard. Although it may differ across different terminals, it is assumed that yard blocks in the yard have a width of 6 rows of containers and can be up to six levels of containers high depending on the gantry crane used. Adjacent yard blocks have a length of 20 containers from end-to-end, while yard blocks in the yard can have up to 30 containers from end-to-end. The number of unloading and loading tasks at a particular yard block is stated in the stowage plan. At a yard block, yard cranes handle containers. Yard cranes can lift containers from trucks onto their assigned yard block and can place containers from the yard block onto trucks for further transportation. Congestion at yard blocks can be caused if a large amount of yard crane and truck activity has to be carried out at one yard block at the same time. After a task is finished, trucks are replaced in their original position allowing them to be used to perform subsequent tasks.

2. Literature review

Research focusing on container terminals has a long history. In recent years, there has been an increasing amount of literature regarding the quay crane scheduling problem with respect to spatial constraints and clearance constraints. However, in line with its novel architecture relatively very few papers on indented berths have been published. In the following section major research conducted on quay crane scheduling problem, and the quay
crane scheduling in the environment of indented berths will be reviewed.

For thorough information and review on container terminal operations we refer the reader to the three papers of excellent detail: Carlo, Vis, and Roodbergen (2014); Steenken, Voß, and Stahlbock (2004) and Bierwirth and Meisel (2015). The work of Steenken et al. (2004) and Carlo et al. (2014) offer important insights into the transportation process and relevant decision problems in relation to overall terminal processes. Bierwirth and Meisel (2015) provides an overview of the research output on berth allocation, quay crane assignment, and quay crane scheduling problems in container terminals. In their paper 52 publications on the quay crane scheduling problem have been reviewed. In this section the key papers on the quay crane scheduling problem will be discussed.

Daganzo (1989) is of the first authors discussing the quay crane scheduling problem. He suggested a linear integer programming formulation for determining the number of cranes to assign to terminals with multiple ships. His objective was to maximize the cost savings by minimizing delays for a situation of up to three ships. Peterkofsky and Daganzo (1990) provided a branch-and-bound algorithm in order to reduce the computation time and obtain the optimal solution. They based their model on the same formulation as in Daganzo (1989) and obtained the number of cranes to assign to ships at a specific time segment. However, both of the studies did not consider interference among quay cranes or relationships between tasks. There are relations between tasks since there are certain aspects in port operation which affect each other, such as the occupation at a yard block affecting crane sequencing and yard operations. Later, Lim, Rodrigues, Xiao, and Zhu (2004) studied the same problem as Daganzo (1989) but included consideration of the interference among quay cranes by integrating spatial constraints and non-simultaneity between tasks. They proposed a probabilistic tabu search and a squeaky wheel optimization heuristic.

Kim and Park (2004) considered the fact that there may be multiple tasks performed at one ship bay for the single ship problem. Their study divided one task into multiple smaller tasks not accounting for clearance constraints. Clearance constraints ensure quay cranes to be positioned at least r bays away from adjacent quay cranes on the left and right of the ship for safety reasons. They then applied a branch-and-bound algorithm to their problem. The branch-and-bound algorithm could not be solved in the time limit they considered to be reasonable. Having noted this, they built a quick GRASP. Therewith they encountered the problem with heuristic approaches: the search procedure does not ensure optimal solutions.

Bish (2003) developed a heuristic for a two ship situation. In this paper the problem was divided into three sub-problems: determining a storage location for each unloaded container, dis-patching vehicles to containers, and scheduling the loading and unloading operations of cranes. Despite the fact that this study encompasses a wider situation at the terminal, the model did not consider many operational constraints such as clearance and congestion.

Choo, Klabjan, and Simchi-Levi (2010) expanded the problem by capturing several requirements such as clearance and yard congestion constraints. They developed a branch-and-price algorithm for the single ship problem. This separates from the model presented by Daganzo (1989), where the single ship problem is solved by branch-and-bound. The set covering formulations in the paper of Choo et al. (2010) are different from the formulations in the paper of Daganzo (1989) and the main reason for the ability to solve large-scale instances which Daganzo (1989) could not solve. The branch-and-price algorithm of Choo et al. (2010) applies to a traditional berth considering only discharging operations.

Imai, Nishimura, Hattori, and Papadimitriou (2007) investigated a hybrid berth allocation problem for indented berths. A hybrid heuristic solves the problem by combining the functionalities of two optimization methods. Imai et al. first construct a new integer linear programming formulation for easier calculation and then the formulation is extended to model the berth allocation problem at a terminal with indented berths. This study is important as to better understand the mechanisms of indented berths and their effects. However, different from the present work, their paper does not consider quay crane scheduling but the efficient berthing of multiple ships. They however very well explain the course of events in indented berths.

Chen, Lee, and Cao (2011) expanded the single ship quay crane problem by accounting for the environment of indented berths. The objective in this problem is to minimize the berthing time considering the non-crossing and safety distance constraints. A tabu search heuristic was developed to solve the proposed problem. In the paper of Wang and Kim (2011) the quay crane scheduling problem was extended to account for decisions for assigning storage yard space to the unloaded containers. Such a quay crane scheduling problem is known as quay crane sequencing. The objective of this problem is not just to minimize berthing time but also to uniformly distribute the workload among the yard blocks. They solved the problem by a meta-heuristic in a traditional berth setting.

Overall, previous research and the current requirements of the shipping industry highlight the need for an efficient algorithm to solve the quay crane scheduling problem with yard congestion and workload safety constraints for an indented berth. In view of the above discussions, this paper provides a branch-and-price algorithm on the quay crane scheduling problem considering the spatial, congestion and workload safety constraints for an indented
berth. To reflect practice efficiently both the unloading and the loading processes are considered. To the best of our knowledge, such research has not yet been conducted and the need for further work on the proposed setting is required considering current industry standards and future of container terminals.

3. Mathematical model

We start with a compact model formulation and continue with a reformulation for the branch-and-price algorithm. The focus of this paper is on the quay crane scheduling of indented berths with yard congestion and workload stability constraints. The objective is to minimize the distance traveled by containers between the ship and the yard block. It is assumed that the planning horizon is divided into small time units and quay crane movements take place at discrete time intervals. As in practice, a vessel is hypothetically divided along its length into segments known as bays, which can be several container rows wide and several lines deep. To ensure the balance constraints, a distinction will be made between the left and right sides of a bay. Accordingly, the bays will be split into two sub-bays: one on the left side of the ship and one on the right side of the ship. It should be noted that this split is only for ease of interpretation and cranes from each side can indeed operate on the whole bay. Scheduling operations of quay cranes should adhere to the following rules:

- A quay crane can only work on one single bay at a time;
- Only one quay crane can work on a bay at a time and no other quay crane can work at the same bay or at the bay located on the opposite side;
- Quay cranes located at the same side of the ship cannot cross each other’s path and, therefore, must be ordered by position at all times;
- As a consequence of the ability to lift the arm of a quay crane, quay cranes located at opposite sides of the ship can cross each other’s path and, therefore, are free from ordering constraints, however cranes at opposite sides cannot work at the same bay;
- The number of quay cranes handling containers at a specific yard block at a particular time cannot exceed a pre-determined number. The term that will be used to describe this pre-determined number is the yard block occupation threshold for the specific yard block.
- The workload on the ship must be in balance at all time in order to keep the ship from tilting.

For this model, the ordering and clearance constraints are disaggregated to help speed up the problem by imposing a tighter problem.

The following assumptions have to be taken into account:

- The gantry distance of a quay crane over the ship is small compared to the container handling distance to the yard blocks. This assumption is often justified and is excluded from the calculations.
- The total number of quay cranes that can be operational during the handling of the vessel is limited.
- There are two different types of storage locations: discharging storage locations and loading storage locations.
- Quay cranes have similar work rates.
- Although balance in port is also affected by passive stability systems, such as the bilge keel, outriggers, anti roll tanks and paravanes, it is assumed that a ship is 100% stable when berthing.
- For simplification of the input variables, merely the number of containers is stated and the weight of a single container is not explicitly taken into account. It is however to possible to incorporate the individual container weights within the input parameter if the information is readily available and correctly given in the stowage plan.
- The differences in terms of number of containers is accounted within the threshold levels imposed.

