New Lower Bound and Exact Method for the Continuous Berth Allocation Problem

Zhou Xu

Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, lgtzx@polyu.edu.hk

Chung-Yee Lee

Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, cylee@ust.hk

We study a continuous berth allocation problem, where incoming vessels need to be assigned a mooring time as well as a berth location on a quay. It is a crucial element in port planning to achieve quick turnaround time for vessels. To solve this problem, many solution methods have been developed in the literature. However, gaps between the best known lower and upper bounds on its optimal solutions are far from close. In this paper, we propose new and more effective solution methods for this important problem. By introducing a novel relaxation of the problem, we have derived a new lower bound that can be computed efficiently in quadratic time. By incorporating this new lower bound with some new heuristic and pruning techniques, we have developed a new exact method, based on a branch and bound approach. To demonstrate general applicability of the proposed methods, we have extended them to a more complicated problem, where decisions on berth allocations are restricted by a quay crane constraint. Extensive computational results have shown that, compared with previous state-of-the-art methods, our new methods have significantly reduced gaps between the lower and upper bounds, and have solved more and larger instances to optimality in significantly less time. We have also performed sensitivity tests to demonstrate how robust the new solutions are against uncertainties in particular input parameters.

Key words: continuous berth allocation; lower bound; branch and bound; exact method.

Subject classifications: Transportation: scheduling; Programming: integer: algorithms; Industries: transportation/shipping.

Area of review: Transportation.

1. Introduction

Today, 65.5% of the delays in ocean transportation occur at ports (Notteboom 2006). Port operators are struggling to enhance efficiency through better utilization of resources, such as berths, yards, cranes, vehicles and workers. Among these resources, berths are by far the most important. By allocating berths to incoming vessels efficiently, the port can reduce the turnaround time of vessels, and can thus increase port throughput as well as improve customer satisfaction. An efficient berth planner is also a critical component in port decision support systems that integrate the planning of various port resources and operations (Bierwirth and Meisel 2010).

This paper studies a berth allocation problem for ports with a continuous quay, known as the *Continuous Berth Allocation Problem* (CBAP), where incoming vessels can berth at arbitrary locations within the boundaries of the quay (Bierwirth and Meisel 2010, Lee et al. 2010). In the CBAP, we are given a quay of length B, as well as a set $V = \{1, 2, ..., n\}$ of n incoming vessels of importance weight w_v, size l_v, arrival time a_v, and handling time t_v for each $v \in V$. For some of these input parameters, their values are often assigned in practice by estimation. For example, importance weights w_v are often estimated based on vessel delays and cargo values, arrival times a_v are often estimated by shipping companies, and handling times t_v are often estimated based on vessel cargo volume and pre-assigned quay cranes (Bierwirth and Meisel 2010, 2015). The CBAP consists of assigning a mooring time x_v and a starting berth location y_v to each vessel v, such that vessels are berthed no earlier than arrival times, vessels occupy only space within the quay, and that no two vessels occupy the same quay space at the same time. Since quick turnaround time at a port is a major concern for seaborne transportation (Imai et al. 2003), the CBAP aims to minimize the total weighted turnaround time $\sum_{v \in V} w_v(x_v + t_v - a_v)$, which is a key issue considered by most studies in the literature (Bierwirth and Meisel 2015).

The CBAP is important because it models a vital objective term, as well as a number of common decisions and constraints faced by port operators. Its solution methods have been widely applied as key components in solving many other port planning problems, including berth allocation problems with a hybrid layout (Moorthy and Teo 2006), berth allocation problems with multiple objectives (Cheong et al. 2010), berth allocation problems with uncertainties (Zhen 2015), simultaneous berth and quay crane allocation problems (Park and Kim 2003, Giallombardo et al. 2010, Li et al. 2015), as well as berth allocation problems with other objective terms (Bierwirth and Meisel 2010, 2015), such as the total cost of tardiness of the vessels (Meisel and Bierwirth 2009), and the total cost of transporting cargo from berths to the yard (Park and Kim 2003). Therefore, as we will demonstrate later in this paper and its appendices, although our study focuses on the CBAP, the newly developed solution methods and analytical results have a large potential for being applied and extended to solve other more complicated port planning problems. Moreover, compared to the berth allocation problem with a discrete layout (or DBAP for short), where the quay is partitioned into segments, with each segment being assigned to at most one vessel (Buhrkal et al. 2011, de Oliveira et al. 2012, Imai et al. 2001, Monaco and Sammarra 2007), although the CBAP is more complicated, solutions to the CBAP can lead to much better utilization of the quay space.

The CBAP is computationally challenging, as it is known to be strongly NP-hard. It can be transformed to a two-dimensional rectangle packing problem in a space-time diagram (Lee et al. 2010), where the horizontal axis with an open right end represents the time units, and the vertical axis with a height of B represents the berth locations. As shown in Figure 1, each vessel $v \in V$ can be viewed as a rectangle with a height of l_v and a width of t_v. The mooring time x_v and the starting berth location y_v correspond to the position of the left-bottom corner of the rectangle in the diagram. Thus, finding a feasible solution to the CBAP is equivalent to determining the position (x_v, y_v) for the rectangle of each vessel v, with x_v being no less than a_v, and $y_v + l_v$ being no larger than B, such that no two rectangles overlap. For each vessel v, its weighted turnaround time $w_v(x_v + t_v - a_v)$ represents the cost for packing the rectangle of v at position (x_v, y_v). Thus, finding an optimal solution to the CBAP is equivalent to finding a packing that minimizes the total packing cost. This is unlike the classic rectangle packing problems that often aim to optimize a min-max objective, e.g., to minimize the highest position occupied by given rectangles for the two-dimensional strip packing problem (Lodi et al. 2002, Hopper and Turton 2001). Therefore, many solution methods and techniques known to be

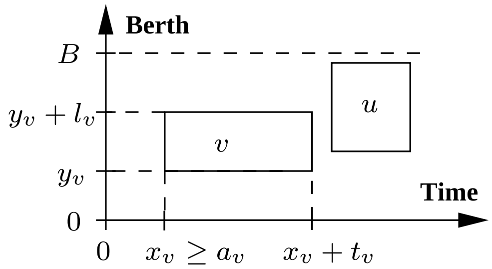


Figure 1 Berth allocations for two vessels v and u represented in a space-time diagram.

effective for the classic rectangle packing problems, such as those in Boschetti and Montaletti (2010), are not applicable or effective when directly applied to solve the CBAP.

The CBAP has been studied extensively (Bierwirth and Meisel 2010). However, performance of its existing solution methods is far behind that for the classic rectangle packing problems. For the two-dimensional strip packing problem, existing branch and bound algorithms can solve instances of 100 rectangles to optimality in an hour (Boschetti and Montaletti 2010, Côté et al. 2014, Martello and Vigo 1998, Pisinger and Sigurd 2007), and gaps between the best known heuristic packings and the best known lower bounds on the optimal packings are usually less than 5% (of the best known lower bounds), even for large instances of one thousand rectangles (Burke et al. 2004, Zhang et al. 2013). In contrast, for the CBAP, the best known branch and bound algorithm, as well as optimization solvers, such as CPLEX, often take hours to solve small instances of only 10∼15 vessels to optimality (Lee et al. 2010, Guan and Cheung 2004), and for median instances of about 30∼80 vessels, gaps between the best known heuristic solutions and the best known lower bounds on the optimal objective values often exceed 20%, and sometimes even exceed 40% (Umang et al. 2013, Ak and Erera 2011, Guan and Cheung 2004, Dai et al. 2008). For large instances of more than 80 vessels, lower bounds on the optimal objective values are seldom reported in the literature. This is mainly because existing lower bounds for the CBAP either have large gaps from the optimal objective values, or cannot be computed efficiently in polynomial time (Ak and Erera 2011, Lee et al. 2010, Dai et al. 2008).

The large gaps between the best known heuristic solutions and the best known lower bounds of the optimal solutions for the CBAP indicate that there is room for improvement, not only in deriving heuristic solutions, but also in computing lower bounds and optimal solutions. In the literature, most of the existing studies focus on developing heuristics and meta-heuristics that aim to produce heuristic solutions for the CBAP in affordable time, including the squeaky wheel optimization (Umang et al. 2013), the greedy randomized adaptive search procedure (Lee et al. 2010), the stochastic beam search algorithm (Wang and Lim 2007), the genetic algorithm (Imai et al. 2003), the simulated annealing algorithm (Dai et al. 2008), and the tabu search (Ak and Erera 2011) etc. These studies follow a similar approach by constructing heuristic solutions in two loops, where the outer loop searches sequences of the vessels, and the inner loop follows a sequence and applies certain greedy methods (Guan and Cheung 2004) to allocate berth locations and mooring times to vessels one by one. Apart from these, several other studies extend some heuristics and meta-heuristics for the DBAP to a restricted case of the CBAP, where the quay is partitioned into segments in advance, such that each vessel can occupy at most three segments (Cordeau et al. 2005, Mauri et al. 2016). Thus, it is of great interest to propose some new approaches that can produce better solutions for the CBAP.

In contrast to the large number of studies on heuristics and meta-heuristics, only a few methods are known to compute optimal solutions or their lower bounds for the CBAP. The most common method is to formulate the CBAP into an integer programming model by introducing binary variables σ_{vu} and δ_{vu} to indicate the relative positions of every two different vessels $v \in V$ and $u \in V$ in the space-time diagram, with $\sigma_{vu} = 1$ implying that v is positioned completely on the right of u, i.e., $x_v \geq x_u + t_u$, and with $\delta_{vu} = 1$ implying that v is positioned completely above u, i.e., $y_v \geq y_u + l_u$ (Lee et al. 2010). *See Online Appendix A for the details of this model, which we refer to as model* $\mathbf{IP_1}$. Commercial solvers, such as CPLEX, can be directly applied to solve IP_1, but it is time-consuming, and can take more than half an hour to solve instances of only ten vessels (Lee et al. 2010). Moreover, the linear programming relaxation of IP_1 provides only a trivial lower bound, equaling the weighted sum of vessel handling times, and often has a large gap from the optimal objective value (Ak and Erera 2011).

Besides IP_1, there is another integer programming model (Dai et al. 2008, Umang et al. 2013), which, however, is only for a discretized version of the CBAP (or the Discretized BAP for short), where the quay is discretized into a set of segments $\mathbb{S}$, and the planning horizon is discretized into a set of periods $\mathbb{T}$, so that each segment in $\mathbb{S}$ cannot be occupied by more than one vessel during the same time period in $\mathbb{T}$. The Discretized BAP can be formulated into an integer programming model by introducing binary variables π_{vxy} for $v \in V$, $x \in \mathbb{T}$ and $y \in \mathbb{S}$ to indicate whether or not vessel v starts berthing in segment y during period x. *See Online Appendix A for the details of this model, which we refer to as model* $\mathbf{IP_2}$.

Exact methods and commercial solvers have been applied to solve IP_2 and its relaxations, but only for instances of the Discretized BAP with a small number of quay segments and time periods (Park and Kim 2002, Guan and Cheung 2004, Umang et al. 2013). The Discretized BAP and the CBAP are not always equivalent. They are only equivalent if the sizes, arrival times and handling times of the vessels are integers and all segments in $\mathbb{S}$ and time periods in $\mathbb{T}$ are of unit size, which leads to $|\mathbb{S}| = B$ and $|\mathbb{T}| = H$, where H is the length of the planning horizon. When the two problems are equivalent, since $|V| = n$, $|\mathbb{S}| = B$ and $|\mathbb{T}| = H$, model IP_2 contains $O(nBH)$ binary variables and $O(n + BH)$ constraints, which are pseudopolynomial but can grow exponentially in the problem size $O(n)$ (Michael and David 1979) (if B and H grows exponentially in n). Thus, model IP_2 and its linear programming relaxation can sometimes be too large to be solved in affordable time. To resolve this issue, Dai et al. (2008) and Umang et al. (2013) suggested scaling down the vessel sizes, arrival times, and handling times, as well as the lengths of the quay and the planning horizon, to multiples of a scaling factor, so that model IP_2 of the scaled problem has a sufficiently small size to solve. By properly rounding up or down the scaled values, the linear programming relaxation of the scaled model can provide a valid lower bound for the CBAP (Dai et al. 2008), and the optimal solution to the scaled model can be transformed to a feasible solution to the CBAP (Umang et al. 2013). However, to achieve an affordable size for the scaled model, the scaling factor needs to be sufficiently large, and can thus grow exponentially in the problem size, which can result in a large deviation from the optimal solution for the lower bound and the feasible solution obtained. Furthermore, Ak and Erera (2011) recently proposed a new relaxation of IP_2 by splitting each vessel v of size l_v and handling time t_v into $l_v \times t_v$ vessels of unit size and unit handling time. However, the algorithm that they developed to compute this lower bound has a pseudopolynomial time complexity of $O(B^3H^3)$ and space complexity of $O(nBH)$, making it still not affordable.

Therefore, it is of considerable importance to develop for the CBAP some new and tight lower bounds that can be computed efficiently, as well as some new exact methods for solving this basic problem more effectively. Success in doing so can generate novel insights into berth allocation methods for various other complicated applications.

1.1. Summary of Our Contributions

The main results and contributions of this paper can be summarized as follows:

1. We have derived a new lower bound by relaxing the CBAP into a novel optimization model on vectors of functions, which we can efficiently solve by a constructive algorithm in quadratic $O(n^2)$ time, even when the sizes, arrival times and handling times of the vessels take continuous values. We have proved that the new lower bound can be significantly better than a widely known linear programming relaxation lower bound. Computational results show that for instances of 40 vessels or more, the new lower bound is significantly better than lower bounds produced by other methods in the literature, with an average improvement of more than 46%. Its gap from the best known heuristic solution is about 10% on average. Therefore, it can be used as a good approximation of the optimal solution in the development of solution methods, as well as in the evaluation of solution qualities.
2. By utilizing the new lower bound, we have developed a new exact method for the CBAP based on the branch and bound approach. It incorporates a new heuristic method of computing an upper bound of the optimal solution, as well as several new dominance rules to reduce search space. Computational results show that the new method outperforms other methods from the literature, and can solve to optimality all the test data of up to 24 vessels in just one hour. Compared with CPLEX, it has not only solved more and larger instances, but has often done so in more than 94% less computing time. Moreover, for instances of 40 vessels or more, the best solutions found by the new exact method within an hour time limit outperform those produced by other meta-heuristic methods from the literature, the improvement being more than 16% on average, as well as achieving a gap of about 10%, on average, from the best known lower bounds.
3. We have conducted extensive computational experiments to compare our solution methods with methods from the literature, some of which have not previously been compared using the same data sets. We have also performed sensitivity tests to demonstrate the robustness of the solutions produced by the new exact method, notably with respect to the uncertainties in various input parameters, such as vessel importance weights, arrival times, and handling times.

4. To demonstrate the general applicability of the proposed solution methods for the CBAP, we have extended them to a more complicated problem, where decisions on berth allocations are also restricted by a quay crane constraint. Computational results show that the extensions significantly outperform various conventional methods. Therefore, although our study focuses on a basic berth allocation problem, the results show great promise for applying and extending our new solution methods and analytical results to optimize berth allocation for other more complicated problems in various applications. This study establishes a sound base for pursuing such challenging and important research directions.

1.2. Outline

In the following, we present our new lower bound in Section 2, the new exact method in Section 3, and their extensions in Section 4. The computational results are illustrated in Section 5, followed by a conclusion in Section 6. All appendices are included in an online companion.

2. New Lower Bound

Let Z^* indicate the optimal objective value of the CBAP. We now propose a new lower bound on Z^*. To this end, we reformulate the CBAP in Section 2.1, and then derive a relaxation of the reformulation in Section 2.2. In Section 2.3, we show that the relaxation can be solved efficiently in $O(n^2)$ time to obtain the new lower bound. In Section 2.4, we prove that the new lower bound equals Z^* for a special case of the CBAP. In Section 2.1 and Section 2.4, we show that the new lower bound is at least as good as, and can be significantly better than, the lower bound from the linear programming relaxation of model IP_1 (described in Online Appendix A).

2.1. Model RF: Reformulation of the CBAP

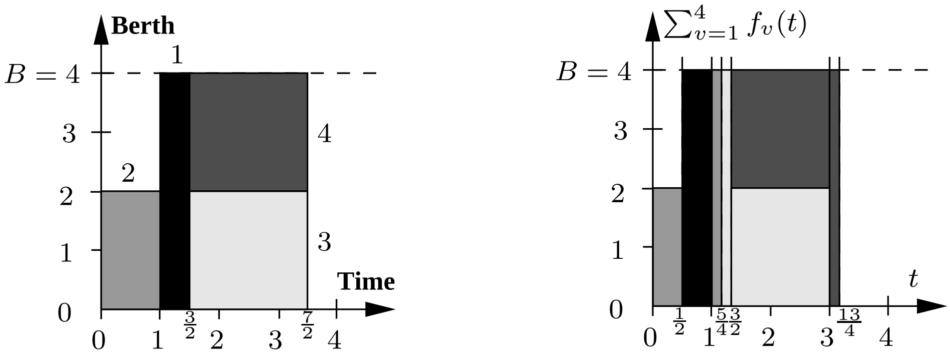
For each vessel $v \in V$, let (v, x_v, y_v) indicate its berth allocation. A solution to the CBAP can be represented by $\{(v, x_v, y_v) : v \in V\}$. Let $\mathbb{Y}$ indicate the set of all feasible solutions to the CBAP that satisfy $x_v \geq a_v$ for $v \in V$, $0 \leq y_v \leq B - l_v$ and the non-overlapping constraint (i.e., for any two different vessels v and u in V, there exists no time $t \geq 0$ and no berth location p with $0 \leq p < B$, such that $x_v \leq t < x_v + t_v$, $x_u \leq t < x_u + t_u$, $y_v \leq p < y_v + l_v$ and $y_u \leq p < y_u + l_u$). Hence, the CBAP can be represented as $\min \sum_{v \in V} w_v(x_v + t_v - a_v)$ subject to $\{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}$.

Before introducing our new lower bound on the optimal objective value Z^*, we need to reformulate the CBAP as follows. Consider any feasible solution $\{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}$. For each vessel $v \in V$, let $f_v(t)$ for $t \geq 0$ denote a function that represents the quay space occupied by v at time t, which can be represented by the following staircase function with three pieces:

$$f_v(t) = \begin{cases} 0, & \text{for } t \in [0, x_v), \\ l_v, & \text{for } t \in [x_v, x_v + t_v), \\ 0, & \text{for } t \in [x_v + t_v, \infty). \end{cases} \tag{1}$$

Using $f_v(t)$, we can represent the turnaround time, $x_v + t_v - a_v$ of v as follows. Note that $\int_{t \in [0,\infty)} t f_v(t) \mathrm{d}t = \int_{t \in [x_v, x_v + t_v)} l_v t \mathrm{d}t = \frac{1}{2} l_v t_v (2x_v + t_v) = l_v t_v x_v + \frac{1}{2} l_v t_v^2$ and $\int_{t \in [0,\infty)} f_v(t) \mathrm{d}t = \int_{t \in [x_v, x_v + t_v)} l_v \mathrm{d}t = l_v t_v$. We can obtain that

$$\begin{aligned} x_v + t_v - a_v &= x_v - a_v + t_v = \frac{1}{l_v t_v} [\int_{t \in [0,\infty)} t f_v(t) \mathrm{d}t - \frac{1}{2} l_v t_v^2] - \frac{a_v}{l_v t_v} \int_{t \in [0,\infty)} f_v(t) \mathrm{d}t + t_v \\ &= \frac{1}{l_v t_v} \int_{t \in [0,\infty)} (t - a_v) f_v(t) \mathrm{d}t + \frac{1}{2} t_v. \end{aligned}$$



(a) An optimal berth allocation with $Z^* = 7$. (b) A stacked bar chart for $[f_1, f_2, f_3, f_4]$ in Example 2.

Figure 2 Illustration of the new lower bound for an instance with $B = 4$, $n = 4$, $a_1 = 1/2$, $a_2 = 0$, $a_3 = a_4 = 1$, $l_1 = 4$, $l_2 = l_3 = l_4 = 2$, $t_1 = 1/2$, $t_2 = 1$, $t_3 = t_4 = 2$, and $w_1 = w_2 = w_3 = w_4 = 1$.

Moreover, by defining W as the following function on vector $[f_1, ..., f_n]$,

$$W(f_1, ..., f_n) := \sum_{v\in V} \frac{w_v}{l_v t_v} \int_{t\in[0,\infty)} (t - a_v) f_v(t) \mathrm{d}t, \tag{2}$$

we can represent the total weighted turnaround time of the vessels as follows:

$$\sum_{v\in V} w_v(x_v + t_v - a_v) = W(f_1, ..., f_n) + \frac{1}{2}\sum_{v\in V} w_v t_v. \tag{3}$$

Hence, the CBAP can be reformulated as the following optimization model:

$$\begin{aligned} \text{(RF)} \quad \min \ & W(f_1, ..., f_n) + \frac{1}{2}\sum_{v\in V} w_v t_v \\ \text{s.t.} \ & (1) \text{ and } \{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}. \end{aligned}$$

Example 1. Figure 2 shows an optimal solution in the space-time diagram for a four vessel instance. It can be seen that the optimal objective value $Z^* = (1 + 1 + 5/2 + 5/2) = 7$, and

$$\begin{aligned} W(f_1, f_2, f_3, f_4) + \frac{1}{2}\sum_{v=1}^{4} w_v t_v = & \frac{1}{4\times 1/2}\int_1^{3/2} 4(t - 1/2)\mathrm{d}t + \frac{1}{2\times 1}\int_0^1 2t\mathrm{d}t + \frac{1}{2\times 2}\int_{3/2}^{7/2} 2(t-1)\mathrm{d}t \\ & + \frac{1}{2\times 2}\int_{3/2}^{7/2} 2(t-1)\mathrm{d}t + \frac{1/2 + 1 + 2 + 2}{2} = 7 = Z^*. \end{aligned}$$

2.2. Relaxing Model RF for New Lower Bound $Z_{\mathbb{F}}$

In order to obtain a new lower bound on the optimal objective value Z^*, we need to derive a relaxation of model RF by replacing its original constraints, (1) and $\{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}$, with the following valid constraints (4)–(8), which are imposed only on vector $[f_1, ..., f_n]$.

Consider any $\{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}$ and $[f_1, ..., f_n]$ that are feasible to model RF and thus satisfy (1). For each vessel $v \in V$, from (1) we know that f_v is a staircase function with three pieces, and each piece is defined on a right-half open interval in $[0, \infty)$. Let $\mathbb{S}$ denote the set of all staircase functions, with each piece defined on a right-half open interval in $[0, \infty)$. Thus, constraint (4) below is valid for model RF:

$$f_v \in \mathbb{S}, \quad \forall v \in V. \tag{4}$$

For each $v \in V$, since $x_v \geq a_v$, and since (1) implies that $f_v(t) = 0$ for $t \in [0, x_v)$, we obtain that constraint (5) below is valid for model RF:

$$f_v(t) = 0, \quad \forall t \in [0, a_v) \text{ and } \forall v \in V. \tag{5}$$

For each $v \in V$, we know that $f_v(t)$ equals l_v if $t \in [x_v, x_v + t_v)$ and equals zero otherwise. Thus, constraint (6) below is valid for model RF:

$$\int_{t \in [0,\infty)} f_v(t) \mathrm{d}t = l_v t_v, \quad \forall v \in V, \tag{6}$$

which means that the area covered by $f_v(t)$ in the space-time diagram equals $l_v t_v$. Moreover, for each $t' \in [a_v, a_v + t_v)$, we have that $\int_{t \in [0,t')} f_v(t) \mathrm{d}t = l_v(t' - a_v)$, implying that constraint (7) below is also valid for model RF:

$$\int_{t \in [0,t')} f_v(t) \mathrm{d}t \leq l_v(t' - a_v), \quad \forall t' \in [a_v, a_v + t_v) \text{ and } \forall v \in V, \tag{7}$$

which means that the area covered by $f_v(t)$ for $0 \leq t < t'$ does not exceed $l_v(t' - a_v)$.

Furthermore, by the non-overlapping constraint, the total quay space occupied by all the vessels in V cannot exceed B for each time t. Thus, (8) below is also valid for model RF:

$$\sum_{v \in V} f_v(t) \leq B, \quad \forall t \in [0, \infty). \tag{8}$$

Notice that valid constraints (4)–(8) for model RF are imposed only on $[f_1, ..., f_n]$. Define $\mathbb{F} := \{[f_1, ..., f_n] \in \mathbb{S}^n : (5)\text{–}(8)\}$. Thus, the optimization model $Z_{\mathbb{F}}$ below is a relaxation of model RF:

$$\begin{aligned} (Z_{\mathbb{F}}) \quad & \min W(f_1, ..., f_n) + \frac{1}{2} \sum_{v \in V} w_v t_v \\ \text{s.t.} \quad & [f_1, ..., f_n] \in \mathbb{F}. \end{aligned} \tag{9}$$

With a slight abuse of the notation, we use $Z_{\mathbb{F}}$ to also denote its optimal objective value.

Hence, by the theorem below, $Z_{\mathbb{F}}$ is a lower bound on Z^* for the CBAP.

Theorem 1. *$Z_{\mathbb{F}} \leq Z^*$.*

Proof. Since $Z_{\mathbb{F}}$ is a relaxation of model RF, we obtain that $Z_{\mathbb{F}} \leq Z^*$. □

Next, let Z_1^{LP} denote the lower bound on Z^* from the linear programming relaxation of IP_1 (described in Online Appendix A). The relaxation model for Z_1^{LP} has an optimal solution with $x_v = a_v$ for $v \in V$ (if the constant M in IP_1 is sufficiently large) (Ak and Erera 2011). Thus,

$$Z_1^{\mathrm{LP}} = \sum_{v \in V} w_v t_v. \tag{10}$$

The following proposition shows that our new lower bound $Z_{\mathbb{F}}$ is at least as good as, and can be better than Z_1^{LP}. (In Section 2.4, we will present a stronger result on the improvement.)

Proposition 1. *(i) $Z_1^{\mathrm{LP}} \leq Z_{\mathbb{F}}$; (ii) $Z_1^{\mathrm{LP}} < Z_{\mathbb{F}}$ for infinitely many instances.*

Proof. See Online Appendix B. □

For every $[f_1,...,f_n]\in\mathbb{F}$, it can be represented by a stacked bar chart. Let A denote the set of the endpoints of all the intervals of $f_v(t)$ for $v\in V$. Sort the endpoints in A according to their positions on the time-axis from left to right. It can be seen that each $f_v(t)$ equals a constant for t between two consecutive endpoints in A. Hence, we can plot a stacked bar chart so that for each interval between two consecutive endpoints in A, it stacks n bars of the same width equal to the interval length and of heights equal to $f_v(t)$ for $v=1,...,n$, respectively.

