European Journal of Operational Research 216 (2012) 47–56

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Berth allocation with time-dependent physical limitations on vessels

Dongsheng Xu [a,1], Chung-Lun Li [b,2], Joseph Y.-T. Leung [c,*]

[a] Department of Management Science, Business School, Sun Yat-sen University, Guangzhou 510275, China
[b] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[c] Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

ARTICLE INFO

Article history:
Received 12 August 2010
Accepted 7 July 2011
Available online 20 July 2011

Keywords:
Scheduling
Berth allocation
Container terminals
Parallel machines
Tidal condition

ABSTRACT

We consider a berth allocation problem in container terminals in which the assignment of vessels to berths is limited by water depth and tidal condition. We model the problem as a parallel-machine scheduling problem with inclusive processing set restrictions, where the time horizon is divided into two periods and the processing sets in these two periods are different. We consider both the static and dynamic cases of the problem. In the static case all of the vessels are ready for service at time zero, while in the dynamic case the vessels may have nonzero arrival times. We analyze the computational complexity and develop efficient heuristics for these two cases. Computational experiments are performed to test the effectiveness of the heuristics and to evaluate the benefits of taking tidal condition into consideration when making berth allocation decisions.

1. Introduction

We consider a berth allocation problem in container terminals in which the assignment of vessels to berths is limited by water depth and tidal condition. The container terminal concerned has multiple berths of different water depths (i.e., different maximum vessel draft restrictions) so that some berths can accommodate all vessel types, while other berths can only accommodate vessels of shallower drafts. In addition, due to different tide levels at different hours of the day, whether a vessel can be assigned to a berth is also dependent on the time when the mooring takes place. We model this by dividing the planning horizon into low-water and high-water periods, where the vessel assignment in a low-water period is more restrictive. The objective is to minimize the total weighted service duration of the vessels, where the "weights" reflect the unit waiting costs of vessels.

This research is motivated by the operation of a terminal operator which manages a number of container terminals in the Pearl River Delta area of China. This company faces big challenges in berth planning due to the tidal effect, which is quite typical to those terminals located along a river. On the one hand, the company needs to provide efficient loading and unloading service to the vessels with its limited terminal capacity. On the other hand, the periodic change in water depth increases the difficulty of serving deep-draft vessels. Careless berth assignments often result in accidents, where stranded vessels can cost the company millions of yuans. Hence, it is desirable to take into account the tidal information when planning the berth usage. Detailed tidal information is available from tide tables, which show the water level at different hours of different days at different locations. For simplicity, in this research we consider only a short planning horizon (say 24 h or less) which contains one high tide and one low tide. In addition, we divide that planning horizon into a low-water period and a high-water period, and ignore the detailed water level variations within each of these two periods (see Section 6 for a discussion of the limitations and possible extensions of our model).

Various berth allocation models have appeared in the literature. A static berth allocation model assumes that all vessels arrive before the berth allocation decision is made, while a dynamic model considers also vessels with known future arrival information. Imai et al. (1997) formulate a static berth allocation problem as a nonlinear integer program to minimize both the total time that the vessels spend at the berth and the degree of dissatisfaction incurred by the berthing order. Imai et al. (2001, 2005a) introduce a dynamic version of the problem, where each vessel has a given arrival time, and the vessels can only be serviced after their arrival. The objective is to minimize the total waiting and handling time of the vessels. Nishimura et al. (2001) study a dynamic berth allocation problem in a public berth system in which a vessel must be served by a berth with acceptable physical conditions and the handling time of each vessel is dependent on the berth it is served. Imai et al. (2003) consider a dynamic berth allocation problem in which different vessels have different service priorities. Cordeau et al. (2005) present a tabu search heuristic for a dynamic berth

* Corresponding author. Tel.: +1 973 596 3387; fax: +1 973 596 5777.
E-mail addresses: xudsh@mail.sysu.edu.cn (D. Xu), lgtclli@polyu.edu.hk (C.-L. Li), leung@cis.njit.edu (J.Y.-T. Leung).
[1] Tel.: +86 20 8411 3192.
[2] Tel.: +852 2766 7410.

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.07.012

allocation problem. Monaco and Sammarra (2007) give a more compact formulation than that of Imai et al. (2001) and solve it via Lagrangian relaxation. Imai et al. (2007b) analyze a two-objective berth allocation problem which minimizes service time and delay time. Cheong and Tan (2008) develop a multi-colony ant algorithm for Nishimura et al.'s (2001) model. Hansen et al. (2008) present a minimum cost berth allocation problem, which is an extension of Imai et al.'s (2003) model, and develop a variable neighborhood search heuristic for solving it. Imai et al. (2008b) study a variation of Imai et al.'s (2001) problem in which an external terminal is available when there is a lack of berth capacity at the operator's own terminal. Barros et al. (2011) develop and analyze a berth allocation model with tidal time windows (i.e., time periods with high tide), where ships can only be served by berths during those time windows. Buhrkal et al. (2011) study several mathematical programming formulations of the dynamic berth allocation problem, and compare their performance via computational experiments. In all these models, it is assumed that there are multiple berths available, each berth can handle one ship at a time, and each ship occupies only one berth. We also make these assumptions in our model.

Brown et al. (1994) and Brown et al. (1997) consider berth allocation models for surface vessels and submarines, respectively. In their models, vessels can be nested alongside one another at a berth; that is, two or more vessels occupying a single berth position is allowed. Imai et al. (2007a) consider a berth allocation problem in which up to two vessels can be served by the same berth simultaneously.

Another line of berth allocation research assumes that berths along a quayside are resources to be shared by different vessels. In other words, each berth is viewed as a continuous line, where a berthing point along this line and a berthing time are needed to be determined for every vessel. This line of research includes the works of Chia et al. (1999), Park and Kim (2002), Kim and Moon (2003), Imai et al. (2005b), Wang and Lim (2007), Dai et al. (2008), Lee and Chen (2009), Tang et al. (2009), Cheong et al. (2010), and Lee et al. (2010), who propose different solution approaches to berth allocation with this "continuous" setting. Some authors model this type of berth allocation problems by discretizing the continuous berth into multiple sections and allowing each vessel to occupy consecutive sections. These include Li et al. (1998) and Guan et al. (2002), who model berth allocation as machine scheduling problems with multiprocessor tasks, as well as Guan and Cheung (2004) who develop efficient heuristics using a discretized berthing section model.

Other related studies include Lai and Shih (1992), who evaluate different berthing policies via simulation, Lim (1998), who analyzes a berth planning problem with an objective of minimizing the maximum amount of space used in a section of a container port, and Moorthy and Teo (2006), who study the optimal allocation of home berths (i.e., preferred berthing locations) to different sets of vessels. Some research also incorporates quay crane scheduling into berth allocation decisions; see Park and Kim (2003), Imai et al. (2008a), Liang et al. (2009a,b), Meisel and Bierwirth (2009), and Blazewicz et al. (2011). Steenken et al. (2004), Stahlbock and Voss (2008), and Bierwirth and Meisel (2010) provide comprehensive surveys on berth allocation related research.

While some of the above-mentioned berth allocation models take physical conditions such as water-depth into consideration (e.g., Nishimura et al., 2001; Barros et al., 2011), none of them considers the assignment of ships to berths to different time periods with different water-depths. In this paper we develop and analyze a new berth allocation model which captures the water-depth and tidal constraints that occur in practice. As mentioned above, we consider a short planning horizon which covers a low-water period and a high-water period. There are multiple berths serving the vessels in parallel, which we can view as a parallel-machine scheduling model. The water-depth constraint is modeled as "inclusive processing sets" (see Leung and Li, 2008 and Ou et al., 2008 for detailed discussions of inclusive processing sets). We consider both the static and dynamic versions of the model, and present different solution methods for them.

The rest of the paper is organized as follows: In the next section we define our problem mathematically. In Section 3 we discuss the computational complexity and approximability of both the static and dynamic versions of the problem. In Section 4 we propose efficient heuristic solution methods for these two versions. Computational experiments are performed and the results are presented in Section 5. Some concluding remarks are provided in Section 6.