The input data to the problem are: (1) the number of quay cranes working on the ship, with the distribution of quay cranes on the left and right side of the ship, (2) the number of bays in the ship, (3) the relative distance from bay handled by a particular quay crane to yard block, (4) the yard block occupation threshold, (5) the safety range in order to keep the ship in balance and (6) the vessel’s stowage plan. These information, although not easily attainable in practice, are assumed to be available.

3.1. Compact model formulation

For the mathematical formulation the following notation is used:

**Indices:**

- **j**: Bay number, in increasing order of their relative location on the ship numbered from front to back and from left to right;
- **k**: Quay crane number, in increasing order of their relative location on the ship numbered from front to back and from left to right starting at the left top of the ship;
- **b**: Yard storage location number. First numbered for discharging storage locations in increasing order of their location in the yard and from left to right and then for loading storage locations in increasing order of their location in the yard and from left to right.
- **t**: Time period index, denoting an interval.
- **r**: Quay crane clearance value, in terms of the number of bays.

**Input data:**

- \( f_{jk} \): The number of containers to be handled for the bay-block combination \((j, b)\);
- \( d_{jk} \): The distance for a container from bay \(j\) handled by crane \(k\) headed for storage location \(b\);
- \( w_b \): Yard block occupation threshold, the maximum number of quay cranes allowed to work on containers scheduled for storage location \(b\) at a particular time;
- \( Q \): Latitudinal-balance threshold, the maximum difference in the number of containers between bays located on the left and right sides of the ship;

**Parameters:**

- \( H_L \): The number of bays on the left side of the ship, hence \( j_L = \{1, \ldots, H_L\} \);
- \( H_R \): The number of bays on the right side of the ship, hence \( j_R = \{H_L + 1, \ldots, H\} \);
- \( H \): The total number of bays in the ship, \( H = H_L + H_R \), note that \( j = j_L + j_R = \{1, \ldots, H_L, H_L + 1, \ldots, H\} \);
- \( C_L \): The number of quay cranes located at the left side of the ship. Note that the set of quay cranes located on the left side of the ship is \( k_L = \{1, \ldots, C_L\} \);
- \( C_R \): The number of quay cranes located at the right side of the ship. Obviously the set of quay cranes located on the right side of the ship is \( k_R = \{C_L + 1, \ldots, C\} \);
- \( C \): Total number of quay cranes allocated. There is \( C = C_L + C_R \), note that \( k = k_L + k_R = \{1, \ldots, C_L, C_L + 1, \ldots, C\} \);
- \( B_D \): The number of discharging yard locations in the yard \( b_D = \{1, \ldots, b_D\} \);
- \( B_L \): The number of loading yard locations in the yard \( b_L = \{1, \ldots, b_L\} \);
- \( B \): The number of storage locations in the yard. Note that \( B = b_D \cup b_L \), \( b_D \cap b_L = \varnothing \), and \( b_D \cup b_L = \{1, \ldots, b_D \cup b_L\} \);
T Number of time periods in the planning horizon;

**Decision Variables:**

\[ x_{jk}(t) \] 1 if quay crane \( k \) is positioned at bay \( j \) at time period \( t \) and 0 otherwise;

\[ \delta_{jkb}(t) \] 1 if quay crane \( k \) is handling a container at bay \( j \) headed for or coming from yard block \( b \) at time period \( t \) and 0 otherwise;

**Fig. 2** demonstrates the interpretation of the decision variables. The bold line around a set of bars represents the fact that no other quay crane can work in this set since there is already a quay crane assigned to handle a task in one of the bays within the set. The distinction between left and right bays is solely made to allow for balancing on the left and right sides. An arrow pointing away from the ship through a quay crane to a yard block indicates a discharging operation at an arbitrary time \( t \). An arrow pointing towards the ship through a quay crane from a yard block indicates a loading operation at the arbitrary time \( t \). For instance in the figure, quay crane 1 discharges a container from bay 2 headed for yard block 1 at time \( t \). At the same time quay crane 3 cannot work at the bay where it is positioned since quay crane 1 is working in a bay with the same relative bay number. The relative bay number is ordered from front to back of the ship and is the number of the bay as if the two sub-bays on the left and right side of the ship were one bay. For example at time \( t \), quay crane 4 is loading a container from yard block 8 to bay \( H_t+4 \) with relative bay number 4.

Having explained the variables, the mathematical formulation for the indented berth quay crane scheduling problem can be given as follows:

\[
\text{minimize} \sum_{j=1}^{H} \sum_{k=1}^{C} \sum_{b=1}^{B} \sum_{t=1}^{T} \delta_{jkb}(t)d_{jkb}
\]

subject to

\[ \sum_{j=1}^{C_t} x_{jk}(t) = 1 \]
\[ \sum_{j=1}^{C_t} x_{jk}(t) + \sum_{j=H_t+1}^{C_t} x_{j+H_t,k}(t) \leq 1 \]
\[ \sum_{k=C_t+1}^{C} x_{jk}(t) + \sum_{k=1}^{C_{t+1}} x_{j+H_t,k}(t) \leq 1 \]
\[ \sum_{b=1}^{B} \sum_{k=1}^{C_t} \delta_{jkb}(t) + \sum_{b=1}^{B} \sum_{k=1}^{C_{t+1}} \delta_{j+H_t,kb}(t) \leq 1 \]
\[ \sum_{b=1}^{B} \delta_{jkb}(t) \leq x_{jk}(t) \]
\[ j \in 1, \ldots, H; k \in 1, \ldots, C; t \in 1, \ldots, T \]

\[ 1 - x_{jk}(t) \geq \sum_{l=\max(1,j-H_t-r)}^{j-H_t} x_{lm}(t) + \sum_{l=\max(H_t+1,j+H_t-r)}^{j-H_t} x_{lm}(t) \]
\[ j \in H_t + 2, \ldots, H; k \in 1, \ldots, C_t; m \in 1, \ldots, C_t, t \in 1, \ldots, T \]

\[ 1 - x_{jk}(t) \geq \sum_{l=\max(1,j-h_t-r)}^{j-H_t} x_{lm}(t) + \sum_{l=\max(H_t+1,j+H_t-r)}^{j-H_t} x_{lm}(t) \]
\[ j \in 2, \ldots, H_t; k \in 1, \ldots, C_t; m \in 1, \ldots, C_t, t \in 1, \ldots, T \]

**Additional Equations**

\[ x_{jk}(t) \leq \sum_{l=1}^{j-H_t-1} x_{lk-1}(t) + \sum_{l=H_t+1}^{j-H_t} x_{lk-1}(t) \]
\[ j \in 2, \ldots, H_t; k \in 1, \ldots, C_t; t \in 1, \ldots, T \]

\[ x_{jk}(t) \leq \sum_{l=1}^{j-H_t-1} x_{lk-1}(t) + \sum_{l=H_t+1}^{j-H_t} x_{lk-1}(t) \]
\[ j \in H_t + 2, \ldots, H; k \in 1, \ldots, C_t; t \in 1, \ldots, T \]

\[ x_{jk}(t) \leq \sum_{l=1}^{j-H_t-1} x_{lk-1}(t) + \sum_{l=H_t+1}^{j-H_t} x_{lk-1}(t) \]
\[ j \in 2, \ldots, H_t; k \in 1, \ldots, C_t; m \in 1, \ldots, C_t, t \in 1, \ldots, T \]
Fig. 2. Interpretation of the variables.