Example 2. For the instance in Figure 2(a), consider the vector $[f_1,f_2,f_3,f_4]$ defined below:

$$f_1(t)=\begin{cases}0,\ t\in[0,\frac{1}{2})\\ 4,\ t\in[\frac{1}{2},1)\\ 0,\ t\in[1,\infty)\end{cases},\ f_2(t)=\begin{cases}2,\ t\in[0,\frac{1}{2})\\ 0,\ t\in[\frac{1}{2},1)\\ 4,\ t\in[1,\frac{5}{4})\\ 0,\ t\in[\frac{5}{4},\infty)\end{cases},\ f_3(t)=\begin{cases}0,\ t\in[0,\frac{5}{4})\\ 4,\ t\in[\frac{5}{4},\frac{3}{2})\\ 2,\ t\in[\frac{3}{2},3)\\ 0,\ t\in[3,\infty)\end{cases},\ f_4(t)=\begin{cases}0,\ t\in[0,\frac{3}{2})\\ 2,\ t\in[\frac{3}{2},3)\\ 4,\ t\in[3,\frac{13}{4})\\ 0,\ t\in[\frac{13}{4},\infty)\end{cases},$$

which is represented by a stacked bar chart in Figure 2(b), where different colors indicate bars for different vessels. It can be easily seen that $f_v(t)$ for $1\le v\le 4$ satisfy (4), (5), (6) and (8), and that $f_1(t)$ and $f_2(t)$ for $v\in\{1,2\}$ satisfy (7). Moreover, we have that

$$\int_{t\in[0,t']}f_3(t)\mathrm{d}t=\begin{cases}0, & t'\in[0,\frac{5}{4})\\ 4(t'-\frac{5}{4}), & t'\in[\frac{5}{4},\frac{3}{2})\\ 1+2(t'-\frac{3}{2}), & t'\in[\frac{3}{2},3)\\ 4, & t'\in[2,\infty)\end{cases}\quad\text{and}\quad\int_{t\in[0,t']}f_4(t)\mathrm{d}t=\begin{cases}0, & t'\in[0,\frac{3}{2})\\ 2(t'-\frac{3}{2}), & t'\in[\frac{3}{2},3)\\ 3+4(t'-3), & t'\in[3,\frac{13}{4})\\ 4, & t'\in[\frac{13}{4},\infty)\end{cases},$$

both of which do not exceed $2(t'-1)$ for $t'\in[1,3)$, and thus satisfy (7). Hence, $[f_1,f_2,f_3,f_4]\in\mathbb{F}$. We will see later in Section 2.3.2 that such $[f_1,f_2,f_3,f_4]$ is optimal to $Z_{\mathbb{F}}$.

Remark 1. It can be seen that a better lower bound than $Z_{\mathbb{F}}$ can be obtained by adding to $\mathbb{F}$ more valid constraints on $[f_1,...,f_n]$. However, this will complicate the relaxation model so that it may be too intractable to be solved. In the following section, we will show that for the relaxation model that we have defined earlier, it is tractable to be solved by an $O(n^2)$ algorithm.

2.3. Computing the New Lower Bound $Z_{\mathbb{F}}$

In this section, we present a novel constructive algorithm that can return $Z_{\mathbb{F}}$ in $O(n^2)$ time.

2.3.1. Framework of the Algorithm By (9), to compute $Z_{\mathbb{F}}$, we need to minimize $W(f_1,...,f_n)+(1/2)\sum_{v\in V}w_vt_v$ over vectors of functions $[f_1,...,f_n]\in\mathbb{F}$, for which it is sufficient to minimize $W(f_1,...,f_n)$. Thus, since each integration term $\int_{t\in[0,\infty)}(t-a_v)f_v(t)\mathrm{d}t$ of $W(f_1,...,f_n)$ in (2) is weighted by $w_v/(l_vt_v)$, it is natural to construct each $f_v(t)$ by minimizing $\int_{t\in[0,\infty)}(t-a_v)f_v(t)\mathrm{d}t$, at each time selecting $f_v(t)$ for v with the largest $w_v/(l_vt_v)$.

Following this intuition, we sort vessels $v\in V$ by a non-increasing order of $w_v/(l_vt_v)$, relabel them as $1,2,...,n$, and construct $f_1,f_2,...,f_n$ sequentially. For each vessel v, given that $f_1,..,f_{v-1}$ have been constructed, we construct f_v by solving the following optimization model $Z^{(v)}(f_1,...,f_{v-1})$, aiming to minimize $\int_{t\in[0,\infty)}(t-a_v)f_v(t)\mathrm{d}t$ over $f_v(t)\in\mathbb{S}$, subject to constraints derived from (4)–(8) of model $Z_{\mathbb{F}}$, where (11)–(14) are equivalent to (4)–(7) on f_v, and (15) is equivalent to $\sum_{u=1}^{v}f_u\le B$, a relaxed constraint of (8):

$$Z^{(v)}(f_1,...,f_{v-1})=\min\int_{t\in[0,\infty)}(t-a_v)f_v(t)\mathrm{d}t$$

Algorithm 1 Computing $Z_{\mathbb{F}}$

1: Sort vessels $v \in V$ in a non-increasing order of $w_v/(l_v t_v)$, and relabel them as 1, 2, ..., n.

2: For each $v = 1, 2, ..., n$, given $f_u(t)$ for $1 \le u < v$, follow the algorithm in Section 2.3.2 to compute an optimal solution f_v to $Z^{(v)}(f_1, ..., f_{v-1})$ in $O(n)$ running time.

3: Return $W(f_1, .., f_n) + \frac{1}{2}\sum_{v \in V} w_v t_v$.

$$\text{s.t.} \quad f_v \in \mathbb{S}, \tag{11}$$

$$f_v(t) = 0, \ \forall t \in [0, a_v), \tag{12}$$

$$\int_{t \in [0,\infty)} f_v(t)\mathrm{d}t = l_v t_v, \tag{13}$$

$$\int_{t \in [0,t')} f_v(t)\mathrm{d}t \le l_v(t' - a_v), \ \forall t' \in [a_v, a_v + t_v), \tag{14}$$

$$f_v(t) \le B - \sum_{u=1}^{v-1} f_u(t), \ \forall t \in [0, \infty). \tag{15}$$

We summarize the above framework for computing $Z_{\mathbb{F}}$ in Algorithm 1. We will later explain in Section 2.3.2 how to solve $Z^{(v)}(f_1, ..., f_{v-1})$ in Step 2 of Algorithm 1 in $O(n)$ running time for each $v \in V$, so as to guarantee an $O(n^2)$ running time for Algorithm 1.

To show that Algorithm 1 returns exactly $Z_{\mathbb{F}}$, we need to derive a sufficient and necessary condition as follows for f_v to be an optimal solution to $Z^{(v)}(f_1, ..., f_{v-1})$, and this condition will also be applied later in Section 2.3.2 to solve $Z^{(v)}(f_1, ..., f_{v-1})$. To minimize $\int_{t \in [0,\infty)}(t - a_v) f_v(t)\mathrm{d}t$ for $Z^{(v)}(f_1, ..., f_{v-1})$, since $t - a_v$ is increasing in t, for each $p \ge 0$ it is natural to assign as large a value as possible to $f_v(p)$ if $f_v(t)$ are fixed for $0 \le t < p$. In other words, for any optimal solution f_v to $Z^{(v)}(f_1, ..., f_{v-1})$, it is natural to expect that its value at p for each $p \ge 0$ is always the maximum among all feasible solutions to $Z^{(v)}(f_1, ..., f_{v-1})$ that have the same values as $f_v(t)$ for $0 \le t < p$. Lemma 1 below shows that this is in fact a sufficient and necessary condition for f_v to be an optimal solution to $Z^{(v)}(f_1, ..., f_{v-1})$.

Lemma 1. *For each $v = 1, ..., n$, given f_u for $1 \le u \le v - 1$, a feasible solution f_v to $Z^{(v)}(f_1, ..., f_{v-1})$ is optimal if and only if it satisfies the condition that for each $p \ge 0$, we have $f_v(p) \ge f'_v(p)$ for every feasible solution f'_v to $Z^{(v)}(f_1, ..., f_{v-1})$ with $f'_v(t) = f_v(t)$ for $0 \le t < p$.*

Proof. See Online Appendix C. □

We can now establish Theorem 2 to show that Algorithm 1 returns $Z_{\mathbb{F}}$ in $O(n^2)$ time.

Theorem 2. *Algorithm 1 returns $Z_{\mathbb{F}}$ in $O(n^2)$ running time.*

Proof Sketch. (See Online Appendix D for details.) As claimed in Step 2 of Algorithm 1, which will be proved later in Section 2.3.2, we can obtain an optimal solution f_v to $Z^{(v)}(f_1, ..., f_{v-1})$ in $O(n)$ time for each $v = 1, ..., n$. Thus, Algorithm 1 runs in $O(n^2)$ time.

To further prove Theorem 2, it is sufficient to show that $W(f_1, ..., f_n) + \frac{1}{2}\sum_{v=1}^{n} w_v t_v = Z_{\mathbb{F}}$, for which it is sufficient to show that $[f_1, .., f_n]$ is an optimal solution to $Z_{\mathbb{F}}$. By contradiction, suppose such $[f_1^*, .., f_n^*]$ is not optimal to $Z_{\mathbb{F}}$. Consider any optimal solution $[f_1^*, ..., f_n^*]$ to $Z_{\mathbb{F}}$.

Let $v \in V$ indicate the smallest vessel index with $f_v^*(t) \neq f_v(t)$ for some $t \geq 0$. Let $p \geq 0$ indicate the smallest time point with $f_v^*(p) \neq f_v(p)$. Without loss of generality, we can assume that the optimal solution $[f_1^*, ..., f_n^*]$ to $Z_{\mathbb{F}}$ is selected in such a way that v is maximized, breaking ties by maximizing p, and then breaking ties by minimizing $|f_v^*(p) - f_v(p)|$. It can be seen that both f_v and f_v^* are feasible solutions to $Z^{(v)}(f_1, ..., f_{v-1})$. Thus, since f_v is an optimal solution to $Z^{(v)}(f_1, ..., f_{v-1})$, by Lemma 1 and $f_v^*(p) \neq f_v(p)$, we obtain $f_v^*(p) < f_v(p)$. Thus, we can change $[f_1^*, .., f_n^*]$, but without changing $f_u^*(t)$ for $1 \leq u < v$ or $f_v^*(t)$ for $0 \leq t < p$, to construct a feasible solution having either a smaller objective value, or the same objective value but a smaller value of $|f_v^*(p) - f_v(p)|$, contradicting the definition of $[f_1^*, ..., f_n^*]$. $\square$

2.3.2. Solving $Z^{(v)}(f_1, ..., f_{v-1})$ for Step 2 of the Algorithm: Illustration of Major Idea For each vessel $v \in V$, given $f_1, ..., f_{v-1}$, where each f_u for $1 \leq u \leq v-1$ is a feasible solution to $Z^{(u)}(f_1, ..., f_{u-1})$, we construct f_v in $O(n)$ running time based on the following idea:

In view of Lemma 1, to construct f_v to be an optimal solution to $Z^{(v)}(f_1, ..., f_{v-1})$, it is natural to assign as large a value as possible to $f_v(t)$ first for the earliest time t. For each $t \geq 0$, from (15) we know that $f_v(t)$ has a maximum possible value equal to $B - g_v(t)$, where $g_v(t) := \sum_{u=1}^{v-1} f_u(t)$ indicates the total value assigned to $f_1(t), ..., f_{v-1}(t)$. Moreover, since each f_u belongs to $\mathbb{S}$ for $1 \leq u \leq v-1$, it can be seen that g_v also belongs to $\mathbb{S}$, that is, g_v is also a staircase function on $[0, \infty)$ over right-half open intervals. Denote the intervals of g_v by $[b_{v0}, b_{v1}), ..., [b_{v,m_v-1}, b_{v,m_v})$ with $b_{v0} = 0 < b_{v1} < ... < b_{v,m_v} = \infty$, where m_v indicates the number of the intervals. Without loss of generality, we can assume that $\{b_{v0}, ..., b_{v,m_v}\}$ contains a_v and $a_v + t_v$, since otherwise the interval that contains a_v or $a_v + t_v$ can be split at a_v or $a_v + t_v$ respectively.

For $1 \leq j \leq m_v$, since $g_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$ equals a constant, denoted by β_{vj}, we have $g_v(t) = \beta_{vj}$ for $t \in [b_{v,j-1}, b_{vj})$. From (15) we know that $f_v(t)$ for all $t \in [b_{v,j-1}, b_{vj})$ has a common maximum possible value equal to $B - \beta_{vj}$, which leads to the following constraint on $f_v(t)$:

$$f_v(t) \leq B - \beta_{vj}, \text{ for } t \in [b_{v,j-1}, b_{vj}). \tag{16}$$

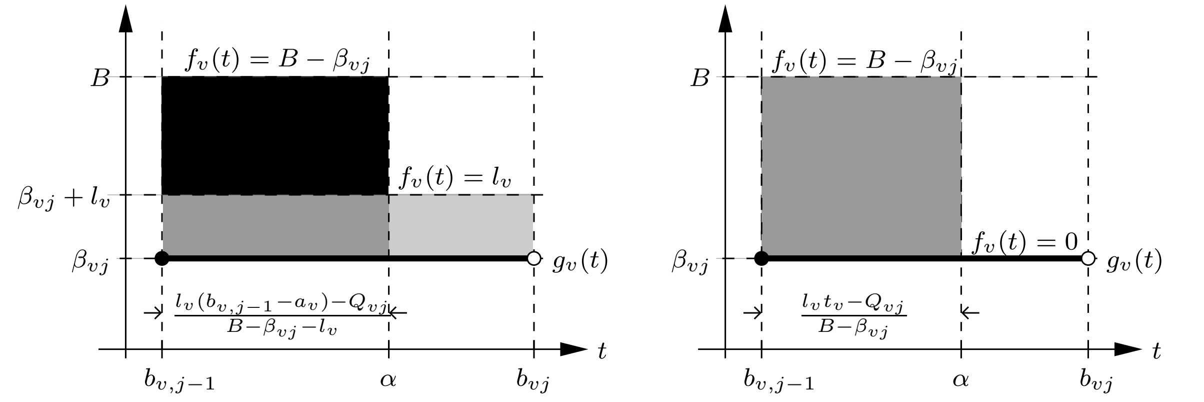
Thus, to save computing time, we can use this common maximum possible value, $B - \beta_{vj}$, to construct $f_v(t)$ as follows for the entire interval $[b_{v,j-1}, b_{vj})$, rather than for each individual t.

For each $j = 1, 2, ..., m_v$, sequentially, given that $f_v(t)$ has been constructed for $t \in [0, b_{v,j-1})$, we can construct $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$ as follows. Following the idea mentioned earlier from Lemma 1, we shall assign the maximum possible value $B - \beta_{vj}$ to $f_v(t)$ as early and as much as we can. However, to ensure that $f_v(t)$ can eventually be feasible to $Z^{(v)}(f_1, ..., f_{v-1})$, our assignment must be restricted by not only constraint (12), i.e., $f_v(t) = 0$ for $0 \leq t < a_v$, but also by the following constraints (17) and (18) that are derived from (13) and (14):

- By (13), the total area in the space-time diagram to be covered by $f_v(t)$ equals $l_v t_v$. Let

$$Q_{vj} := \int_{t \in [0, b_{v,j-1})} f_v(t) \mathrm{d}t$$

indicate the total area that has been covered by $f_v(t)$ for $0 \leq t < b_{v,j-1}$. Thus, $l_v t_v - Q_{vj}$, referred to as *the total remaining area of f_v at $b_{v,j-1}$*, indicates the area that needs to be covered by



(a) For Case I: $b_{vj} \leq a_v + t_v$, where the black area is covered by $f_v(t) - l_v$.

(b) For Case II: $b_v > a_v + t_v$.

Figure 3 Construction of an optimal solution $f_v(t)$ to $Z^{(v)}(f_1, ..., f_{v-1})$ for $t \in [b_{v,j-1}, b_{vj})$, where gray areas are covered by the newly assigned $f_v(t)$, and α is assumed to be less than b_{vj}.

$f_v(t)$ for $t \geq b_{v,j-1}$, implying that $l_v t_v - Q_{vj}$ is a maximum possible area that can be covered by $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$. This leads to the following constraint (17) on $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$:

$$\int_{t \in [b_{v,j-1}, b_{vj})} f_v(t) \mathrm{d}t \leq l_v t_v - Q_{vj}. \tag{17}$$

• According to (14), for each $t' \in [a_v, a_v + t_v)$, the total area covered by $f_v(t)$ for $0 \leq t < t'$ cannot exceed $l_v(t' - a_v)$. Thus, we refer to $l_v(t' - a_v)$ as *the available area* of f_v at t'. If $t' \in [b_{v,j-1}, b_{vj})$, then from (14) we can obtain the following constraint (18) on $f_v(t)$ for $t \in [b_{v,j-1}, t')$:

$$\int_{t \in [b_{v,j-1}, t')} f_v(t) \mathrm{d}t \leq l_v(t' - a_v) - Q_{vj}, \tag{18}$$

where the right hand side indicates the remaining area that is available at t' but has not been covered by $f_v(t)$ for $0 \leq t < b_{v,j-1}$, implying that $l_v(t' - a_v) - Q_{vj}$ is a maximum possible area that can be covered by $f_v(t)$ for $t \in [b_{v,j-1}, t')$.

Hence, in the following construction, we will assign the maximum possible value $B - \beta_{vj}$ as early and as much as possible to $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$, subject to constraints (12), (17) and (18). First, for any of the following three situations, we have to set $f_v(t) = 0$ for all $t \in [b_{v,j-1}, b_{vj})$:

• If $Q_{vj} = l_v t_v$, then $f_v(t)$ for $0 \leq t \leq b_{v,j-1}$ has covered all the area $l_v t_v$, and its total remaining area at $b_{v,j-1}$ becomes zero, which, together with (17), implies that $f_v(t) = 0$ for $t \in [b_{v,j-1}, b_{vj})$.

• If $\beta_{vj} = B$ for $t \in [b_{v,j-1}, b_{vj})$, then the total value assigned to $f_1(t), ..., f_{v-1}(t)$ has reached B, and thus from (16) we obtain that $f_v(t) = 0$ for $t \in [b_{v,j-1}, b_{vj})$.

• If $a_v \geq b_{vj}$, then each time $t \in [b_{v,j-1}, b_{vj})$ is earlier than the arrival time a_v, and thus from (12) we also obtain $f_v(t) = 0$ for $t \in [b_{v,j-1}, b_{vj})$.

We next need to consider only the situation where $Q_{vj} < l_v t_v$, $\beta_{vj} < B$ and $a_v < b_{vj}$. Since $a_v \in \{b_{v0}, ..., b_{v,m_v}\}$ and $a_v < b_{vj}$, we know $a_v \leq b_{v,j-1}$, implying that each $t \in [b_{v,j-1}, b_{vj})$ satisfies $a_v \leq t$. Thus, in this situation, constraint (12) is not imposed on the construction of $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$, and only constraints (17) and (18) need to be considered. We now continue our construction for the following two cases, as illustrated in Figure 3:

- Case I: $b_{vj} \le a_v + t_v$. Since $a_v \le b_{v,j-1}$, we have $[b_{v,j-1}, b_{vj}) \subseteq [a_v, a_v + t_v)$. Thus, if $l_v \ge B - \beta_{vj}$, then even if we assign $B - \beta_{vj}$ to all $f_v(t)$ with $t \in [b_{v,j-1}, b_{vj})$, constraints (17) and (18) are still satisfied, and therefore we set $f_v(t) = B - \beta_{vj}$ for all $t \in [b_{v,j-1}, b_{vj})$.

Otherwise, $l_v < B - \beta_{vj}$. Since $[b_{v,j-1}, b_{vj}) \subseteq [a_v, a_v + t_v)$, each $t' \in [b_{v,j-1}, b_{vj})$ satisfies that $l_v(t' - a_v) - Q_{vj} \le l_v t_v - Q_{vj}$, which implies that if (18) is satisfied, then (17) is also satisfied. Thus, in this case, only constraint (18) needs to be considered.

Suppose we now assign $B - \beta_{vj}$ to all $f_v(t)$ with $t \in [b_{v,j-1}, t')$ for some $t' \in [b_{v,j-1}, b_{vj})$. To satisfy (18), t' must satisfy $(t' - b_{v,j-1})(B - \beta_{vj}) \le l_v(t' - a_v) - Q_{vj}$, which implies that

$$(B - \beta_{vj} - l_v)(t' - b_{v,j-1}) \le l_v(b_{v,j-1} - a_v) - Q_{vj}. \tag{19}$$

In (19), the right hand side is the remaining area available to f_v but not yet covered at $b_{v,j-1}$, which allows $f_v(t)$ for $t \in [b_{v,j-1}, t')$ to be greater than l_v, and thus it is a maximal possible area that can be covered by $[f_v(t) - l_v]$ for $t \in [b_{v,j-1}, t')$. The left hand side is the actual area covered by $[f_v(t) - l_v]$ for $t \in [b_{v,j-1}, t')$, where $f_v(t) - l_v = B - \beta_{vj} - l_v$. See Figure 3(a). By (19), $t' \le [l_v(b_{v,j-1} - a_v) - Q_{vj}]/(B - \beta_{vj} - l_v) + b_{v,j-1}$. Thus, from $t' \le b_{vj}$, we obtain $t' \le \alpha$, where

$$\alpha := \min\{b_{vj}, \frac{l_v(b_{v,j-1} - a_v) - Q_{vj}}{B - \beta_{vj} - l_v} + b_{v,j-1}\}.$$

This implies that we can set $f_v(t) = B - \beta_{vj}$ for t from $b_{v,j-1}$ and only up to α (not including α), without violating constraint (18).

Hence, if $\alpha = b_{vj}$, we set $f_v(t) = B - \beta_{vj}$ for all $t \in [b_{v,j-1}, b_{vj})$. Otherwise, $\alpha < b_{vj}$. We then set $f_v(t) = B - \beta_{vj}$ for all $t \in [b_{v,j-1}, \alpha)$, which implies that (18) is satisfied at equality for $t' = \alpha$. This means that $f_v(t)$ *for* $t \in [0, \alpha)$ *has covered all its available area* $l_v(\alpha - a_v)$. Thus, to satisfy (18) for $t' \in [\alpha, b_{vj})$, we have $f_v(t) \le l_v$ for $t \in [\alpha, b_{vj})$. It can be seen that even if we assign l_v to all $f_v(t)$ with $t \in [\alpha, b_{vj})$, constraint (18) is still satisfied. Thus, we set $f_v(t) = l_v$ for $t \in [\alpha, b_{vj})$.

- Case II: $b_{vj} > a_v + t_v$. Since $b_{vj} > a_v + t_v$ and $(a_v + t_v) \in \{b_{v0}, ..., b_{v,m_v}\}$, we have $b_{v,j-1} \ge a_v + t_v$, which implies $[a_v, a_v + t_v) \subseteq [0, b_{v,j-1})$. Thus, each $t \in [b_{v,j-1}, b_{vj})$ satisfies $t \ge a_v + t_v$, so that (18) is not imposed on $f_v(t)$ for $t \in (b_{v,j-1}, b_{vj}]$. Thus, all the remaining area $l_v t_v - Q_{vj}$ is available to $f_v(t)$ for $t \in (b_{v,j-1}, b_{vj}]$, and only constraint (17) needs to be considered.

We can now determine the maximum value of $t' \in [b_{v,j-1}, b_{vj})$ as follows, such that even if we assign $B - \beta_{vj}$ to all $f_v(t)$ with $t \in [b_{v,j-1}, t')$, constraint (17) is still satisfied. By (17), t' must satisfy $(t' - b_{v,j-1})(B - \beta_{vj}) \le l_v t_v - Q_{vj}$. Thus, we have $t' \le (l_v t_v - Q_{vj})/(B - \beta_{vj}) + b_{v,j-1}$, which, together with $t' \le b_{vj}$, implies $t' \le \alpha$, where

$$\alpha := \min\{b_{vj}, \frac{l_v t_v - Q_{vj}}{B - \beta_{vj}} + b_{v,j-1}\}.$$

Thus, we can set $f_v(t) = B - \beta_{vj}$ for t from $b_{v,j-1}$ and only up to α (not including α), without violating constraint (17).

Hence, similar to Case I, if $\alpha = b_{vj}$, we set $f_v(t) = B - \beta_{vj}$ for all $t \in [b_{v,j-1}, b_{vj})$. Otherwise, $\alpha < b_{vj}$. We then set $f_v(t) = B - \beta_{vj}$ for all $t \in [b_{v,j-1}, \alpha)$, which implies that $f_v(t)$ *for* $t \in [0, \alpha)$ *has covered all the area* $l_v t_v$*, and its total remaining area at* α *becomes zero*, which, together with (17), implies that we shall set $f_v(t) = 0$ for $t \in [\alpha, b_{vj})$.

Algorithm 2 Computing an optimal solution f_v to $Z^{(v)}(f_1, ..., f_{v-1})$.

Let $g_v(t) = \sum_{u=1}^{v-1} f_u(t)$, which is a staircase function over intervals $[b_{v0}, b_{v1}), ..., [b_{v,m_v-1}, b_{v,m_v})$ with $0 = b_{v0} < ... < b_{v,m_v} = \infty$ and $\{a_v, a_v + t_v\} \subseteq \{b_{v0}, ..., b_{v,m_v}\}$, and for each $1 \le j \le m_v$, let β_{vj} indicate the constant value of $g_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$.
for $j = 1, 2, .., m_v$ **do**
  Let $Q_{vj} = \int_{t\in[0,b_{v,j-1})} f_v(t)\mathrm{d}t$.
  if $Q_{vj} = l_v t_v$, or $\beta_{vj} = B$, or $a_v \ge b_{vj}$ **then**
    Set $f_v(t) = 0$ for $t \in [b_{v,j-1}, b_{vj})$.
  else if $b_{vj} \le a_v + t_v$ **then** ▷ Case I:
    if $l_v \ge B - \beta_{vj}$ **then**
      Set $f_v(t) = B - \beta_{vj}$ for $t \in [b_{v,j-1}, b_{vj})$.
    else
      Let $\alpha = \min\{b_{vj}, [l_v(b_{v,j-1} - a_v) - Q_{vj}]/(B - \beta_{vj} - l_v) + b_{v,j-1}\}$.
      If $\alpha = b_{vj}$, then set $f_v(t) = B - \beta_{vj}$ for $t \in [b_{v,j-1}, b_{vj})$.
      If $\alpha < b_{vj}$, then set $f_v(t) = B - \beta_{vj}$ for $t \in [b_{v,j-1}, \alpha)$, and $f_v(t) = l_v$ for $t \in [\alpha, b_{vj})$.
    end if
  else ▷ Case II:
    Let $\alpha = \min\{b_{vj}, (l_v t_v - Q_{vj})/(B - \beta_{vj}) + b_{v,j-1}\}$.
    If $\alpha = b_{vj}$, then set $f_v(t) = B - \beta_{vj}$ for $t \in [b_{v,j-1}, b_{vj})$.
    If $\alpha < b_{vj}$, then set $f_v(t) = B - \beta_{vj}$ for $t \in [b_{v,j-1}, \alpha)$, and $f_v(t) = 0$ for $t \in [\alpha, b_{vj})$.
  end if
end for
Return f_v.

