OR Spectrum
DOI 10.1007/s00291-006-0036-5

REGULAR ARTICLE

Rajeeva Moorthy · Chung-Piaw Teo

Berth management in container terminal: the template design problem

Abstract One of the foremost planning problems in container transshipment operation concerns the allocation of *home berth* (preferred berthing location) to a set of vessels scheduled to call at the terminal on a weekly basis. The home berth location is subsequently used as a key input to yard storage, personnel, and equipment deployment planning. For instance, the yard planners use the home berth template to plan for the storage locations of transshipment containers within the terminal. These decisions (yard storage plan) are in turn used as inputs in actual berthing operations, when the vessels call at the terminal. In this paper, we study the economical impact of the home berth template design problem on container terminal operations. In particular, we show that it involves a delicate trade-off between the service (waiting time for vessels) and cost (movement of containers between berth and yard) dimension of operations in the terminal. The problem is further exacerbated by the fact that the actual arrival time of the vessels often deviates from the scheduled arrival time, resulting in last-minute scrambling and change of plans in the terminal operations. Practitioners on the ground deal with this issue by building (capacity) buffers in the operational plan and to scramble for additional resources if needs be. We propose a framework to address the home berth design problem. We model this as a rectangle packing problem on a cylinder and use a sequence pair based simulated annealing algorithm to solve the problem. The sequence pair approach allows us to optimize over a large class of packing efficiently and decomposes the home berth problem with data uncertainty into two smaller subproblems that can be readily handled using techniques from stochastic project scheduling. To evaluate the quality of a template, we use a dynamic berth allocation package developed recently by Dai et al. (unpublished manuscript,

Part of this work was done when the second author was at the SKK Graduate School of Business, Sungkyunkwan University, South Korea.

R. Moorthy · C.-P. Teo (✉)
Department of Decision Sciences, NUS Business School, National University of Singapore,
1 Business Link, 117592, Singapore
E-mail: bizteocp@nus.edu.sg, rajeeva.moorthy@gmail.com

2004) to obtain various berthing statistics associated with the template. Extensive computational results show that the proposed model is able to construct efficient and robust template for transshipment hub operations.

Keywords Container logistics · Transshipment hub · Sequence pair · Project management

1 Introduction

Mega container terminals around the world routinely handle more than 10 million TEU of cargo and serve thousands of vessels in a year. Efficiency of container operations (along berth and within yard), to certain extent, determines the competitiveness of the terminals within the global shipping network. This depends on a delicate coordination of various expensive resources, including the deployment of quay cranes and crews, allocation of prime movers and drivers, planning and deployment of yard resources etc.

Port operations planning can be broadly classified into the following categories:

– *Strategic* planning deals with long-term issues, such as strategic alliances with shipping lines, infrastructure development to support volume growth, etc. A major exercise in this phase is to identify proper allocation of major/feeder services to different terminals or various sections within a terminal to ensure quick vessel turnaround and transship containers in short time windows.
– *Tactical* planning deals primarily with midterm berth and yard planning issues. A berth template and an associated yard template are usually drawn so as to minimize berthing delays and operational bottlenecks. The tactical plans follow the general guidelines laid out in the strategic plans and is a primary driver of the operational planning phase.
– *Operational* planning involves more detailed equipment and manpower deployment plans, taking into consideration real time operational constraints.

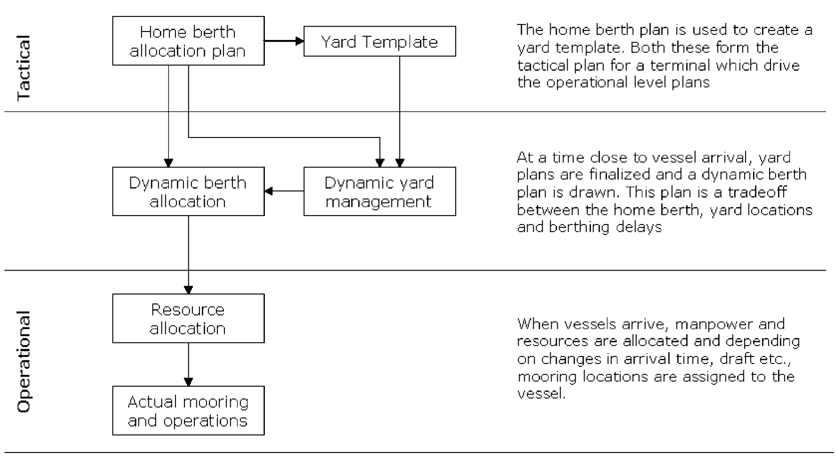

Fig. 1 Tactical and operational planning before mooring a vessel

These plans broadly follow the tactical plans and changes are dynamically made so as to satisfy customer service demands.