2. Model description

To define our problem mathematically, we first divide the time line $[0,\infty)$ into two intervals $[0,T]$ and $[T,\infty)$, where $[0,T]$ is a low-water period and $[T,\infty)$ is a high-water period. Another scenario is when $[0,T]$ is a high-water period and $[T,\infty)$ is a low-water period. The solution methods described in this paper can handle both scenarios. Here, T represents the time point at which the tide level has reached a certain threshold where the berthing of vessels becomes either less restrictive or more restrictive. In our model we define the second time period to be infinitely long. In reality, the tide level goes up and down periodically. Thus, to apply this model, the problem needs to be solved in a rolling horizon basis such that the schedule is updated periodically, say every T time units.

There are n vessels (i.e., vessels $1,2,\ldots,n$) and m berths (i.e., berths $1,2,\ldots,m$). For $j=1,2,\ldots,n$, vessel j has a given processing time $p_j>0$, arrival time $a_j\geqslant 0$, weight $w_j>0$, "high-water berth index" H_j, and "low-water berth index" L_j, where $H_j, L_j \in \{1,2,\ldots,m\}$. During a high-water period, vessel j can be served by any one of berths $H_j, H_j+1,\ldots,m$, but not by any of berths $1,2,\ldots,H_j-1$. During a low-water period, vessel j can be served by any one of berths $L_j, L_j+1,\ldots,m$, but not by any of berths $1,2,\ldots,L_j-1$. Thus, the berths are linearly indexed in such a way that at any point in time, berth i can serve all those vessels that berth $i-1$ can serve ($i=2,3,\ldots,m$). In other words, the water depth at berth i is no less than that at berth $i-1$. In addition, $L_j\geqslant H_j$ for $j=1,2,\ldots,n$. Thus, if vessel j can be served by berth i in a low-water period, then it can also be served by berth i in a high-water period. Note that in our model we are expressing the tidal constraints in terms of high-water berth indices and low-water berth indices. In practice, when the draft of each vessel and the water depths in time intervals $[0,T]$ and $[T,\infty)$ are given, it is easy to convert such information into high-water berth indices and low-water berth indices (see example below).

Let C_j denote the completion time of vessel j. The objective is to determine a feasible schedule for the berths to serve all vessels such that $\sum_j w_j(C_j-a_j)$ is minimized. Here, C_j-a_j is the service duration (i.e., waiting time plus processing time) of vessel j, while w_j reflects the unit waiting cost of vessel j.

We assume that each berth can handle one vessel at a time, and that each vessel is served by only one berth. Once a vessel j has started its service at a berth, it will stay there for the next p_j units of time without interruption. We also assume that m is fixed, and that T, p_j, w_j, and a_j ($j=1,2,\ldots,n$) are all integer valued. We denote this problem as $\mathbf{P}$. We refer to the general problem $\mathbf{P}$ as the "dynamic case." We refer to the special case with $a_j=0$ ($j=1,2,\ldots,n$) as the "static case." Here, the static case represents the situation where the scheduling is done by considering only the vessels that have already arrived at the port, while the future arrivals are ignored (see, e.g., Bierwirth and Meisel, 2010).

Note that problem $\mathbf{P}$ can be viewed as a parallel-machine scheduling problem, with a vessel being a job and a berth being a machine. Job j has a processing time p_j, a release date a_j, a weight w_j, a set of machines $\{H_j, H_j+1,\ldots,m\}$ that are capable of processing job j during a high-water period, and a set of machines $\{L_j, L_j+1,\ldots,m\}$ that are capable of processing job j during a low-water period. The machine subsets $\{H_j, H_j+1,\ldots,m\}$ and $\{L_j, L_j+1,\ldots,m\}$ are the "processing sets" of job j. Problem **P** differs from the traditional scheduling problem with inclusive processing set restrictions in that the processing set of a job is now time-dependent (see, e.g., Ou et al., 2008 for the traditional scheduling model with inclusive processing set restrictions). More specifically, the processing set of a job now depends on whether any portion of the job is processed during the low-water period.

Consider an example with $m = 3$, $n = 6$, $T = 12$, and other data as shown in Fig. 1(a), where $[0,T]$ is a low-water period and $[T,\infty)$ is a high-water period. In this example, all vessel arrival times are zero. Hence, it is a static case. During the low-water period, the drafts of vessels 1 and 6 are less than or equal to the water depths of all three berths, and therefore $L_1 = L_6 = 1$; the drafts of vessels 2, 3, 4, and 5 are greater than the water depths of berths 1 and 2 but are less than or equal to the water depth of berth 3, and therefore $L_2 = L_3 = L_4 = L_5 = 3$. During the high-water period, the drafts of vessels 1 and 6 are less than the water depths of all three berths, and therefore $H_1 = H_6 = 1$; the drafts of vessels 2, 3, and 5 are greater than the water depth of berth 1 but are less than the water depths of berths 2 and 3, and therefore $H_2 = H_3 = H_5 = 2$; the draft of vessel 4 is greater than the water depths of berths 1 and 2 but is less than the water depth of berth 3, and therefore $H_4 = 3$. The L_j and H_j values are summarized in Fig. 1(b). Note that in our model we have ignored the change in vessel draft during the cargo loading and unloading process. When our model is applied, the vessel draft is defined as the maximum draft during the service time of the vessel at the container terminal.

A feasible schedule of this example is depicted in Fig. 1(c). In this schedule, the completion times of the vessels are $C_1 = 6$, $C_2 = 23$, $C_3 = 19$, $C_4 = 14$, $C_5 = 8$, and $C_6 = 13$. The total weighted vessel completion time is $(4)(6) + (1)(23) + (4)(19) + (4)(14) + (8)(8) + (2)(13) = 269$. Note that, for example, vessel 3 can only be served by berth 3 during low-water period $[0,T]$, and therefore if it is assigned to berth 2, then it can only start its processing at or after time T. Note also that during the low-water period $[0,T]$, vessel 4 can only be served by berth 3, and therefore it cannot be assigned to berth 2 even though berth 2 is idle during the time interval $[6,12]$.

Problem **P** can be formulated as a mixed integer linear program (MILP). Let $x_{ij} = 1$ if vessel j is assigned to berth i; let $x_{ij} = 0$ otherwise. Let $I_{ijj'} = 1$ if vessels j, j' are both assigned to berth i and vessel j is processed before vessel j'; let $I_{ijj'} = 0$ otherwise. Let s_j be the start time of processing of vessel j. Let M be a large constant. Then, the following MILP formulation is for the scenario where $[0,T]$ is the low-water period and $[T,\infty)$ is the high-water period:

$$\text{minimize} \sum_{j=1}^{n} w_j(s_j + p_j - a_j) \tag{1}$$

$$\text{subject to} \sum_{i=1}^{m} x_{ij} = 1 \quad (j = 1,2,\ldots,n) \tag{2}$$

$$s_j \geqslant a_j \quad (j = 1,2,\ldots,n) \tag{3}$$

$$s_{j'} \geqslant s_j + p_j - M(1 - I_{ijj'}) \quad (j, j' = 1,2,\ldots,n \text{ s.t. } j \neq j'; i = 1,2,\ldots,m) \tag{4}$$

$$I_{ijj'} + I_{ij'j} \leqslant \frac{1}{2}(x_{ij} + x_{ij'}) \quad (j, j' = 1,2,\ldots,n \text{ s.t. } j < j'; i = 1,2,\ldots,m) \tag{5}$$

vessel j	1	2	3	4	5	6
a_j	0	0	0	0	0	0
w_j	4	1	4	4	8	2
p_j	6	9	7	6	8	13
draft*	10	16	18	20	18	10

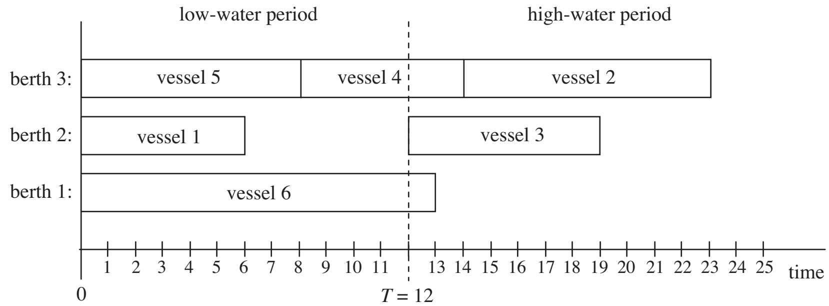
*The draft of a vessel includes the safety vertical distance for berthing

berth i	1	2	3
water depth (low-water period)	10	15	20
water depth (high-water period)	14	19	24