\[
x_{jk}(t) \leq \sum_{l=j+1}^{H} x_{l,k-1}(t) + \sum_{l=H_{k}+j+1}^{H_{k}} x_{l,k+1}(t)
\]

\[j \in H_0 + 1, \ldots, H - 1; k = 1, \ldots, C_l - 1; t = 1, \ldots, T\]  \hspace{1cm} (19)

\[
x_{jk}(t) \leq \sum_{l=j+1}^{H} x_{l,k-1}(t) + \sum_{l=H_{k}+j+1}^{H_{k}} x_{l,k+1}(t)
\]

\[j \in 1, \ldots, H_0 - 1; k = C_l + 1, \ldots, C - 1; t = 1, \ldots, T\]  \hspace{1cm} (20)

\[
x_{jk}(t) \leq \sum_{l=j+1}^{H} x_{l,k-1}(t) + \sum_{l=H_{k}+j+1}^{H_{k}} x_{l,k+1}(t)
\]

\[j \in H_0 + 1, \ldots, H - 1; k = C_l + 1, \ldots, C - 1; t = 1, \ldots, T\]  \hspace{1cm} (21)

\[
x_{1,k,t} = 0 \hspace{1cm} k \in 2, \ldots, C_l; t = 1, \ldots, T\]  \hspace{1cm} (22)

\[
x_{1,k,t} = 0 \hspace{1cm} k \in C_l + 2, \ldots, C; t = 1, \ldots, T\]  \hspace{1cm} (23)

\[
x_{H_{k}+1,k,t} = 0 \hspace{1cm} k \in 2, \ldots, C_l; t = 1, \ldots, T\]  \hspace{1cm} (24)

\[
x_{H_{k}+1,k,t} = 0 \hspace{1cm} k \in C_l + 2, \ldots, C; t = 1, \ldots, T\]  \hspace{1cm} (25)

\[
x_{H_l,k,t} = 0 \hspace{1cm} k \in 1, \ldots, C_l - 1; t = 1, \ldots, T\]  \hspace{1cm} (26)

\[
x_{H_l,k,t} = 0 \hspace{1cm} k \in C_l + 1, \ldots, C - 1; t = 1, \ldots, T\]  \hspace{1cm} (27)

\[
x_{H_{k}+1,k,t} = 0 \hspace{1cm} k \in 1, \ldots, C_l - 1; t = 1, \ldots, T\]  \hspace{1cm} (28)

\[
x_{H_{k}+1,k,t} = 0 \hspace{1cm} k \in C_l + 1, \ldots, C - 1; t = 1, \ldots, T\]  \hspace{1cm} (29)

\[
\sum_{j=1}^{H} \sum_{k=1}^{C_l} \delta_{jk}(t) \leq w_b
\]

\[b \in 1, \ldots, B; t = 1, \ldots, T\]  \hspace{1cm} (30)

\[
\sum_{j=1}^{H} \sum_{k=1}^{C_l} \sum_{b=1}^{B} \delta_{ijk}(t) - \sum_{j=1}^{H} \sum_{k=1}^{C_l} \sum_{b=1}^{B} \delta_{ijk}(t) \leq Q
\]

\[t \in 1, \ldots, T\]  \hspace{1cm} (31)

\[
\sum_{j=1}^{H} \sum_{k=1}^{C_l} \sum_{b=1}^{B} \delta_{ijk}(t) - \sum_{j=1}^{H} \sum_{k=1}^{C_l} \sum_{b=1}^{B} \delta_{ijk}(t) \leq Q
\]

\[t \in 1, \ldots, T\]  \hspace{1cm} (32)
\[ \sum_{k=1}^{C} \sum_{t=1}^{T} \delta_{jkb}(t) = f_{jb} \]

\[ j \in 1, \ldots, H; b \in 1, \ldots, B; \]

(33)

\[ x_{jk}(t), \delta_{jkb}(t) \quad \text{binary} \]

\[ j \in 1, \ldots, H; k \in 1, \ldots, C; b \in 1, \ldots, B; t \in 1, \ldots, T \]

(34)

The objective function evaluates the sum of the total travel distance of the quay cranes between bays and storage locations. Constraint (1) rules that a quay crane can only be positioned at a bay at all times. Constraints (2) and (3) ensure that one and only one of the quay cranes mounted on the same side can be positioned at a bay at a time. Due to the possibility of lifting the arm of the quay crane, two quay cranes on opposing sides are allowed to be positioned at the same bay at a time. However, two opposing quay cranes cannot work at the same bay at a time, therefore constraint (4) ensures that only one quay crane can work at a (bay,bay) combination (either on the left bay or on the right bay). Constraint (5) guarantees that a quay crane must be positioned at a bay if it is working there. Constraints (6)–(13) demonstrate the clearance constraints, stating that quay cranes on the same side are restricted to being positioned r bays to the left and right if a quay crane is positioned at a particular bay. Constraints (14)–(21) assure the ordering conditions for the quay cranes positioned at the left and right sides of the ship. As stated before, quay cranes positioned on opposing sides are free from clearance and ordering constraints. Therefore there are separate clearance and ordering constraints for quay cranes on opposite sides. It is possible for quay cranes on opposing sides to be positioned at one bay at a time while constraint (4) ensures that only one quay crane can perform a task at one bay at a time. The clearance constraints for quay cranes mounted on the same side with respect to the ship are split into 4 separate constraints. First there are 2 constraints ensuring that higher-numbered quay cranes cannot be positioned within r bays with a lower relative bay number. Then, there are 2 constraints ensuring that lower-numbered quay cranes cannot be positioned within r bays with a lower relative bay number. The ordering constraints for quay cranes positioned on the same side are also split into 4 separate constraints. The first 2 constraints ensure that higher-numbered quay cranes cannot be positioned to bays with a lower relative bay number. The second 2 constraints ensure that lower-numbered quay cranes cannot be positioned to bays with a higher relative bay number. It is important to note that both clearance and ordering constraints take into account the relative bay numbering and not bay numbering as in the increasing order of their location on the ship numbered from front to back and from left to right. Constraint (22)–(29) state that the outermost bays of the ship are unreachable for some quay cranes. Constraint (30) makes sure that the total amount of quay crane work performed at a particular storage location b at a time t does not exceed the yard block occupation threshold. Constraints (31) and (32) fulfill the balance conditions, the weight of work on one side cannot differ from the other side more than Q containers. Constraint (33) is the work completion constraint. This is the agreed deadline where the vessel must be served. All the requested container jobs must be completed within the predefined planning horizon. All decision variables are binary as stated in (34).

Numerical experiments show that solving this compact model leads to large computational times, being inconvenient for the problem. A branch-and-price algorithm is therefore built in order to improve the computational times.

3.2. Branch-and-price algorithm for the indented berth layout

Branch-and-price algorithms modify the basic branch-and-bound algorithm by attempting to strengthen the linear programming relaxation. In short, the branch-and-price integrates the branch-and-bound and column generation methods. This leads to columns being added to the algorithm, which may be possible since most of them will have their associated variables equal to zero in an optimal solution. The particular algorithm built in this paper is designed for the indented berth structure. Throughout the rest of this paper, the abbreviation B&PIB is used to refer to the Branch-and-Price Indented Berth algorithm.

3.2.1. Reformulation

It is necessary to reformulate the compact model into column generation form resulting in the so-called master problem. The key principle of the reformulation is to set up an entire quay crane position-to-(bay,block) assignment per time period as a decision variable. Each coordinate represents a quay crane position-to-(bay,block) assignment and each row symbolizes a (bay,block)-combination. To account for the fact that a task at a (bay,block)-combination can be accomplished from two sides, each odd row represents a (bay,block)-combination from the left side and each even row represents a (bay,block)-combination from the right side. Hence, every (bay,block)-combination is represented by an odd row and an even row. Stated differently, each odd row is numbered ‘1’ if a quay crane from the left is performing a task in a particular (bay,block)-combination and each even row is numbered ‘1’ if a quay crane from the right side is performing a task in that particular (bay,block)-combination. These assignments can be further exemplified by the following example: Suppose there are 2 quay cranes of which 1 positioned to the left and 1 positioned to the right, and 4 bays, 2 left and 2 right numbered counterclockwise beginning at the left front of the ship. Then the following column (0, 1, 1, 0, 0, 0, 0, 0)’ denotes that the first right quay crane is working on bay 1 and the first left quay crane is working on bay 2. Such assignments must obey the ordering, congestion and balance constraints.