We have by now completed the construction of $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$, which can be repeated for $j = 1, 2, ..., m_v$ to obtain $f_v(t)$ for $t \in [0, \infty)$. The pseudocode of the construction is in Algorithm 2. We can establish Theorem 3 to show that f_v obtained is optimal to $Z^{(v)}(f_1, ..., f_{v-1})$.

Theorem 3. *Algorithm 2 can return an optimal solution f_v to $Z^{(v)}(f_1, ..., f_{v-1})$ in $O(n)$ time.*

Proof Sketch. (See Online Appendix E for details.) We can first verify that f_v is a feasible solution to $Z^{(v)}(f_1, ..., f_{v-1})$, and then prove that f_v is optimal by verifying the condition in Lemma 1. Moreover, it can be seen that Algorithm 2 runs in $O(m_v)$ time. Thus, we can prove that Algorithm 2 runs in $O(n)$ time by showing that m_v is in $O(n)$. □

See Online Appendix F that illustrates an example as to how Algorithm 2 is used by Algorithm 1 to construct f_1, f_2, f_3, and f_4, sequentially, in the computation of $Z_{\mathbb{F}}$ for the instance in Figure 2, where the resulting $[f_1, f_2, f_3, f_4]$ is the same as that in Example 2 and Figure 2(b).

2.4. Special Case of the CBAP Satisfying $Z_{\mathbb{F}} = Z^*$

Consider a special case of the CBAP, where $l_v = B$ and $a_v = a$ for $v \in V$, and $a \ge 0$ is a constant, that is, all vessels have sizes equal to the quay length, and have equal arrival times. This special case is equivalent to the classic single machine scheduling problem for minimizing the total weighted completion time, and thus it is optimal to process vessels with the largest w_v/t_v first (Pinedo 2012). Proposition 2 below shows that in this special case, our new lower bound $Z_{\mathbb{F}}$ equals the optimal objective value Z^*.

Proposition 2. *If $l_v = B$ and $a_v = a$ for $v \in V$, where $a \geq 0$ is a constant, then $Z_{\mathbb{F}} = Z^*$.*

Proof. See Online Appendix G. □

From Proposition 2, we can derive the following corollary, which strengthens Proposition 1 and implies that the improvement of $Z_{\mathbb{F}}$ against Z_1^{LP}, measured by the gap ratio $(Z_{\mathbb{F}} - Z_1^{\mathrm{LP}})/Z_1^{\mathrm{LP}}$, can be arbitrarily large when the number of vessels n grows to infinity.

Corollary 1. *$(Z_{\mathbb{F}} - Z_1^{\mathrm{LP}})/Z_1^{\mathrm{LP}}$ can be as large as $(n-1)/2$ for any $n \geq 1$.*

Proof. Consider any instance with $l_v = B$, $a_v = a$, $t_v = t$ and $w_v = w$ for $v \in V$, where $a \geq 0$, $t > 0$ and $w > 0$. Since vessels are identical and their sizes all equal the quay length, it is optimal to handle vessels one by one. Thus, $Z^* = w(1+2+...+n)t = w(n+1)nt/2$. By (10) and Proposition 2, we have $Z_1^{\mathrm{LP}} = wnt$ and $Z_{\mathbb{F}} = Z^*$. Thus, $(Z_{\mathbb{F}} - Z_1^{\mathrm{LP}})/Z_1^{\mathrm{LP}} = (n-1)/2$. □

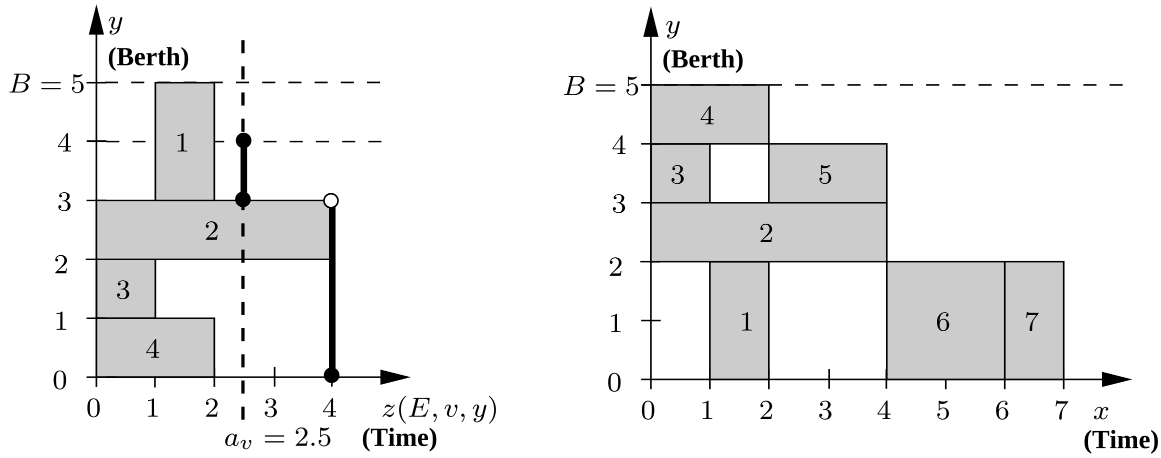
3. New Exact Method

Our exact method follows a branch and bound approach that has been commonly adopted for the two dimensional bin packing problem (Martello et al. 2000). In Section 3.1, we present its branching rules. In Section 3.2, we illustrate its bounding procedure, which makes use of the new lower bound introduced in Section 2, and applies a new heuristic to compute upper bounds. To reduce the search space, our exact method also adopts several new dominance rules for pruning nodes of the search tree, and these are introduced in Section 3.3.

3.1. Branching Rules

As it goes down the search tree from a root node, the exact method assigns positions in the space-time diagram to vessels one by one. When exploring the search tree, among all the nodes visited but not branched, it selects a node p, and then branches node p to generate new nodes by assigning positions to one more vessel. There are two strategies commonly used to select node p, including a *depth-first* strategy that selects a node furthest from the root, and a *breadth-first* strategy that selects a node nearest to the root. Compared with the breadth-first strategy, the depth-first strategy requires less computer memory, but it is often stuck in branches that do not lead to good solutions. We therefore adopt a *mixed strategy* that applies the breadth-first strategy as long as the memory consumption is below a certain threshold, and applies the depth-first strategy otherwise.

We now consider any selected node p, and let E denote the current partial solution at node p. We still assume that the horizontal axis of the space-time diagram represents the time, and the vertical axis represents the berth locations. Let $V(E)$ indicate the set of vessels whose positions in the space-time diagram have been assigned by E. Descendant nodes of p can be generated by selecting in turn each unassigned vessel $v \in V \setminus V(E)$, assigning v all its



(a) $z(E,v,y)$, shown in bold lines, for a partial solution E with four vessels, 1, 2, 3, and 4, and for vessel v with $a_v = 2.5$ and $l_v = 1$.

(b) A berth allocation of 7 vessels, where $a_1 = 1$, $a_4 = a_5 = 0$, $a_6 = 4$, $a_7 = 0$, $w_1 = 1$, $w_4 = 1$, $w_5 = 2$, and $w_6 = w_7 = 2$.

Figure 4 Examples for illustration of the new exact algorithm.

"admissible" positions (x_v, y_v) in the space-time diagram, and increasing E by including each new assignment (v, x_v, y_v).

To define the admissible positions, we follow an argument similar to that in Martello et al. (2000) for the two-dimensional bin packing problem, which implies that for every optimal solution $\{(u, x_u, y_u) : u \in V\}$ to the CBAP, there exists an ordering of the vessels such that for $u \in V$ and $r \in V$ with $u < r$, either $y_r \geq y_u + l_u$ or $x_r \geq x_u + t_u$. Thus, unassigned vessels may be placed only at positions above or on the right of any assigned vessel in the space-time diagram. In effect, the assigned vessels define an "envelope" that separates the two areas where unassigned vessels may or may not be placed, and this can be represented by $z'(E,y)$, indicating the latest departure time of the assigned vessels that occupy berth locations at or above y:

$$z'(E,y) := \max\{x_u + t_u : \ \forall u \in V(E) \text{ with } y < y_u + l_u\}, \text{ for } y \in [0, B). \tag{20}$$

Hence, each unassigned vessel $v \in V \setminus V(E)$ can only be placed at positions (x_v, y_v) with $z'(E, y_v) \leq x_v$, $a_v \leq x_v$, and $y_v \leq B - l_v$. Thus, for each $0 \leq y \leq B - l_v$, we can use $Z(E,v,y)$ to indicate the earliest mooring time for v to be assigned y as its starting berth location, where

$$z(E,v,y) := \max\{z'(E,y), a_v\}, \text{ for } y \in [0, B - l_v]. \tag{21}$$

It can be seen that each $Z(E,v,y)$ is a staircase function for $y \in [0, B - l_v]$. See Figure 4(a) for an example of $z(E,v,y)$.

Moreover, by the argument in Guan and Cheung (2004), every feasible solution to the CBAP can, without increasing the objective value, be transformed to a feasible solution where no vessel can be moved leftward or downward in the space-time diagram. Thus, for each unassigned vessel $v \in V \setminus V(E)$ that is selected to be placed, it may be placed only at the positions on $z(E,v,y)$ where the slope of $z(E,v,y)$ changes from vertical to horizontal. See positions (2.5,3) and (4,0) in Figure 4(a) for a vessel v with $a_v = 2.5$ and $l_v = 1$. We define such positions as "admissible" positions for v. They can be computed in $O(|V(E)| \log |V(E)|)$ time by an algorithm similar to that proposed by Martello et al. (2000).

3.2. Bounding Procedure with a New Heuristic

Consider the current selected node p of the search tree and its corresponding partial solution E. Let UB indicate the objective value of the current best feasible solution that has been found, and LB indicate the value of the current best lower bound that has been found for the optimal solution. If LB equals UB, then the current best feasible solution is optimal.

For each decedent node of p that is newly generated by assigning an admissible position (x_v, y_v) to an unassigned vessel $v \in V \setminus V(E)$, we can compute a lower bound as follows for the remaining problem that decides on positions for other unassigned vessels in $V \setminus V(E) \setminus \{v\}$. Let $E' = E \cup \{(v, x_v, y_v)\}$. For each $t \geq 0$, let $\rho(E', t)$ indicate the highest berth location that the assigned vessels in $V(E')$ occupy at or after time t. We can see that $\rho(E', t)$ is a staircase function of at most $O(n)$ pieces. For each $u \in V \setminus V(E')$, we know from the branching rules that u can be positioned only at (x_u, y_u) with $y_u \geq \rho(E', x_u)$. By following the argument in Section 2, we can obtain that $[\sum_{u \in V(E')} w_u(x_u + t_u - a_u) + Z_{\mathbb{F}}(E')]$ is a lower bound on the remaining problem, where $Z_{\mathbb{F}}(E')$ is a revised model of $Z_{\mathbb{F}}$ that replaces V with $V \setminus V(E')$, and B with $B - \rho(E', t)$ in (4)–(9). Moreover, Algorithm 1 can be revised to solve $Z_{\mathbb{F}}(E')$ by replacing V with $V \setminus V(E')$, and using a revised Algorithm 2 in Step 2. The revised Algorithm 2 requires $b_{v0}, ..., b_{v,m_v}$, the endpoints of the pieces of $g_v(t)$ in Step 1, to include the endpoints of the pieces of $\rho(E', t)$, and requires the replacement of B with $B - \rho_j$ in the loop from Step 2 to Step 19, where ρ_j for each $1 \leq j \leq m_v$ indicates the constant value of $\rho(E', t)$ for $t \in [b_{v,j-1}, b_{vj})$. Since $\rho(E', t)$ is a staircase function of at most $O(n)$ pieces, the revised algorithm can still return the lower bound in $O(n^2)$ time. If the lower bound is greater than LB, then LB will be updated. If the lower bound is not smaller than UB, then the new node can be pruned so that it will not be explored further.

From the partial solution E of node p, we can extend it to obtain a feasible solution by a new heuristic method as follows. If the objective value of the feasible solution is smaller than UB, then UB will be updated. Our new heuristic for the CBAP adopts a best-fit strategy and follows a one-loop approach, as shown in Algorithm 3, where the best-fit unassigned vessel v^* from $V \setminus V(E)$ and its best-fit position $(x^*_{v^*}, y^*_{v^*})$ in the space-time diagram are dynamically selected, so as to gradually extend the partial solution E to a feasible solution. To save on computing time, we add the restriction that no unassigned vessel in $V \setminus V(E)$ can occupy the same quay space earlier than any assigned vessel. In other words, no unassigned vessel can be placed directly on the left of any assigned vessel in the space-time diagram. As a result, those assigned vessels define an "envelope" that separates the two areas where the unassigned vessels may or may not be placed. The envelope can be represented by $h(E, y)$, which indicates the departure time of the latest assigned vessel that occupies the berth location $y \in [0, B)$:

$$h(E, y) := \max\{x_v + t_v : \forall v \in V(E) \text{ with } y \in [y_v, y_v + l_v)\}, \text{ for } y \in [0, B). \tag{22}$$

For example, for the instance in Figure 4(a), we have $h(E, y) = 2$ for $y \in [0, 1)$ and $y \in [3, 5)$, $h(E, y) = 1$ for $y \in [1, 2)$, and $h(E, y) = 4$ for $y \in [2, 3)$. Next, we can use $S(E)$ to indicate the

Algorithm 3 Framework of the best-fit heuristic to extend a partial solution E

1: **for** $i = 1, 2, ..., n$ **do**
2: Among all unassigned vessels $v \in V \setminus V(E)$ and all valid positions $(x_v, y_v) \in S(E)$ of v, choose v^* and $(x^*_{v^*}, y^*_{v^*})$ that minimizes a certain fitness evaluation function.
3: Add the new assignment $(v^*, x^*_{v^*}, y^*_{v^*})$ to E.
4: **end for**
5: Return E.

area where unassigned vessels may be placed, which corresponds to the area on the right of $h(E, y)$ in the space-time diagram:

$$S(E) := \{(x, y) : x \geq h(E, y),\ 0 \leq y < B\}. \tag{23}$$

By minimizing a certain fitness evaluation function, the heuristic then selects the best-fit unassigned vessel v^* from $V \setminus V(E)$ and its best-fit position $(x^*_{v^*}, y^*_{v^*})$ from $S(E)$, and inserts assignment $(v^*, x^*_{v^*}, y^*_{v^*})$ to extend E. In our implementation of the heuristic, we have carefully designed the fitness evaluation function so that the heuristic has a polynomial time complexity, and that it can always produce optimal solutions for a special case of the CBAP with $l_v = B$ and $a_v = a$ for $v \in V$. See Online Appendix H for the implementation details of the new heuristic.

3.3. Dominance Rules

Using the branching rules mentioned earlier, the exact method may generate a decision node q that is dominated by a previously encountered node q', in the sense that the best solution that can be created by completing the partial solution of q has an objective value not smaller than the best solution that can be created by completing the partial solution of q'. In such a case, node q can be pruned so as to reduce the search space.

Consider each decedent node q of a current decision node p, where the partial solution E' of q extends the partial solution E of p by assigning (x_v, y_v) to $v \in V \setminus V(E)$. We propose the following rules to quickly identify dominance for node q.

Rule 1 (for better sequencing): Let $u \in V(E)$ indicate the vessel assigned just before v is assigned. Node q may be dominated by a node that assigns u and v in a different sequence. Consider the partial solution for vessels 1–5 shown in Figure 4(b), which first places vessel $u = 4$ at $(0, 4)$, and then vessel $v = 5$ at $(2, 3)$. However, one could also first place vessel 5 at $(1, 3)$ to start berthing earlier, and then place vessel 4 at $(0, 4)$, which leads to a better partial solution for vessels 1–5. To identify such dominance for node q, we check the condition that $y_v + l_v \leq y_u$. If the condition is satisfied, which implies that v is placed below u, then due to the branching rules, v must be placed on the right of u, i.e., $x_u + t_u \leq x_v$, and thus one can first place v at a position (x', y_v) with $x' \leq x_u + t_u$, before placing u at position (x_u, y_u). In this case, since x' is no later than x_v, node q is dominated and can be pruned.

It is noted that Guan and Cheung (2004) proposed to prune node q if $y_v + l_v \leq y_u$ and (x_v, y_v) is also an admissible position of v in node p, which is more restrictive than our condition above.

For example, their condition is not satisfied by the partial solution of vessels 1–5 in Figure 4(b) for $u = 4$ and $v = 5$. Proposition 3 below shows that our condition ensures that among nodes with the same partial solution, at most one will be explored to generate new nodes.

Proposition 3. *For any two nodes reached by the exact method, if they are associated with the same partial solution, then at least one of them satisfies $y_v + l_v \leq y_u$, and thus must be pruned, where v is an assigned vessel, and u is the vessel assigned just before v is assigned.*

Proof. See Online Appendix I. □

Rule 2 (for better placement of unassigned vessels): Consider any unassigned vessel $r \in V \setminus V(E')$ of node q. Let $x_r^{\min}$ indicate the earliest possible mooring time for r such that $t \geq a_r$ and $t \geq \min_{y \in [0, B-l_r]} z(E', r, y)$. By the branching rules, we know that in the future completion of E', r cannot be assigned any mooring time earlier than $x_r^{\min}$. Thus, we check whether or not there exists a position (x'_r, y'_r) with $a_r \leq x'_r < x_r^{\min}$, such that r can be placed at (x'_r, y'_r) without overlapping any assigned vessels in $V(E')$. If such a position (x'_r, y'_r) exists for r, then we can obtain a partial solution P by adding (r, x'_r, y'_r) to E. Accordingly, for every solution that can be created by completing E', by placing r at (x'_r, y'_r) it can be transformed to a solution that can be created by completing P, which, by $x'_r < x_r^{\min}$, has a smaller objective value. Thus, node q is dominated, and can be pruned. See Figure 4(b) with E' being the partial solution for vessels 1–3, and with $r = 7$, $x_7^{min} = 1$ and $(x'_7, y'_7) = (0, 0)$.

Rule 3 (for eliminating symmetry): For each feasible solution $P = \{(u, x_u, y_u) : u \in V\}$, its reflection symmetry about the quay, denoted by $\hat{P} = \{(u, x_u, B - y_u - l_u) : u \in V\}$, is also a feasible solution with the same objective value. To avoid reaching both P and $\hat{P}$, we introduce a new dominance relation as follows. Let $L(P) := \{r \in V : x_r < x_u + t_u, \forall u \in V \setminus \{r\}\}$ indicate a subset of vessels with mooring times no later than the departure times of all other vessels. We define the *bottom vessel* of $L(P)$ as the vessel $s \in L(P)$ of the lowest starting berth location y_s, and the *top vessel* of $L(P)$ as the vessel $\hat{s} \in L(P)$ of the highest starting berth location $y_{\hat{s}}$. It can be seen that $\hat{s}$ and s are unique for P. Since $L(P) = L(\hat{P})$, we can see that $\hat{s}$ and s are in turn the bottom and the top vessels of $L(\hat{P})$. Define that P is *dominated by symmetry* if $s > \hat{s}$. It can be seen that if P is dominated by symmetry, $\hat{P}$ is not dominated by symmetry.

Example 3. Consider the berth allocation P of vessels 1, 2, 3 and 4 in Figure 4(a). Its reflection symmetry $\hat{P}$ equals the berth allocation of vessels 1, 2, 3 and 4 shown in Figure 4(b). It can be seen that $L(P) = \{2, 3, 4\}$, and the bottom vessel and top vessel of $L(P)$ are 4 and 2, respectively, which are the top and the bottom vessels of $L(\hat{P})$, respectively. Thus, since $4 > 2$, P is dominated by symmetry, but $\hat{P}$ is not dominated by symmetry.

To reach only feasible solutions not dominated by symmetry, we check each new node q and its partial solution E'. Let $L(E') := \{r \in V(E') : x_r < x_u + t_u, \forall u \in V(E') \setminus \{r\}\}$. Let s denote the vessel in $L(E')$ that has the lowest starting berth location, and let $\hat{s}$ denote the vessel in $L(E')$ that has the highest starting berth location. For every feasible solution P that can be created by completing E', according to the branching rules, (i) s must be the bottom vessel of $L(P)$, if each $u \in V \setminus V(E')$ has either $x_s < a_u + t_u$ or $z(E', u, x_s - t_u) + l_u > B$, so that no unassigned vessel can depart when or before s starts berthing; and (ii) $\hat{s}$ must be the top vessel of $L(P)$, if each $u \in V \setminus V(E')$ has either $x_{\hat{s}} + t_{\hat{s}} \le a_u$ or $z(E', u, x) + l_u > B$ for all $x < x_{\hat{s}} + t_{\hat{s}}$, so that no unassigned vessel can start berthing before $\hat{s}$ departs. Thus, if both (i) and (ii) are satisfied, and $s > \hat{s}$, then P must be dominated by symmetry, and node q can be pruned.

Rule 4 (for better placement of vessels with equal sizes): Consider any assigned vessel $r \in V(E') \setminus \{v\}$ other than the newly assigned vessel v of node q, such that $l_r = l_v$, $y_r = y_v$ and $x_r + t_r = x_v$. If such r exists, then r and v occupy the same berth locations, and v starts berthing right after the departure of r. Thus, node q may be dominated by another node where r starts berthing right after the departure of v. See the partial solution for vessels 1–7 shown in Figure 4(b), where the total weighted turnaround time of vessel $r = 6$ and vessel $v = 7$ equals $2 \times 2 + 2 \times 7 = 18$, and it can be reduced to $2 \times 3 + 2 \times 5 = 16$ if vessel 6 is placed at $(5, 0)$ and vessel 7 is placed at $(4, 0)$. To identify such dominance, we check the condition that $w_r/t_r < w_v/t_v$ and $a_v \le x_r$. If the condition is satisfied, then one can change the mooring times of v and r to x_r and $x_r + t_v$, respectively, which, by $w_r/t_r < w_v/t_v$, leads to a better partial solution that costs less than E'. Thus, node q is dominated, and can be pruned.

Rule 5 (for better placement of vessels with equal sizes and handling times): Consider any unassigned vessel $r \in V \setminus V(E')$ and assigned vessel $u \in V(E')$ of node q with $l_r = l_u$ and $t_r = t_u$. Note that r and u have equal sizes and handling times. Thus, node q may be dominated by another node that assigns (x_u, y_u) to r instead of to u. To identify such dominance, we check the condition that $w_u < w_r$ and $a_r \le x_u < x_r^{\min}$, where $x_r^{\min}$ indicates the earliest possible mooring time for r, as defined in rule 2. If it is satisfied, one can replace (u, x_u, y_u) in E' with (r, x_u, y_u) to obtain another partial solution P. Every solution that can be created by completing E' can be transformed by swapping positions of v and r to a solution that can be created by completing P, which, by $w_u < w_r$ and $x_u < x_r^{\min}$, has a smaller objective value. Thus, node q is dominated, and can be pruned. See the partial solution for vessels 1–4 shown in Figure 4(b) with $r = 5$ and $u = 4$, where $w_4 < w_5$ and $a_5 = 0 = x_4 < 2 = x_5^{\min}$.

Remark 2. Rules 4 and 5 are effective only for instances that have some vessels with equal sizes or equal handling times. Such instances often appear when vessel sizes or handling times are measured by integral multiples of some constant factors, such as one hundred meters or one hour. In such a case, vessel sizes or handling times can be scaled down by the constant factors, and rounded to some small integers, which are likely to be equal for some vessels.

4. Extensions to the CBAP with a Quay Crane Constraint

As we mentioned in Section 1, vessel handling times are often estimated based on vessel cargo volume and pre-assigned quay cranes. For each vessel $v \in V$, let k_v indicate the number of quay cranes pre-assigned to v. Let K indicate the total number of available quay cranes. Thus, it is natural to impose a quay crane constraint on berth allocation, so that the total number of quay cranes used at the same time cannot exceed K. We refer to the CBAP with this additional quay crane constraint as the CBAPQ for short. The CBAPQ is more complicated than the CBAP, but solutions to the CBAPQ can be used to improve the quay crane assignment, which can in turn be used to improve the solutions to the CBAPQ. Such an iterative approach, known as "feedback-loop" integration, can be applied to jointly optimize the berth allocation and quay crane assignment (Bierwirth and Meisel 2010, 2015).

We are now going to show that our new lower bound and exact method for the CBAP can be extended to the CBAPQ. To extend the lower bound, we first formulate the CBAPQ as follows by using functions $f_v(t)$ defined in (1) for vessels $v \in V$, where each $f_v(t)$ represents the quay space occupied by vessel v at time t. From (1) we know that $(k_v/l_v)f_v(t)$ indicates the number of quay cranes used by vessel v at time t. Thus, due to the quay crane constraint, we have:

$$\sum_{v \in V} (k_v/l_v) f_v(t) \leq K, \ \forall t \in [0, \infty). \tag{24}$$

Adding (24) to model RF of the CBAP in Section 2.1, we can formulate the CBAPQ by the following model, denoted by Q:

$$\begin{aligned} \text{(Q)} \quad & \min \ W(f_1, ..., f_n) + \frac{1}{2} \sum_{v \in V} w_v t_v \\ & \text{s.t.} \ \ (1), (24), \text{ and } \{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}. \end{aligned}$$

Next, we introduce a parameter $\theta \in [0, 1]$, and for each value of θ, we will construct an instance of the CBAP with revised vessel sizes, so that the relaxation $Z_{\mathbb{F}}$ (defined in Section 2.2) of this CBAP instance is a relaxation of model Q of the CBAPQ. To revise vessel sizes, we let $\lambda := B/K$, and then transform the quay crane constraint (24) equivalently to (25) as follows:

$$\sum_{v \in V} \lambda (k_v/l_v) f_v(t) \leq B, \ \forall t \in [0, \infty). \tag{25}$$

Combining (25) with the quay space constraint (8), we have that $\sum_{v \in V} [\theta\lambda(k_v/l_v) + (1 - \theta)] f_v(t) \leq B, \ \forall t \in [0, \infty)$, from which, by letting $\gamma_v(\theta) := \theta\lambda(k_v/l_v) + (1 - \theta)$, we obtain a valid constraint (26) for model Q as follows:

$$\sum_{v \in V} \gamma_v(\theta) f_v(t) \leq B, \ \forall t \in [0, \infty). \tag{26}$$

We can now obtain an instance of the CBAP with vessel sizes revised to $\hat{l}_v(\theta) := \gamma_v(\theta) l_v = \theta\lambda k_v + (1 - \theta) l_v$ for $v \in V$, where other input parameters are the same as those of the CBAPQ.