(a) problem data

vessel j	1	2	3	4	5	6
L_j	1	3	3	3	3	1
H_j	1	2	2	3	2	1

(b) the L_j and H_j values derived from the problem data



(c) a feasible solution to problem **P**

Fig. 1. A numerical example.

$$I_{ijj'} + I_{ij'j} \geqslant x_{ij} + x_{ij'} - 1 \quad (j, j' = 1, 2, \ldots, n \text{ s.t.} j < j'; i = 1, 2, \ldots, m) \tag{6}$$

$$x_{ij} = 0 \quad (j = 1, 2, \ldots, n; i = 1, 2, \ldots, H_j - 1) \tag{7}$$

$$s_j \geqslant T x_{ij} \quad (j = 1, 2, \ldots, n; i = 1, 2, \ldots, L_j - 1) \tag{8}$$

$$x_{ij} \in \{0, 1\} \quad (j = 1, 2, \ldots, n; i = 1, 2, \ldots, m) \tag{9}$$

$$I_{ijj'} \in \{0, 1\} \quad (j, j' = 1, 2, \ldots, n \text{ s.t. } j \neq j'; i = 1, 2, \ldots, m) \tag{10}$$

Objective function (1) minimizes $\sum_j w_j (C_j - a_j)$, where the completion time of vessel j is given as $C_j = s_j + p_j$. Constraint (2) requires each vessel to be assigned to one berth. Constraint (3) requires that each vessel can start its processing only after it has arrived at the terminal. Constraint (4) states that if vessels j, j' are both assigned to berth i and vessel j is processed before vessel j' (i.e., $I_{ijj'} = 1$), then the start time of vessel j' must be no earlier than $s_j + p_j$. Constraints (5) and (6) ensure that one of $I_{ijj'}$ and $I_{ij'j}$ equals 1 if vessels j and j' are both assigned to berth i. They also ensure that $I_{ijj'} = I_{ij'j} = 0$ if one of vessels j and j' is not assigned to berth i. Constraint (7) disallows vessel j from being assigned to berths $1, 2, \ldots, H_j - 1$. Constraint (8) disallows vessel j from being processed by berths $1, 2, \ldots, L_j - 1$ during period $[0, T]$.

An MILP formulation for the scenario with high-water period $[0, T]$ and low-water period $[T, \infty)$ can be obtained by replacing constraint (8) with the following constraint:

$$s_j \leqslant T - p_j + M'(1 - x_{ij}) \quad (j = 1, 2, \ldots, n; i = 1, 2, \ldots, L_j - 1), \tag{11}$$

where M' is a large constant.

3. Computational complexity and approximability

In this section we analyze the computational complexity of both the static and dynamic cases of problem **P**. Notice that since $\sum_j w_j a_j$ is a constant, minimizing $\sum_j w_j (C_j - a_j)$ is equivalent to minimizing $\sum_j w_j C_j$, which is the total weighted completion time of the vessels. When $m = 2$, $L_j = H_j = 1$, and $a_j = 0$ for all j, problem **P** becomes the two-parallel-machine minimum weighted total job completion time scheduling problem; that is, problem $P2\|\sum w_j C_j$ (see, e.g., Pinedo, 2002 for the notations of different machine scheduling models). Because $P2\|\sum w_j C_j$ is NP-hard, the static case of problem **P** is also NP-hard. Thus, we have the following theorem:

Theorem 1. *Problem* **P** *is NP-hard even when $a_j = 0$ for all $j = 1, 2, \ldots, n$.*

The following lemma provides some important properties of the optimal solution of both the static and dynamic cases of problem **P**. The validity of Theorems 2, 3 and 6 below is based on these properties.

Lemma 1. *Consider any berth i, any time interval $[t_1, t_2]$, and any two vessels j, j' such that both vessels j and j' can be served by berth i throughout the period $[t_1, t_2]$. (i) In any optimal solution, if both vessels j and j' are served consecutively by berth i within time interval $[t_1, t_2]$, then there is no idle time between these two vessels. (ii) In any optimal solution, if $w_j/p_j > w_{j'}/p_{j'}$ and both vessels j and j' are served consecutively by berth i within time interval $[t_1, t_2]$, then vessel j must be served before vessel j'.*

Lemma 1 implies that there exists an optimal solution to the static case of **P** in which those vessels served by berth i during the time interval $[0, T]$, as well as those vessels served by berth i during the time interval $[T, \infty)$, are sequenced in nonincreasing order of w_j/p_j with no idle time between vessels. Sequencing jobs in nonincreasing order of w_j/p_j is the so-called weighted shortest processing time first (WSPT) rule (see Pinedo, 2002). The validity of Lemma 1 is obvious, and its proof is omitted. The next theorem implies that the static case is only NP-hard in the ordinary sense. The proof of the theorem is given in the Appendix.

Theorem 2. *Problem* **P** *is solvable in pseudo-polynomial time when $a_j = 0$ for all $j = 1, 2, \ldots, n$.*

Next, we consider the possibility of developing good approximation algorithms for problem **P**. For simplicity, we consider a slightly restricted case where $p_{max} \leqslant cT$ for some positive constant c. The condition "$p_{max} \leqslant cT$" is reasonable, as we only consider a short planning horizon. The following theorem states that under the scenario where the low-water period is followed by the high-water period, a polynomial-time ϵ-approximation algorithm exists for the static case of the problem. The proof of the theorem is given in the Appendix.

Theorem 3. *If $p_{max} \leqslant cT$ for some positive constant c and if the scenario is low-water period $[0, T]$ followed by high-water period $[T, \infty)$, then there exists a fully polynomial time approximation scheme (FPTAS) for the static case of problem* **P**.

Under the scenario where the high-water period is followed by the low-water period, the existence of an FPTAS is highly unlikely, as stated in the following theorem, where the proof is given in the Appendix.

Theorem 4. *If the scenario is high-water period $[0, T]$ followed by low-water period $[T, \infty)$, then there is no FPTAS for the static case of problem* **P**, *unless P = NP. This property remains valid even if the problem is restricted to the case where $p_{max} \leqslant T$.*

Theorems 3 and 4 indicate that there is a difference in difficulty in finding an effective solution for the two scenarios. We now consider the dynamic case of problem **P**. We have the following theorem:

Theorem 5. *The general (dynamic) case of problem* **P** *is NP-hard in the strong sense.*

The proof of Theorem 5 is straightforward, as problem **P** is a generalization of the single-machine scheduling problem "$1|r_j|\sum(C_j - r_j)$," which is known to be strongly NP-hard (see Brucker, 2007, p. 106, and notice that problem $1|r_j|\sum(C_j - r_j)$ differs from problem $1|r_j|\sum C_j$ only by a constant in the objective function). Theorem 5 implies that the existence of a pseudo-polynomial time optimal algorithm or the existence of an FPTAS for the dynamic case is highly unlikely.

4. Simple heuristics

As discussed in Section 3, the dynamic case of problem **P** is computationally intractable. Although the static case can be solved in pseudo-polynomial time, the computational complexity of such a pseudo-polynomial time algorithm is quite high (see the proof of Theorem 2). In addition, the FPTAS stated in Theorem 3 only applies to one of the scenarios of the static case. Hence, in this section we present simple heuristic solution methods that enable us to obtain good solutions to the problem efficiently.

4.1. A heuristic for the static case

We first present a heuristic for the static case. The heuristic assigns vessels to berths one by one, while taking into consideration the low-water and high-water berth indices of the vessel being assigned, the position to which the vessel should be inserted, and the increment of the objective function value resulted from the assignment. When looking for the position where a vessel is inserted, the heuristic applies the property of Lemma 1. We first focus on the scenario where period $[0,T]$ is the low-water period and period $[T,\infty)$ is the high-water period.