In this paper, we study a tactical problem motivated by the operation in a large container terminal, where close to 80% of the containers handled are designated for transshipment to other destinations. Figure 1 shows the major activities and the influence of the tactical plans on the operational level planning and execution.

Planning the yard is critical because the containers, after being unloaded from a vessel, will be moved to an area in the yard to wait for the arrival of a connecting vessel. The designated storage location in the yard and its distance from the mooring positions of the connecting vessels along the quay determine to a large extent the workload needed to carry out the transshipment operations. In general, finding a proper storage plan for the transshipment containers is the main challenge confronting the yard planners in the terminal. Designing these storage plans, however, requires prior knowledge of the mooring locations and time-of-arrival of the vessels. To address this issue, the terminal operator currently assigns a *home berth* location for each vessel calling at the terminal on a weekly basis. Note that in the current operational environment, almost all vessels calling the terminal follow a fixed cycle of 7, 10, or 14 days, of which majority arrive on a weekly schedule. The storage plan for the transshipment containers is designed by assuming that these vessels will be moored at the designated home berths upon arrival.

Figure 2 shows two possible solutions for the home berth allocation problem. The horizontal axis shows the scheduled time of call, which depends on the ship's schedule (thus not entirely within the control of the terminal operator) and scheduled departure time of a vessel, whereas the vertical axis shows the berthing location assigned. There are two groups of vessels (i.e., $\{3,4\}$ and $\{1,2,5\}$) belonging to two shipping lines that call at the terminal. In this example, let us assume that containers are exchanged only between vessels within a group. The key question we would like to address in this paper is this: *Which one of the two templates should we use for the home berth allocation*?

To minimize the amount of work for container operations, the template on the left is desirable because vessels belonging to the same group are moored in close proximity to each other. The transshipment containers in each group can thus be

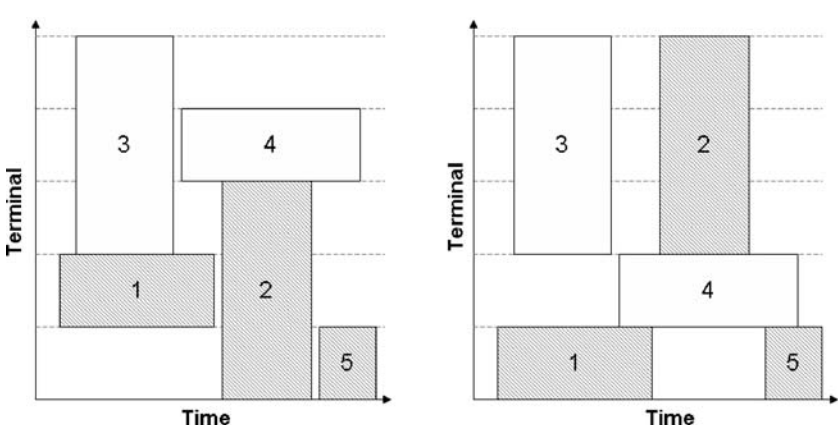

Fig. 2 Two solutions to the home berth allocation problem

stored in the same area of the yard, reducing the amount of work and distance covered by the prime movers. However, in reality, the actual time of call of each vessel will normally deviate from its scheduled time, and the processing time needed to service a vessel may vary. This results in the necessity to adjust the mooring locations of the vessels along the quay. For instance, if the departure of vessel 1 is delayed for an extended time, the operation for vessel 2 will be adversely affected. The delays may propagate and spread to other vessels (such as 5). In practice, the planners may have to moor a vessel far away from its home berth, just to cut down the waiting time of the vessel. In this case, the template on the right of Fig. 2 may be more desirable, as there are sufficient gaps (buffers) in front of the scheduled time-of-call of each vessel. The template is thus more *robust* to unexpected delays experienced by the vessels, and this leads to a more stable and reliable mooring plan, even though it may result in higher container handling cost.