For the instance of the CBAP obtained above with the revised vessel sizes $\hat{l}_v(\theta)$ for $v \in V$, we can see from (1) that $\gamma_v(\theta) f_v(t)$ equals the quay space occupied by each vessel v at time t, and thus, (26) is equivalent to the quay space constraint. Let $Z_{\mathbb{F}}(\theta)$ denote the relaxation $Z_{\mathbb{F}}$ (defined in Section 2.2) of this CBAP instance, where l_v is replaced with $\hat{l}_v(\theta)$ for each $v \in V$. We can establish Theorem 4 as follows to show that model $Z_{\mathbb{F}}(\theta)$ is a relaxation of model Q.

Theorem 4. *For each $\theta \in [0, 1]$, model $Z_{\mathbb{F}}(\theta)$ is a relaxation of model Q.*

Proof. See Online Appendix J. □

By Theorem 4, for each $\theta \in [0,1]$, $Z_{\mathbb{F}}(\theta)$ provides a lower bound on the optimal solution to the CBAPQ. Since $Z_{\mathbb{F}}(\theta)$ equals $Z_{\mathbb{F}}$ of the CBAP instance with the revised vessel sizes $\hat{l}_v(\theta)$ for $v \in V$, we can directly use the algorithm in Section 2.3 to compute $Z_{\mathbb{F}}(\theta)$ in $O(n^2)$ time.

Moreover, with different values of θ, the lower bound $Z_{\mathbb{F}}(\theta)$ may have different values. Thus, given a subset $\Theta \subseteq [0,1]$ of values for θ, among all $Z_{\mathbb{F}}(\theta)$ with $\theta \in \Theta$ we can choose the maximum as the best lower bound for the CBAPQ, denoted by $Z_{\mathbb{F}}(\Theta)$ as follows:

$$Z_{\mathbb{F}}(\Theta) := \max_{\theta \in \Theta} Z_{\mathbb{F}}(\theta). \tag{27}$$

See Online Appendix K for an illustrative example of $Z_{\mathbb{F}}(\Theta)$.

With $Z_{\mathbb{F}}(\Theta)$ defined above, we can extend the new exact method of the CBAP to solve the CBAPQ. The extension is direct, since we can still follow the branch-and-bound approach, use the same mixed-strategy to select nodes to be branched, and extend $Z_{\mathbb{F}}(\Theta)$ to obtain lower bounds for new nodes generated. By incorporating the newly introduced quay crane constraint, we can also extend the admissible positions for the branching rules, and extend the best-fit heuristic in Algorithm 3 for computing upper bounds. The dominance rules derived for the CBAP can also be applied or extended. See Online Appendix L for details.

Nevertheless, in some situations the number of quay cranes assigned to each vessel can be changed from time to time during the handling period. This is known as a "variable-in-time" quay crane assignment, which complicates the berth allocation (Bierwirth and Meisel 2010, Giallombardo et al. 2010). However, as shown in Online Appendix M, the above new lower bound and exact method for the CBAPQ can be extended to this more complicated case.

5. Experiments and Results

This section reports on the results of the experiments that were conducted to test how our proposed new exact method and lower bound perform in comparison to published methods and bounds. All the algorithms were coded in C++ and run on a PC with 1.0GB of RAM and an Intel Xeon CPU at 2.80GHz. The test data are based on those from the literature, with the number of vessels ranging from 5 to 160, which, according to a container terminal operator in Hong Kong, covers the needs for various berth plans, such as daily or weekly plans, or for those ports with different quay lengths or traffic density. More details of the computational results can be found at `http://www.mypolyuweb.hk/~lgtzx/cbap/result.htm`.

5.1. Comparing Exact Methods

For the new exact method proposed in Section 3, we want to compare its performance with conventional methods that use optimization solvers to solve integer programming models IP_1 and IP_2 (described in Online Appendix A) (Lee et al. 2010, Guan and Cheung 2004, Park and Kim 2002, Umang et al. 2013). We adopted ILOG CPLEX 12.2 as the optimization solver.

For the experiment, we used 308 instances of test data from the following sixteen data sets in three different classes, where the number of vessels ranges from 5 to 24. In Class A, two of the data sets, denoted by a05 and a10, are from Lee et al. (2010), where each set contains 30

Table 1 Test data statistics for data sets of Class A, Class B, and Class C.

Class	B	$\overline{l}$	σ_l	$\overline{t}$	σ_t	$\overline{a}$	σ_a	$B/\overline{l}$	$\sigma_l/\overline{l}$	$\sigma_t/\overline{t}$	$\sigma_a/\overline{t}$
A	80	28.0	12.7	50.0	0.4	10.0	10.0	2.9	**0.5**	0.2	**0.1**
B	10	5.5	2.6	9.0	0.2	0.0	0.0	1.8	0.5	0.3	0.0
C	600	240.1	41.8	31.2	0.2	78.3	78.3	2.5	0.2	0.2	1.6

instances, with the number of vessels $n \in \{5, 10\}$ and the quay length $B = 80$. In addition, we followed the same approach as Lee et al. (2010) in generating another two data sets for Class A, denoted by a12 and a15, having 10 instances in each, with larger $n \in \{12, 15\}$ and the same $B = 80$. To include instances with smaller B, we followed an approach of Guan and Cheung (2004) to generate another three data sets for Class B, denoted by b10, b15 and b20, where each set contains 10 instances, with $n \in \{10, 15, 20\}$ and $B = 10$. For Class C, we used another data set, denoted by c, of 44 instances with $11 \leq n \leq 24$ and $B = 600$, which is based on actual operational data at the Pusan Port from Park and Kim (2002). For all eight data sets above, vessel importance weights w_v are the same for all $v \in V$. To include instances with different vessel importance weights, we changed each w_v randomly to a value in $\{5, 6, 7, 8, 9, 10\}$, and obtained another eight data sets, denoted by a05w, a10w, a12w, a15w, b10w, b15w, b20w and cw, for Class A, Class B, and Class C, respectively.

Table 1 summarizes the test data statistics for the three classes of data sets, where the mean and the variance of vessel sizes, handling times, and arrival times are indicated by $\overline{l}$, σ_l, $\overline{t}$, σ_t, $\overline{a}$, and σ_a, respectively. Since, in different classes of data sets, units of measurement for lengths and times can be different, we use $B/\overline{l}$ to measure how large the quay is with respect to the average vessel size, and use $\sigma_a/\overline{t}$ to measure how concentrated vessel arrivals are with respect to the vessel average handling time. Smaller values of $B/\overline{l}$ and $\sigma_a/\overline{t}$ indicate that the port is more crowded. Moreover, we use $\sigma_l/\overline{l}$ and $\sigma_t/\overline{t}$ to indicate relative deviations of vessel sizes and handling times from their means. Larger values of $\sigma_l/\overline{l}$ and $\sigma_t/\overline{t}$ imply that vessels are more heterogeneous. Table 1 shows that, compared with the data in Class C, test data in Class A and Class B have significantly smaller $\sigma_a/\overline{t}$ and larger $\sigma_l/\overline{l}$, implying that they are more crowded, and are comprised of more heterogeneous vessels.

Table 2 compares the results produced by the new exact method (New BB) and by CPLEX on IP_1 and IP_2. For each data set, we show in column #Inst the total number of instances. For each method, we report in column S the number of instances solved to optimality, and in column $G\%$ the average percentage gap between the best lower bound lb and the best feasible solution ub found by the method, i.e., $G\% = (\text{ub} - \text{lb})/\text{ub}$. Columns T_0, T_1 and T_2 report the average computing times in seconds for test data solved to optimality by the new exact method, by CPLEX on IP_1, and by CPLEX on IP_2, respectively. The time limit was set to be 3600 seconds. The symbol '-' indicates that no feasible solutions were found within the time limit.

Table 2 shows that *the new exact method solves all 308 instances to optimality within the time limit, significantly outperforming CPLEX.* For test data in Class A and Class B with 15 vessels or more, CPLEX on IP_1 solves only one to optimality, and its average percentage gap exceeds 34%. For test data with $B \in \{80, 600\}$, CPLEX on IP_2 always runs out of memory before finding any feasible solutions, because there are too many variables and constraints in

Table 2 Comparison of the new exact method (New BB) with CPLEX on IP_1 and IP_2.

Set	n	B	#inst	New BB				CPLEX on IP_1			CPLEX on IP_2		
				T_0	T_1	T_2	S	T_1	$G\%$	S	T_2	$G\%$	S
a5	5	80	30	0	**0**	-	**30**	0	0.0	30	-	-	0
a5w	5	80	30	0	**0**	-	**30**	0	0.0	30	-	-	0
a10	10	80	30	2	**2**	-	**30**	171	0.1	29	-	-	0
a10w	10	80	30	2	**2**	-	**30**	186	0.0	30	-	-	0
a12	12	80	10	32	**21**	-	**10**	2003	7.2	3	-	-	0
a12w	12	80	10	38	**25**	-	**10**	1214	5.6	4	-	-	0
a15	15	80	10	655	-	-	**10**	-	32.3	0	-	-	0
a15w	15	80	10	522	-	-	**10**	-	28.8	0	-	-	0
b10	10	10	10	0	**0**	**0**	**10**	18	0.0	10	3	0.0	10
b10w	10	10	10	0	**0**	**0**	**10**	8	0.0	10	3	0.0	10
b15	15	10	10	39	-	**7**	**10**	-	19.3	0	157	0.3	8
b15w	15	10	10	32	**2**	**6**	**10**	2318	18.0	1	203	0.4	8
b20	20	10	10	521	-	**446**	**10**	-	51.3	0	1289	0.1	9
b20w	20	10	10	462	-	**376**	**10**	-	48.4	0	432	0.2	8
c	11–24	600	44	6	**3**	-	**44**	27	0.1	43	-	-	0
cw	11–24	600	44	5	**5**	-	**44**	97	0.0	44	-	-	0
Total			308				**308**			234			53

Table 3 Comparisons among variants of the new exact method with alternative settings.

Set	#inst	Not using the new lower bound			Not using the new heuristic			Using depth-first strategy			Not using any dominance rules		
		S	ΔT	ΔN	S	ΔT	ΔN	S	ΔT	ΔN	S	ΔT	ΔN
a5	30	30	0.0	0.5	30	6.8	62.9	30	0.2	0.3	30	-0.1	0.3
a5w	30	30	-0.2	0.3	30	3.8	61.7	30	0.0	0.1	30	-0.1	0.3
a10	30	30	3.2	8.6	30	17.9	151.4	30	2.3	7.0	30	1.7	2.1
a10w	30	30	2.9	7.6	30	18.0	152.2	30	2.2	6.3	30	1.9	2.0
a12	10	10	10.3	27.7	10	1.0	11.4	10	11.0	28.9	10	2.8	2.5
a12w	10	10	8.2	28.1	10	1.2	20.3	10	7.6	27.4	10	2.5	2.9
a15	10	0	-	-	9	0.4	1.7	0	-	-	5	4.4	3.8
a15w	10	0	-	-	10	0.2	1.3	0	-	-	7	4.7	4.4
b10	10	10	6.5	236.9	10	689.2	$1 \cdot 10^5$	10	6.1	262.2	10	3.0	4.0
b10w	10	10	7.9	148.0	10	$1 \cdot 10^3$	$1 \cdot 10^5$	10	6.4	147.2	10	3.9	3.4
b15	10	9	178.2	330.6	10	2.5	43.3	9	141.4	602.8	8	32.4	45.6
b15w	10	9	191.7	805.4	10	9.0	85.9	9	154.1	528.1	9	23.9	22.1
b20	10	0	-	-	9	1.2	4.5	0	-	-	1	14.4	20.7
b20w	10	0	-	-	3	0.3	2.8	0	-	-	1	3.9	3.1
c	44	42	38.6	77.8	21	$4 \cdot 10^4$	$3 \cdot 10^5$	42	116.5	281.2	38	$2 \cdot 10^3$	$4 \cdot 10^3$
cw	44	42	39.4	54.8	21	$7 \cdot 10^4$	$3 \cdot 10^5$	41	94.4	146.8	38	$2 \cdot 10^3$	$3 \cdot 10^3$
Total	308	262			253			261			267		

IP_2. Moreover, as shown in columns T_1 and T_2 of Table 2, *the average computing time of the new exact method is much shorter than CPLEX*, more than 94% less for most data sets.

Table 2 also shows that test data in Class A are more difficult to solve than test data in Class B and Class C, since no instances with $n \geq 15$ in Class A can be solved to optimality by CPLEX. This may be because test data in Class A have a larger value of B than those in Class B, and as shown earlier, they are also more crowded and are comprised of more heterogeneous vessels than those in Class C.

Moreover, to prove computationally the effectiveness of the enhancements introduced for our new exact method, namely the new lower bound, heuristic, mixed node selection strategy, and dominance rules, we have further tested and compared four more variants of the new method.

Each variant excludes one of these enhancements but includes the other three. For the variant not using the mixed node selection strategy, it adopts the depth-first strategy, since the breadth-first strategy has an obvious weakness due to its extensive memory consumption. For each variant, column S in Table 3 reports the number of instances solved to optimality. Column ΔT reports the relative increase in computing time, expressed as a ratio against that of the new method with all the enhancements. Column ΔN reports the relative increase in the number of explored nodes, expressed as a ratio against that of the new method with all the enhancements. The results show that, except for test data in a5 and a5w where $n = 5$, the four variants all spent significantly longer time and explored many more nodes than the new method with all the enhancements. Therefore, *using all the enhancements together is effective in reducing both the computing time and the number of explored nodes.*

5.2. Comparing the New Exact Method with Heuristic Methods

For instances with a large number of vessels, although their optimal solutions can barely be guaranteed, the new exact method can still produce the best found feasible solutions, together with the best found lower bounds, within a certain time limit, which was set to be 3600 seconds. We have run the new exact method on 240 instances of test data from the following eight data sets. Among them, four data sets, denoted by a40, a80, a120, and a160, are from Lee et al. (2010), where each set contains 30 instances, with $n \in \{40, 80, 120, 160\}$ and $B = 80$. For these instances, the vessel importance weights w_v are the same for all $v \in V$. To include instances with different vessel importance weights, we changed each w_v randomly to a value in $\{5, 6, 7, 8, 9, 10\}$, and obtained another four data sets, denoted by a40w, a80w, a120w, and a160w. All these data sets belong to Class A, whose setting, as shown in Section 5.1, leads to more difficult instances than Class B and Class C, and therefore can allow for a more exhaustive performance test.

We want to compare the performance of the new exact method with those best known heuristic methods of the CBAP, including the squeaky wheel optimization (SWO) (Umang et al. 2013), the greedy randomized adaptive search procedure (GRASP) (Lee et al. 2010), and the stochastic beam search algorithm (SBS) (Wang and Lim 2007). For SWO, we have implemented it according to Umang et al. (2013), and set the time limit to 3600 seconds. For GRASP and SBS, the test results are from Lee et al. (2010) and are only available for data sets a40, a80, a120, and a160.

Table 4 compares the results of the new exact method and various heuristic methods. For each method and each data set, we report in column $G^{\mathrm{ub}}_{\mathrm{lb}^*}\%$ the average percentage gap between the objective value of its solution ub and the best known lower bound lb*, i.e., $G^{\mathrm{ub}}_{\mathrm{lb}^*}\% = (\mathrm{ub} - \mathrm{lb}^*)/\mathrm{lb}^*$. For GRASP and SBS, their average computing times in seconds are reported in column T according to Lee et al. (2010), based on experiments on a PC with a CPU at 3.00 GHZ, about 1.1 times faster than our PC. For the new BB and SWO, their computing times are always equal to the time limit of 3600 seconds.

According to Table 4, *the new exact method always produces significantly better solutions within the 3600 second time limit than other heuristics from the literature, reducing the average*

Table 4 Comparison of the new exact method with various heuristic methods.

Set	n	B	#inst	New BB		SWO		GRASP		SBS	
				$G^{\text{ub}}_{\text{lb}^*}\%$	T	$G^{\text{ub}}_{\text{lb}^*}\%$	T	$G^{\text{ub}}_{\text{lb}^*}\%$	T	$G^{\text{ub}}_{\text{lb}^*}\%$	T
a40	40	80	30	**11.1**	3600	20.9	3600	42.1	37	45.3	188
a40w	40	80	30	**10.8**	3600	24.7	3600	-	-	-	-
a80	80	80	30	**10.8**	3600	27.1	3600	44.5	271	54.2	790
a80w	80	80	30	**10.1**	3600	30.3	3600	-	-	-	-
a120	12	80	30	**10.2**	3600	29.3	3600	44.0	882	54.7	911
a120w	12	80	30	**9.8**	3600	32.1	3600	-	-	-	-
a160	160	80	30	**9.7**	3600	30.8	3600	44.3	2137	56.7	1703
a160w	160	80	30	**9.2**	3600	33.8	3600	-	-	-	-

percentage gap from 20–57% to only 9–11%, with the average improvement being more than 16%. Among the heuristics taken from the literature, solutions produced by SWO have the best quality. However, they still deviate significantly from the best lower bounds, by at least 20% on average, even for instances with only 40 vessels. This is consistent with the findings of Umang et al. (2013), who showed that solutions of SWO had an average percentage gap of about 20% for their test data of 40 vessels and $B = 30$. This also confirms that our implementation of SWO is comparable to the one in Umang et al. (2013).

5.3. Comparing Lower Bounds

For the new lower bound $Z_{\mathbb{F}}$ proposed in Section 2, we want to compare it with the lower bounds from the linear programming relaxations of IP_1 and IP_2, denoted by Z_1^{LP} and Z_2^{LP}. For the experiment, we have used all the 608 instances of test data introduced in Sections 5.1 and 5.2.

By (10), computing $Z_1^{\text{LP}} = \sum_{v\in V} w_v t_v$ is easy. However, computing Z_2^{LP} is difficult, particularly for instances with large values of the number of vessels n, quay length B, and length of the planning horizon H. We adopted ILOG CPLEX 12.2 to solve linear programming models. When computing Z_2^{LP}, except for instances in Class B, CPLEX always runs out of memory. Therefore, we have instead followed a scaling approach, proposed in Dai et al. (2008), to compute a lower bound Z_2^{SLP}, by first scaling down vessel sizes, arrival times and handling times, as well as values of B and H, by certain constant factors, then rounding them to integers and solving the linear programming relaxation of IP_2 for the scaled instance. For instances in Class B, since n, B and H are small, we set the scaling factors to 1, implying $Z_2^{\text{SLP}} = Z_2^{\text{LP}}$. For other instances, we set the scaling factors to greater than 1, so that the values of H and B after scaling do not exceed 80 (if $n \leq 20$) or 40 (if $n > 20$), to avoid CPLEX running out of memory.

Table 5 compares the lower bounds $Z_{\mathbb{F}}$, Z_1^{LP} and Z_2^{SLP}, as well as the best lower bounds found by CPLEX on IP_1 within 3600 seconds. For each lower bound lb, we report in column T its average computing time (in seconds), and in column $G^{\text{lb}}_{\text{ub}^*}$ its average percentage gap from the best known upper bound ub^*, i.e., $G^{\text{lb}}_{\text{ub}^*}\% = (\text{ub}^* - \text{lb})/\text{ub}^*$. It can be seen that $Z_{\mathbb{F}}$ *is close to the best known upper bound of the optimal solution*, with an average percentage gap of about 10%. For those median and large instances with $n \geq 40$, when comparing with Z_1^{LP}, CPLEX on IP_1, and Z_2^{SLP}, $Z_{\mathbb{F}}$ *significantly improves the lower bounds by about 90%, 88% and 47%, respectively.*

Table 5 also shows that *the computation of $Z_{\mathbb{F}}$ is very fast, as it takes less than a second even for large instances in a160 and a160w.* Moreover, although other lower bounds, such as Z_2^{SLP}

Table 5 Comparison of the new lower bound $Z_{\mathbb{F}}$ with the lower bounds from methods in the literature.

Sets	n	B	$Z_{\mathbb{F}}$		Z_1^{LP}		CPLEX on IP_1		Z_2^{SLP}	
			$G_{\text{ub}^*}^{\text{lb}}\%$	T	$G_{\text{ub}^*}^{\text{lb}}\%$	T	$G_{\text{ub}^*}^{\text{lb}}\%$	T	$G_{\text{ub}^*}^{\text{lb}}\%$	T
a5,a5w	5	80	12.2	**0**	24.0	0	0.0	0	9.7	185
a10,a10w	10	80	13.2	**0**	46.0	0	0.0	236	13.3	163
a12,a12w	12	80	11.6	**0**	51.5	0	6.4	2884	12.0	167
a15,a15w	15	80	**10.9**	**0**	58.9	0	29.4	3600	13.8	141
b10,b10w	10	10	10.4	**0**	65.6	0	0.0	13	3.0	0
b15,b15w	15	10	8.8	**0**	70.9	0	18.4	3536	4.3	0
b20,b20w	20	10	7.0	**0**	78.0	0	48.9	3600	2.2	1
c,cw	11–24	600	8.9	**0**	13.1	0	0.0	103	30.8	35
a40,a40w	40	80	**10.0**	**0**	82.1	0	76.5	3600	25.6	149
a80,a80w	80	80	**9.5**	**0**	90.5	0	88.0	3600	40.5	134
a120,a120w	120	80	**9.1**	**0**	93.6	0	92.2	3600	60.4	127
a160,160w	160	80	**8.6**	**0**	95.1	0	94.3	3600	75.1	71

and Z_2^{LP}, can produce better lower bounds for small instances, it can still be more effective to use $Z_{\mathbb{F}}$ to compute lower bounds, particularly in branch and bound algorithms, when the saving in computing time can more than offset the slight decrease of the lower bounds. This is exactly the case that can be observed from the test of the exact methods on instances in Class B, where our new exact method computes lower bounds based on $Z_{\mathbb{F}}$, and it performs significantly better than CPLEX on IP_2, which computes lower bounds based on Z_2^{LP}.

5.4. Sensitivity Tests for Solutions Produced by the New Method

In the CBAP, all input parameters are assumed to be known. However, in practice, due to uncertainties, one may not know actual values of some parameters in advance, such as the vessel importance weights w_v (which often depend on vessel delays and cargo values), vessel arrival times a_v, and handling times t_v. We therefore want to test the sensitivity of solution qualities for the new exact method towards various deviations of these parameter values.

We conducted sensitivity tests for w_v, and for a_v and t_v, separately. In each test, we generated new test data randomly from existing ones that have different vessel importance weights w_v, by introducing random deviations to the values of the parameters. We use $\delta\%$ to control the extent of deviations, varying from 5%, 10%, ..., to 30%, so that the maximum deviation of w_v equals $\delta\%$ of the original value of w_v, and the maximum deviations of a_v and t_v both equal $\delta\%$ of the original value of t_v. For each new instance I' generated from an existing instance I, we applied the new exact method with a 3600 second time limit to obtain the best found feasible solution $S_{I'}$. We use ub to indicate the objective value of solution $S_{I'}$ for instance I, and use ub^* to denote the best known upper bound for I obtained in the experiments in Section 5.1 and Section 5.2. It can be seen that solution $S_{I'}$ corresponds to a berth allocation made before actual parameter values are known, that ub is the actual objective value that $S_{I'}$ achieves for the instance of actual parameter values, and that ub^* is the best objective value that we can obtain when all parameter values are known in advance. Therefore, we can measure the quality of $S_{I'}$ with respect to I by the percentage gap $G_{\text{ub}^*}^{\text{ub}}\% = (\text{ub} - \text{ub}^*)/\text{ub}^*$.

For each test data set and each deviation $\delta\%$, we report the average percentage gaps $G_{\text{ub}^*}^{\text{ub}}\%$ in Tables 6(a) and 6(b). Table 6(a) shows that *the qualities of solutions obtained by the new*

Table 6 Comparison of solutions produced by the new exact method for instances with various deviations of (a) vessel importance weights w_v, or (b) vessel arrival times a_v and handling times t_v.

(a) $G^{\text{ub}}_{\text{ub}*}\%$ for instances with $\delta\%$ deviations of w_v.

δ:	5	10	15	20	25	30
a5w	5.9	6.1	6.0	6.1	6.3	6.2
a10w	1.6	1.7	1.8	1.9	2.1	2.6
a12w	1.3	1.4	1.4	2.1	2.6	1.9
a15w	0.1	0.1	0.3	0.2	0.7	2.3
b10w	0.0	0.1	0.5	0.7	0.9	1.1
b15w	2.5	3.0	2.8	3.1	3.6	3.9
b20w	1.3	1.4	1.6	1.9	1.9	2.8
cw	0.5	0.5	0.5	0.7	0.8	0.8
Average	1.7	1.8	1.9	2.1	2.4	**2.7**
a40w	0.5	0.7	0.8	0.8	1.4	1.7
a80w	0.5	0.6	0.6	0.8	1.2	1.4
a120w	0.3	0.1	0.4	0.6	0.9	1.2
a160w	0.2	0.3	0.4	0.7	1.0	1.5
Average	0.4	0.4	0.6	0.7	1.1	**1.5**

(b) $G^{\text{ub}}_{\text{ub}*}\%$ for instances with $\delta\%$ deviations of a_v and t_v.

δ:	5	10	15	20	25	30
a5w	0.1	0.4	1.0	1.5	1.4	2.2
a10w	0.3	0.7	1.3	2.6	2.7	5.3
a12w	0.4	0.7	1.7	2.8	2.7	6.1
a15w	0.5	1.2	1.6	5.6	4.5	6.8
b10w	0.0	0.9	0.7	1.8	2.1	5.6
b15w	0.2	1.6	2.1	2.9	4.1	4.6
b20w	0.2	1.5	2.0	2.3	2.8	5.7
cw	0.3	0.8	0.9	2.4	2.6	4.4
Average	0.3	1.0	**1.4**	2.7	2.9	5.1
a40w	0.8	1.9	2.7	3.4	5.2	6.6
a80w	0.8	1.6	2.4	4.1	6.2	7.6
a120w	0.6	1.8	2.8	3.9	5.7	7.5
a160w	0.8	1.9	2.8	4.2	5.9	**8.1**
Average	0.8	1.8	**2.7**	3.9	5.7	7.4

exact method are robust even in scenarios where there are some large deviations of w_v, with an average percentage gap of only about 2% from the best known upper bounds for instances with up to 30% deviations in w_v. Table 6(b) shows that *the qualities of solutions obtained by the new exact method are robust even in scenarios where there are some mild deviations of a_v and t_v*, with an average percentage gap of about 2% from the best known upper bounds for instances with up to 15% deviations in a_v and t_v. However, solutions obtained by the new exact method can have an average percentage gap of about 8% when deviations $\delta\%$ reach 30% in a_v and t_v. Although such solution may leave room for improvement, they are satisfactory, particularly for large sized instances, since from the experiments in Section 5.2 we know that, even when actual parameter values are known in advance, the meta-heuristic SWO can only produce solutions with an average percentage gap of more than 16% from the best known upper bounds.