Algorithm 1

1. Let $\Phi := \{1,2,\ldots,n\}$.
2. Let $L'_1 < L'_2 < \cdots < L'_k$ be all the berth indices such that for each $j = 1,2,\ldots,k$, there is a vessel in Φ whose low-water berth index is L'_j. (In other words, $L'_1, L'_2, \ldots, L'_k$ are all the distinct low-water berth indices.)
3. For $j := k$ down to 1, do:
   (A) Let $\Psi := \{i \in \Phi | L_i = L'_j\}$. Let $\Phi := \Phi \backslash \Psi$.
   (B) Sort the vessels in Ψ in nonincreasing order of the ratio of weight versus processing time, i.e., in WSPT order. Denote these vessels as vessels $g_1, g_2, \ldots, g_q$.
   (C) For $u := 1$ to q, do:
      (i) For $v := L'_j$ to m, do:
         (a) Let w'_i and p'_i be the weight and processing time, respectively, of the ith vessel at berth v in the current schedule. Search for the index q_v such that $w'_{q_v-1}/p'_{q_v-1} \geqslant w_{g_u}/p_{g_u} \geqslant w'_{q_v}/p'_{q_v}$ (with ties broken arbitrarily). Here, we use the convention that $w'_0/p'_0 = \infty$ and $w'_{z+1}/p'_{z+1} = 0$, where z is the number of vessels that have been assigned to berth v.
         (b) Calculate the increment of the objective value I_{v,g_u} if vessel g_u is inserted into the q_vth position of berth v.

      (ii) Let $v' := \arg\min_{v=L'_j,L'_j+1,\ldots,m}\{I_{v,g_u}\}$ (with ties broken arbitrarily). Assign vessel g_u to berth v' and insert it into the $q_{v'}$th position.

4. Scan all vessels of each berth in the current schedule. If the start time of vessel j is greater than or equal to T, then remove it from the current schedule and put it back into Φ.
5. Let $H'_1 < H'_2 < \cdots < H'_\ell$ be all the berth indices such that for each $j = 1,2,\ldots,\ell$, there is a vessel in Φ whose high-water berth index is H'_j.
6. For $j := \ell$ down to 1, do:
   (A) Let $\Psi := \{i \in \Phi | H_i = H'_j\}$. Let $\Phi := \Phi \backslash \Psi$.
   (B) Sort the vessels in Ψ in nonincreasing order of the ratio of weight versus processing time, i.e., in WSPT order. Denote these vessels as vessels $g_1, g_2, \ldots, g_r$.
   (C) For $u := 1$ to r, do:
      (i) For $v := H'_j$ to m, do:
         (a) In the current partial schedule, find the first vessel assigned to berth v of which the start time τ is greater than or equal to T. If no such vessel exists, define $\tau = T$. Consider only the positions of which the start time is greater than or equal to τ. Let w'_i and p'_i be the weight and processing time, respectively, of the vessel in the ith position of berth v. Search for the index q_v such that $w'_{q_v-1}/p'_{q_v-1} \geqslant w_{g_u}/p_{g_u} \geqslant w'_{q_v}/p'_{q_v}$.
         (b) Calculate the increment of the objective value I_{v,g_u} if vessel g_u is inserted into the q_vth position of berth v.

      (ii) Let $v' := \arg\min_{v=H'_j,H'_j+1,\ldots,m}\{I_{v,g_u}\}$ (with ties broken arbitrarily). Assign vessel g_u to berth v' and insert it into the $q_{v'}$th position.
7. Scan the final schedule. Remove idle time if there is any, without violating any constraint.

In Steps 3(C)(i)(a) and 6(C)(i)(a), if the search for a position is done via linear search, then the overall running time of Algorithm 1 is $O(n^2)$. If binary search is used in Steps 3(C)(i)(a) and 6(C)(i)(a), then the running time can be reduced to $O(n\log n)$.

For the scenario of high-water period $[0,T]$ followed by low-water period $[T,\infty)$, the algorithm is similar. In this case, we assign the vessels in Step 3 based on their high-water berth indices, while we assign the vessels in Step 6 based on their low-water berth indices. A major difference is in Step 4, where we also need to remove those vessels that start before time T, complete after time T, and are not allowed to be served after time T by the berths to which they are assigned (note: a vessel is not allowed to be served by its assigned berth after time T if its low-water berth index is greater than the index of the berth).

4.2. A heuristic for the dynamic case

We now present a heuristic for the dynamic case. This heuristic, which will be called Algorithm 2, is similar to Algorithm 1. It differs from Algorithm 1 only in Steps 3 and 6. In the dynamic case, when a vessel is inserted into the existing schedule, the start time of the vessel is constrained by the vessel's arrival time. Thus, Steps 3 and 6 are modified by taking into account this constraint. The running time of Algorithm 2 is $O(n^2)$. Again, in the following presentation we will focus on the scenario where period $[0,T]$ is the low-water period and period $[T,\infty)$ is the high-water period. The heuristic can be modified easily for the other scenario as explained in Section 4.1.

Algorithm 2

1. Same as Step 1 of Algorithm 1.
2. Same as Step 2 of Algorithm 1.
3. For $j = k$ down to 1, do:
   (A) Let $\Psi := \{i \in \Phi | L_i = L'_j\}$. Let $\Phi := \Phi \backslash \Psi$.
   (B) Sort the vessels in Ψ in WSPT order. Denote these vessels as vessels $g_1, g_2, \ldots, g_q$.
   (C) For $u := 1$ to q, do:
      (i) For $v := L'_j$ to m, do the following: Let t denote the number of vessels assigned to berth v in the current partial schedule, and let $x_1, x_2, \ldots, x_t$ denote the t vessels at berth v. Let s'_b and p'_b denote the start time and processing time, respectively, of vessel x_b, $1 \leqslant b \leqslant t$. Denote $s'_0 = p'_0 = 0$. For $b = 1,2,\ldots,t+1$, calculate the increment in objective function value if vessel g_u is inserted into the bth position of berth v with its start time set equal to $\max\{a_{g_u}, s'_{b-1} + p'_{b-1}\}$. Select the position with the smallest increment and denote the position as q_v. Let the increment of the objective function value be I_{v,g_u} if vessel g_u is inserted into the q_vth position of berth v.
      (ii) Let $v' := \arg\min_{v=L'_j,L'_j+1,\ldots,m}\{I_{v,g_u}\}$ (with ties broken arbitrarily). Insert vessel g_u into the $q_{v'}$th position of berth v', with its start time set equal to $\max\left\{a_{g_u}, s'_{q_{v'}-1} + p'_{q_{v'}-1}\right\}$.

4. Same as Step 4 of Algorithm 1.
5. Same as Step 5 of Algorithm 1.
6. For $j := \ell$ down to 1, do:
   (A) Let $\Psi := \left\{i \in \Phi | H_i = H'_j\right\}$. Let $\Phi := \Phi \backslash \Psi$.
   (B) Sort the vessels in Ψ in WSPT order. Denote these vessels as vessels $g_1, g_2, \ldots, g_r$.

(C) For $u := 1$ to r, do:
 (i) For $v := H'_j$ to m, do the following: In the current partial schedule, find the first vessel assigned to berth v of which the start time τ is greater than or equal to T. If no such vessel exists, define $\tau = T$. Consider only the vessels of which the start time is greater than or equal to τ. Let t denote the number of such vessels that are assigned to berth v in the current schedule, and let $x_1, x_2, \ldots, x_t$ denote the t vessels at berth v. For $b = 1,2,\ldots,t$, let s'_b and p'_b denote the start time and processing time, respectively, of vessel x_b. Denote $s'_0 = \tau$ and $p'_0 = 0$. For $b = 1,2,\ldots,t+1$, calculate the increment in objective function value if vessel g_u is inserted into the bth position of berth v with its start time set equal to $\max\{a_{g_u}, s'_{b-1} + p'_{b-1}\}$. Select the position with the smallest increment and denote the position as q_v. Let the increment of the objective function value be I_{v,g_u} if vessel g_u is inserted into the q_vth position of berth v.
 (ii) Let $v' := \arg\min_{v=H'_j, H'_j+1,\ldots,m}\{I_{v,g_u}\}$ (with ties broken arbitrarily). Insert vessel g_u into the $q_{v'}$th position of berth v', with its start time set equal to $\max\left\{a_{g_u}, s'_{q_{v'}-1} + p'_{q_{v'}-1}\right\}$.