Finding a good home berth template is clearly a difficult combinatorial problem, because we have to search through exponentially many different ways to assign a location to each vessel. Note that the time-of-call of the vessels are preannounced by the shipping lines and cannot be changed by the template designer. We also need to address the associated problem of finding a good way to evaluate the *robustness* of a home berth template. To this end, we identify two primary objectives used in container operations:

- *Service level-waiting time*. This is defined as the time elapsed between the actual time-of-call at the port and the beginning of the mooring operation along the berth. A vessel is said to be *berthed-on-arrival* (BOA) if the mooring operation commences within 2 h of arrival. The BOA statistics is often used as a proxy to gauge the quality of service provided by the port operator.
- *Operational cost-connectivity*. The actual movement cost (containers move from quay to yard storage location and, subsequently, from storage to quayside to be loaded to the connecting vessel) is difficult to estimate and depends also on the storage plan of the containers. As a proxy, we approximate the movement cost with the following: Let x_i and x_j denote the berthing locations of vessels i and j (measured with respect to the midpoint of the vessels) and c_{ij} the number of containers to be exchanged between vessels i and j. If vessel i arrives before j, then c_{ij} denotes the number of containers that need to be transferred from i to j. On the other hand, if vessel j arrives before i, then c_{ij} denotes the number of containers that need to be transferred from j to i. The connectivity cost is defined to be $c_{ij} \times d(x_i, x_j)$, where $d(\cdot, \cdot)$ is a properly selected distance function. In reality, the effort required to transport containers depends on their storage locations in the yard. However, as a policy, most of the containers are stored close to the berth where the vessel on which they should be loaded will be moored. Hence, using the berthing locations of the vessels to compute connectivity cost is acceptable in reality.

We use the two opposing objectives to develop an approach to evaluate the relative performance of different home berth templates. Figure 3 shows the efficient frontiers for two different templates: B is clearly a better design compared to A, as it attains a higher service level at a lower operating cost.

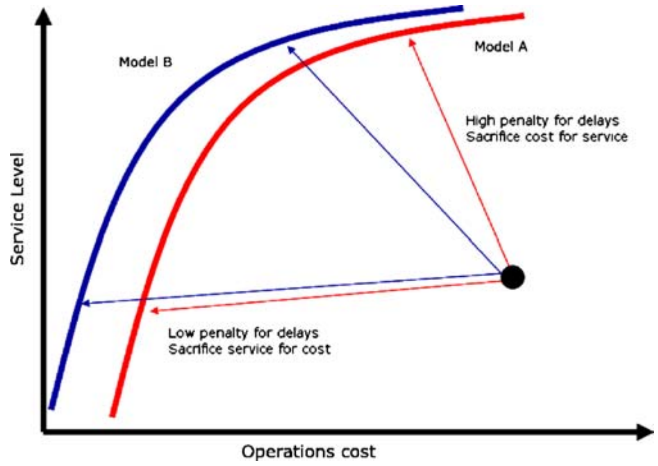

Fig. 3 The efficient frontier for two different templates

In the rest of this paper, we develop a methodology to design a good home berth template, and we use the dynamic berth allocation planning package developed in Dai et al. (unpublished manuscript, 2004) to estimate the efficient frontier attained.

1.1 Literature review

For an introduction to the terminal operations, we refer to Murty et al. (2005). The operational issues involved in managing a container terminal are vividly described, and they identify a number of operational planning problems. The foremost of these is the berth assignment problem. It is also highlighted that data uncertainty plays a critical role in decision making, and the authors specifically recommend techniques to solve some of the operational issues. The paper, however, deals only with operational issues in the terminal and furthermore does not solve the berth planning problem.