5.5. Testing the Extensions to the CBAPQ

For the extensions of our new lower bound and exact method to the CBAPQ, proposed in Section 4, we want to compare their performance with conventional bounds and methods. To generate test data for the CBAPQ, we extended the data for the CBAP with different w_v, by including the numbers of pre-assigned quay cranes k_v for $v \in V$, where each k_v was randomly drawn from $\{1, 2, 3, 4, 5, 6\}$, and the total number of available quay cranes $K = 10$. For all solution methods to be tested for the CBAPQ, we set their time limits to 3600 seconds.

We conducted experiments on test data with $5 \le n \le 24$ for the CBAPQ, to compare the extension of the new exact method (Extended New BB) with using CPLEX on integer programming models IPQ_1 and IPQ_2 of the CBAPQ. See Online Appendix N for details of the two models. Similar to Table 2, the results shown in Online Appendix O indicate that *the extension of the new exact method significantly outperforms CPLEX for solving more and larger instances to optimality in much less computing time.* Among all the 154 instances, it solves 150 instances to optimality, and for the other four instances, it produces solutions with an average percentage gap less than only 8% from the best known lower bound.

We also conducted experiments on test data with $40 \leq n \leq 160$ for the CBAPQ, to compare the extension of the new exact method with an extension of the meta-heuristic SWO (Extended SWO) for the CBAPQ, and to compare the extension of the new lower bound $Z_{\mathbb{F}}(\Theta)$, where $\Theta = \{0.0, 0.01, ..., 1.0\}$, with a lower bound Z_1^{LPQ} derived from the linear programming relaxation of model IPQ_1. Similar to Table 4, the results shown in Online Appendix O indicate that *the extension of the new exact method always produces near optimal solutions for large sized test data, with an average percentage gap less than* 9%, *outperforming the extension of the SWO.* Similar to Table 5, the results shown in Online Appendix O also show that *the extension of the new lower bound* $Z_{\mathbb{F}}(\Theta)$ *outperforms* Z_1^{LPQ}*, not only for being much closer to the best known upper bound, but also for requiring much less computing time.*

6. Conclusions

In this paper, we have proposed a new lower bound and a new exact method for the CBAP. For the new lower bound, we have derived a novel relaxation of the problem, which we can efficiently solve in quadratic time. For the new exact method, we have used our new lower bound, together with a new heuristic and some new dominance rules. Computational results have shown that the new lower bound and exact method significantly outperform existing bounds and methods. We have also extended our new lower bound and exact method to a more complicated problem CBAPQ, where decisions on berth allocations are restricted by a quay crane constraint. Its solutions are of significant practical value for integrating with the quay crane assignment.

Due to uncertainties in certain input parameters, we have also conducted extensive computational experiments to test the sensitivity of solution qualities for our new exact method of the CBAP. The results show that they are robust even in scenarios with some large deviations in vessel importance weights, as well as in scenarios with some mild deviations in vessel arrival and handling times. For problems with high uncertainties in vessel arrival and handling times, one can often formulate them into stochastic programming models or robust optimization models. As demonstrated in Zhen (2015), Shang et al. (2016), several bounds and solution methods for these models closely rely on solutions to certain deterministic berth allocation problems. Therefore, our new solution methods for the CBAP can also be applied or extended to these models. Moreover, in a dynamic setting, where values of input parameters are updated from time to time, we can follow a rolling-horizon approach (Cordeau et al. 2005) to re-allocate berths whenever values of certain input parameters are changed. Although such berth re-allocation problems may include additional objectives to minimize recovery costs for the changes from the original berth allocation (Zeng et al. 2011), their decisions and constraints are similar to those of the CBAP. Therefore, extending our new solution methods for the CBAP to them also shows great promise.

This paper has opened up several directions for future research. First, although in the research for this paper we have reduced the gap between the best known lower bound and heuristic solution, there is still room for improvement. Such improvement can be achieved by further enhancing the lower bounds. In future studies, we will follow this direction to strengthen our lower bound by including more valid constraints into the relaxations, which may involve solving even more complex optimization models on vectors of functions.

Second, this paper mainly focuses on new solution methods for the basic berth allocation problem. As we have discussed earlier, these new methods can be useful in solving some more complicated problems, including those integrating the berth allocation with quay crane assignment, and those with uncertainties in input parameters. Apart from these, there are other issues that have not been considered in this study, but that sometimes arise in real-world applications, such as the total cost of tardiness of the vessels (Meisel and Bierwirth 2009), the total cost of transporting cargo from berths to the yard (Park and Kim 2003), the hybrid layout of berths (Moorthy and Teo 2006), and the joint optimization of berth allocation with yard planning (Zhen et al. 2011). Since the CBAP is often a special case of these more complicated problems, the new lower bound proposed in this paper is often valid for them as well. However, it needs to be further strengthened, so as to well approximate their optimal solutions. To achieve this, we may extend our reformulation of the CBAP to reformulate these more complicated problems into optimization models on vectors of functions, and following this we need to carefully relax these models, so that the relaxations can be not only computationally tractable, but also effective in providing good lower bounds on the optimal solutions (see Online Appendix P for detailed illustrations.). This task is challenging but of significant research value, and we therefore leave it to our future study. Moreover, for these more complicated problems, the new exact method proposed in this paper can also be extended, but future research is required to develop valid dominance rules for pruning, as well as effective heuristics for computing upper bounds.

Third, the CBAP can be formulated as a rectangle packing problem that aims to minimize the total packing cost. This is different from the classic rectangle packing problems, which mainly aim to optimize min-max objectives (Hopper and Turton 2001). Therefore, by changing the objective function, the new relaxation model proposed in this paper for the CBAP can be revised to derive new lower bounds for such classic rectangle packing problems. It will be very interesting to investigate how to compute these lower bounds, and how good they actually are.

Acknowledgments

The authors are grateful to the area editor, associate editor, and two reviewers for helpful comments. This research was supported by grants from the Research Grants Council of the HKSAR, China, T32-620/11 and 152186, and the Hong Kong Polytechnic University, 1-ZVDS.

References

A. Ak and A. L. Erera. A nested tabu search algorithm for the berth allocation problem, 2011. Working paper.

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

C. Bierwirth and F. Meisel. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3):675–689, 2015.

M.A. Boschetti and L. Montaletti. An exact algorithm for the two-dimensional strip-packing problem. *Operations Research*, 58(6):1774–1791, 2010.

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

E.K. Burke, G. Kendall, and G. Whitwell. A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52(4):655–671, 2004.

C.Y. Cheong, K.C. Tan, D.K. Liu, and C.J. Lin. Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180(1):63–103, 2010.

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

J.-F. Côté, M. DellAmico, and M. Iori. Combinatorial Benders' cuts for the strip packing problem. *Operations Research*, 62(3):643–661, 2014.

J. Dai, W. Lin, R. Moorthy, and C.P. Teo. Berth allocation planning optimization in container terminals. In C. Tang, C.P. Teo, and K.K. Wei, editors, *Supply Chain Analysis*, volume 119 of *International Series in Operations Research & Management Science*, pages 69–104. Springer US, 2008.

R. M. de Oliveira, G. R. Mauri, and Nogueira L.A.N. Clustering search for the berth allocation problem. *Expert Systems with Applications*, 39(5):5499–5505, 2012.

G. Giallombardo, L. Moccia, M. Salani, and I. Vacca. Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological*, 44(2):232–245, 2010.

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

E. Hopper and B.C.H. Turton. A review of the application of meta-heuristic algorithms to 2D strip packing problems. *Artificial Intelligence Review*, 16(4):257–300, 2001.

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

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

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

F. Li, J.-B. Sheu, and Z.-Y. Gao. Solving the continuous berth allocation and specific quay crane assignment problems with quay crane coverage range. *Transportation Science*, 49(4):968–989, 2015.

A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: a survey. *European Journal of Operational Research*, 141(2):241–252, 2002.

S. Martello and D. Vigo. Exact solution of the two-dimensional finite bin packing problem. *Management Science*, 44(3):388–399, 1998.

S. Martello, D. Pisinger, and D. Vigo. The three-dimensional bin packing problem. *Operations Research*, 48(2):256–267, 2000.

G.R. Mauri, G.M. Ribeiro, L.A.N. Lorena, and G. Laporte. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. *Computers & Operations Research*, 70: 140–154, 2016.

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

R.G. Michael and S.J. David. Computers and intractability: a guide to the theory of NP-completeness. *W. H. Freeman and Company, San Francisco*, 1979.

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

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

T.E. Notteboom. The time factor in liner shipping services. *Maritime Economics & Logistics*, 8(1): 19–39, 2006.

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

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

M. Pinedo. *Scheduling: theory, algorithms, and systems*. Springer, 2012.

D. Pisinger and M. Sigurd. Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem. *INFORMS Journal on Computing*, 19(1):36–51, 2007.

X.T. Shang, J.X. Cao, and J. Ren. A robust optimization approach to the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 94:44–65, 2016.

N. Umang, M. Bierlaire, and I. Vacca. Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research Part E: Logistics & Transportation Review*, 54:14–31, 2013.

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

Q. Zeng, Z. Yang, and X. Hu. Disruption recovery model for berth and quay crane scheduling in container terminals. *Engineering Optimization*, 43(9):967–983, 2011.

D. Zhang, L. Wei, S.C.H. Leung, and Q. Chen. A binary search heuristic algorithm based on randomized local search for the rectangular strip-packing problem. *INFORMS Journal on Computing*, 25(2): 332–345, 2013.

L. Zhen. Tactical berth allocation under uncertainty. *European Journal of Operational Research*, 247(3): 928–944, 2015.

L. Zhen, E.P. Chew, and L.H. Lee. An integrated model for berth template and yard template planning in transshipment hubs. *Transportation Science*, 45(4):483–504, 2011.

Online Appendix A: Integer Programming Models for the CBAP

There are two integer programming models that are well-known in the literature for the CBAP. The first model, IP_1, is based on the representation of the CBAP in a space-time diagram (Lee et al. 2010), and can be represented as follows:

$$(\mathrm{IP}_1) \quad \min \sum_{v\in V} w_v(x_v + t_v - a_v) \tag{28}$$

$$\text{s.t.} \quad x_v - (x_u + t_u) \geq (\sigma_{vu} - 1)M, \ \forall v \in V, u \in V \setminus \{v\}, \tag{29}$$

$$y_v - (y_u + l_u) \geq (\delta_{vu} - 1)M, \ \forall v \in V, u \in V \setminus \{v\}, \tag{30}$$

$$\sigma_{vu} + \sigma_{uv} + \delta_{vu} + \delta_{uv} \geq 1, \ \forall v \in V, u \in V \setminus \{v\}, \tag{31}$$

$$x_v \geq a_v, 0 \leq y_v \leq B - l_v, \ \forall v \in V, \tag{32}$$

$$\sigma_{vu} \in \{0,1\}, \delta_{vu} \in \{0,1\}, \ \forall v \in V, u \in V \setminus \{v\}, \tag{33}$$

where the binary variables σ_{vu} and δ_{vu} indicate the relative positions of every two different vessels v and u in the space-time diagram, with $\sigma_{vu} = 1$ implying that v is positioned completely on the right of u, i.e., $x_v \geq x_u + t_u$, and with $\delta_{vu} = 1$ implying that v is positioned completely above u, i.e., $y_v \geq y_u + l_u$. (29)–(31) are equivalent to the condition that no two vessels overlap in the space-time diagram, where M represents a sufficiently large constant. (32) restricts the position of each vessel due to its arrival time and the length of the quay.

The second model, IP_2, which is defined below, is only for a discretized version of the CBAP (or the Discretized BAP for short), where the quay is discretized into a set of segments denoted by $\mathbb{S}$, and the planning horizon is discretized into a set of periods denoted by $\mathbb{T}$, so that each segment in $\mathbb{S}$ cannot be occupied by more than one vessel during the same time period in $\mathbb{T}$. For each vessel $v \in V$, let $\mathbb{S}_v$ and $\mathbb{T}_v$ indicate the set of segments and the set of time periods that are feasible to v. For each time period $x \in \mathbb{T}_v$ and each segment $y \in \mathbb{S}_v$, we use π_{vxy} to denote a binary variable that indicates whether or not vessel v starts berthing in segment y during period x. Let c_{vxy} denote the weighted turnaround time of vessel v if v starts berthing in segment y during period x. Let $\mathbb{A}_{vxy}$ indicate the set of pairs (x', y') with $x' \in \mathbb{T}$ and $y' \in \mathbb{S}$ such that vessel v will occupy segment y' during period x' if it starts berthing in segment y during period x. The Discretized BAP can be formulated as the following integer programming model:

$$(\mathrm{IP}_2) \quad \min \sum_{v\in V} \sum_{x\in T_v} \sum_{y\in S_v} c_{vxy}\pi_{vxy} \tag{34}$$

$$\text{s.t.} \quad \sum_{x\in T_v} \sum_{y\in S_v} \pi_{vxy} = 1, \ \forall v \in V, \tag{35}$$

$$\sum_{v\in V} \sum_{\forall x\in \mathbb{T}_v, y\in \mathbb{S}_v : (x',y')\in \mathbb{A}_{vxy}} \pi_{vxy} \leq 1, \ \forall x' \in \mathbb{T}, \ y' \in \mathbb{S}, \tag{36}$$

$$\pi_{vxy} \in \{0,1\}, \ \forall v \in V, x \in \mathbb{T}_v, y \in \mathbb{S}_v. \tag{37}$$

In this model, (35) specifies that each vessel must be assigned exactly one pair of starting berth segment and time period, and (36) specifies that each segment cannot be occupied by more than one vessel during the same period. It is worthy of note that Park and Kim (2002) and Guan and Cheung (2004) have formulated the Discretized BAP with a similar integer programming model that differs from the above model only in the formulation of constraints (36).

Online Appendix B: Proof of Proposition 1

Proof. To show $Z_1^{\mathrm{LP}} \leq Z_{\mathbb{F}}$, consider $[f_1^*, ..., f_n^*] \in \mathbb{F}$ that is optimal to model $Z_{\mathbb{F}}$. We have

$$Z_{\mathbb{F}} = W(f_1^*, ..., f_n^*) + \frac{1}{2}\sum_{v\in V} w_v t_v. \tag{38}$$

From (5) and (6), we have

$$\begin{aligned}
&\int_{t\in[0,\infty)} (t-a_v) f_v^*(t)\mathrm{d}t \geq \int_{t\in[a_v,a_v+t_v)} (t-a_v) f_v^*(t)\mathrm{d}t + t_v \int_{t\in[a_v+t_v,\infty)} f_v^*(t)\mathrm{d}t \qquad (39)\\
&= \int_{t\in[a_v,a_v+t_v)} (t-a_v) f_v^*(t)\mathrm{d}t + t_v[l_v t_v - \int_{t\in[a_v,a_v+t_v)} f_v^*(t)\mathrm{d}t]\\
&= -\int_{t\in[a_v,a_v+t_v)} \int_{s\in[t,a_v+t_v]} f_v^*(t)\mathrm{d}s\mathrm{d}t + l_v t_v^2 = -\int_{s\in[a_v,a_v+t_v)} \int_{t\in[a_v,s]} f_v^*(t)\mathrm{d}t\mathrm{d}s + l_v t_v^2.
\end{aligned}$$

Thus, since (5) and (7) implies that

$$-\int_{s\in[a_v,a_v+t_v)} \int_{t\in[a_v,s]} f_v^*(t)\mathrm{d}t\mathrm{d}s + l_v t_v^2 \geq -\int_{s\in[a_v,a_v+t_v)} l_v(s-a_v)\mathrm{d}s + l_v t_v^2 = l_v t_v^2/2,$$

we obtain $\int_{t\in[0,\infty)}(t-a_v)f_v^*(t)\mathrm{d}t \geq l_v t_v^2/2$. This, together with (10), (2) and (38), implies that

$$Z_1^{\mathrm{LP}} = \sum_{v\in V} \frac{w_v}{l_v t_v}(l_v t_v^2/2) + \sum_{v\in V} \frac{1}{2} w_v t_v \leq W(f_1^*, ..., f_n^*) + \frac{1}{2}\sum_{v\in V} w_v t_v = Z_{\mathbb{F}}.$$

Hence, (i) of Proposition 1 is proved.

Next, consider any instance with $n > t'$, $l_v = B/t'$, $a_v = 0$ and $t_v = t'$ for $v \in V$ and for some $t' > 0$. Let $[f_1^*, ..., f_n^*] \in \mathbb{F}$ denote an optimal solution to model $Z_{\mathbb{F}}$. By (6), $n > t'$ and (8),

$$\sum_{v\in V} \int_{t\in[0,\infty)} f_v^*(t)\mathrm{d}t = nB > t'B = \int_{t\in[0,t')} B\mathrm{d}t \geq \int_{t\in[0,t')} \sum_{v\in V} f_v^*(t)\mathrm{d}t = \sum_{v\in V} \int_{t\in[0,t')} f_v^*(t)\mathrm{d}t,$$

which implies that there exists at least one vessel $v \in V$ such that $f_v^*(p) > 0$ for some $p \geq t'$. Moreover, since $f_v^*(t) \in \mathbb{S}$, there must exist $\delta > 0$ such that $f_v^*(t) = f_v^*(p) > 0$ for $t \in [p, p+\delta)$, which, together with $p \geq t' = t_v$ and $a_v = 0$, implies that $p \geq a_v + t_v$, and that

$$\int_{t\in[p,p+\delta)} (t-a_v) f_v^*(t)\mathrm{d}t = \int_{t\in[p,p+\delta)} t f_v^*(t)\mathrm{d}t > p\int_{t\in[p,p+\delta)} f_v^*(t)\mathrm{d}t \geq t_v \int_{t\in[p,p+\delta)} f_v^*(t)\mathrm{d}t.$$

Thus, it can be seen that the left-hand side of (39) must be strictly greater than the right-hand side, which implies that $Z_1^{\mathrm{LP}} < Z_{\mathbb{F}}$. Hence, (ii) of Proposition 1 is proved. □

Online Appendix C: Proof of Lemma 1

Proof. To show that the condition in Lemma 1 is necessary, by contradiction, suppose there exists an optimal solution f_v to $Z^{(v)}(f_1, ..., f_{v-1})$, exists $p \in [0, \infty)$, and exists a feasible solution f_v' to $Z^{(v)}(f_1, ..., f_{v-1})$ with $f_v(t) = f_v'(t)$ for all $0 \leq t < p$, such that $f_v(p) < f_v'(p)$. Since both f_v and f_v' are in $\mathbb{S}$, there must exist $\delta_1 > 0$ such that $f_v(t) = f_v(p)$ and $f_v'(t) = f_v'(p)$ for $t \in [p, p+\delta_1)$.

This, together with $f_v(t)=f'_v(t)$ for $t\in[0,p)$, and $\int_{t\in[0,\infty)} f_v(t)\mathrm{d}t=\int_{t\in[0,\infty)} f'_v(t)\mathrm{d}t=l_vt_v$ due to (13), implies that there must exist $p'\geq p+\delta_1$ and $\delta_2>0$ such that $f_v(t)\leq f'_v(t)$ for $t\in[p+\delta_1,p')$, that $f_v(p')>f'_v(p')$, and that $f_v(t)=f_v(p')$ and $f'_v(t)=f'_v(p')$ for $t\in[p',p'+\delta_2)$. Thus, define $\delta:=\min\{\delta_1,\delta_2\}>0$. We obtain that $f_v(t)=f_v(p)<f'_v(p)=f'_v(t)$ for $t\in[p,p+\delta)$, $f_v(t)\leq f'_v(t)$ for $t\in[p+\delta,p')$, and $f_v(t)=f_v(p')>f'_v(p')=f'_v(t)\geq 0$ for $t\in[p',p'+\delta)$.

Since (12) implies $f'_v(p)=0$ for $p\leq a_v$, by $f'_v(p)>f_v(p)\geq 0$ we obtain $p\geq a_v$, implying $p'>p\geq a_v$. Define $\theta:=\min\{f'_v(p)-f_v(p),f_v(p')\}>0$. It can be seen that increasing $f_v(t)$ by θ for $t\in[p,p+\delta)$, and decreasing $f_v(p')$ by θ for $t\in[p',p'+\delta)$, do not violate (13). Since $p\leq a_v$ and $f_v(t)-\theta=f_v(p')-\theta\geq 0$ for $t\in[p',p'+\delta)$, it can be seen that such changes do not violate (11) and (12). Moreover, since $f_v(t)+\theta=f_v(p)+\theta\leq f'_v(p)=f'_v(t)$ for $t\in[p,p+\delta)$, $f_v(t)\leq f'_v(t)$ for $t\in[p+\delta,p')$, and $f_v(t)=f'_v(t)$ for $t\in[0,p)$, it can be seen that such changes do not violate (14) and (15). Thus, we obtain that such changes of f_v lead to another feasible solution to $Z^{(v)}(f_1,...,f_{v-1})$, whose objective value, however, is decreased by $\int_{p'}^{p'+\delta}(t-a_v)\theta\mathrm{d}t-\int_{p}^{p+\delta}(t-a_v)\theta\mathrm{d}t=(p'-p)\delta\theta>0$, which contradicts the assumption that f_v is optimal to $Z^{(v)}(f_1,...,f_{v-1})$. Hence, the condition in Lemma 1 is necessary.

To show that the condition in Lemma 1 is sufficient, consider any feasible solution f_v to $Z^{(v)}(f_1,...,f_{v-1})$ that satisfies $f_v(p)\geq f'_v(p)$ for each $p\in[0,\infty)$ and for each feasible solution f'_v to $Z^{(v)}(f_1,...,f_{v-1})$ with $f'_v(t)=f_v(t)$ for $t\in[0,p)$. By contradiction, suppose f_v is not optimal. Consider an optimal solution f^*_v to $Z^{(v)}(f_1,...,f_{v-1})$. There must exist $p\in[0,\infty)$, such that $f^*_v(p)\neq f_v(p)$ and $f^*_v(t)=f_v(t)$ for $t\in[0,p)$. By the assumption on f_v, we have $f_v(p)\geq f^*_v(p)$. However, since we have shown that the condition in Lemma 1 is satisfied by every optimal solution to $Z^{(v)}(f_1,...,f_{v-1})$, it can be seen that $f^*_v(p)\geq f_v(p)$. Thus, we obtain $f_v(p)=f^*_v(p)$, which contradicts $f^*_v(p)\neq f_v(p)$. Hence, the condition in Lemma 1 is sufficient. □

Online Appendix D: Proof of Theorem 2

Proof. It is easy to see that Algorithm 1 runs in $O(n^2)$ time. Consider $[f_1,...,f_n]$ where f_v is an optimal solution to $Z^{(v)}(f_1,...,f_{v-1})$ for each $v=1,...,n$. To further prove Theorem 2, it is sufficient to show that $[f_1,..,f_n]$ is an optimal solution to $Z_{\mathbb{F}}$. By contradiction, suppose this is not true. Consider an optimal solution $[f^*_1,...,f^*_n]\in\mathbb{F}$ to $Z_{\mathbb{F}}$. Let $v\in V$ indicate the smallest vessel index with $f^*_v(t)\neq f_v(t)$ for some $t\geq 0$. Let $p\geq 0$ indicate the smallest time point with $f^*_v(p)\neq f_v(p)$. Thus, $f^*_v(p)\neq f_v(p)$, $f^*_u(t)=f_u(t)$ for $1\leq u\leq v-1$ and $t\geq 0$, and $f^*_v(t)=f_v(t)$ for $t\in[0,p)$. Without loss of generality, we can assume that the optimal solution $[f^*_1,...,f^*_n]$ to $Z_{\mathbb{F}}$ is selected in such a way that v is maximized, breaking ties by maximizing p, and then breaking ties by minimizing $|f^*_v(p)-f_v(p)|$.

Since $[f^*_1,...,f^*_n]\in\mathbb{F}$, it can be seen that f^*_v is a feasible solution to $Z^{(v)}(f^*_1,...,f^*_{v-1})$, which, together with that f_u and f^*_u are equal for $1\leq u\leq v-1$, implies that f^*_v is also a feasible solution to $Z^{(v)}(f_1,...,f_{v-1})$. Thus, since f_v is an optimal solution to $Z^{(v)}(f_1,...,f_{v-1})$, by Lemma 1 and $f^*_v(p)\neq f_v(p)$, we obtain $f^*_v(p)<f_v(p)$. Thus, since (13) implies that $\int_{t\in[0,\infty)} f^*_v(t)\mathrm{d}t=\int_{t\in[0,\infty)} f_v(t)\mathrm{d}t=l_vt_v$, and since (11) implies that both f^*_u and f_u for $u\in V$ are in $\mathbb{S}$, there must exist $\delta>0$ and $p'\geq p+\delta$, such that $f^*_v(t)=f^*_v(p)<f_v(p)=f_v(t)$ for $t\in[p,p+\delta)$, $f^*_v(t)\leq f_v(t)$

for $t \in [p+\delta, p')$, and $f_v^*(t) = f_v^*(p') > f_v(p') = f_v^*(t)$ for $t \in [p', p'+\delta)$, and that for all $u \in V$, $f_u^*(t) = f_u^*(p)$ and $f_u(t) = f_u(p)$ for $t \in [p, p+\delta)$, and $f_u^*(t) = f_u^*(p')$ and $f_u(t) = f_u(p')$ for $t \in [p', p'+\delta)$.