7. Same as Step 7 of Algorithm 1.

5. Computational experiments

We have conducted computational experiments to test the performance of Algorithms 1 and 2, and to evaluate the benefits of using our model for decision making as opposed to solving the problem without considering the tidal effect. The test data are generated randomly, while the parameter settings are selected in such a way that the actual terminal operating conditions are reflected. To evaluate the effectiveness of the heuristics, we compare the heuristic solution values with the lower bound values generated by the IBM ILOG CPLEX Optimizer (CPLEX).

5.1. Problem generation

Because computing the lower bound values requires a substantial amount of computer time and memory space, we need to select the parameter settings of the test instances carefully so that the experiments can be completed within a reasonable amount of time, and yet the parameters can capture the main characteristics of a real-life terminal operation. Hence, throughout the computational study, we define one time unit as one hour and assume that all vessel processing times and arrival times are multiples of one hour. Furthermore, we only consider the situation where the tides are diurnal (i.e., one tidal cycle per day), and therefore we set $T = 12$.

We consider both the static and dynamic cases of problem **P**. In each of these two cases, we consider the two scenarios: low-water period $[0,T]$ followed by high-water period $[T,\infty)$ (denoted by L-H), and high-water period $[0,T]$ followed by low-water period $[T,\infty)$ (denoted by H-L). We consider problem instances with six different sizes, namely problems with $(m,n) = (3,9)$, $(4,12)$, $(5,15)$, $(6,18)$, $(7,21)$, and $(8,24)$. We also consider the situation where the difference between L_j and H_j is relatively small (for $j = 1,2,\ldots,n$), as well as the situation where the difference is relatively big. In the first situation the tidal effect has a smaller impact on the vessel assignment, while in the second situation the tidal effect has a bigger impact on the vessel assignment (the generation of L_j and H_j for these two situations is described below). In sum, there are $2 \times 6 \times 2 = 24$ different parameter settings in each of static and dynamic cases.

For each combination of m and n, we generate ten random problem instances as follows: The processing time p_j of each vessel j is randomly generated according to a discrete uniform distribution within $\{3,4,\ldots,12\}$ (this is because in the application that we know, most vessels occupy the berth for three to twelve hours). The weight w_j of each vessel is randomly generated according to a discrete uniform distribution within $\{1,2,\ldots,10\}$ (in practice, w_j is determined by a number of factors such as vessel size, customer importance, urgent requests, etc.; for simplicity, we assume that it is independent of p_j). For the dynamic case, the arrival time a_j of each vessel j is randomly generated according to a discrete uniform distribution within $\{0,1,\ldots,T\}$.

The low-water berth indices and high-water berth indices of the vessels are generated as follows: There are two vessel groups, and each vessel has a probability of 0.5 belonging to each group. For the first group of vessels, the low-water berth index L_j is randomly generated according to a discrete uniform distribution within $\{1,2,\ldots,m\}$, and the high-water berth index H_j is set equal to $\max\{L_j - 1, 1\}$ (resp. $\max\{L_j - 2, 1\}$) for the case where the tidal impact is small (resp. big). For the second group of vessels, the high-water berth index H_j is randomly generated according to a discrete uniform distribution within $\{1,2,\ldots,m\}$, and the low-water berth index L_j is set equal to $\min\{H_j + 1, m\}$ (resp. $\min\{H_j + 2, m\}$) for the case where the tidal impact is small (resp. big).

5.2. Lower bounds on the optimal solution value

To evaluate the effectiveness of the heuristics, we compare the heuristic solution values with the solution generated by CPLEX. For large-size problem instances, CPLEX cannot generate optimal solutions within a reasonable time limit. In such a case, we use the lower bound on the optimal solution value generated by CPLEX for comparison. However, such a lower bound can be very loose if we simply apply CPLEX to solve the MILP model described in (1)–(10). To generate tighter lower bounds, we add some valid inequalities to the MILP formulation before solving it with CPLEX. In the following, we present several sets of valid inequalities (the validity of these valid inequalities are available in the Appendix; see Theorem 6 and its proof).

For any berth i and any two vessels j and j', define $\Omega_{ijj'} = \{k \mid k \neq j; k \neq j'; i \geqslant L_k\}$. If $\Omega_{ijj'} \neq \emptyset$, then let $p^{\min}_{ijj'} = \min\{p_k \mid k \in \Omega_{ijj'}\}$.

Valid Inequality 1. For any berth i and any two vessels j and j', if $i \geqslant L_j, i \geqslant L_{j'}, a_j \leqslant a_{j'}, w_j/p_j > w_{j'}/p_{j'}$, and $\Omega_{ijj'} \neq \emptyset$, then

$$s_{j'} + p_{j'} - s_j + p^{\min}_{ijj'} \leqslant M(2 - x_{ij} - x_{ij'} + I_{ijj'}),$$

where M is a large constant. If $i \geqslant L_j$, $i \geqslant L_{j'}$, $a_j \leqslant a_{j'}$, $w_j/p_j > w_{j'}/p_{j'}$, and $\Omega_{ijj'} = \emptyset$, then

$$x_{ij} + x_{ij'} - I_{ijj'} \leqslant 1.$$

Valid Inequality 1 states that if vessels j and j' are allowed to be served by berth i throughout the time horizon, and if $a_j \leqslant a_{j'}$ and $w_j/p_j > w_{j'}/p_{j'}$, then either (i) vessel j is processed before vessel j' or (ii) vessel j completes at least $p^{\min}_{ijj'}$ time units after vessel j', so that some other vessel is processed in between these two vessels. (If $\Omega_{ijj'} = \emptyset$, then only case (i) applies.)

Next, for $j = 1,2,\ldots,n$, define

$$E_j = \sum_{k=1}^{n} p_k - p_j$$

and

$$D_j = \left[\sum_{k=1}^{n} p_k - p_j\right] + (n-1)(p_j - 1).$$

For the L-H scenario of the static case, let

$$U_j = \begin{cases} T + \frac{E_j-(T-p_j+1)(m-L_j+1)}{m-H_j+1}, & \text{if } E_j \geqslant (T-p_j+1)(m-L_j+1); \\ \frac{E_j}{m-L_j+1}, & \text{otherwise}. \end{cases}$$

For the H-L scenario of the static case, let

$$U_j = \begin{cases} T-p_j+1+\frac{E_j-(T-p_j+1)(m-H_j+1)}{m-L_j+1}, \\ \quad \text{if } E_j \geqslant (T-p_j+1)(m-H_j+1); \\ \frac{E_j}{m-H_j+1}, \\ \quad \text{otherwise}. \end{cases}$$

For the L-H scenario of the dynamic case, let

$$U_j = \begin{cases} T+\frac{D_j-(T-a_j)(m-L_j+1)}{m-H_j+1}, \\ \quad \text{if } T-a_j \geqslant 0 \quad \text{and} \quad D_j \geqslant (T-a_j)(m-L_j+1); \\ a_j+\frac{D_j}{m-L_j+1}, \\ \quad \text{if } T-a_j \geqslant 0 \quad \text{and} \quad D_j < (T-a_j)(m-L_j+1); \\ a_j+\frac{D_j}{m-H_j+1}, \\ \quad \text{otherwise}. \end{cases}$$

For the H-L scenario of the dynamic case, let

$$U_j = \begin{cases} T-p_j+1+\frac{D_j-(T-a_j-p_j+1)(m-H_j+1)}{m-L_j+1}, \\ \quad \text{if } T-a_j \geqslant p_j-1 \text{ and } D_j \geqslant (T-a_j-p_j+1)(m-H_j+1); \\ a_j+\frac{D_j}{m-H_j+1}, \\ \quad \text{if } T-a_j \geqslant p_j-1 \text{ and } D_j < (T-a_j-p_j+1)(m-H_j+1); \\ a_j+\frac{D_j}{m-L_j+1}, \\ \quad \text{otherwise}. \end{cases}$$

Valid Inequality 2. $s_j \leqslant \lfloor U_j \rfloor$ for any vessel j.