Let P be the set of all feasible QC work assignments. For p ∈ P, the binary variables \( y_p \) are created, the number of times the quay crane position-to-(bay, yard position) assignment p is selected. Hence, the decision variable \( y_p \) depends on \( (s, j, b) \), where s represents the position, or in other words side, of a quay crane with respect to the ship. Hence \( s \in \{ \text{left}, \text{right} \} \). Because the work completion constraint has to be satisfied, the set partitioning structure has the right-hand side vector \( f \), which is the vessel’s stowage plan for each \((j, b)\) pair. The set-partitioning formulation is given by:

\[ \text{minimize} \sum_{p \in P} y_p \]

subject to

\[ \sum_{p \in P : (s, j, b) = p} y_p = f_{jb} \]

\[ j \in 1, \ldots, H; b \in 1, \ldots, B \]

(35)

\[ y_p \] nonnegative integer

(36)

where \( d_p \) represents the traveling distance of one assignment, calculated from the total distance of all bay-to-block pairs appearing in that assignment and \((s, j, b) \in p \) represents the fact that bay j yard block b is assigned to a quay crane assignment in p.

The objective function minimizes the total traveling distance from the quay crane position-to-(bay,block) assignments needed to handle all the jobs in the stowage plan. Constraint (35) captures the work completion requirement: for every bay j and block b combination at least \( f_{jb} \) assignments must be selected regardless of the side from which the containers are handled.
In set partitioning problems, constraints are equality constraints whereas in set covering problems, constraints are inequality. Set covering formulations are preferred over set partitioning problems in a branch-and-price algorithm since their LP relaxation is more stable, and thus easier to solve (Milano, 2004). To account for this, the master formulation will be modified to a set covering formulation. In general, if any subcolumn of a feasible column defines another feasible column with lower cost, an optimal solution to the set covering problem will be an optimal set partitioning (Barnhart, Johnson, Nemhauser, Savelsbergh, & Vance, 1998). Without loss of optimality, the set partitioning formulation can be written as a set covering formulation.

Let the optimal number of jobs in the set-partitioning problem be \( y_P \) and the number of jobs in the set-covering problem be \( y_C \). Let \( S_P \) be the set of set partitioning solutions and \( S_C \) the set of set covering solutions. Since every set partitioning solution is covered in the set of set covering solutions, \( S_P \subseteq S_C \). Suppose \( y_C - y_P = D \), hence the number of jobs is exceeded by \( D \) jobs. Then the number of jobs can arbitrarily set back by resetting \( D \) times a \( \delta \) from ‘1’ to ‘0’ in the solution. Since quay crane positions \( x_s \)’s are independent of quay crane jobs \( \delta \)’s, the quay crane position constraints are not violated. Since reduced workload positions makes other constraints less tight, the feasibility of the problem is not affected. Therefore, it is possible to modify the master formulation to a set covering formulation.

The set-covering formulation of the pricing problem is now given by:

\[
\begin{align*}
\text{minimize} & \quad \sum_{p \in P} y_p d_p^{\text{dual}} \\
\text{subject to} & \quad \sum_{p \in \{ (i,j,b) \} \cap P} y_p \geq f_{jb} \\
& \quad j \in 1, \ldots, H; \ b \in 1, \ldots, B \\
& \quad y_p \text{non-negative integer}
\end{align*}
\]

(37)

(38)

After the B&P algorithm, the amount of surplus at a particular (bay,block)-combination can easily be found and the excess \( \delta \)’s that are ‘1’ in the original solution can be reset to ‘0’. In this paper, the term that will be used to describe phenomenon of the excess \( \delta \)’s that are reset is excess assignments. As explained previously, resetting excess assignments to ‘0’ will not change the optimal solution, since the excess assignments and thus the excess distance will not be covered.

### 3.2.2. Initial column pool

Before the start of the column generation, an initial RMP has to be established with a feasible solution. Hence, a small set of columns will be generated which provide a feasible basis. This small set will be referred to as the initial column pool. The initial column pool is generated on account of the initial RMP such that it has a feasible, but most likely non-optimal, LP solution. The initial column pool needs to be a set of columns that covers all (bay,block)-combinations with non-zero workloads. Otherwise, the initial master problem will be infeasible causing the whole problem to be infeasible. The initial column pool must consider constraints such as the clearance constraints and the fact that only one quay crane can work on a bay at a time. Any set of columns obeying these covering all bay-block combinations once will suffice. Here we follow a simple heuristic where we begin from an opposite end of the vessel on each side and proceed with the remaining bays in the physical order. In each phase, we add assignments step by step making certain that the restrictions are met. We further use the knowledge that most quay cranes on the left side are presumably closer to lower numbers of discharging and loading blocks than quay cranes on the right side. Containers corresponding to blocks on the left side are at least once handled by left quay cranes and containers corresponding to blocks on the right side are at least once handled by right quay cranes, since we use the approach that ‘The distance between a bay and storage location is minimal if the handling quay crane is located at the same side as the yard block.’ Accordingly, the initial column pool generation procedure generates a feasible basis and stores the set of (bay,block)-combinations.

#### 3.2.3. Pricing

The pricing problem identifies a column that can enter the basis favorably or prove that no such column exists. The pricing problem constructs a path in the assignments, taking decisions on whether to include a typical edge in a path or not. In this problem the objective of the pricing problem is to find an assignment with the highest negative value. The dual variable \( y_{jb} \) associated with constraint (37) is used to solve the problem. If the cost of the solution is less than the dual variable of the constraint, the column has a negative reduced cost. The columns generated by the pricing problem are then added to the RMP and the problem is resolved.

The pricing problem reads:

\[
\begin{align*}
\text{minimize} & \quad - \sum_{s \in S} \sum_{j=1}^{H} \sum_{b=1}^{B} d_{sjb} \delta_{sjb} y_{jb} \\
\text{subject to} & \quad \sum_{j=1}^{H} x_{s,j} = c_L \\
& \quad \sum_{j=1}^{H} x_{s,j} = c_R \\
& \quad \delta_{sjb} \leq x_{sj} \\
& \quad s \in S; j \in 1, \ldots, H; b \in 1, \ldots, B_L + B_D \\
& \quad \sum_{s \in S} \sum_{j=1}^{H} \delta_{sjb} \leq w_b \\
& \quad b \in 1, \ldots, B \\
& \quad \sum_{s \in S} \sum_{j=1}^{H} \sum_{b=1}^{B} \delta_{sjb} - \sum_{s \in S} \sum_{j=1}^{H} \sum_{b=1}^{B} \delta_{sjb} \leq Q \\
& \quad \sum_{s \in S} \sum_{j=1}^{H} \sum_{b=1}^{B} \delta_{sjb} - \sum_{s \in S} \sum_{j=1}^{H} \sum_{b=1}^{B} \delta_{sjb} \leq Q \\
& \quad \sum_{s \in S} \sum_{j=1}^{H} \sum_{b=1}^{B} \delta_{sjb} + \sum_{s \in S} \sum_{j=1}^{H} \sum_{b=1}^{B} \delta_{sjb} \leq 1 \\
& \quad j \in 1, \ldots, H; j \in 1, H_L \\
& \quad \sum_{s \in S} \sum_{b=1}^{B} \delta_{sjb} + \sum_{s \in S} \sum_{b=1}^{B} \delta_{sjb} \leq 1 \\
& \quad j \in H_L + 1, \ldots, H 
\end{align*}
\]
\[ C_l(1 - x_{1,j}) \geq \min_{j \in 1, \ldots, H_l} x_{1,j} + \sum_{l = j + H_l + 1}^\infty \min_{j \in H_l + r} x_{1,j} \]
\[ j \in 1, \ldots, H_l - 1 \quad (47) \]