Since (12) implies $f_v^*(t) = 0$ for $t \le a_v$, from $0 \le f_v^*(p) < f_v(p)$ and $p' \ge p+\delta$, we obtain $p \ge a_v$ and $p' > a_v$. By (15) we have $\sum_{u\in V} f_u^*(p) \le B$. Thus, based on p, p' and δ defined above, we can derive contradictions to complete the proof of Theorem 2 by the following two cases:

- Case 1: $\sum_{u\in V} f_u^*(p) < B$. Let $\theta_1 = \min\{B - \sum_{u\in V} f_u^*(p), f_v(p) - f_v^*(p), f_v^*(p')\}$. Since $\sum_{u\in V} f_u^*(p) < B$, $f_v^*(p) < f_v(p)$, and $f_v^*(p') > f_v(p') \ge 0$, we have $\theta_1 > 0$. Since $B - \sum_{u\in V} f_u^*(p) \ge \theta$, $p \ge a_v$, and $f_v^*(p') \ge \theta$, it can be seen that increasing $f_v^*(t)$ by θ_1 for $t \in [p, p+\delta)$, and decreasing $f_v^*(t)$ by θ_1 for $t \in [p', p'+\delta)$ do not violate (4)–(6) and (8). Moreover, since $f_v^*(t) + \theta_1 \le f_v(t)$ for $t \in [p, p+\delta)$, $f_v^*(t) \le f_v(t)$ for $t \in [p+\delta, p')$, and $f_v^*(t) = f_v(t)$ for $t \in [0, p)$, it can be seen that such changes do not violate (7). Thus, we obtain that such changes of f_v^* lead to another feasible solution to $Z_{\mathbb{F}}$, with $W(f_1^*, ..., f_n^*)$ decreased by $[w_v/(l_v t_v)](p'-p)\delta\theta_1 > 0$. This contradicts the definition of $[f_1^*, ..., f_n^*]$.
- Case 2: $\sum_{u\in V} f_u^*(p) = B$. Since $f_v^*(p) < f_v(p)$, and $f_u^*(p) = f_u(p)$ for $1 \le u \le v-1$, we have $\sum_{u=1}^{v} f_v^*(p) < \sum_{u=1}^{v} f_v(p) \le B$. This, together with $\sum_{u\in V} f_u^*(p) = B$, implies that there must exist $e \ge v+1$ with $f_e^*(p) > 0$. Since $e > v$, we have $w_e/(l_e t_e) \le w_v/(l_v t_v)$. Let $\theta_2 = \min\{f_v(p) - f_v^*(p), f_e^*(p), f_v^*(p')\}$. Since $f_v^*(p) < f_v(p)$, $f_e^*(p) > 0$, and $f_v^*(p') > f_v(p') \ge 0$, we have $\theta_2 > 0$. Thus, following similar reasoning to that in Case 1, it can be seen that increasing $f_v^*(t)$ and decreasing $f_e^*(t)$ by θ_2 for $t \in [p, p+\delta)$, and decreasing $f_v^*(t)$ and increasing $f_e^*(t)$ by θ_2 for $t \in [p', p'+\delta)$ do not violate (4)–(8), leading to another feasible solution to $Z_{\mathbb{F}}$. Notice that with such changes, values of $f_u^*(t)$ for $1 \le u < v$ or values of $f_v^*(t)$ for $0 \le t < p$ are not changed, but $W(f_1^*, ..., f_n^*)$ is decreased by $[w_v/(l_v t_v) - w_e/(l_e t_e)](p'-p)\delta\theta_2 \ge 0$, and $|f_v^*(p) - f_v(p)|$ is decreased by $\theta_2 > 0$. This contradicts the definition of $[f_1^*, ..., f_n^*]$. $\square$

Online Appendix E: Proof of Theorem 3

Proof. First, we prove as follows that f_v is a feasible solution to $Z^{(v)}(f_1, ..., f_{v-1})$. As we have shown earlier, f_v satisfies (12), (16), (17) and (18). From (18) and (16) we have that f_v satisfies (14) and (15). Moreover, since $b_{v,m_v} = \infty$, we have $a_v < b_{v,m_v}$, $\beta_{v,m_v} < B$, and $a_v + t_v < b_{v,m_v}$. Thus, $f_v(t)$ for $t \in [b_{v,m_v-1}, b_{v,m_v})$ must be constructed by Case II, which, together with $b_{v,m_v} = \infty$, implies that α must equal $(l_v t_v - Q_v)/(B - \beta_{v,m_v})$, and thus $f_v(t)$ must cover all the remaining area, i.e., $\int_{t\in[b_{v,m_v-1}, b_{v,m_v})} f_v(t)\mathrm{d}t = l_v t_v - Q_{v,m_v}$. Hence, f_v also satisfies (13). Moreover, it can be seen that for each $1 \le j \le m_v$, $f_v(t)$ constructed for $t \in [b_{v,j-1}, b_{vj})$ is a staircase function with at most two pieces, and each piece is defined on a right-half open interval, which implies that f_v also satisfies (11). Hence, f_v is a feasible solution to $Z^{(v)}(f_1, ..., f_{v-1})$.

Thus, to show f_v is optimal, it is sufficient to show that f_v satisfies the condition in Lemma 1. By contradiction, suppose f_v does not satisfy the condition. There must exist $p \in [0, \infty)$ and a feasible solution f'_v to $Z^{(v)}(f_1, ..., f_{v-1})$ with $f_v(t) = f'_v(t)$ for $t \in [0, p)$, such that $f_v(p) < f'_v(p)$.

Consider the construction of $f_v(t)$ in Algorithm 2. Let j indicate the unique index of the interval $[b_{v,j-1}, b_{vj})$ defined in Step 1 that includes p, i.e., $p \in [b_{v,j-1}, b_{vj})$. Consider the construction of $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$ in Steps 2–19. If $Q_{vj} = l_v t_v$, or $a_v \ge b_{vj}$, or $\beta_{vj} = B$, then since $f'_v(t) = f_v(t)$ for $t \in [0, p)$, it can be seen, from (12), (13) and (15) for f'_v and f_v, that $f'_v(t) = f_v(t) = 0$ for $t \in [b_{v,j-1}, b_{vj})$, implying that $f'_v(p) = f_v(p)$, which contradicts $f'_v(p) \neq f_v(p)$. Thus, we have $Q_{vj} < l_v t_v$, $a_v < b_{vj}$ and $\beta_{vj} < B$, implying that $f_v(t)$ for $t \in [b_{v,j-1}, b_{vj})$ must be constructed in Steps 6–18 of Algorithm 2.

We can now complete the proof of the fact that f_v is indeed optimal by showing $f'_v(p) \leq f_v(p)$, which contradicts $f_v(p) < f'_v(p)$: If $f_v(p) = B - \beta_{vj}$, since $f'_v(p) \leq B - \sum_{u=1}^{v-1} f'_u(p) = B - \sum_{u=1}^{v-1} f_u(p) = B - \beta_{vj}$, we have $f'_v(p) \leq f_v(p)$. Otherwise, by (15), $f_v(p) < B - \beta_{vj}$, and we can consider the following two cases:

- If $f_v(p)$ is assigned in Case I (Steps 6–13). Since $f_v(p) < B - \beta_{vj}$, we know that $f_v(p)$ must be set to l_v in Step 12. Thus, $p \in [\alpha, b_{vj})$ and $p < b_{vj} \leq a_v + t_v$. According to the argument earlier for Case I, we know that by now $f_v(t)$ for $t \in [0, \alpha)$ has covered all its available area $l_v(\alpha - a_v)$, and so has $f'_v(t)$. Thus, from (12) and (14) on f'_v, we can obtain $f'_v(p) \leq l_v = f_v(p)$.

- If $f_v(p)$ is assigned in Case II (Steps 14–18). Since $f_v(p) < B - \beta_{vj}$, it can be seen that $f_v(p)$ must be set to zero in Step 17, which implies that $p \in [\alpha, b_{vj})$. According to the argument earlier for Case II, we know that by now $f_v(t)$ for $0 \leq t \leq \alpha$ has covered all the area $l_v t_v$, and so has $f'_v(t)$. Thus, from (12) and (13) on f'_v, we can obtain $f'_v(t) = 0 \leq 0 = f_v(p)$.

Next, we are going to show that Algorithm 2 for each $v \in V$ can achieve a running time of $O(n)$. To show this, since the running time of Algorithm 2 for each $v \in V$ is $O(m_v)$, it is sufficient to show as follows that m_v is in $O(n)$ for each $v \in V$.

Sort elements in $\{a_v : v \in V\} \cup \{a_v + t_v : v \in V\} \cup \{0\} \cup \{\infty\}$ in an increasing order, denoted by $0 = e_0 < e_1 < ... < e_m = \infty$. It can be seen that $m \leq 2n + 2$. Thus, to prove that m_v is in $O(n)$ for each $v \in V$, it is sufficient to show that for each $v \in V$, there exist $p_{vi} \in [e_{i-1}, e_i)$ for $1 \leq i \leq m$ that satisfy the following two conditions: (i) $g_v(t) = B$ for $t \in [e_{i-1}, p_{vi})$, and (ii) $g_v(t)$ equals a constant less than or equal to B for $t \in [p_{vi}, e_i)$, and for $1 \leq i \leq m$. This is because if, for each $v \in V$, such p_{vi} for $1 \leq i \leq m$ exist, then $g_v(t)$ can be represented as a staircase function over at most $2m$ intervals, $[e_0, p_{v1}), [p_{v1}, e_1), ..., [e_{m-1}, p_{vm}), [p_{vm}, e_m)$, such that the set of endpoints of the intervals always includes a_v and $a_v + t_v$, satisfying the requirements of Step 1 of Algorithm 2, and implying that $m_v \leq 2m \leq 4n + 4$ is in $O(n)$.

Thus, all we are now left to prove is the existence of such p_{vi} for $1 \leq i \leq m$ that satisfy the above conditions (i) and (ii) for each $v \in V$. For $v = 1$, since $g_1(t) = 0$ for $t \in [0, \infty)$, setting $p_{1i} = e_{i-1}$ for $1 \leq i \leq m$ satisfy the conditions (i) and (ii). By induction, supposing that such p_{vi} for $1 \leq i \leq m$ exist for $1 \leq v \leq n - 1$, we can construct $p_{v+1,i}$ for $1 \leq i \leq m$ as follows to satisfy the conditions (i) and (ii) for $v + 1$. Consider each interval $[e_{i-1}, e_i)$ where $1 \leq i \leq m$. For each $t \in [e_{i-1}, p_{vi})$, since $g_v(t) = B$, it can be seen from Step 5 of Algorithm 2 that $f_v(t) = 0$, which implies that $g_{v+1}(t) = g_v(t) = B$. Now, consider interval $[p_{vi}, e_i)$ for the following two cases:

- If $Q_{vj} = l_v t_v$, or $\beta_{vj} = B$, or $a_v \geq e_i$, then from Step 5 of Algorithm 2, it can be seen that $f_v(t) = 0$ for $t \in [p_{vi}, e_i)$, which implies $g_{v+1}(t) = g_v(t)$ for $t \in [p_{vi}, e_i)$. Thus, $g_{v+1}(t) = g_v(t)$ for $t \in [e_{i-1}, e_i)$. Hence, by setting $p_{v+1,i} = p_{vi}$ it satisfies the conditions (i) and (ii).

- Otherwise, by Steps 6–18 of Algorithm 2, there exists α such that $g_{v+1}(t) = B$ for $t \in [p_{vi}, \alpha)$, and that $g_{v+1}(t)$ equals a constant not exceeding B for $t \in [\alpha, e_i)$. Thus, by setting $p_{v+1,i} = \alpha$, it satisfies the conditions (i) and (ii).

Hence, we have obtained $p_{v+1,i}$ for $1 \le i \le m$ that satisfy the conditions (i) and (ii) for $v+1$. Thus, such p_{vi} for $1 \le i \le m$ must exist for all $v \in V$, implying that Algorithm 2 for each $v \in V$ can achieve a running time of $O(n)$. This completes the proof of Theorem 3. □

Online Appendix F: An Illustrative Example for Computing $Z_{\mathbb{F}}$

We apply Algorithm 1 on the instance in Figure 2, with Algorithm 2 used in Step 2. Since vessels 1, 2, 3 and 4 are in a non-increasing order of $w_v/(l_v t_v)$, Algorithm 1 will construct f_v for $v = 1, 2, 3, 4$, sequentially. (The resulting $[f_1, f_2, f_3, f_4]$ is the same as that in Example 2 and Figure 2(b).)

For $v = 1$, since $g_1(t) = 0$ for $t \in [0, \infty)$, $a_1 = 1/2$, $t_1 = 1/2$, and $l_1 = 4 = B$, it is easy to see that Algorithm 2 assigns $f_1(t) = l_1 = 4$ for $t \in [1/2, 1)$ to cover all the area of $4 \times 1/2 = 2$, and sets $f_1(t) = 0$ for $t \in [0, 1/2)$ and $t \in [1, \infty)$.

For $v = 2$, we have $g_2(t) = f_1(t)$, i.e., $g_2(t) = 0$ for $t \in [0, 1/2)$, $g_2(t) = 4$ for $t \in [1/2, 1)$, and $g_2(t) = 0$ for $t \in [1, \infty)$. Endpoints of the three intervals include $a_2 = 0$ and $a_2 + t_2 = 1$. Thus, $m_2 = 3$, $b_{20} = 0$, $b_{21} = 1/2$, $b_{22} = 1$, $b_{23} = \infty$, $\beta_{21} = 0$, $\beta_{22} = 4$, and $\beta_{23} = 0$ in Step 1 of Algorithm 2. Next, consider each iteration j of Steps 2–19 of Algorithm 2 for $j = 1, 2, 3$, respectively:

- For $j = 1$, we have $Q_{21} = 0 < 2 = l_2 t_2$, $\beta_{21} = 0 < 4 = B$, $a_2 = 0 < 1/2 = b_{21}$, and $b_{21} = 1/2 < 1 = a_2 + t_2$, implying that Algorithm 2 constructs $f_2(t)$ for $t \in [0, 1/2)$ by Case I (Steps 6–13). Since $l_2 = 2 < 4 = B - \beta_{21}$, and $\alpha = \min\{1/2, 0\} = 0$ (by Step 10), Algorithm 2 sets $f_2(t) = l_2 = 2$ $t \in [0, 1/2)$ in Step 12.
- For $j = 2$, since $\beta_{22} = 4 = B$, Algorithm 2 sets $f_2(t) = 0$ for $t \in [1/2, 1)$ in Step 5.
- For $j = 3$, we have $Q_{23} = 1 < 2 = l_2 t_2$, $\beta_{23} = 0 < 4 = B$, $a_2 = 0 < \infty = b_{23}$, and $b_{23} = \infty > 1 = a_2 + t_2$, implying that Algorithm 2 constructs $f_2(t)$ for $t \in [1, \infty)$ by Case II (Steps 14–18). Noticing that $B - \beta_{23} = 4$ is the current maximal possible value that can be assigned, by (17) we can set $f_2(t) = B - \beta_{23} = 4$ for t from 1 only up to α, and set $f_2(t) = 0$ for all $t \ge \alpha$, where by Step 15, $\alpha = \min\{\infty, (2 \times 1 - 1)/(4 - 0) + 1\} = 5/4$. Thus, Step 17 sets $f_2(t) = 4$ for $t \in [1, 5/4)$, and $f_2(t) = 0$ for $t \in [5/4, \infty)$.

For $v = 3$, we have $g_3(t) = f_1(t) + f_2(t)$, implying that $g_3(t) = 2$ for $t \in [0, 1/2)$, $g_3(t) = 4$ for $t \in [1/2, 1)$, $g_3(t) = 4$ for $t \in [1, 5/4)$, $g_3(t) = 0$ for $t \in [5/4, 3)$, and $g_3(t) = 0$ for $t \in [3, \infty)$. Endpoints of the five intervals include $a_3 = 1$ and $a_3 + t_3 = 1 + 2 = 3$. Thus, we obtain $m_3 = 5$, $b_{30} = 0$, $b_{31} = 1/2$, $b_{32} = 1$, $b_{33} = 5/4$, $b_{34} = 3$, $b_{35} = \infty$, $\beta_{31} = 2$, $\beta_{32} = 4$, $\beta_{33} = 4$, $\beta_{34} = 0$, and $\beta_{35} = 0$ in Step 1 of Algorithm 2. Next, consider each iteration j of Steps 2–19 of Algorithm 2 for $j = 1, 2, 3, 4, 5$, respectively:

- For $j = 1, 2, 3$, since $a_3 = 1 = b_{32} > b_{31}$ and $\beta_{33} = 4 = B$, Algorithm 2 sets $f_3(t) = 0$ for $t \in [0, 1/2)$, $t \in [1/2, 1)$, and $t \in [1, 5/4)$ in Step 5.
- For $j = 4$, we have $Q_{34} = 0 < 4 = l_3 t_3$, $\beta_{34} = 0 < 4 = B$, $a_3 = 1 < 3 = b_{34}$, and $b_{34} = 3 = a_3 + t_3$. Thus, Algorithm 2 constructs $f_3(t)$ for $t \in [5/4, 3)$ by Case I (Steps 6–13). Since $l_3 = 2 < 4 =$

$B - \beta_{34}$, by (18), we can set $f_3(t) = B - \beta_{34} = 4$ for t from $5/4$ only up to α, and set $f_3(t) = l_3 = 2$ for t from α to 3, where by Step 10, $\alpha = \min\{3, [2(5/4-1)-0]/(4-0-2)+5/4\} = 3/2$. Thus, Step 12 sets $f_3(t) = 4$ for $t \in [5/4, 3/2)$, and $f_3(t) = 2$ for $t \in [3/2, 3)$.

- For $j = 5$, we have $Q_{35} = 4 = l_3 t_3$, implying that all the area is covered. Thus, Algorithm 2 sets $f_3(t) = 0$ for $t \in [3, \infty)$ in Step 8.

For $v = 4$, we have $g_4(t) = f_1(t) + f_2(t) + f_3(t)$, implying that $g_4(t) = 2$ for $t \in [0, 1/2)$, $g_4(t) = 4$ for $t \in [1/2, 1)$, $g_4(t) = 4$ for $[1, 3/2)$, $g_4(t) = 2$ for $t \in [3/2, 3)$, and $g_4(t) = 0$ for $t \in [3, \infty)$. Endpoints of the five intervals include $a_4 = 1$ and $a_4 + t_4 = 3$. Thus, we obtain that $m_4 = 5$, $b_{40} = 0$, $b_{41} = 1/2$, $b_{42} = 1$, $b_{43} = 3/2$, $b_{44} = 3$, $b_{45} = \infty$, $\beta_{41} = 2$, $\beta_{42} = 4$, $\beta_{43} = 4$, $\beta_{44} = 2$, and $\beta_{45} = 0$ in Step 1 of Algorithm 2. Next, consider each iteration j in Steps 2–19 of Algorithm 2 for $j = 1, 2, 3, 4, 5$, respectively:

- For $j = 1, 2, 3$, since $a_4 = 1 = b_{42} > b_{41}$ and $\beta_{43} = 4 = B$, Algorithm 2 sets $f_4(t) = 0$ for $t \in [0, 1/2)$, $t \in [1/2, 1)$, and $t \in [1, 3/2)$ in Step 5.
- For $j = 4$, we have $Q_{44} = 0 < 4 = l_4 t_4$, $\beta_{44} = 2 < 4 = B$, $a_4 = 1 < 3 = b_{44}$, and $b_{44} = 3 = a_4 + t_4$, implying that Algorithm 2 constructs $f_4(t)$ for $t \in [3/2, 3)$ by Case I (Steps 6–13). Since $l_4 = 2 = B - \beta_{44}$, Algorithm 2 sets $f_4(t) = B - \beta_{44} = 2$ for $t \in [3/2, 3)$ in Step 8.
- For $j = 5$, we have $Q_{45} = 3 < 4 = l_4 t_4$, $\beta_{45} = 0 < 4 = B$, $b_{45} = \infty > a_4$, and $a_4 + t_4 = 3 < \infty = b_{45}$, implying that Algorithm 2 constructs $f_4(t)$ for $t \in [3, \infty)$ by Case II (Steps 14–18). By (17), we can set $f_4(t) = B - \beta_{45} = 4$ for t from 3 only up to α, and set $f_4(t) = 0$ for all $t \geq \alpha$, where by Step 15, $\alpha = \min\{\infty, (2 \times 2 - 3)/(4-0) + 3\} = 13/4$. Thus, Step 17 sets $f_4(t) = 4$ for $t \in [3, 13/4)$, and $f_4(t) = 0$ for $t \in [13/4, \infty)$.

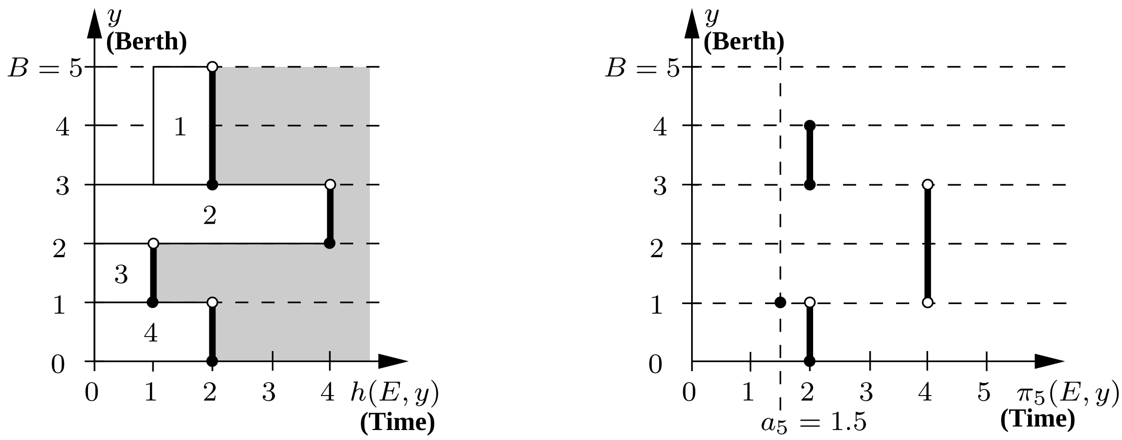
The vector $[f_1, f_2, f_3, f_4]$ that we finally obtained is the same as that in Example 2 and Figure 2(b). By Theorem 3, each f_v for $v = 1, 2, 3, 4$ is an optimal solution to $Z^{(v)}(f_1, ..., f_{v-1})$. By (2), we have that

$$W(f_1, f_2, f_3, f_4) = \frac{1}{4 \times 1/2} \int_{1/2}^{1} 4(t - 1/2)\mathrm{d}t + \frac{1}{2 \times 1}[\int_0^{1/2} 2(t-0)\mathrm{d}t + \int_1^{5/4} 4(t-0)\mathrm{d}t]$$
$$+\frac{1}{2 \times 2}[\int_{5/4}^{3/2} 4(t-1)\mathrm{d}t + \int_{3/2}^{3} 2(t-1)\mathrm{d}t] + \frac{1}{2 \times 2}[\int_{3/2}^{3} 2(t-1)\mathrm{d}t + \int_3^{13/4} 4(t-1)\mathrm{d}t] = \frac{55}{16}.$$

Thus, since $\frac{1}{2}\sum_{v=1}^4 w_v t_v = \frac{11}{4}$, by Theorem 2 and Theorem 1 we obtain that $Z_{\mathbb{F}} = \frac{55}{16} + \frac{11}{4} = 6\frac{3}{16}$ is a lower bound on $Z^* = 7$, greater than $Z_1^{\mathrm{LP}} = \sum_{v=1}^4 w_v t_v = (1/2 + 1 + 2 + 2) = 5\frac{1}{2}$ by (10).

Online Appendix G: Proof of Proposition 2

Proof. Without loss of generality, assume that vessels $v \in V$ are sorted by a non-increasing order of w_v/t_v. Consider the special case of the CBAP with $l_v = B$ and $a_v = a$ for all $v \in V$, where $a \geq 0$ is a constant. First of all, let us apply Algorithm 1 on this special case to construct the optimal solution $[f_1, ..., f_n]$ to $Z_{\mathbb{F}}$. For $v = 1$, since $l_1 = B$, it can be seen that the optimal solution to $Z^{(1)}$ satisfies that $f_1(t) = B$ if $t \in [a, a+t_1)$, and $f_1(t) = 0$, otherwise. This implies that $g_2(t) = B$ for $t \in [a, a+t_1)$. Thus, for $v = 2$, since $l_2 = B$ and $g_2(t) = B$ for $t \in [a, a+t_1)$, it can be seen that the optimal solution to $Z^{(2)}(f_1)$ satisfies that $f_2(t) = B$ if $t \in [a+t_1, a+t_1+t_2)$, and $f_2(t) = 0$, otherwise. This implies that $g_3(t) = B$ for $t \in [a, a+t_1+t_2)$. Thus, by induction,



(a) $h(E,y)$ and $S(E)$ for a partial solution E with four vessels, where $h(E,y)$ is a staircase function shown in bold lines, and $S(E)$ is the shadow area.

(b) $\pi_5(E,y)$, shown in bold lines, for E shown in Figure 5(a) and for vessel 5 with $l_5 = 1$ and $a_5 = 1.5$.

Figure 5 Illustration of functions $h(E,y)$, $S(E)$ and $\pi_u(P,y)$

we can obtain that $f_v(t) = B$ if $t \in [a + t_1 + ... + t_{n-1}, a + t_1 + ... + t_n)$, and $f_v(t) = 0$, otherwise, for $v = 1, 2, ..., n$. Moreover, by Theorem 2, we have $Z_{\mathbb{F}} = W(f_1, ..., f_n) + \sum_{v \in V} w_v t_v / 2$.

On the other hand, as mentioned earlier, for this special case of the CBAP, it is optimal to handle vessels $1, 2, .., n$, sequentially, implying that $\{(v, x_v^*, y_v^*) : v \in V\}$, with $y_v^* = 0$ for $v \in V$, $x_1^* = a$ and $x_v^* = a + t_1 + ... + t_{v-1}$ for $2 \le v \le n$, is an optimal solution. It can be seen that $[f_1, ..., f_n]$ constructed above satisfies that $f_v(t) = B$ if $t \in [x_v^*, x_v^* + t_v)$, and $f_v(t) = 0$, otherwise. Hence, $\{(v, x_v^*, y_v^*) : v \in V\}$ and $[f_1, ..., f_n]$ is an optimal solution to model RF, which implies that $Z^* = W(f_1, ..., f_n) + \sum_{v \in V} w_v t_v / 2 = Z_{\mathbb{F}}$. $\square$

Online Appendix H: Implementation Details of the New Heuristic

In the following, we first illustrate the design of the fitness evaluation function in Section H.1, and then describe the polynomial time implementation of the new heuristic in Section H.2.

H.1. Design of the Fitness Evaluation Function

For any partial solution E, we are given an unassigned vessel $v \in V \setminus V(E)$ and its valid position $(x_v, y_v) \in S(E)$ with $a_v \le x_v$ and $0 \le y_v \le B - l_v$, where, as we defined in (23) and (22), $S(E) = \{(x, y) : x \ge h(E, y),\ 0 \le y < B\}$ with $h(E, y) = \max\{x_v + t_v : \forall v \in V(E) \text{ with } y \in [y_v, y_v + l_v)\}$ for $y \in [0, B)$. See the example of $h(E,y)$ and $S(E)$ in Figure 5(a). To evaluate the fitness of the assignment (v, x_v, y_v), the new heuristic adopts a fitness evaluation function $\phi_E^{(2)}(v, x_v, y_v)$, which is modified from another fitness evaluation function $\phi_E^{(1)}(v, x_v, y_v)$, as illustrated below.