Valid Inequality 2 provides an upper limit on the start time of a vessel. The next set of valid inequalities applies to the static case only. Define $p^{\min} = \min\{p_1, p_2, \ldots, p_n\}$. For the L-H scenario, let $K_j = L_j$ ($j = 1, 2, \ldots, n$). For the H-L scenario, let $K_j = H_j$ ($j = 1, 2, \ldots, n$).

Valid Inequality 3. If $a_j = 0$ for $j = 1, 2, \ldots, n$, then there exist $R_1, R_2, \ldots, R_n \in \{0, 1\}$ such that

$$\min\{p^{\min}, T\} \cdot R_j \leqslant s_j \leqslant MR_j \textit{ for } j = 1, 2, \ldots, n$$

and

$$\sum_{j \text{ s.t. } K_j \geqslant i} (1-R_j) \leqslant m-i+1 \textit{ for } i = 1, 2, \ldots, m,$$

where M is a large constant.

Theorem 6. *There exists an optimal solution to formulation (1)–(10) which satisfies Valid Inequalities 1, 2, and 3.*

The proof of Theorem 6 is available in the Appendix. To obtain a lower bound on the optimal solution value of problem **P**, we add Valid Inequalities 1, 2, and 3 to the MILP formulation (1)–(10) and then solve it using CPLEX with a prespecified time limit. If CPLEX completes its execution within the time limit, then an optimal solution value is available. If the running time exceeds the time limit, then we use the lower bound generated by CPLEX as our lower bound value.

5.3. Computational results

Let $Z^H(I)$ denote the heuristic solution value of the random problem instance I, let $Z^*(I)$ denote the optimal solution value of I, and let $LB(I)$ denote the lower bound on $Z^*(I)$. If CPLEX successfully generates the optimal solution of instance I, then the relative error of the heuristic solution is given by

$$e = \frac{Z^H(I) - Z^*(I)}{Z^*(I)} \times 100\%.$$

If the optimal solution value $Z^*(I)$ is unavailable, then the lower bound is used to estimate the performance of the heuristics, and the estimate of the relative error of the solution is given by

$$e = \frac{Z^H(I) - LB(I)}{LB(I)} \times 100\%.$$

For each parameter setting, we let $avg(e)$ denote the average value of e from the ten random instances.

Besides testing the effectiveness of the heuristics, we are also interested in evaluating the benefits of using our model for making berth allocation decisions. To do so, for each test instance I, we first solve it as a two-period model, which comprises a low-water period and a high-water period, using our heuristic (i.e., using either Algorithm 1 or Algorithm 2, depending on whether it is the static or dynamic case). Then, we solve the problem again by setting the entire planning horizon to a low-water period (i.e., we conservatively do the berth allocation through treating the water depth as "low water" throughout the time horizon), and we let $Z^H(I_{NT})$ denote the solution value generated by such approach. The benefit of using our model for decision making is given as

$$b = \frac{Z^H(I_{NT}) - Z^H(I)}{Z^H(I_{NT})} \times 100\%.$$

For each parameter setting, we let $avg(b)$ denote the average value of b from the ten random instances.

The computational experiments were conducted on a computer with a Duo CPU with two 2.53 GHz processors and 2.0 GB RAM. The IBM ILOG CPLEX Optimizer Version 12.2 was used to generate the lower bound, and we set a time limit of 10 min per instance.

Tables 1 and 2 summarize the computational results of testing the effectiveness of Algorithms 1 and 2, respectively. From these results, we observe that the heuristics can generate fairly effective solutions in a very short time. On the average, the algorithms generate solutions with estimated errors of 10.7% and 9.9% in the static and dynamic cases, respectively. We also observe that the percentage errors tend to increase as the problem size increases. In addition, our heuristics have a slightly lower performance under H-L scenario compared to L-H scenario. This is consistent with our findings from the theoretical analysis performed in Section 3, where we stated that it is a lot more difficult to develop good approximation algorithms, say FPTASs, under the scenario where a high-water period is followed by a low-water period.

Considering those parameter settings for which the optimal solutions are available (i.e., those entries in Tables 1 and 2 with asterisks), the percentage errors in the dynamic case are higher than those in the corresponding static case. This is because the existence of vessel arrival times may conflict with the schedule implied by the WSPT rule, and therefore it degrades the heuristic's performance. For those cases where the optimal solutions are not available, the estimated percentage errors in the dynamic case tend to be lower than those in the corresponding static case. This is because in the dynamic case the vessels' nonzero arrival times make the feasible region of the MILP smaller, and therefore CPLEX is able to search for a tighter lower bound more efficiently. As a result, the CPLEX lower bound is tighter in the dynamic case than in the static case.

Tables 3 and 4 report the results obtained by comparing the two-period model with the model where we set the entire planning horizon to a low-water period. The results show that our

Table 1
Estimates of percentage error generated by Algorithm 1 in the static case.

Tidal periods	Problem size	Tidal effect	$avg(e) \times 100\%$ (%)	Avg. heuristic running time per instance (seconds)	Avg. CPLEX running time per instance (seconds)
L-H	$m = 3$, $n = 9$	small	0.97[a]	0.00	3.32
L-H	$m = 3$, $n = 9$	Big	1.80 [a]	0.00	3.64
L-H	$m = 4$, $n = 12$	Small	1.27 [a]	0.00	12.78
L-H	$m = 4$, $n = 12$	Big	2.90 [a]	0.00	12.54
L-H	$m = 5$, $n = 15$	Small	1.94	0.01	160.80
L-H	$m = 5$, $n = 15$	Big	3.40 [a]	0.01	93.19
L-H	$m = 6$, $n = 18$	Small	8.21	0.02	443.63
L-H	$m = 6$, $n = 18$	Big	8.17	0.02	500.96
L-H	$m = 7$, $n = 21$	Small	16.72	0.03	600.00
L-H	$m = 7$, $n = 21$	Big	15.88	0.02	600.00
L-H	$m = 8$, $n = 24$	Small	19.71	0.04	600.00
L-H	$m = 8$, $n = 24$	Big	21.26	0.06	600.00
H-L	$m = 3$, $n = 9$	Small	3.84 [a]	0.01	2.62
H-L	$m = 3$, $n = 9$	Big	4.98 [a]	0.01	2.73
H-L	$m = 4$, $n = 12$	Small	3.46 [a]	0.01	16.23
H-L	$m = 4$, $n = 12$	Big	7.24 [a]	0.00	20.73
H-L	$m = 5$, $n = 15$	Small	6.62	0.04	196.79
H-L	$m = 5$, $n = 15$	Big	9.05	0.02	209.43
H-L	$m = 6$, $n = 18$	Small	12.56	0.03	472.67
H-L	$m = 6$, $n = 18$	Big	15.86	0.03	490.72
H-L	$m = 7$, $n = 21$	Small	19.59	0.04	600.00
H-L	$m = 7$, $n = 21$	big	21.78	0.07	600.00
H-L	$m = 8$, $n = 24$	Small	21.08	0.15	600.00
H-L	$m = 8$, $n = 24$	Big	27.62	0.08	600.00

[a] Percentage error with an asterisk represents the average of $[Z^H(I) - Z^*(I)]/Z^*(I) \times 100\%$; percentage error without an asterisk represents the average of $[Z^H(I) - LB(I)]/LB(I) \times 100\%$.

Table 2
Estimates of percentage error generated by Algorithm 2 in the dynamic case.