\[ C_l(1 - x_{1,j}) \geq \sum_{l = j + H_l + 1}^\infty \min_{j \in H_l + r} x_{1,j} \]
\[ j \in H_l + 1, \ldots, H_l - 1 \quad (48) \]

\[ C_l(1 - x_{1,j}) \geq \sum_{l = j + H_l + 1}^\infty \min_{j \in H_l + r} x_{1,j} \]
\[ j \in H_l + 2, \ldots, H_l \quad (49) \]

\[ C_l(1 - x_{2,j}) \geq \min_{j \in 1, \ldots, H_l} x_{2,j} + \sum_{l = j - H_l}^\infty \min_{j \in H_l - r} x_{2,j} \]
\[ j \in 1, \ldots, H_l - 1 \quad (50) \]

\[ C_l(1 - x_{2,j}) \geq \sum_{l = j - H_l}^\infty \min_{j \in H_l - r} x_{2,j} \]
\[ j \in H_l + 1, \ldots, H_l - 1 \quad (51) \]

\[ C_l(1 - x_{2,j}) \geq \sum_{l = j - H_l}^\infty \min_{j \in H_l - r} x_{2,j} \]
\[ j \in H_l + 2, \ldots, H_l \quad (52) \]

\[ C_l(1 - x_{2,j}) \geq \sum_{l = j - H_l}^\infty \min_{j \in H_l - r} x_{2,j} \]
\[ j \in H_l + 2, \ldots, H_l \quad (53) \]

\[ \delta_{ij} \text{ binary} \]
\[ s \in S; \ j \in 1, \ldots, H; b \in 1, \ldots, B \quad (55) \]

The decision variable \( \delta_{ij} \) is 1 if a QC from side \( s \) is handling at bay \( j \) for yard block \( b \) and 0 otherwise; and the decision variable \( x_{ij} \) is 1 if a QC from side \( s \) is positioned at bay \( j \). In the model \( S \) denotes both sides left and right.

The objective function captures the reduced cost of a non-basic column where the non-basic column vector is represented by \( \delta_{ij} \), \( s \). Constraints (39) and (40) ensure that all quay cranes are assigned. Constraint (41) guarantees that a quay crane is positioned at a bay if it is working there. Constraint (42) embodies the yard congestion constraint. Constraints (43) and (44) impose the balance constraints. Constraints (45) and (46) ensure that only one quay crane can be working at a bay. Constraints (47)–(54) impose the ordering and clearance constraints.

3.2.4. Branching

Based on the RMP and the pricing problem, branching is applied. In this paper branching is applied to the least fractional column. In case of ties, the variable with the lowest index number is selected. Branching continues until a column that is regenerated is encountered. Here, depth-first search is adopted where integer solutions are found quickly.

3.2.5. Translating the solution

The result of a successful run of the B&PIB algorithm is \( y_{\text{best}} \), the number of times a feasible quay crane handling-assignment ‘pattern’ is used. The \( y_{\text{best}} \)'s can be rearranged to quay crane assignments \( \delta_S \), the decision variables of the original model for each time period. Here the column pool is used to find the quay crane assignments per time period. As an example, in every column the first ‘1’ in an odd row corresponds to the first quay crane on the left side, the second ‘1’ correspond to the second quay crane on the left side. Then in the same procedure, in every column the first ‘1’ in an even row corresponds to the first quay crane on the right side, the second ‘1’ correspond to the second quay crane on the right side.

As previously stated, the RMP is a set-covering formulation. The LP relaxation for a set-covering formulation is stronger than the LP relaxation for a set-partitioning formulation. A consequence is that a quay crane is most likely scheduled to carry out more tasks than executed in reality. The wastage of scheduled work can easily be computed by using the task workload of a bay.

It is possible to set back scheduled \( \delta \)'s to 0 in order to precisely meet the work completion constraints. The \( \delta \)'s that will be set back to 0 are assignments that will be deleted from the schedule in reality. As noted earlier, this type of assignment will be defined by the term ‘excess assignment’. The number of excess assignments in a (bay,block)-combination is equal to the amount of wastage in the (bay,block)-combination. Here we specify the excess assignments in a (bay,block)-combination with the largest distances where they do not belong in the solution sequence.

The following example is given to illustrate the translation of the solution. Consider the simple scenario: There are 2 quay cranes, 1 located on the left and 1 on the right. There are 6 bays with duration being 3 time periods. Lastly, there is only 1 block \( b \). We note that since the workload balance constraints cannot be taken into account for 1 block (being either a loading or discharging block) and the distance between cranes and blocks are fixed, this simple example will not represent the full features but will serve the purpose of illustrating the translation process. The following Fig. 3 illustrates this translation.

This translation completes the entire B&PIB algorithm in its search for the optimal solution to the quay crane scheduling with yard congestion and stability constraints for an indented berth.

4. Numerical experiments

The first part of this section gives a brief overview of the various problem instances used for numerical experiments on the compact model and the B&PIB algorithm. It will then continue to provide a summary of the main findings and to explain the result of the numerical experiments. Furthermore a sensitivity analysis is conducted.

4.1. Experimental settings

To test the models, datasets for 6 main problem instances were created, arranged in the order of the increase in bays within a ship, which varies from 20 to 60. Within the 6 problem instances several sub-cases are created, ordered by increase in number of cranes. An instance is defined by an IB-number relative to the number of bays and cranes, where IB stands for Indented Berth. The instances are used to compare the computational performances of the B&PIB to the exact model. Thereafter, a sensitivity analysis is performed to test the robustness of the B&PIB algorithm.

A stowage plan was created per instance and remains the same for the sub-cases. The tasks that need to be executed are stated in the stowage plan. One task is equal to performing desired actions.
on approximately 3 containers. The number of containers per bay-block combination is generated randomly but takes into account the balance conditions. In order to compare the effect of the size of the stowage plan IB3 and IB4 are created. These two instances contain the same characteristics but have a different stowage plan. The size of the stowage plan of IB4 is larger relative to the stowage plan of IB3, causing more tasks to be carried out in IB4 than in IB3. In an attempt to simulate very large problem instances, the instances IB5 and IB6 are generated. The table below illustrates some of the main characteristics of the problem instances:

In the problem instances 7 is regulated such that a quay crane can handle approximately 3 containers. This implies that if a quay crane is assigned to perform a task at a bay at some time period \( t \in T \), the quay crane handles 3 containers jobs at that bay in one particular direction (either unloading direction or discharging direction). Since as stated before the technical performance of a quay crane in operation is in the range of 22–33 containers per hour and on the basis of the results from Vis and van Anholt (2010), a makespan for a large container ship berth between 24 and 48 hours is considered as a plausible realistic berthing time for a container ship, where a large container ship consists approximately 2000 container jobs per port.

All the problem instances are tested at three congestion levels determined by the value of the yard block occupation threshold. These threshold levels in practice may indeed vary depending on the number/layout of lanes, safety considerations, number of vehicles and such. In the low congestion level the yard block occupation threshold is set as 4 for each block. The yard block occupation threshold at this congestion level most likely does not restrict the feasible region, since the number of quay cranes lies between 4 and 8 for all problem instances causing the yard block occupation threshold to have little effect on the sequence of quay cranes handling containers headed for and coming from a particular block. For the medium congestion level the yard block occupation threshold is bounded by 2 for all blocks and for the high congestion level the yard block occupation threshold is limited to 1.