Most of the papers in the berth planning literature focus on the combinatorial complexity of the static berth planning problem, where the objective is to obtain a nonoverlapping berth plan within a given scheduling window (cf. Brown et al. (1994); Lim (1998); Chen and Hsieh (1999) and the references therein). In all these papers, it is assumed that some preferred berth location is already known. The focus is to penalize deviation from the preferred locations and to penalize for excessive waiting times. The connectivity cost component is not considered. Dai et al. (unpublished manuscript, 2004) solve the berth planning problem for the dynamic case when vessels arrive over time. They provide a set of stability conditions when the arrival information is random using the tools from stochastic processing networks. They also provide a local search method for obtaining good berth plans using the sequence pair approach.

The berth template problem is also related to a variant of the two-dimensional rectangle packing problem, where the objective is to place a set of rectangles in the plane without overlap so that a given cost function will be minimized. In typical rectangle packing papers, the objectives are normally to minimize the height or area used in the packing solution. Imahori et al. (2003), building on a long series of work by Murata et al. (1996, 1998), Tang et al. (2000) etc., propose a local search method for this problem, using an encoding scheme called sequence pair. Their approach is able to address the rectangle packing problem with spatial cost function of the type $g(max_i p_i(x_i), max_i q_i(t_i))$, where p_i, q_i are general cost functions and can be discontinuous or nonlinear, and g is nondecreasing in its parameters. Given a fixed sequence pair, Imahori et al. (2003) showed that the associated optimization problem can be solved efficiently using a Dynamic Programming framework.

The technique in this paper builds on the sequence pair concept in this series of works. We extend the concept to handle the situation of rectangle packing on a cylinder (instead of the plane). We use this technique to handle the complexity inherent in the combinatorial explosion of the number of different possible template designs. We also integrate this method with results in stochastic project scheduling, which allows us to analyze the impact of arrival time uncertainties on waiting times of the vessels for a class of templates. Note that the problem of evaluating expected delay in a project network can be cast as one of determining the expected longest path in a network with random arc lengths. The problem is well-studied in the project management community.

2 The sequence pair approach

In this section, we describe how we structure the search over all possible template designs, using an encoding scheme called sequence pair (represented by a pair of permutations of the vessels). Each pair of permutation corresponds to *a class* of templates, where the connectivity cost and waiting time objectives can be evaluated efficiently. We use the standard simulated annealing scheme to search the space of all sequence pairs.

Note that a container terminal is divided into a number of linear stretches of berthing space called wharfs, which are further subdivided into sections, called berths, depending on the draft.

In the template, the berthing time is fixed at the scheduled arrival time of the vessels. The decision variables in the home berth problem is essentially x_i, the berthing location of vessel i within the terminal. However, as we are also interested in the delays experienced by each vessel, we let t_i (another decision variable) denote the *planned* berthing time of vessel i, where t_i may be larger than $\mu(r_i)$, the scheduled (expected) arrival time of the vessel. The difference between t_i and $\mu(r_i)$ is the planned delays for vessel i. The actual delays experienced by each vessel are more complicated, as they depend on the (random) arrival time of several other vessels calling at the terminal. We will develop a technique in the next section to capture the expected delays due to the home berth allocation decisions. (See Table 1 for the list of notations used in this paper.)

A priori, it is not clear whether we can design a home berth template where all the vessels are nonoverlapping, because there may be instances where the demand for terminal space exceed the total available space within the terminal. Another complication lies in the periodic schedule operated by the vessels, as we need to ensure that the packing near the two boundaries are properly aligned to avoid excessive overlapping of terminal space, as one move from the end of 1 week to the beginning of the next.

Table 1 Notations used in the paper

l_i	Length of vessel i
p_i:	Expected length of port stay upon berthing by vessel i
w_i:	Cost for delaying vessel i. This can be interpreted as the vessel's priority class.
P:	Number of hours in the planning horizon (we use 24×7 h for a weekly template)
M:	Number of berths in the terminal
L_i:	Length of berth i, $i = 1, \ldots, M$

If we ignore the inherent periodicity of the weekly arrival schedule and the issue of overlapping vessels in the template, the layout of the home berth template (cf. Fig. 2) can be conveniently viewed as a packing of rectangles in a two-dimensional (time–space) plane, where the berthing time of the vessel i is taken to be $\mu(r_i)$, the scheduled (expected) arrival time of the vessel. However, the periodicity of the schedule introduces added complexity to the problem—the template is more suitably viewed as packing rectangles on a cylinder with circumference P.