First, we derive $\phi_E^{(1)}(v, x_v, y_v)$ as follows to estimate the difference between the objective values of feasible solutions that can be obtained from E before and after the assignment. To illustrate this, consider any partial solution P, unassigned vessel $u \in V \setminus V(P)$, and starting berth location y with $0 \le y \le B - l_u$. If vessel u is assigned y as its starting berth location, then to avoid overlapping or placing u directly on the left of some assigned vessel, u must be berthed no earlier than $\max\{h(P,z) : y \le z < y + l_u\}$. Thus, since u cannot be berthed earlier than its arrival time a_u, we can represent the earliest possible mooring time of u by $\pi_u(P,y)$, where

$$\pi_u(P,y) := \max\{a_u, \max\{h(P,z) : y \le z < y + l_u\}\}, \text{ for } y \in [0, B - l_u]. \tag{40}$$

See an example of $\pi_u(P,y)$ illustrated in Figure 5(b) with $h(E,y)$ shown in Figure 5(a), where $l_5 = 1$, $a_5 = 1.5$, $\pi_5(E,y) = \max\{1.5, 2\} = 2$ for $y \in [0,1)$, $\pi_5(E,y) = \max\{1.5, 1\} = 1.5$ for $y = 1$, $\pi_5(E,y) = \max\{1.5, 4\} = 4$ for $y \in (1,3)$, and $\pi_5(E,y) = \max\{1.5, 2\} = 2$ for $y \in [3,4]$. Based on (40), we can represent the average earliest possible mooring time of u by $\overline{\pi}_u(P)$, where

$$\overline{\pi}_u(P) := \frac{1}{B - l_u} \int_0^{B-l_u} \pi_u(P,y) \mathrm{d}y, \tag{41}$$

which is defined as equaling $\pi_u(P,0)$ if $l_u = B$. Using $\overline{\pi}_u(P)$ to estimate the mooring time of each unassigned vessel $u \in V \setminus V(P)$, we define

$$\phi(P) := \sum_{u \in V(P)} w_u(x_u + t_u - a_u) + \sum_{u \in V \setminus V(P)} w_u[\overline{\pi}_u(P) + t_u - a_u], \tag{42}$$

to estimate the objective value of the feasible solutions that can be obtained from P. Hence, $\phi(E \cup \{(v, x_v, y_v)\}) - \phi(E)$ can be used to estimate the difference between the objective values of the feasible solutions that can be obtained from E before and after the assignment. Based on this, we obtain our first fitness evaluation function $\phi_E^{(1)}(v, x_v, y_v)$ as follows:

$$\phi_E^{(1)}(v, x_v, y_v) := \phi(E \cup \{(v, x_v, y_v)\}) - \phi(E). \tag{43}$$

Remark 3. Consider the special case of the CBAP with $l_v = B$ and $a_v = a$ for all $v \in V$, where $a \ge 0$ is a constant. As explained in Section 2.4, for this special case it is always optimal to process vessels with the largest w_v/t_v first. Thus, we are interested in investigating how our new heuristic performs in this special case, when $\phi_E^{(1)}$ is used as the fitness evaluation function. For each iteration of the heuristic, consider the partial solution E at its beginning. For each $u \in V \setminus V(E)$, since $l_u = B$ and $a_u = a$, from (22), (23), (40), and (41) we can obtain that $\overline{\pi}_u(E) = \pi_u(E,0) = \max\{a, h(E,0)\}$. Similarly, for any $v \in V \setminus V(E)$ and its valid position $(x_v, y_v) \in S(E)$, and for any $u \in V \setminus V(E) \setminus \{v\}$, we can obtain $\overline{\pi}_u(E \cup \{(v, x_v, y_v)\}) = \pi_u(E \cup \{(v, x_v, y_v)\}, 0) = \max\{a, h(E \cup \{(v, x_v, y_v)\}, 0)\}$. Since $l_v = B$ and $x_v \ge a$, we have $h(E \cup \{(v, x_v, y_v)\}, 0) = x_v + t_v \ge a$, implying that $\overline{\pi}_u(E \cup \{(v, x_v, y_v)\}) = x_v + t_v$. Thus, from (43) and (42) we have

$$\begin{aligned}
\phi_E^{(1)}(v, x_v, y_v) &= w_v[x_v - \overline{\pi}_v(E)] + \sum_{u \in V \setminus V(E) \setminus \{v\}} w_u[\overline{\pi}_u(E \cup \{(v, x_v, y_v)\}) - \overline{\pi}_u(E)] \\
&= w_v[x_v - \max\{a, h(E,0)\}] + \sum_{u \in V \setminus V(E) \setminus \{v\}} w_u[x_v + t_v - \max\{a, h(E,0)\}] \\
&= (\sum_{u \in V \setminus V(E)} w_u)[x_v - \max\{a, h(E,0)\}] + (\sum_{u \in V \setminus V(E)} w_u - w_v)t_v.
\end{aligned} \tag{44}$$

Since $x_v \ge a$ and $(x_v, y_v) \in S(E)$, we have $x_v \ge \max\{a, h(E,0)\}$. Thus, it can be seen that $\phi_E^{(1)}(v, x_v, y_v)$ is minimized when $x_v = \max\{a, h(E,0)\}$ and v minimizes $(\sum_{u \in V \setminus V(E)} w_u - w_v)t_v$. Hence, for this special case, using $\phi_E^{(1)}$ as the fitness evaluation function, our heuristic may not return an optimal solution, unless vessels have equal weights. $\square$

Next, to ensure that the heuristic can guarantee optimal solutions for the above special case with $l_v = B$ and $a_v = a$ for $v \in V$, where $a \geq 0$, even if vessels have different weights, we modify $\phi_E^{(1)}(v, x_v, y_v)$ to obtain the other fitness evaluation function $\phi_E^{(2)}(v, x_v, y_v)$ as follows. Notice that if we divide $\phi_E^{(1)}(v, x_v, y_v)$ by $[w_v(\sum_{u \in V \setminus V(E)} w_u - w_v)]$, then (44) will be changed to

$$(\frac{1}{\sum_{u \in V \setminus V(E)} w_u - w_v} + \frac{1}{w_v})[x_v - \max\{a, h(E, 0)\}] + \frac{t_v}{w_v}. \tag{45}$$

It can be seen that (45) is minimized when $x_v = \max\{a, h(E, 0)\}$ and v maximizes w_v/t_v. Thus, with (45) as the fitness evaluation function, the heuristic always chooses to process vessels with the largest w_v/t_v first. Due to this, we define $\phi_E^{(2)}(v, x_v, y_v)$ as follows:

$$\phi_E^{(2)}(v, x_v, y_v) := \frac{\phi_E^{(1)}(v, x_v, y_v)}{w_v(\sum_{u \in V \setminus V(E)} w_u - w_v)}. \tag{46}$$

By the arguments above, the heuristic with $\phi_E^{(2)}$ as the evaluation function can guarantee optimal solutions to the special case of the CBAP with $l_v = B$ and $a_v = a$ for $v \in V$. Moreover, our preliminary computational experiments show that, using $\phi_E^{(2)}$, the heuristic can always produce better solutions than using $\phi_E^{(1)}$. Therefore, in the bounding procedure of the new exact method, we use the heuristic with $\phi_E^{(2)}$ as the evaluation function to construct feasible solutions and obtain upper bounds of the optimal solutions.

H.2. Polynomial Time Implementation

The key to the implementation of the new heuristic is the minimization of the fitness evaluation function, $\phi_E^{(1)}(v, x_v, y_v)$ or $\phi_E^{(2)}(v, x_v, y_v)$, for the partial solution E of each iteration in Algorithm 3. This can be achieved by first enumerating every unassigned $v \in V \setminus V(E)$, then determining (x_v^*, y_v^*) that minimizes $\phi_E^{(1)}(v, x_v, y_v)$ or $\phi_E^{(2)}(v, x_v, y_v)$ for v, and then choosing v^* that minimizes $\phi_E^{(1)}(v, x_v^*, y_v^*)$ or $\phi_E^{(2)}(v, x_v^*, y_v^*)$. From (46), it can be seen that when v is fixed, the ratio of $\phi_E^{(2)}(v, x_v, y_v)$ and $\phi_E^{(1)}(v, x_v, y_v)$ is also fixed, equal to $w_v(\sum_{u \in V \setminus V(E)} w_u - w_v)$. Thus, the minimization of $\phi_E^{(2)}(v, x_v, y_v)$ for v is equivalent to the minimization of $\phi_E^{(1)}(v, x_v, y_v)$ for v. Therefore, to show that our heuristic with $\phi_E^{(1)}(v, x_v, y_v)$ or $\phi_E^{(2)}(v, x_v, y_v)$ as the fitness evaluation function can achieve a polynomial time complexity, it is sufficient to show that the minimization of $\phi_E^{(1)}(v, x_v, y_v)$ for v can be solved in polynomial time, as follows.

Given $v \in V \setminus V(E)$, due to the continuous domain, minimizing $\phi_E^{(1)}(v, x_v, y_v)$ over (x_v, y_v) is not trivial. To tackle this, we first show that $\phi_E^{(1)}(v, x_v, y_v)$ is non-decreasing in x_v. By (22) and (40), $\pi_u(E \cup \{(v, x_v, y_v)\}, y)$ is non-decreasing in x_v. By (41), $\overline{\pi}_u(E \cup \{(v, x_v, y_v)\})$ for $u \in V \setminus V(E)$ is non-decreasing in x_v. Thus, by (42) and (43), $\phi_E^{(1)}(v, x_v, y_v)$ is non-decreasing in x_v.

Therefore, given v and y_v, to minimize $\phi_E^{(1)}(v, x_v, y_v)$ over x_v, it is sufficient to minimize x_v. Moreover, from (40) we know that $\pi_v(E, y_v)$ indicates the minimal possible value of x_v if v is assigned y_v as its starting berth location. Thus, for a given v, to minimize $\phi_E^{(1)}(v, x_v, y_v)$ over (x_v, y_v), it is sufficient to fix x_v to be $\pi_v(E, y_v)$, and to minimize $\phi_E^{(1)}(v, \pi_v(E, y_v), y_v)$ over y_v.

Given v, we next explain how to minimize $\phi_E^{(1)}(v, \pi_v(E, y_v), y_v)$ over y_v. To simplify the notation, we use E_y to denote the partial solution that adds assignment $(v, \pi_v(E, y), y)$ to E, i.e.,

$$E_y := E \cup \{(v, \pi_v(E, y), y)\}, \text{ for any } 0 \leq y \leq B - l_v. \tag{47}$$

According to (42) and (43), we can rewrite $\phi_E^{(1)}(v, \pi_v(E, y_v), y_v)$ as follows:

$$\phi_E^{(1)}(v, \pi_v(E, y_v), y_v) = w_v[\pi_v(E, y_v) - \overline{\pi}_v(E)] + \sum_{u \in V \setminus V(E) \setminus \{v\}} w_u[\overline{\pi}_u(E_{y_v}) - \overline{\pi}_u(E)]. \quad (48)$$

Thus, to compute $\phi_E^{(1)}(v, \pi_v(E, y_v), y_v)$, it needs to compute $\overline{\pi}_v(E)$, as well as $\overline{\pi}_u(E)$ and $\overline{\pi}_u(E_{y_v})$ for each $u \in V \setminus V(E) \setminus \{v\}$, which, by (41), needs to integrate $\pi_v(E, y)$, $\pi_u(E, y)$ and $\pi_u(E_{y_v}, y)$ for $0 \le y \le B - l_u$. By Proposition 4 below, each of these integrations can be computed, piece by piece, in $O(n)$ time. Thus, $\phi_E^{(1)}(v, \pi_v(E, y_v), y_v)$ can be computed in $O(n^2)$ time.

Proposition 4. *Consider any partial solution P, and any $u \in V \setminus V(P)$. Let $0 = q_{u0} < ... < q_{u,n_u} = B - l_u$ indicate distinct elements in $\{y_j - l_u, y_j, y_j + l_j, y_j + l_j - l_u : j \in V(P)\} \cup \{0\} \cup \{B - l_u\}$ that belong to interval $[0, B - l_u]$. Thus, $n_u \le 4n + 2$. Then, $\pi_u(P, y)$ is a staircase function of y on intervals $(q_{u,i-1}, q_{ui})$ for $1 \le i \le n_u$, and on endpoints q_{ui} for $0 \le i \le n_u$, such that* $\max\{\pi_u(P, q_{u,i-1}), \pi_u(P, q_{ui})\} \le \pi_u(P, y)$ *for each $y \in (q_{u,i-1}, q_{ui})$.*

Proof. For each $1 \le i \le n_u$, it can be proved as follows that $h(P, y)$ equals a constant for all $y \in [q_{u,i-1}, q_{ui})$, and that $h(P, y)$ equals a constant for all $y \in [q_{u,i-1} + l_u, q_{ui} + l_u)$:

- Consider any e and e' with $q_{u,i-1} \le e < e' < q_{ui}$, which implies that there exists no vessel $j \in V(P)$ with $e < y_j \le e'$ or $e < y_j + l_j \le e'$. In other words, for each vessel $j \in V(P)$, either j occupies no berth location in $[e, e']$, or j occupies all berth locations in $[e, e']$. Thus, $\{j \in V(P) : e \in [y_j, y_j + l_j)\} = \{j \in V(P) : e' \in [y_j, y_j + l_j)\}$. By (22), we obtain $h(P, e) = h(P, e')$.
- Consider any e and e' with $q_{u,i-1} + l_u \le e < e' < q_{ui} + l_u$, which implies $q_{u,i-1} \le e - l_u < e' - l_u < q_{ui}$. Thus, there exists no vessel $j \in V(P)$ with $e - l_u < y_j - l_u \le e' - l_u$ or $e - l_u < y_j + l_j - l_u \le e' - l_u$, (i.e., $e < y_j \le e'$ or $e < y_j + l_j \le e'$). Thus, by the same argument as that for the above case where $q_{u,i-1} \le e < e' < q_{ui}$, we can also obtain $h(P, e) = h(P, e')$.

Thus, for $1 \le i \le n_u$, and for any e and e' with $q_{u,i-1} < e < e' < q_{ui}$, we have $h(P, s) = h(P, e')$ for $s \in [q_{u,i-1}, e']$, and $h(P, s) = h(P, q_{u,i-1} + l_u)$ for $s \in [q_{u,i-1} + l_u, e' + l_u]$. Since $q_{u,i-1} < e < e' < q_{ui}$, we obtain $\max\{h(P, s) : e \le s < e + l_u\} = \max\{h(P, s) : e' \le s < e + l_u\} = \max\{h(P, s) : e' \le s < e' + l_u\}$. From this and (40), we obtain $\pi_u(P, e) = \pi_u(P, e')$. Hence, $\pi_u(P, y)$ is a staircase function of y on intervals $(q_{u,i-1}, q_{ui})$ for $1 \le i \le n_u$, and on endpoints q_{ui} for $0 \le i \le n_u$.

Moreover, for each $1 \le i \le n_u$, consider any $y \in (q_{u,i-1}, q_{ui})$. Since $h(P, s)$ equals a constant for $s \in [q_{u,i-1}, q_{ui})$, $\max\{h(P, s) : q_{u,i-1} \le s < q_{u,i-1} + l_u\} \le \max\{h(P, s) : y \le s < y + l_u\}$. Thus, by (40), we have $\pi_u(P, q_{u,i-1}) \le \pi_u(P, y)$. Since $h(P, s)$ equals a constant for $s \in [q_{u,i-1} + l_u, q_{ui} + l_u)$, $\max\{h(P, s) : q_{ui} \le s < q_{ui} + l_u\} \le \max\{h(P, s) : y \le s < y + l_u\}$. Thus, by (40), we have $\pi_u(P, q_{ui}) \le \pi_u(P, y)$. Hence, $\max\{\pi_u(P, q_{u,i-1}), \pi_u(P, q_{ui})\} \le \pi_u(P, y)$. $\square$

For given E and $v \in V \setminus V(E)$, let $0 = e_1 < ... < e_m = B - l_v$ indicate distinct elements in $\{q_{ui} - l_v, q_{ui}, q_{ui} + l_u : u \in V \setminus V(E), 1 \le i \le n_u\} \cup \{0\} \cup \{B - l_v\}$ that belong to interval $[0, B - l_v]$, where $0 = q_{u0} < ... < q_{u,n_u} = B - l_u$ for each $u \in V \setminus V(E)$ indicate the endpoints of the intervals of $\pi_u(E, y)$ on $y \in [0, B - l_u]$, as defined in Proposition 4, which implies $n_u \le 4n + 2$. Thus, $m \le (2 + 3\sum_{u \in V \setminus V(E)} n_u)$ is in $O(n^2)$. From Proposition 5 below, it can be seen that to minimize $\phi_E^{(1)}(v, \pi_v(E, y_v), y_v)$ over y_v, it is sufficient to consider only values of y_v in $\{e_i : 0 \le i \le m\}$.

Proposition 5. *Given any partial solution E, and any $v \in V \setminus V(E)$, define $c(y) := \phi_E^{(1)}(v, \pi_v(E,y), y)$. For each $1 \le i \le m$, we have* $\min\{c(e_{i-1}), c(e_i)\} \le c(y)$ *for any* $y \in (e_{i-1}, e_i)$.

Proof. Consider any i and y with $1 \le i \le m$ and $y \in (e_{i-1}, e_i)$. By Proposition 4, $\pi_r(E, e_{i-1}) \le \pi_r(E, y)$ and $\pi_r(E, e_i) \le \pi_r(E, y)$ for each $r \in V \setminus V(E)$. Thus, by (48) and (41), we have

$$c(e_{i-1}) - c(y) \le \sum_{u \in V \setminus V(E) \setminus \{v\}} \frac{w_u}{B - l_u} \int_0^{B-l_u} [\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s)] \mathrm{d}s. \tag{49}$$

For each $u \in V \setminus V(E) \setminus \{v\}$, and each $p \in [0, B - l_v]$, it can be seen from (40) and (22) that

$$\pi_u(E_p, s) = \begin{cases} \pi_u(E,s), & \text{if } s \in [0, p - l_u], \text{ or } s \in [p + l_v, B - l_u], \\ \max\{\pi_u(E,s), \pi_v(E,p) + t_v\}, & \text{if } s \in (p - l_u, p + l_v). \end{cases}$$

Thus, we can obtain the value of $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s)$ for each $s \in [0, B - l_u]$ as follows:

- For $s \in [0, e_{i-1} - l_u]$ and $s \in [y + l_v, B - l_u]$, $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s) = \pi_u(E,s) - \pi_u(E,s) = 0$.
- For $s \in (e_{i-1} - l_u, y - l_u]$, $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s) = \max\{\pi_u(E,s), \pi_v(E, e_{i-1}) + t_v\} - \pi_u(E,s)$. Since $e_{i-1} < s + l_u \le y < e_i$, there exists no index i with $0 \le i \le n_u$, such that $s + l_u \le q_{ui} + l_u \le y$ (or $s \le q_{ui} \le y - l_u$), and thus by Proposition 4, we have $\pi_u(E,s) = \pi_u(E, y - l_u)$. Thus, since $\pi_v(E, e_{i-1}) \le \pi_v(E,y)$, we obtain $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s) = \max\{0, \pi_v(E, e_{i-1}) + t_v - \pi_u(E, y - l_u)\} \le \max\{0, \pi_v(E,y) + t_v - \pi_u(E, y - l_u)\}$.
- For $s \in (y - l_u, e_{i-1} + l_v)$, $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s) = \max\{\pi_u(E,s), \pi_v(E, e_{i-1}) + t_v\} - \max\{\pi_u(E,s), \pi_v(E,y) + t_v\}$. By $\pi_v(E, e_{i-1}) \le \pi_v(E,y)$, we have $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s) \le 0$.
- For $s \in [e_{i-1} + l_v, y + l_v)$, $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s) = \pi_u(E,s) - \max\{\pi_u(E,s), \pi_v(E,y) + t_v\}$. Since $e_{i-1} < s - l_v \le y < e_i$, there exists no index i with $0 \le i \le n_u$, such that $s - l_v \le q_{ui} - l_v \le y$ (or $s \le q_{ui} \le y + l_v$), and thus by Proposition 4, we have $\pi_u(E,s) = \pi_u(E, y + l_v)$, implying that $\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s) = -\max\{0, \pi_v(E,y) + t_v - \pi_u(E, y + l_v)\}$.

Define $\Delta_u := [\max\{0, \pi_v(E,y) + t_v - \pi_u(E, y - l_u)\} - \max\{0, \pi_v(E,y) + t_v - \pi_u(E, y + l_v)\}]$. From the above, we have $\int_0^{B-l_u} [\pi_u(E_{e_{i-1}}, s) - \pi_u(E_y, s)] \mathrm{d}s \le (y - e_{i-1}) \Delta_u$. Thus, by (49), we obtain

$$c(e_{i-1}) - c(y) \le (y - e_{i-1}) \sum_{u \in V \setminus V(E) \setminus \{v\}} \frac{w_u \Delta_u}{B - l_u}. \tag{50}$$

Similarly, noticing that by Proposition 4, each $r \in V \setminus V(E)$ satisfies that $\pi_r(E, e_i) \le \pi_r(E, y)$, $\pi_r(E,s) = \pi_r(E, y - l_r)$ for $s \in (y - l_r, e_i - l_r]$, and $\pi_r(E,s) = \pi_r(E, y + l_v)$ for $s \in [y + l_v, e_i + l_v)$, we can also obtain

$$c(e_i) - c(y) \le (e_i - y) \sum_{u \in V \setminus V(E) \setminus \{v\}} \frac{-w_u \Delta_u}{B - l_u}. \tag{51}$$

By (50), (51) and $e_{i-1} < y < e_i$, we have that $c(e_{i-1}) \le c(y)$, if $\sum_{u \in V \setminus V(E) \setminus \{v\}} [w_u \Delta_u / (B - l_u)] \le 0$, and $c(e_i) \le c(y)$ otherwise. Thus, we obtain $\min\{c(e_{i-1}), c(e_i)\} \le c(y)$. $\square$

We can now summarize the implementation of the new heuristic with $\phi_E^{(1)}$ or $\phi_E^{(2)}$ as the fitness evaluation function: In each of the n iterations in Step 2 of Algorithm 3, we enumerate $v \in$

$V\setminus V(E)$, and for each v, we enumerate $y_v \in \{e_i : 0 \le 1 \le m\}$, so as to determine v^* and $y^*_{v^*}$ that minimize $\phi^{(1)}_E(v, \pi_v(E, y_v), y_v)$ or $\phi^{(2)}_E(v, \pi_v(E, y_v), y_v)$, and then we set $x^*_{v^*}$ to be $\pi_{v^*}(E, y^*_{v^*})$. Note that $\phi^{(2)}_E(v, \pi_v(E, y_v), y_v)$ for each v and y_v equals $\phi^{(1)}_E(v, \pi_v(E, y_v), y_v)/[w_v(\sum_{u\in V\setminus V(E)} w_u - w_v)]$. Thus, since the computation of $\phi^{(1)}_E(v, \pi_v(E, y_v), y_v)$ needs $O(n^2)$ time for each v and y_v, and since m is in $O(n^2)$ and $|V \setminus V(E)| \le n$, it can be seen that the total running time of the implementation, whether using $\phi^{(1)}_E$ or $\phi^{(2)}_E$, is $O(n^6)$ in the worst case.

Online Appendix I: Proof of Proposition 3

Proof. Consider any two nodes q and q' associated with the same partial solution, reached by the exact method by following two different vessel sequences. Without loss of generality, assume the two sequences differ in their last vessels, denoted by r and r', respectively, with $y_r \le y_{r'}$. To prove Proposition 3, we need to show as follows that either q or q' satisfies $y_v + l_v \le y_u$, and thus must be pruned, where v is an assigned vessel, and u is the vessel assigned just before v is assigned.

In node q, since r' is assigned before r, noting that $y_r \le y_{r'}$ implies that r is not above r' in the space-time diagram, due to the branching rules, r must be on the right of r'. Let v indicate the first vessel that is assigned and placed on the right of r' after r' is assigned. Let u indicate the vessel that is assigned just before v is assigned. Thus, either u equals r', implying $y_u = y_{r'}$, or u is assigned after r' is assigned, implying that u is not placed on the right of r', and therefore must be placed above r', which implies that $y_u \ge y_{r'} + l_{r'} > y_{r'}$. Thus, we obtain $y_{r'} \le y_u$.

In node q', v must be assigned before r'. Thus, since v is on the right of r', due to the branching rules, r' must be placed above v, implying that $y_v + l_v \le y_{r'}$.