Tidal periods	Problem size	Tidal effect	$avg(e) \times 100\%$ (%)	Avg. heuristic running time per instance (seconds)	Avg. CPLEX running time per instance (seconds)
L-H	$m = 3$, $n = 9$	small	3.00[a]	0.01	1.36
L-H	$m = 3$, $n = 9$	big	3.94 [a]	0.03	1.91
L-H	$m = 4$, $n = 12$	Small	5.34 [a]	0.01	5.48
L-H	$m = 4$, $n = 12$	Big	6.08 [a]	0.03	6.72
L-H	$m = 5$, $n = 15$	Small	2.56 [a]	0.01	26.92
L-H	$m = 5$, $n = 15$	Big	4.67 [a]	0.01	63.39
L-H	$m = 6$, $n = 18$	Small	8.43	0.01	205.42
L-H	$m = 6$, $n = 18$	Big	7.29	0.01	332.88
L-H	$m = 7$, $n = 21$	Small	9.97	0.01	446.26
L-H	$m = 7$, $n = 21$	Big	10.10	0.07	537.52
L-H	$m = 8$, $n = 24$	Small	18.06	0.08	515.99
L-H	$m = 8$, $n = 24$	Big	19.83	0.05	544.81
H-L	$m = 3$, $n = 9$	Small	5.35 [a]	0.03	0.49
H-L	$m = 3$, $n = 9$	Big	11.68 [a]	0.02	0.42
H-L	$m = 4$, $n = 12$	Small	10.08 [a]	0.01	2.10
H-L	$m = 4$, $n = 12$	Big	12.99 [a]	0.02	1.15
H-L	$m = 5$, $n = 15$	Small	9.26 [a]	0.02	8.35
H-L	$m = 5$, $n = 15$	Big	13.95 [a]	0.02	5.79
H-L	$m = 6$, $n = 18$	Small	10.59	0.04	95.28
H-L	$m = 6$, $n = 18$	Big	6.78 [a]	0.03	49.60
H-L	$m = 7$, $n = 21$	Small	10.50	0.03	251.01
H-L	$m = 7$, $n = 21$	Big	12.72	0.05	134.95
H-L	$m = 8$, $n = 24$	Small	18.71	0.10	487.61
H-L	$m = 8$, $n = 24$	Big	15.30	0.08	422.14

[a] Percentage error with an asterisk represents the average of $[Z^H(I) - Z^*(I)]/Z^*(I) \times 100\%$; percentage error without an asterisk represents the average of $[Z^H(I) - LB(I)]/LB(I) \times 100\%$.

two-period model consistently generates better results than solving the problem without taking tidal effect into account (i.e., all the percentage benefits are positive). On the average, the benefits are 10.6% and 9.1% in the static case and dynamic case, respectively. As expected, the benefits are more significant when the tidal effect is big. When the tidal effect is big, in our two-period model the vessels have more candidate berths that we can schedule to, and therefore, the assignment enjoys more flexibility.

Under the L-H scenario, we observe that the benefits of our two-period model are more significant in the dynamic case than in the static case. This is because the existence of nonzero vessel arrival times increases the number of vessels that are served in the high-water period, and therefore in the dynamic case our two-period model provides more benefits over the single-period model. However, under the H-L scenario, the benefits of our two-period model are less significant in the dynamic case than in the static

Table 3
Benefits of solving the problem using a two-period model (static case).

Tidal periods	Problem size	Tidal effect	$avg(b) \times 100\%$ (%)
L-H	$m = 3, n = 9$	Small	6.76
L-H	$m = 3, n = 9$	Big	17.71
L-H	$m = 4, n = 12$	Small	6.05
L-H	$m = 4, n = 12$	Big	15.61
L-H	$m = 5, n = 15$	Small	4.41
L-H	$m = 5, n = 15$	Big	14.55
L-H	$m = 6, n = 18$	Small	4.92
L-H	$m = 6, n = 18$	Big	11.08
L-H	$m = 7, n = 21$	Small	2.06
L-H	$m = 7, n = 21$	Big	11.26
L-H	$m = 8, n = 24$	Small	3.41
L-H	$m = 8, n = 24$	Big	8.56
H-L	$m = 3, n = 9$	Small	6.81
H-L	$m = 3, n = 9$	Big	17.49
H-L	$m = 4, n = 12$	Small	11.98
H-L	$m = 4, n = 12$	Big	14.41
H-L	$m = 5, n = 15$	Small	10.60
H-L	$m = 5, n = 15$	Big	27.63
H-L	$m = 6, n = 18$	Small	9.18
H-L	$m = 6, n = 18$	Big	16.94
H-L	$m = 7, n = 21$	Small	4.84
H-L	$m = 7, n = 21$	Big	14.05
H-L	$m = 8, n = 24$	Small	4.42
H-L	$m = 8, n = 24$	Big	9.39

Table 4
Benefits of solving the problem using a two-period model (dynamic case).

Tidal periods	Problem size	Tidal effect	$avg(b) \times 100\%$ (%)
L-H	$m = 3, n = 9$	Small	14.34
L-H	$m = 3, n = 9$	Big	29.13
L-H	$m = 4, n = 12$	Small	13.45
L-H	$m = 4, n = 12$	Big	17.11
L-H	$m = 5, n = 15$	Small	9.72
L-H	$m = 5, n = 15$	Big	18.05
L-H	$m = 6, n = 18$	Small	6.17
L-H	$m = 6, n = 18$	Big	16.17
L-H	$m = 7, n = 21$	Small	4.10
L-H	$m = 7, n = 21$	Big	12.19
L-H	$m = 8, n = 24$	Small	2.50
L-H	$m = 8, n = 24$	Big	12.74
H-L	$m = 3, n = 9$	Small	1.73
H-L	$m = 3, n = 9$	Big	6.21
H-L	$m = 4, n = 12$	Small	1.14
H-L	$m = 4, n = 12$	Big	11.01
H-L	$m = 5, n = 15$	Small	5.76
H-L	$m = 5, n = 15$	Big	12.37
H-L	$m = 6, n = 18$	Small	3.13
H-L	$m = 6, n = 18$	Big	7.02
H-L	$m = 7, n = 21$	Small	2.59
H-L	$m = 7, n = 21$	Big	5.57
H-L	$m = 8, n = 24$	Small	2.10
H-L	$m = 8, n = 24$	Big	4.27

case. This is because under this scenario, the existence of nonzero vessel arrival times increases the number of vessels that are served in the low-water period, and therefore in the dynamic case our two-period model provides less significant benefits over the single-period model. Note that if we extend our model to cover a longer planning horizon which includes more than one high tide and one low tide, then we expect that the benefits of the multiple-period model over the single-period model will be equally significant in both the static and dynamic cases.

6. Conclusions

In this paper we have proposed a berth allocation model that takes into account water depth and tidal effect. The model can be viewed as a parallel-machine scheduling model with time-dependent inclusive processing sets. We have explored the computational complexity of the model, and have developed heuristics for both the static and dynamic cases of the problem. Computational results have indicated that the heuristics are fairly effective. We have also demonstrated that our model can help terminal operators make better decisions by taking in account the change in water depth over time when assigning vessels to berths.

Our work represents a first step in the detailed study of berth allocation with tidal effect considerations. Our current study, however, has some limitations. First, our model assumes that there is only one high-water period and one low-water period. More accurate results can be obtained if we further divide the planning horizon into multiple time periods with varying water depth. In fact, Algorithms 1 and 2 can be extended to deal with the situation where the planning horizon is divided into τ time periods $[0,T_1],[T_1,T_2],\ldots,[T_{\tau-2},T_{\tau-1}],[T_{\tau-1},\infty)$, where each period has a different water level. For example, suppose we have three time periods $[0,T_1]$, $[T_1,T_2]$ and $[T_2,\infty)$ (i.e., $\tau = 3$), where the first and third time periods are low-water periods while the second time period is a high-water period. In this case, we change Algorithm 1 as follows: Step 7 is changed to scanning the schedule to remove any job that finishes later than T_2. Then, we add Steps 8 and 9 which are identical to Steps 2 and 3, respectively. Finally, we add Step 10 to scan the final schedule and remove idle time without violating any constraint. Similarly, Algorithm 2 can be modified by the same strategy. However, such an extension of the model will drastically increase the difficulty in finding a good lower bound on the optimal solution value.

Second, our model assumes that there are multiple berths available, and that each berth can handle one ship at a time. However, in some applications multiple vessels with different lengths may be served simultaneously at a berth. Hence, an interesting future research topic is to extend our model to such a setting, where berths along a quayside are resources to be shared by different vessels.

Note that although Algorithms 1 and 2 are simple and efficient, more effective heuristics can be developed to solve the problem. In particular, developing meta-heuristics (e.g., extending Nishimura et al.'s 2001 genetic algorithm) for our problem and conducting computational studies to compare the performance of those heuristics is also an interesting future research direction.