For ease of exposition, we will first review the sequence pair concept for packing on a plane. Consider the template of two vessels as shown in Fig. 4. We define a pair of permutations H and V associated with the template and construct them with the following properties:

- If vessel a is on the right of vessel b in H, then vessel b *does not see* vessel a on its LEFT-UP view.
- Similarly, if vessel i is on the right of vessel j in V, then vessel j *does not* see vessel i on its LEFT-DOWN view.

It is clear that, given any (nonoverlapping) template, we can construct a pair (H, V) (need not be unique) satisfying the above properties. For the rest of the paper, we write $a <_H b$ (and $a <_V b$) if a is placed on the left of b in H (respectively in V). For any two vessels a and b, the ordering of a, b in H, V essentially determines the relative placement of vessels in the packing.

- If $a <_H b$ and $a <_V b$, then a does not see b in LEFT-DOWN or LEFT-UP, i.e., vessel b is to the right of vessel a. In other words, vessel b can only be berthed after vessel a leaves the terminal.

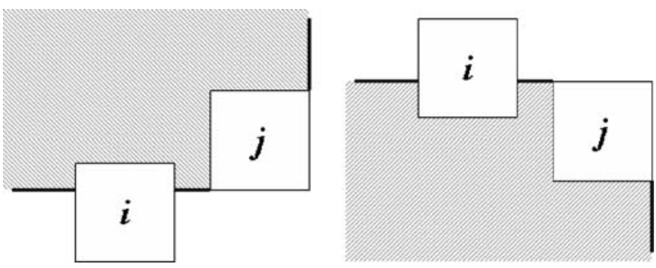

Fig. 4 The figure on the *left* shows the LEFT-UP view of vessel j, whereas the figure on the *right* shows the LEFT-DOWN view of vessel j in the time–space plane. In the former, j cannot be on the right of i in H, whereas in the latter, j cannot be on the right of i in V

– If $a <_H b$ and $b <_V a$, then a does not see b in LEFT-UP and b does not see a in the LEFT-DOWN view, i.e., vessel b is berthed below vessel a in the terminal.

For any H and V, either one of the above holds, i.e., either vessel a and vessel b do not overlap in time (one is to the right of the other) or do not overlap in space (one is on top of the other).

Note that every sequence pair (H, V) corresponds to a *class* of templates satisfying the above properties. The constraints imposed by the sequence pairs splits into two classes: constraints of the type $x_i + l_i/2 \leq x_j - l_j/2$ (in the space variables) or of the type $t_i + p_i \leq t_j$ (in the time variables). In this way, finding the optimal packing in this class, given a fixed sequence pair, decomposes into two subproblems: *space* and *time* (cf. Fig. 5). In the space and time graphs, each vessel is represented by a node, and the constraints imposed by the sequence pair are represented by directed arcs.

This decomposition provides the flexibility to address the stochastic issues in the time problem and the connectivity cost minimization in the space problem separately. This feature turns out to be extremely useful for our problem, as the delay experienced by each vessel, for a fixed sequence pair, no longer depends on the home berth location decision, but on the precedence constraints imposed by the time-constraint graph.

With this insight, we obtain the optimal packing by searching among all permutations of H and V. For each sequence pair (H, V), the procedure for solving the time problem is described in the "Estimating the expected delays" section. "Estimating the connectivity cost" section describes a model to solve the space problem. The periodicity of vessel arrivals introduces additional complexity in solving the template problem and we introduce the virtual wharf mark technique in the "Rectangle packing on cylinder" section to address this issue. In "Searching over the sequence pairs," a simulated annealing-based search procedure is described to obtain the optimal sequence pair.