Hence, we have $y_v + l_v \le y_{r'} \le y_u$. Proposition 3 is proved. □

Online Appendix J: Proof of Theorem 4

Proof. According to the definition of $Z_{\mathbb{F}}(\theta)$ (in Section 4), by replacing l_v with $\hat{l}_v(\theta)$ in the relaxation $Z_{\mathbb{F}}$ (in Section 2.2) we can obtain $Z_{\mathbb{F}}(\theta)$ as follows:

$$
\begin{aligned}
Z_{\mathbb{F}}(\theta) = \ & \min \ \sum_{v\in V} \frac{w_v}{\hat{l}_v(\theta)t_v} \int_{t\in[0,\infty)} (t-a_v)\hat{f}_v(t)\mathrm{d}t + \frac{1}{2}\sum_{v\in V} w_v t_v \\
& \text{s.t.} \ \ \hat{f}_v \in \mathbb{S}, \quad \forall v \in V, \\
& \qquad \hat{f}_v(t) = 0, \quad \forall t \in [0, a_v) \text{ and } \forall v \in V, \\
& \qquad \int_{t\in[0,\infty)} \hat{f}_v(t)\mathrm{d}t = \hat{l}_v(\theta)t_v, \quad \forall v \in V, \\
& \qquad \int_{t\in[0,t')} \hat{f}_v(t)\mathrm{d}t \le \hat{l}_v(\theta)(t'-a_v), \quad \forall t' \in [a_v, a_v+t_v) \text{ and } \forall v \in V, \\
& \qquad \sum_{v\in V} \hat{f}_v(t) \le B.
\end{aligned}
$$

Here we use $\hat{f}_v(t)$ instead of $f_v(t)$ for ease of presentation below. Noting that constraints (4)–(7) are valid for model Q (in Section 4), since (26) is also a valid constraint, we can obtain a relaxation of model Q by replacing (1), (24), and $\{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}$ with (4)–(7) and (26) as follows:

$$\begin{aligned} &\text{(R')} \quad \min W(f_1, ..., f_n) + \frac{1}{2}\sum_{v\in V} w_v t_v \\ &\text{s.t.} \quad (4)\text{–}(7), \text{ and } (26). \end{aligned}$$

To prove Theorem 4, we only need to show as follows that the relaxation R' above is equivalent to $Z_{\mathbb{F}}(\theta)$: First, it can be seen that for each $v \in V$, $f_v \in \mathbb{S}$ if and only if $\gamma_v(\theta) f_v \in \mathbb{S}$, implying that (4) is equivalent to

$$\gamma_v(\theta) f_v \in \mathbb{S}, \text{ for } v \in V. \tag{52}$$

Next, by multiplying both the left and right sides of (5)–(7) by $\gamma_v(\theta)$, and noting $\hat{l}_v(\theta) = \gamma_v(\theta) l_v$, we obtain their equivalent equalities and inequalities as follows:

$$\gamma_v(\theta) f_v(t) = 0, \quad \forall t \in [0, a_v) \text{ and } \forall v \in V, \tag{53}$$

$$\int_{t\in[0,\infty)} \gamma_v(\theta) f_v(t)\mathrm{d}t = \hat{l}_v(\theta) t_v, \quad \forall v \in V, \tag{54}$$

$$\int_{t\in[0,t')} \gamma_v(\theta) f_v(t)\mathrm{d}t \le \hat{l}_v(\theta)(t' - a_v), \quad \forall t' \in [a_v, a_v + t_v) \text{ and } \forall v \in V. \tag{55}$$

Moreover, from (2) we have that

$$W(f_1, ..., f_n) = \sum_{v\in V} \frac{w_v}{[\gamma_v(\theta) l_v] t_v} \int_{t\in[0,\infty)} (t - a_v)\gamma_v(\theta) f_v(t)\mathrm{d}t = \sum_{v\in V} \frac{w_v}{\hat{l}_v(\theta) t_v} \int_{t\in[0,\infty)} (t - a_v)\gamma_v(\theta) f_v(t)\mathrm{d}t.$$

Thus, we obtain that the relaxation R' of model Q is equivalent to:

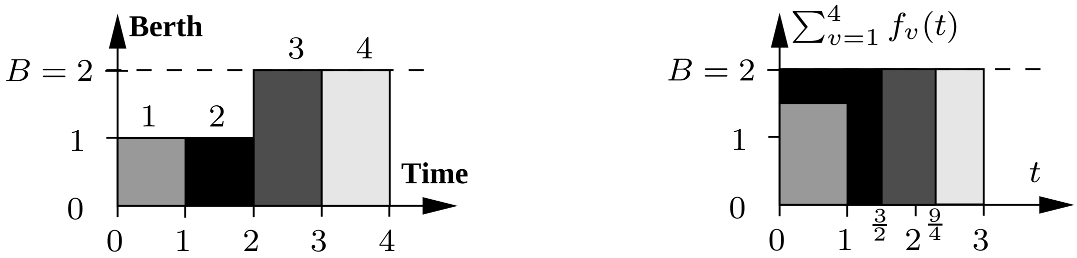
$$\begin{aligned} \min \quad & \sum_{v\in V} \frac{w_v}{\hat{l}_v(\theta) t_v} \int_{t\in[0,\infty)} (t - a_v)\gamma_v(\theta) f_v(t)\mathrm{d}t + \frac{1}{2}\sum_{v\in V} w_v t_v \\ \text{s.t.} \quad & (52)\text{–}(55), (26), \end{aligned}$$

which can be equivalently transformed to $Z_{\mathbb{F}}(\theta)$ by replacing $\gamma_v(\theta) f_v(t)$ with $\hat{f}_v(t)$. This completes the proof of Theorem 4. □

Online Appendix K: An Illustrative Example for Computing $Z_{\mathbb{F}}(\Theta)$ of the CBAPQ

Consider an instance of the CBAPQ with four vessels, where $n = 4$, $a_1 = a_2 = a_3 = a_4 = 0$, $t_1 = t_2 = t_3 = t_4 = 1$, $l_1 = l_2 = 1$, $l_3 = l_4 = 2$, $k_1 = k_2 = 2$, $k_3 = k_4 = 1$, $w_1 = w_2 = w_3 = w_4 = 1$, and $B = K = 2$. Its optimal berth allocation is to handle vessels 1, 2, 3, and 4 sequentially, as shown in Figure 6, with an objective value $Z^* = 1 + 2 + 3 + 4 = 10$. Let $\Theta = \{0, 1, 0.5\}$. To compute $Z_{\mathbb{F}}(\Theta) = \max_{\theta\in\Theta} Z_{\mathbb{F}}(\theta)$, we calculate $Z_{\mathbb{F}}(\theta)$ for $\theta = 0$, 1, and 0.5 as follows:

For $\theta = 0$, we have $Z_{\mathbb{F}}(\theta) = Z_{\mathbb{F}}(0)$, which equals the lower bound $Z_{\mathbb{F}}$ of the CBAP. Noting that $w_v/(l_v t_v) = 1$, 1, 0.5, and 0.5 for $v = 1$, 2, 3, and 4, respectively, we can apply Algorithm 1 to obtain an optimal solution $[f_1, f_2, f_3, f_4]$ to $Z_{\mathbb{F}}$, by setting $f_1(t) = f_2(t) = 1$ for $t \in [0, 1)$, $f_3(t) = 2$ for $t \in [1, 2)$, $f_4(t) = 2$ for $t \in [2, 3)$, and others to zero. Thus, $Z_{\mathbb{F}}(0) = 1 + 1 + 2 + 3 = 7$.



(a) An optimal berth allocation with $Z^* = 10$. (b) A stacked bar chart for an optimal solution $[f_1, f_2, f_3, f_4]$ to $Z_{\mathbb{F}}(0.5)$.

Figure 6 Illustration of the computation of $Z_{\mathbb{F}}(\Theta)$ for the instance of the CBAPQ in Section K.

For $\theta = 1$, we have $Z_{\mathbb{F}}(\theta) = Z_{\mathbb{F}}(1)$. By $B = K = 2$, we have $\lambda = B/K = 1$. Thus, the revised vessel sizes $\hat{l}_v(1) = k_v/l_v$ equal 2, 2, 1, 1 for $v = 1$, 2, 3, and 4, respectively. This implies that $w_v/[\hat{l}_v(1)t_v] = 0.5$, 0.5, 1, and 1 for $v = 1$, 2, 3, and 4, respectively. Based on this, we can apply Algorithm 1 to obtain an optimal solution $[f_1, f_2, f_3, f_4]$ to $Z_{\mathbb{F}}(1)$, by setting $f_3(t) = f_4(t) = 1$ for $t \in [0, 1)$, $f_1(t) = 2$ for $t \in [1, 2)$, $f_2(t) = 2$ for $t \in [2, 3)$, and others equal to zero. Thus, $Z_{\mathbb{F}}(1) = 1 + 1 + 2 + 3 = 7$.

When $\theta = 0.5$, we have $Z_{\mathbb{F}}(\theta) = Z_{\mathbb{F}}(0.5)$. By $B = K = 2$, we have $\lambda = B/K = 1$. Thus, the revised vessel sizes $\hat{l}_v(0.5) = 0.5k_v + 0.5l_v$ equal 1.5 for all $v \in \{1, 2, 3, 4\}$. This implies that $w_v/[\hat{l}_v(0.5)t_v] = 2/3$ for all $v \in \{1, 2, 3, 4\}$. Based on this, we can apply Algorithm 1 to obtain an optimal solution $[f_1, f_2, f_3, f_4]$ to $Z_{\mathbb{F}}(0.5)$, by setting $f_1(t) = 1.5$ for $t \in [0, 1)$, setting $f_2(t) = 0.5$ for $t \in [0, 1)$ and $f_2(t) = 2$ for $t \in [1, 1.5)$, setting $f_3(t) = 2$ for $t \in [1, 2.25)$, setting $f_4(t) = 2$ for $t \in [2.25, 3)$, and setting others equal to zero. See Figure 6(b). Thus, we obtain that

$$\begin{aligned} Z_{\mathbb{F}}(0.5) &= \frac{1}{1.5}(1.5 \times 0.5t^2\big|_0^1 + 0.5 \times 0.5t^2\big|_0^1 + 2 \times 0.5t^2\big|_1^{1.5} + 2 \times 0.5t^2\big|_{1.5}^{2.25} + 2 \times 0.5t^2\big|_{2.25}^{3}) + 0.5 \times 4 \times 1 \\ &= (1/1.5)(0.75 + 0.25 + 1.25 + 1.5 \times 4.5) + 2 = 8. \end{aligned}$$

Hence, for $\Theta = \{0, 1, 0.5\}$, we obtain $Z_{\mathbb{F}}(\Theta) = \max_{\theta \in \Theta} Z_{\mathbb{F}}(\theta) = 8$.

Online Appendix L: Details on Extending the New Exact Method to the CBAPQ

With the lower bound $Z_{\mathbb{F}}(\Theta)$ derived above we can extend our exact method as follows to solve the CBAPQ. We still follow the branch-and-bound approach by using the same mixed-strategy to select nodes to be branched.

From any selected node p with its partial solution E, we follow similar branching rules to generate new nodes by selecting in turn each unassigned vessel $v \in V \setminus V(E)$, and then assigning v all its admissible positions (x_v, y_v) in the space-time diagram. Let $k(E, t)$ indicate the number of quay cranes used by assigned vessels in E at time t. Due to the quay crane constraint, the admissible positions (x_v, y_v) must satisfy that $k(E, t) + k_v \leq K$ for all $t \in [x_v, x_v + t_v)$. With this, we can extend the admissible positions defined for the CBAP to the CBAPQ.

The bounding procedure of the exact method for the CBAPQ is similar to that for the CBAP. For each new node generated with a partial solution E', we can compute a lower bound $Z_{\mathbb{F}}(\Theta, E')$ on the remaining problem by extending the method for computing $Z_{\mathbb{F}}(\Theta)$. To balance the lower bound quality and the time consumption, we set Θ to be $\{0.0, 0.01, ..., 1.0\}$ for the computation of $Z_{\mathbb{F}}(\Theta) = \max_{\theta\in\Theta} Z_{\mathbb{F}}(\theta)$ in the root node of the search tree, and let θ^* indicate the value of $\theta \in \Theta$ that maximizes $Z_{\mathbb{F}}(\theta)$, and then, for partial solutions E' of other non-root nodes in the search tree, we restrict Θ to be $\{0.0, \theta^*, 1.0\}$ for the computation of $Z_{\mathbb{F}}(\Theta, E')$.

To compute an upper bound from any selected node p, we can extend its partial solution E to a feasible solution by a best-fit heuristic similar to Algorithm 3, where positions (x_v, y_v) to be examined for each unassigned vessel v must satisfy $k(E,t) + k_v \le K$ for all $t \in [x_v, x_v + t_v)$, due to the quay crane constraint.

Moreover, among the dominance rules derived for the CBAP, it can be verified that rule 1 and rule 3 are still valid for the CBAPQ. Rules 4 and 5 are valid for the CBAPQ only for vessels r and v that have an equal number of pre-assigned quay cranes. Although rule 2 is not valid for the CBAPQ, it can be revised by taking into account the quay crane constraint.

Online Appendix M: The CBAPQ with Variable-in-Time Pre-Assigned Quay Cranes

In the CBAPQ, the number of quay cranes assigned to each vessel is fixed during its handling period. However, in some situations the number of quay cranes assigned to a vessel can vary from time to time. For these situations, we need to solve a more generalized berth allocation problem, referred to as the GCBAPQ for short, where each vessel $v \in V$ is assigned a set of pairs $\{(h_{v1}, k_{v1}), ..., (h_{vm_v}, k_{vm_v})\}$ (with $h_{v1} = 0$), indicating that for each $1 \le j \le m_v$, the number of quay cranes assigned to v is changed to k_{vj} after h_{vj} time units from the mooring time of v.

To derive a lower bound on the optimal solution to the GCBAPQ, we construct an instance of the CBAPQ with fixed numbers of quay cranes assigned to the vessels, by splitting each vessel into m_v vessels, where for each vessel (v,j) with $v \in V$ and $1 \le j \le m_v$, its length $l_{vj} := l_v$, handling time $t_{vj} := h_{v,j+1} - h_{vj}$ (with $h_{v,m_v+1} := t_v$), arrival time $a_{vj} := a_v + h_{vj}$, importance weight $w_{vj} := w_v/m_v$, and its number of pre-assigned quay cranes equals k_{vj}. It can be seen that every solution $\{(x_v, y_v) : v \in V\}$ to the GCBAPQ can be transformed to a feasible solution to this CBAPQ instance, by setting $x_{vj} = x_v + h_{vj}$ and $y_{vj} = y_v$ for $v \in V$ and $1 \le j \le m_v$, so that their total weighted turnaround times, $\sum_{v\in V} w_v(x_v + t_v - a_v)$ and $\sum_{v\in V}\sum_{j=1}^{m_v} w_{vj}(x_{vj} + t_{vj} - a_{vj})$, have a difference of $\sum_{v\in V}(m_v - 1)w_v t_v/m_v$. Thus, using the lower bound $Z_{\mathbb{F}}(\Theta)$ defined in (27) for the CBAPQ instance, we can obtain that $Z_{\mathbb{F}}(\Theta) + \sum_{v\in V}(m_v - 1)w_v t_v/m_v$ is a valid lower bound for the GCBAPQ.

Moreover, we can extend the exact method for the CBAPQ to the GCBAPQ by following the same mixed-strategy to select nodes to be branched, adopting similar dominance rules, and using the lower bound derived above. For the branching rules, as well as the heuristic used in the bounding procedure, we need to revise the quay crane constraint for assigning a position (x, y) to an unassigned vessel v for a partial solution E, as follows:

$$k(E,t) + k_{vj} \le K, \text{ for } x + h_{vj} \le t < x + h_{v,j+1},\ 1 \le j \le m_v.$$

Online Appendix N: Integer Programming Models for the CBAPQ

First, we extend model IP_1 of the CBAP (described in Online Appendix A) to the CBAPQ. For the K quay cranes, we index them by 1, 2, ..., and K. The quay crane constraint is then equivalent to that each crane k for $1 \le k \le K$ can handle at most one vessel at any time. For each $v \in V$ and $1 \le k \le K$, we introduce a new binary variable z_{vk} to indicate whether or not vessel v is handled by crane k. To ensure that each vessel v is assigned k_v cranes, we have:

$$\sum_{k=1}^{K} z_{vk} = k_v, \text{ for } v \in V, \tag{56}$$

and due to the quay crane constraint, we have:

$$z_{vk} + z_{uk} \le \sigma_{vu} + \sigma_{uv} + 1, \text{ for } 1 \le k \le K, \tag{57}$$

where the binary variable σ_{vu}, as defined in IP_1, equals 1 if vessel v is positioned completely on the right of vessel u in the space-time diagram. Thus, adding (56), (57), and $z_{vk} \in \{0,1\}$ for $v \in V$ and $1 \le k \le K$ to IP_1, we obtain the following integer programming model for the CBAPQ:

$$\begin{aligned} (\text{IPQ}_1) \quad & \min \sum_{v \in V} w_v(x_v + t_v - a_v) \\ \text{s.t.} \quad & (29) - (33), (56), \text{ and } (57), \\ & z_{vk} \in \{0,1\}, \text{ for } v \in V \text{ and } 1 \le k \le K. \end{aligned}$$

From the computational results, we know that, similar to model IP_1 of the CBAP, the optimal objective value of model IPQ_1 and its linear programming relaxation often have a large gap.

Next, we extend model IP_2 of the CBAP (described in Online Appendix A) to the CBAPQ, given that the quay is discretized into a set of segments $\mathbb{S}$, and that the planning horizon is discretized into a set of periods $\mathbb{T}$. For this, let $\mathbb{T}_{vxy}$ indicate the set of time periods $t \in \mathbb{T}$ such that vessel v will be processed during period t if it starts berthing in quay segment y during time period x. Thus, the quay crane constraint can be represented by:

$$\sum_{v \in V} \sum_{\forall x \in \mathbb{T}, y \in \mathbb{S}_v : t \in \mathbb{T}_{vxy}} k_v \pi_{vxy} \le K, \ \forall t \in \mathbb{T}, \tag{58}$$

where the binary variable π_{vxy}, as defined in IP_2, equals 1 if vessel v starts berthing in quay segment y during time period x. Thus, adding (58) to IP_2, we obtain the following mixed integer programming model for the CBAPQ:

$$\begin{aligned} (\text{IPQ}_2) \quad & \min \sum_{v \in V} \sum_{x \in \mathbb{T}_v} \sum_{y \in \mathbb{S}_v} c_{vxy} \pi_{vxy} \\ \text{s.t.} \quad & (35)\text{–}(37), \text{ and } (58). \end{aligned}$$

It can be seen that, similar to model IP_2 of the CBAP, model IPQ_2 contains $O(nBH)$ variables and $O(n + BH)$ constraints, where $B = |\mathbb{S}|$ and $H = |\mathbb{T}|$. Therefore, it can be very time consuming to solve IPQ_2 and its linear programming relaxation when B and H are large.

Online Appendix O: Computational Results for the Test of the Extensions to the CBAPQ

Table 7 reports the results on test data with $5 \leq n \leq 24$ for the CBAPQ, to compare the extension of the new exact method (Extended New BB) with using CPLEX on integer programming models IPQ_1 and IPQ_2 of the CBAPQ. See Online Appendix N for details of the two models.

Table 7 Computational results on CBAPQ instances with $5 \leq n \leq 24$.

Set	n	B	#inst	Extended New BB					CPLEX on IPQ_1			CPLEX on IPQ_2		
				T_0	T_1	T_2	$G\%$	S	T_1	$G\%$	S	T_2	$G\%$	S
a5wq	5	80	30	0	**0**	-	**0.0**	**30**	0	0.0	30	-	-	0
a10wq	10	80	30	3	**2**	-	**0.0**	**30**	469	3.2	19	-	-	0
a12wq	12	80	10	54	-	-	**0.0**	**10**	-	20.3	0	-	-	0
a15wq	15	80	10	1507	-	-	**0.7**	**9**	-	41.8	0	-	-	0
b10wq	10	10	10	0	**0**	**0**	**0.0**	**10**	40	0.0	10	30	0.0	10
b15wq	15	10	10	86	-	**60**	**0.0**	**10**	-	28.4	0	2026	2.5	3
b20wq	20	10	10	1540	-	-	**2.5**	**7**	-	59.3	0	-	3.8	0
cwq	11–24	600	44	25	**0**	-	**0.0**	**44**	34	0.2	43	-	-	0
Total			154					**150**			102			13

Table 8 reports the results on test data with $40 \leq n \leq 160$ for the CBAPQ, to compare the extension of the new exact method with an extension of the meta-heuristic SWO (Extended SWO) for the CBAPQ, and to compare the extension of the new lower bound $Z_{\mathbb{F}}(\Theta)$, where $\Theta = \{0.0, 0.01, ..., 1.0\}$, with a lower bound Z_1^{LPQ} derived from the linear programming relaxation of model IPQ_1.

Table 8 Computational results on CBAPQ instances with $40 \leq n \leq 160$.

Set	n	B	#inst	Extended New BB		Extended SWO		$Z_{\mathbb{F}}$		Z_1^{LPQ}	
				$G_{\text{lb*}}^{\text{ub}}\%$	T	$G_{\text{lb*}}^{\text{ub}}\%$	T	$G_{\text{ub*}}^{\text{lb}}\%$	T	$G_{\text{ub*}}^{\text{lb}}\%$	T
a40wq	40	80	30	**8.6**	3600	23.2	3600	7.2	0	83.7	1
a80wq	80	80	30	**7.3**	3600	20.1	3600	6.5	0	91.4	37
a120wq	12	80	30	**6.7**	3600	19.8	3600	6.2	0	94.2	86
a160wq	160	80	30	**6.1**	3600	18.8	3600	5.7	0	95.6	241

Online Appendix P: Extending the Reformulation of the CBAP for Other Objective Terms

The new model RF proposed in Section 2.1 of this paper for the CBAP can be extended to reformulate other berth allocation problems with various objective terms in addition to the total weighted turnaround time of the vessels. To demonstrate this, let us consider the following two additional objective terms: (i) The total cost of tardiness of the vessels, and (ii) the total cost of transporting cargo from berths to the yard, both of which have been studied in the existing literature (see Meisel and Bierwirth (2009), Park and Kim (2003) for examples).

Consider any feasible solution $\{(v, x_v, y_v) : v \in V\} \in \mathbb{Y}$. For each $v \in V$, let τ_v indicate the requested departure time of vessel v. It can be seen that the tardiness of vessel v equals $\max\{x_v + t_v - \tau_v, 0\}$. Following Meisel and Bierwirth (2009), we consider linear tardiness costs for the vessels. Let $w_v^{(\text{i})}$ indicate the unit cost of tardiness for each vessel v. We obtain that the additional objective term (i) for the total cost of tarindess can be represented in (59) as follows:

$$\sum_{v \in V} w_v^{(\text{i})} \max\{x_v + t_v - \tau_v, 0\}. \tag{59}$$

Moreover, for each $v \in V$, let $\hat{y}_v$ indicate the preferred starting berth location for the least cost, denoted by $\hat{d}_v$, for transporting cargo of vessel $v \in V$ to the yard. It can be seen that the distance between the preferred starting berth location and the allocated location of vessel v equals $|y_v - \hat{y}_v|$. Following Park and Kim (2003), we consider the additional cost for transporting cargo between the allocated location y_v and the yard to be linear in $|y_v - \hat{y}_v|$. Let $w_v^{(\mathrm{ii})}$ indicate the additional transportation cost per unit distance. We obtain that the additional objective term (ii) for the total cost of transporting cargo between berths and the yard can be represented in (60) as follows:

$$\sum_{v \in V} (\hat{d}_v + w_v^{(\mathrm{ii})} |y_v - \hat{y}_v|). \tag{60}$$

As a result, model IP_1 of the CBAP can be extended to the problem with the above two additional objective items by revising the objective to

$$\min \sum_{v \in V} w_v (x_v + t_v - a_v) + \sum_{v \in V} w_v^{(\mathrm{i})} \max\{x_v + t_v - \tau_v, 0\} + \sum_{v \in V} (\hat{d}_v + w_v^{(\mathrm{ii})} |y_v - \hat{y}_v|). \tag{61}$$

Next, we are going to show how model RF for the CBAP can be extended to incorporate the two additional objective terms represented in (59) and (60). To reformulate (59), we introduce a new function $r_v(t)$ for each $v \in V$ and $t \geq 0$, to indicate the quay space occupied by vessel v at time t or later, which can be represented by the following staircase function with two pieces:

$$r_v(t) = \begin{cases} l_v, & \text{for } t \in [0, x_v + t_v), \\ 0, & \text{for } t \in [x_v + t_v, \infty). \end{cases} \tag{62}$$

Since $\int_{t \in [\tau_v, \infty)} r_v(t) \mathrm{d}t = l_v \max\{x_v + t_v - \tau_v, 0\}$, we can reformulate (59) as a function on $[r_1, ..., r_n]$ as follows:

$$\sum_{v \in V} w_v^{(\mathrm{i})} \min\{x_v + t_v - \tau_v, 0\} = \sum_{v \in V} \frac{w_v^{(\mathrm{i})}}{l_v} \int_{t \in [\tau_v, \infty)} r_v(t) \mathrm{d}t. \tag{63}$$

To reformulate (60), note that $|y_v - \hat{y}_v| = \max\{\hat{y}_v - y_v, 0\} + \max\{y_v - \hat{y}_v, 0\}$. For each $v \in V$, we introduce a new function $\eta_v(s)$ for $s \in [0, B)$, to indicate the length of duration for vessel v to occupy any berth location at or below s, which can be represented by the following staircase function with two pieces:

$$\eta_v(s) = \begin{cases} 0, & \text{for } s \in [0, y_v), \\ t_v, & \text{for } s \in [y_v, B). \end{cases} \tag{64}$$

It can be seen that

$$\int_{s \in [0, \hat{y}_v)} \eta_v(s) \mathrm{d}s = t_v \max\{\hat{y}_v - y_v, 0\}. \tag{65}$$

Moreover, for each $v \in V$, we introduce another binary function $\xi_v(s)$ for $s \in [0, B)$, to indicate the length of duration for vessel v to occupy any berth location at or above s:

$$\xi_v(s) = \begin{cases} t_v, & \text{for } s \in [0, y_v + l_v), \\ 0, & \text{for } s \in [y_v + l_v, B). \end{cases} \tag{66}$$

It can be seen that

$$\int_{s\in[\hat{y}_v+l_v,B)} \xi_v(s)\mathrm{d}s = t_v \max\{(y_v+l_v)-(\hat{y}_v+l_v),0\} = t_v \max\{y_v-\hat{y}_v,0\}. \tag{67}$$

By (65) and (67), we can reformulate (60) as a function on $[\eta_1,...,\eta_n]$ as follows:

$$\sum_{v\in V}(\hat{d}_v + w_v^{(\mathrm{ii})}|y_v-\hat{y}_v|) = \sum_{v\in V}\frac{w_v^{(\mathrm{ii})}}{t_v}\int_{s\in[0,\hat{y}_v)}\eta_v(s)\mathrm{d}s + \sum_{v\in V}\frac{w_v^{(\mathrm{ii})}}{t_v}\int_{s\in[\hat{y}_v+l_v,B)}\xi_v(s)\mathrm{d}s + \sum_{v\in V}\hat{d}_v. \tag{68}$$

Define $W^{(\mathrm{i})}$ and $W^{(\mathrm{ii})}$ as the following functions on $[r_1,...,r_n]$ and $[\xi_1,...,\xi_n]$, respectively:

$$W^{(\mathrm{i})}(r_1,...,r_n) = \sum_{v\in V}\frac{w_v^{(\mathrm{i})}}{l_v}\int_{t\in[\tau_v,\infty)} r_v(t)\mathrm{d}t, \tag{69}$$

$$W^{(\mathrm{ii})}(\xi_1,...,\xi_n) = \sum_{v\in V}\frac{w_v^{(\mathrm{ii})}}{t_v}\int_{s\in[0,\hat{y}_v)}\eta_v(s)\mathrm{d}s + \sum_{v\in V}\frac{w_v^{(\mathrm{ii})}}{t_v}\int_{s\in[\hat{y}_v+l_v,B)}\xi_v(s)\mathrm{d}s. \tag{70}$$

By (63) and (68), we can extend model RF to incorporate the two additional objective terms as follows:

$$\begin{aligned}
\text{(ERF)}\quad & \min\ W(f_1,...,f_n) + \frac{1}{2}\sum_{v\in V} w_v t_v + W^{(\mathrm{i})}(r_1,...,r_n) + W^{(\mathrm{ii})}(\xi_1,...,\xi_n) + \sum_{v\in V}\hat{d}_v \\
& \text{s.t.}\ \ (1),\ (62),\ (64),\ (66),\ \text{and}\ \{(v,x_v,y_v): v\in V\}\in\mathbb{Y}.
\end{aligned}$$

Remark 4. It can be seen that the objective function in model ERF above is linear in $f_v(t)$, $r_v(t)$, $\eta_v(s)$, and $\xi_v(s)$. The new lower bound proposed in this paper for the RF is still valid for model ERF, since the objective function of model RF is always less than or equal to that of model ERF. To strengthen the lower bound, we can further derive various relaxations of model ERF by following or extending our approach proposed in this paper for model RF. This requires identifying a number of valid constraints on only vectors $[f_1,..,f_n]$, $[r_1,...,f_n]$, and $[\xi_1,...,\xi_n]$, and using them to replace the constraints (1), (62), (64), (66), and $\{(v,x_v,y_v): v\in V\}\in\mathbb{Y}$ of model ERF. The valid constraints have to be included properly, so that solving the relaxation can be not only computationally tractable, but also effective in providing tight lower bounds on the optimal solutions to model ERF. Achieving this is challengning, but of significant research value, and we therefore leave it to our future study.