Experiments with these problem instances were conducted on a Intel®Core™2 Duo CPU 3.00 GigaHertz personal computer with 4.00 GHz of RAM operating on a Windows 7 Enterprise operating system using Mosel Xpress-IVE Version 1.24.00 64 bit.

4.2. Comparison of the models

The compact model formulation and B&PIB formulation have the same optimal solution and are compared by performance, measured in runtime. Maximum runtime is set as one hour. In case no solution is found within this time limit this is interpreted with a blank line in Table 2. The set of results for the different problem instances is given in the following table where the runtimes are averaged over 6 runs for the B&PIB algorithm:

It can be seen in Table 2 that the compact model is unable to solve large instances. Since large instances are also a reflection of reality especially in the case of indented berth layout, it is inconvenient to use the compact model. Due to the large and complex mathematical formulation of the compact model, it is also not able to solve the other remaining instances within a reasonable time. Solely based on the runtime, these results are evidence for the need of the B&PIB algorithm. Regarding the runtime, the B&PIB outperforms the compact model in order to find the optimal solution. The setup time and the initial column pool generation time are also provided in the table. The increase in computation times is also affected by the size of the column pool as the instances become larger. The depth-first branching approach allows for integer solutions to be found quickly resulting in fewer number of nodes explored in the branching tree which relates to the reduction of the computation times of the algorithm. To decrease the computational workload in the branch-and-price approach, instead of repeatedly solving the restricted master problem at each iteration, the basis of the optimal set of columns from previous iterations is used as an initial basis. Regarding the computation time, the set covering formulation enables more stable and easier to solve LP relaxations. Accordingly, the runtime is significantly different from the compact model and stays under 1 second even for the largest instances.

A brief inspection of differences in the results between the distinct difficulty levels shows that the yard block occupation constraint neither increases nor decreases the optimal distances. These results are in agreement with expectations and might be explained by the following: Certain (bay,block)-combinations cannot be handled by a quay crane with the shortest distance in a certain time period as the yard congestion constraint will be violated otherwise. As required, the (bay,block)-combination will eventually be
Table 1

<table>
<thead>
<tr>
<th>Instance code</th>
<th>$H_1$</th>
<th>$H_2$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$Q$</th>
<th>$T$</th>
<th>$B$</th>
<th>No. of constraints</th>
<th>No. of variables</th>
<th>No. of container jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IB1-1</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>11</td>
<td>200</td>
<td>82,560</td>
</tr>
<tr>
<td>IB1-2</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>11</td>
<td>200</td>
<td>159,360</td>
<td>1,144,000</td>
</tr>
<tr>
<td>IB2-1</td>
<td>30</td>
<td>15</td>
<td>15</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>250</td>
<td>156,690</td>
<td>216,000</td>
</tr>
<tr>
<td>IB2-2</td>
<td>30</td>
<td>15</td>
<td>15</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>15</td>
<td>250</td>
<td>305,490</td>
<td>270,000</td>
</tr>
<tr>
<td>IB3-1</td>
<td>40</td>
<td>20</td>
<td>20</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>18</td>
<td>300</td>
<td>252,920</td>
<td>405,000</td>
</tr>
<tr>
<td>IB3-2</td>
<td>40</td>
<td>20</td>
<td>20</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>18</td>
<td>300</td>
<td>494,120</td>
<td>468,000</td>
</tr>
<tr>
<td>IB4-1</td>
<td>40</td>
<td>20</td>
<td>20</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>20</td>
<td>300</td>
<td>494,120</td>
<td>648,000</td>
</tr>
<tr>
<td>IB4-2</td>
<td>40</td>
<td>20</td>
<td>20</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>20</td>
<td>300</td>
<td>803,720</td>
<td>2400</td>
</tr>
<tr>
<td>IB5-1</td>
<td>50</td>
<td>25</td>
<td>25</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>23</td>
<td>350</td>
<td>725,350</td>
<td>1,152,000</td>
</tr>
<tr>
<td>IB5-2</td>
<td>50</td>
<td>25</td>
<td>25</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>23</td>
<td>350</td>
<td>1,180,950</td>
<td>1,260,000</td>
</tr>
<tr>
<td>IB6-1</td>
<td>60</td>
<td>30</td>
<td>30</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>25</td>
<td>400</td>
<td>1,008,880</td>
<td>1,296,000</td>
</tr>
<tr>
<td>IB6-2</td>
<td>60</td>
<td>30</td>
<td>30</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>25</td>
<td>400</td>
<td>1,627,680</td>
<td>1,728,000</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>'Low'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP1-1</td>
<td>2248</td>
<td>19,874</td>
<td>0.05</td>
<td>0.092</td>
<td>2</td>
</tr>
<tr>
<td>IP1-2</td>
<td>2248</td>
<td>22,015</td>
<td>0.04</td>
<td>0.103</td>
<td>5</td>
</tr>
<tr>
<td>IP2-1</td>
<td>4982</td>
<td>28,624</td>
<td>0.08</td>
<td>0.170</td>
<td>2</td>
</tr>
<tr>
<td>IP2-2</td>
<td>4982</td>
<td>81,728</td>
<td>0.08</td>
<td>0.205</td>
<td>3</td>
</tr>
<tr>
<td>IB1-1</td>
<td>7244</td>
<td>138,437</td>
<td>0.12</td>
<td>0.291</td>
<td>2</td>
</tr>
<tr>
<td>IB1-2</td>
<td>7244</td>
<td>487,471</td>
<td>0.13</td>
<td>0.317</td>
<td>5</td>
</tr>
<tr>
<td>IB2-1</td>
<td>7959</td>
<td>507,905</td>
<td>0.12</td>
<td>0.284</td>
<td>4</td>
</tr>
<tr>
<td>IB2-2</td>
<td>7959</td>
<td>676,346</td>
<td>0.17</td>
<td>0.423</td>
<td>5</td>
</tr>
<tr>
<td>IB3-1</td>
<td>10,920</td>
<td>1,048,952</td>
<td>0.20</td>
<td>0.521</td>
<td>9</td>
</tr>
<tr>
<td>IB3-2</td>
<td>14,244</td>
<td>569,975</td>
<td>0.28</td>
<td>0.798</td>
<td>16</td>
</tr>
<tr>
<td>'Medium'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IB1-1</td>
<td>2448</td>
<td>20,126</td>
<td>0.05</td>
<td>0.113</td>
<td>2</td>
</tr>
<tr>
<td>IB1-2</td>
<td>2448</td>
<td>31,871</td>
<td>0.04</td>
<td>0.103</td>
<td>5</td>
</tr>
<tr>
<td>IB2-1</td>
<td>4982</td>
<td>29,054</td>
<td>0.08</td>
<td>0.170</td>
<td>2</td>
</tr>
<tr>
<td>IB2-2</td>
<td>4982</td>
<td>87,925</td>
<td>0.15</td>
<td>0.205</td>
<td>5</td>
</tr>
<tr>
<td>IB3-1</td>
<td>7244</td>
<td>132,308</td>
<td>0.13</td>
<td>0.291</td>
<td>4</td>
</tr>
<tr>
<td>IB3-2</td>
<td>7244</td>
<td>512,195</td>
<td>0.13</td>
<td>0.317</td>
<td>4</td>
</tr>
<tr>
<td>IB4-1</td>
<td>7959</td>
<td>533,666</td>
<td>0.12</td>
<td>0.284</td>
<td>4</td>
</tr>
<tr>
<td>IB4-2</td>
<td>7959</td>
<td>705,208</td>
<td>0.12</td>
<td>0.423</td>
<td>5</td>
</tr>
<tr>
<td>IB5-1</td>
<td>10,920</td>
<td>1,279,175</td>
<td>0.17</td>
<td>0.521</td>
<td>9</td>
</tr>
<tr>
<td>IB5-2</td>
<td>10,920</td>
<td>o</td>
<td>0.17</td>
<td>0.798</td>
<td>10</td>
</tr>
<tr>
<td>'High'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IB1-1</td>
<td>2448</td>
<td>20,936</td>
<td>0.02</td>
<td>0.047</td>
<td>1</td>
</tr>
<tr>
<td>IB1-2</td>
<td>2448</td>
<td>43,979</td>
<td>0.05</td>
<td>0.119</td>
<td>4</td>
</tr>
<tr>
<td>IB2-1</td>
<td>4982</td>
<td>29,613</td>
<td>0.09</td>
<td>0.204</td>
<td>2</td>
</tr>
<tr>
<td>IB2-2</td>
<td>4982</td>
<td>90,205</td>
<td>0.07</td>
<td>0.173</td>
<td>3</td>
</tr>
<tr>
<td>IB3-1</td>
<td>7244</td>
<td>145,348</td>
<td>0.14</td>
<td>0.291</td>
<td>5</td>
</tr>
<tr>
<td>IB3-2</td>
<td>7244</td>
<td>772,277</td>
<td>0.17</td>
<td>0.335</td>
<td>7</td>
</tr>
<tr>
<td>IB4-1</td>
<td>7959</td>
<td>932,506</td>
<td>0.19</td>
<td>0.390</td>
<td>8</td>
</tr>
<tr>
<td>IB4-2</td>
<td>7959</td>
<td>2,714,539</td>
<td>0.13</td>
<td>0.462</td>
<td>9</td>
</tr>
<tr>
<td>IB5-1</td>
<td>10,920</td>
<td>1,655,353</td>
<td>0.21</td>
<td>0.483</td>
<td>16</td>
</tr>
<tr>
<td>IB5-2</td>
<td>10,920</td>
<td>o</td>
<td>0.20</td>
<td>0.504</td>
<td>18</td>
</tr>
<tr>
<td>IB6-1</td>
<td>14,244</td>
<td>o</td>
<td>0.29</td>
<td>0.655</td>
<td>18</td>
</tr>
<tr>
<td>IB6-2</td>
<td>14,244</td>
<td>o</td>
<td>0.30</td>
<td>0.694</td>
<td>19</td>
</tr>
</tbody>
</table>