Acknowledgments

The authors thank three anonymous referees for their helpful comments and suggestions. We would also like to thank Mr. Allen Du of the Guangzhou South China Oceangate Container Terminal Company Limited and Mr. Peng Zhong of the SinoTrans Guangdong Company Limited for explaining to us the details of tidal condition and its impact on terminal operation. This research was supported in part by The Hong Kong Polytechnic University under Grant J-BB7H and by the Research Grants Council of Hong Kong under Grant PolyU5236/09E. The first author was supported in part by the National Natural Science Foundation of China under Grant 71001109, and the third author was supported in part by the NSF Grant CMMI-0969830.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2011.07.012.

References

Barros, V.H., Costa, T.S., Oliveira, A.C.M., Lorena, L.A.N., 2011. Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. Computers and Industrial Engineering 60 (4), 606–613.

Bierwirth, C., Meisel, F., 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research 202 (3), 615–627.

Blazewicz, J., Cheng, T.C.E., Machowiak, M., Oguz, C., 2011. Berth and quay crane allocation: A moldable task scheduling model. Journal of the Operational Research Society 62 (7), 1189–1197.

Brown, G.G., Cormican, K.J., Lawphongpanich, S., Widdis, D.B., 1997. Optimizing submarine berthing with a persistence incentive. Naval Research Logistics 44 (4), 301–318.

Brown, G.G., Lawphongpanich, S., Thurman, K.P., 1994. Optimizing ship berthing. Naval Research Logistics 41 (1), 1–15.

Brucker, P., 2007. Scheduling Algorithms, 5th ed. Springer, Berlin.

Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., Lusby, R., 2011. Models for the discrete berth allocation problem: A computational comparison. Transportation Research Part E 47 (4), 461–473.

Cheong, C.Y., Tan, K.C., 2008. A multi-objective multi-colony ant algorithm for solving the berth allocation problem. In: Liu, Y., Sun, A., Loh, H.T., Lu, W.F., Lim, E.-P. (Eds.), A dvances of Computational Intelligence in Industrial Systems. Springer-Verlag, Berlin, pp. 333–350.

Cheong, C.Y., Tan, K.C., Liu, D.K., Lin, C.J., 2010. Multi-objective and prioritized berth allocation in container ports. Annals of Operations Research 180, 63–103.

Chia, J.T., Lau, H.C., Lim, A., 1999. Ant colony optimization for the ship berthing problem. Lecture Notes in Computer Science 1742, 359–370.

Cordeau, J.-F., Laporte, G., Legato, P., Moccia, L., 2005. Models and tabu search heuristics for the berth-allocation problem. Transportation Science 39 (4), 526–538.

Dai, J., Lin, W., Moorthy, R., Teo, C.-P., 2008. Berth allocation planning optimization in container terminals. In: Tang, C.S., Teo, C.-P., Wei, K.-K. (Eds.), Supply Chain Analysis: A Handbook on the Interaction of Information, System and Optimization. Springer, New York, pp. 69–104.

Guan, Y., Cheung, R.K., 2004. The berth allocation problem: Models and solution methods. OR Spectrum 26 (1), 75–92.

Guan, Y., Xiao, W.-Q., Cheung, R.K., Li, C.-L., 2002. A multiprocessor task scheduling model for berth allocation: Heuristic and worst-case analysis. Operations Research Letters 30 (5), 343–350.

Hansen, P., Oguz, C., Mladenovic, N., 2008. Variable neighborhood search for minimum cost berth allocation. European Journal of Operational Research 191 (3), 636–649.

Imai, A., Chen, H.C., Nishimura, E., Papadimitriou, S., 2008a. The simultaneous berth and quay crane allocation problem. Transportation Research Part E 44 (5), 900–920.

Imai, A., Nagaiwa, K., Chan, W.T., 1997. Efficient planning of berth allocation for container terminals in Asia. Journal of Advanced Transportation 31 (1), 75–94.

Imai, A., Nishimura, E., Hattori, M., Papadimitriou, S., 2007a. Berth allocation at indented berths for mega-containerships. European Journal of Operational Research 179 (2), 579–593.

Imai, A., Nishimura, E., Papadimitriou, S., 2001. The dynamic berth allocation problem for a container port. Transportation Research Part B 35 (4), 401–417.

Imai, A., Nishimura, E., Papadimitriou, S., 2003. Berth allocation with service priority. Transportation Research Part B 37 (5), 437–457.

Imai, A., Nishimura, E., Papadimitriou, S., 2005a. Corrigendum to "The dynamic berth allocation problem for a container port [Transportation Research Part B 35 (2001) 401–417]. Transportation Research Part B 39 (3), 197.

Imai, A., Nishimura, E., Papadimitriou, S., 2008b. Berthing ships at a multi-user container terminal with a limited quay capacity. Transportation Research Part E 44 (1), 136–151.

Imai, A., Sun, X., Nishimura, E., Papadimitriou, S., 2005b. Berth allocation in a container port: Using a continuous location space approach. Transportation Research Part B 39 (3), 199–221.

Imai, A., Zhang, J.-T., Nishimura, E., Papadimitriou, S., 2007b. The berth allocation problem with service time and delay time objectives. Maritime Economics and Logistics 9 (4), 269–290.

Kim, K.H., Moon, K.C., 2003. Berth scheduling by simulated annealing. Transportation Research Part B 37 (6), 541–560.

Lai, K.K., Shih, K., 1992. A study of container berth allocation. Journal of Advanced Transportation 26 (1), 45–60.

Lee, D.-H., Chen, J.H., Cao, J.X., 2010. The continuous berth allocation problem: A greedy randomized adaptive search solution. Transportation Research Part E 46 (6), 1017–1029.

Lee, Y., Chen, C.-Y., 2009. An optimization heuristic for the berth scheduling problem. European Journal of Operational Research 196 (2), 500–508.

Leung, J.Y.-T., Li, C.-L., 2008. Scheduling with processing set restrictions: A survey. International Journal of Production Economics 116 (2), 251–262.

Li, C.-L., Cai, X., Lee, C.-Y., 1998. Scheduling with multiple-job-on-one-processor pattern. IIE Transactions 30 (5), 433–445.

Liang, C., Huang, Y., Yang, Y., 2009a. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. Computers and Industrial Engineering 56 (3), 1021–1028.

Liang, C., Lin, L., Jo, J., 2009b. Multiobjective hybrid genetic algorithm for quay crane scheduling in berth allocation planning. International Journal of Manufacturing Technology and Management 16 (1–2), 127–146.

Lim, A., 1998. The berth planning problem. Operations Research Letters 22 (2–3), 105–110.

Meisel, F., Bierwirth, C., 2009. Heuristics for the integration of crane productivity in the berth allocation problem. Transportation Research Part E 45 (1), 196–209.

Monaco, M.F., Sammarra, M., 2007. The berth allocation problem: A strong formulation solved by a Lagrangian approach. Transportation Science 41 (2), 265–280.

Moorthy, R., Teo, C.-P., 2006. Berth management in container terminal: The template design problem. OR Spectrum 28 (4), 495–518.

Nishimura, E., Imai, A., Papadimitriou, S., 2001. Berth allocation planning in the public berth system by genetic algorithms. European Journal of Operational Research 131 (2), 282–292.

Ou, J., Leung, J.Y.-T., Li, C.-L., 2008. Scheduling parallel machines with inclusive processing set restrictions. Naval Research Logistics 55 (4), 328–338.

Park, K.T., Kim, K.H., 2002. Berth scheduling for container terminals by using a sub-gradient optimization technique. Journal of the Operational Research Society 53 (9), 1054–1062.

Park, Y.-M., Kim, K.H., 2003. A scheduling method for berth and quay cranes. OR Spectrum 25 (1), 1–23.

Pinedo, M.L., 2002. Scheduling: Theory, Algorithms, and Systems, 2nd ed. Prentice-Hall, Upper Saddle River, NJ.

Stahlbock, R., Voss, S., 2008. Operations research at container terminals: A literature update. OR Spectrum 30 (1), 1–52.

Steenken, D., Voss, S., Stahlbock, R., 2004. Container terminal operation and operations research – A classification and literature review. OR Spectrum 26 (1), 3–49.

Tang, L., Li, S., Liu, J., 2009. Dynamically scheduling ships to multiple continuous berth spaces in an iron and steel complex. International Transactions in Operational Research 16 (1), 87–107.

Wang, F., Lim, A., 2007. A stochastic beam search for the berth allocation problem. Decision Support Systems 42 (4), 2186–2196.