handled by a quay crane with the shortest distance in a different time period. This is possible since the makespan for the ship is equal within the instance classes. As a result, every difficulty level will result in the same optimal distance.

A graphical representation of changing the congestion level on the position of crane 1 in problem instance IB1-1 is illustrated in Fig. 4. The x-axis denotes each time period $T$ and the y-axis denotes the bay number relative to one side of the ship. For example, bay number 1 represents bay 1 and bay 11, bay number 2 represents bay 2 and bay 12, and so forth until the last bay number 10 represents bay 10 and bay 20. The lines indicate the position of crane 1 and the dots indicate tasks performed by crane 1. What is interesting from the figure is that there is no difference in the optimal sequence for the low and medium congestion levels. On the other hand, changing the congestion level to high does have an effect on the optimal sequence. This indicates that at certain time periods the yard block occupation threshold restricts the crane to handling containers for a particular bay. The effect of changing other input parameters is discussed in the next section, where a sensitivity analysis will be performed.

A graphical representation of the optimal quay crane positioning and handling scheme of problem instance IB1-1 with low congestion level is depicted in Figs. 5 and 6. Fig. 5 illustrates the optimal solution for crane 1 and crane 2 positioned on the left and Fig. 6 shows the solutions applied to crane 3 and crane 4 positioned on the right. Again, the x-axis denotes each time period $T$ and the y-axis represents the bay number relative to one side of the ship. The lines indicate the position of the quay cranes relative to the relative bay number and the crosses and triangles indicate if a quay crane is performing a task at a particular bay.
It is apparent from the figures that the lines of the cranes mounted on the same side never cross and abide by the clearance constraints at all times. It can be concluded that the ordering and non-crossing constraints are satisfied in all time periods. Please note that the solutions follow the assumption of relatively small gantry distances. Another important constraint is the constraint ensuring that only one quay crane can handle a bay and no other quay crane can handle the bay with the same relative bay number on the opposite side of the ship. The following figure (Fig. 7) shows that this constraint is met at all times for IB1-1. Similarly, the y-axis denotes the relative bay number and the x-axis denotes time period. A square, triangle, cross, or rectangle indicates that a crane related to the particular marker is handling a container at the relative bay number corresponding to the y-axis.

Summarizing the numerical results, we can conclude that the B&PIB algorithm is able to solve the given practical instances for
quay crane scheduling for an indented berth with yard congestion and workload stability constraints. The method abides by the constraints at all times. It is apparent from this section that for the cases that a solution is obtainable by the compact model, the B&PIB performances are significantly better compared to the performances of the compact model. In the following section the effect of changing input parameters on the runtime will be investigated by a sensitivity analysis.

4.3. Sensitivity analysis with respect to runtime

Sensitivity analysis is used to determine how input parameters influence the runtime of the compact model and the B&PIB model. The objective is to investigate the effect of $H$, $C_{T}$, $T$, $B$, $r$ and the number of containers on the runtime of the models. In order to perform a sensitivity analysis, a baseline benchmark case is selected and the input parameters are varied one by one. The baseline benchmark case is chosen to allow both models to provide a reasonable runtime. Hence, in this set of experiments, we do not work on large instances in order to be able to analyze the solutions obtained by the compact model. We set the baseline case with $H = 30$, $C_{T} = 2$, $C_{R} = 2$, $T = 275$, $B = 8$, $r = 1$, with the number of containers as 480. In the instances created for the sensitivity analysis one task is equal to handling three containers and a time period corresponds to seven minutes, causing the real makespan for the benchmark case to be 32 hours and the number of containers in the stowage plan to be 1440. In order to illustrate the outcomes, the results are depicted in figures. In all figures corresponding to this section, the x-axis is related to the value of the input parameter of interest and all figures have two different y-axes. The right y-axis corresponds to the runtime of the compact model and the left y-axis corresponds to the runtime of the B&PIB.

**Sensitivity Analysis of Bays.** The container shipping industry continues to build larger vessels. In an indented berth ships of great sizes can berth. A bigger ship contains more bays than a smaller ship. In order to identify the effect of a change in the input parameter bay, $H$, two approaches are adopted. The first approach is to increase the number of bays keeping the number of containers in the stowage plan constant and the second approach is to increase the number of bays and the number of containers. Both cases can be experienced in real life practice. The process of increasing bays is done in steps of 5 bays per side, resulting in steps of 10 bays in total each time. The range of the total number of bays in the ship increases from 10 up to 60. In the left chart of Fig. 8, the runtimes of both methods on the first approach are depicted with an exponential $y$-axis. As can be seen, the runtime of the compact model does increase with the enlargement of the number of bays without increasing the predetermined number of containers, whereas the effect on the runtimes of the B&PIB algorithm remains very small. In the right chart of Fig. 8, the runtimes of both methods on the second approach are depicted with an exponential $y$-axis. In every step 160 containers are added to the load profile. In the chart there is a clear trend of increasing runtimes for the compact model. In contrast, increasing the number of bays and thus the number of containers has a small but increasing effect on the runtime of the B&PIB.

**Sensitivity Analysis of Cranes.** In an indented berth as many as twelve cranes, six on each side, can be positioned on top of the holds. This is double the number of quay cranes that can work on a
ship in traditional berths. The effect of changing the input parameter for cranes, C, is depicted in the figure below (Fig. 9). The number of cranes ranges from 4 to 12. From the chart, it can be seen that the change in runtime of the B&PIB algorithm is almost negligible as the number of cranes are increased. On the other hand, increasing the number of cranes has a clear effect on the runtime of the compact model. The runtime of the compact model increases with an increase in the number of cranes. One reason for the increasing trend of the runtime with an increase in the number of quay cranes is that the models have to deal with more constraints, however the B&PIB algorithm is more robust to this change.

**Sensitivity Analysis of Clearance.** Fig. 10 shows the results from the sensitivity analysis of the clearance value. Increasing the clearance value, means that quay cranes are restricted to being more bays away to the left and right if a quay crane is positioned at a particular bay. To compare the runtimes of the compact model and the B&PIB, the clearance value ranges from 0 up to 4. The figure (Fig. 10) illustrates that the compact model is affected slightly by a change in the parameter, while the runtime of the B&PIB is almost negligible to increases. The lack of increase in the runtime may partly be explained by the fact that an increase in clearance value does not increase the number of constraints and is not sufficient enough to constrain the problem further for this specific instance where we have deliberately allowed for a less complex problem for comparison purposes with the compact formulation.

**Sensitivity Analysis of Blocks.** The effect of changing the number of yard blocks, B, is illustrated in Fig. 11. It has commonly been assumed that the number of yard blocks in the yard is 4. One big yard block contains one unloading section and one loading section. The size of the yard blocks differs relative to their position in the yard. For a sensitivity analysis, the minimum number of yard blocks is assumed to be 2. Hence in the smallest instance only the yard blocks in the yard are included. In the next step of the sensitivity analysis the number of yard blocks is 3. Hence, the two yard blocks in the yard and one yard block adjacent to the ship are included. The last step is the benchmark case, including all 4 yard blocks: two in the yard and two adjacent to the ship. As can be seen from the figure below, the trend of the runtime for the compact model is increasing. From the figure a slight increase in the trend of the runtime of the B&PIB can also be observed.

**Sensitivity Analysis of Containers.** The characteristics of the sensitivity analysis on the number of containers in the load profile are somewhat different than the characteristics of the other sensitivity analyses. This is because the different sizes of the yard blocks impose a limit on the increase of the number of containers in adjacent yard blocks where as the limit of the increase of the number of containers in yard blocks in the yard might not have been reached, causing the increase in the yard blocks to be disproportional. As stated previously, the yard consists of 4 big yard blocks divided into 2 subsections each. In one subsection unloaded containers will be stacked and in the other subsection containers for loading are stacked. The range of number of containers for a sensitivity analysis is from 1440 to 3600, in steps of approximately 270 containers. Another difference with the other sensitivity analyses is that the benchmark case is changed slightly and that after there is a maximum number of containers in yard blocks adjacent to the ship.

The original benchmark case is changed slightly, by setting the input parameter $T = 400$. The reason for this deviation is that the largest instance for the sensitivity analysis contains 3600 containers. In every time period a quay crane can handle 3 containers in the same bay. It is only possible to handle 3600 containers in 200 time periods if in every time period all quay cranes are assigned and no wastage is sequenced. Therefore, the benchmark case for a sensitivity analysis of the number of containers in the load profile is $H = 30$, $C_l = 2$, $C_R = 2$, $T = 400$, $B = 8$, $r = 1$, and the number of containers is 480 (see Fig. 12).

It can be safely concluded that increasing any of the input parameters causes a significant increase in the runtime of the compact model. In contrast, the effect on the B&PIB is almost
negligible. No significant increases were found when increasing cranes, time, clearance or the number of containers. An increasing but very small effect is recorded when increasing the bays or blocks. The B&PIB algorithm is therefore assumed to be a robust method for solving the quay crane scheduling problem for an indented berth.

5. Industrial point of view on the indented berth layout

The proposed model is based on optimizing quay crane scheduling for an indented berth. The B&PIB algorithm results in rapid runtimes guaranteeing optimal solutions. This indented berth layout is a novel structure and although there is a need for such game changers the concept is not yet common. However, it is expected to be implemented, considering its performance compared to traditional settings. In this section we would like to discuss its previous implementation and the important aspects for the industry to consider.

The first indented berth was built in the Amsterdam Container Terminal formerly known as the Ceres Paragon Terminal. The Ceres Paragon Terminal was in operation from 2001 to 2013. The terminal was designed to handle large containerships with an annual capacity of 950,000 containers. With 9 quay cranes available to handle one ship, the overall quay crane capacity was about 250–300 container units per hour. Compared to traditional berths such as in the Rotterdam ECT Terminal, where physical limitations impose a maximum of 5 quay cranes per ship, the Ceres Paragon indented berth allowed for roughly double the amount of handling equipment. This increase was achieved by doubling the ship-to-shore interface with an indented berth allowing the containers to be handled from both sides. The performance in movements per hour of traditional berths is a maximum of 110 container units per hour, which is much lower than the indented berths. In spite of this, the Ceres Paragon Terminal was closed in 2013. This decision was due to external factors around the terminal such as the navigational difficulties in reaching the port, the limited sea canal draught and the closely located Port of Rotterdam (Young, 2012). This may help to understand the closing of the Ceres Paragon Terminal and suggests that the fate of indented berths also relies on nautical access and hinterland connections. Decision makers in the industry should therefore carefully consider the market structure next to the performance gains possible with the novel infrastructure.

We would like to further compare the possible use of the model proposed by this paper with a traditional berth setting to give an indication of the differences between the two approaches.

In a traditional berth, ships can be handled only from one side. This causes the distance between bays and yard locations to be approximately the same for all quay cranes. Therefore, the aim of minimizing the distances traveled may not be of great concern. The terminals then struggle to meet the deadlines imposed by the shippers. On the other hand, yard congestion plays a greater role in an indented berth setting as the number of containers being transferred within the yard is approximately doubled. For both layouts, the workload on the ship must be in balance at all times in order to keep the ship from tilting. However, it should be noted that this would be less of a concern in a traditional berth setting as there is less complexity during the operations. An important aspect that should be considered with the traditional berth setting is the need for additional time to be able to complete the same number of tasks which would otherwise lead to infeasibilities. As demonstrated in practice, this need can be almost twice as great depending on the yard congestion levels and vessel specific balance threshold levels.

6. Conclusion

Indented berths are of major interest since they allow ships to be handled by two rows of quay cranes on both sides. Hence, in comparison to a traditional berth, indented berths significantly increase the ship-to-shore interface and the maximum number of quay cranes handling a vessel. This new architecture in the container terminal layout requires new models and solution techniques to be built. Mainly, the new structure brings differences in handling yard operations and crane constraints. The distances traveled within the yard for the transportation of the containers becomes more significant and opportunities to minimize this arises. There is also an added challenge in preserving the stability of the vessel. Considering balance constraints at an indented berth is crucial since this prevents a ship from tilting while at berth by restricting to a latitudinal-balance threshold. Up to now, little attention has been paid to the problem and to the best of our knowledge, no dominant model considering these features has been built. Accordingly, this paper contributes to the literature by providing a model formulation and developing an effective solution technique based on branch-and-price algorithm in order to solve the indented berth quay crane scheduling problem.

We first provide a compact formulation of the problem and then develop a technique based on branch-and-price algorithm. Findings from the compact model show that runtime increases considerably as problem instances become larger. In fact, the compact model is unable to solve larger instances within time limits. These findings suggest that a solution method has to be established in order to find an optimal solution for large instances. Hence, we provide optimal solutions for practical instances to the quay crane scheduling for an indented berth with yard congestion and balance constraints through a branch-and-price approach. The B&PIB algorithm decreases the runtime significantly with respect to the compact model. We provide thorough experiments to analyze the impact of different vessel sizes, load profile, handling equipment and terminal layout. Increasing the size of the problem has significant effect on the performance of the compact model. The branch-and-price approach is much more robust reacting mainly to the changes in the vessel size. Experiments validate the applicability of the solution technique. As a future work, a different perspective on the indented berth layout setting can be followed considering different performance criteria. In practical settings where agreed deadlines are not set, reducing the service time as much as possible may provide opportunities for future handling tasks. The problem setting should indeed be formed regarding the relevance in the practical problem under analysis.