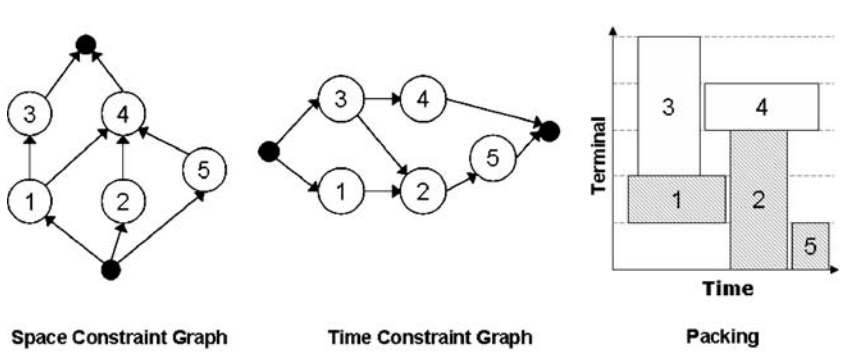

Fig. 5 Directed graphs on the space and time variables along with a packing arising from the sequence pair, H:{3,4,1,2,5} and V:{1,3,2,5,4}

3 Evaluation of the template

3.1 Estimating the expected delays

In this section, we present a derivation for estimating the expected waiting time in a general project management network with release time constraints. The associated precedence relationship in time is shown as the *time-constraint graph* (cf. Fig. 5).

Let r_i and p_i represent the release time and processing time, respectively, for each job in the network. We assume that both r_i and p_i are normal r.v., with mean $\mu(r_i)$, $\mu(p_i)$, and variance $\sigma(r_i)^2$, $\sigma(p_i)^2$. Let S_i represent a minimal set of vessels that should complete before vessel i can be processed. Note that S_i depends on the template obtained. In our example in Fig. 5, $S_1 = \emptyset$, $S_3 = \emptyset$, $S_2 = \{1, 3\}$, $S_4 = \{3\}$, and $S_5 = \{2\}$.

The planned berthing time t_i is easy to determine in this case: it is simply the (random) earliest possible starting time of vessel i. The value t_i depends on the completion time of the jobs in S_i, as

$$t_i = \max\left(r_i, \max_{j \in S_i} (t_j + p_j)\right).$$

The expected waiting time for job i is thus $E(t_i - r_i)$, where

$$t_i - r_i = \max\left(0, \max_{j \in S_i} (t_j + p_j) - r_i\right).$$

Finding the exact distribution of the maximum of a multivariate distribution with an arbitrary covariance structure is a difficult computational problem. In the project management area, a common technique, called the Project Evaluation and Review Technique, identifies a critical (longest) path in the network and uses certain, carefully chosen distribution (such as Beta or Normal distribution) to approximate the longest path duration in the stochastic network. The parameters in the distribution for $t_i - r_i$ are chosen with mean $\mu(t_i - r_i)$ and variance $\sigma^2(t_i - r_i)$,

where

$$\mu(t_i - r_i) = \max\left(0, \max_{j \in S_i} (E(t_j) + E(p_j)) - \mu(r_i)\right), \tag{1}$$

and $\sigma^2(t_i - r_i) = 0$ if $\mu(r_i) \geq \max_{j \in S_i} (E(t_j) + E(p_j))$, otherwise

$$\sigma^2(t_i - r_i) = (\sigma^2(t_{j^*}) + \sigma^2(p_{j^*})) + \sigma^2(r_i) \text{ with } j^* = \operatorname{argmax}\left(E(t_j) + E(p_j) : j \in S_i\right). \tag{2}$$

This approach works well when there is a dominant longest path, i.e., a unique solution to Eq. 1 exists, and this path attains the largest value in most realization of the stochastic longest path problem. However, the variance estimation of the longest path is too conservative, especially if there are many paths with mean path length close to $\mu(t_i - r_i)$ in Eq. 1. Unfortunately, this is normally the case in the template design problem, especially in a heavily congested environment. It is crucial to distinguish between templates where there are only one vs several potential paths, which may delay the berthing of a particular vessel. For this purpose, we need a more refined estimate of the variance of the longest path distribution.

When the underlying distributions are Gaussian, the following result is well-known: Let $X = (X_1, \ldots, X_n)$ be a multivariate normal distribution, with identical mean $E(X_j) = m$, and variance $\sigma^2(X_j)$ for all $j = 1, \ldots, n$. Let $\max(X)$ denote $\max(X_1, \ldots, X_n)$.

Proposition (Borel's Inequality) For all $\lambda > 0$,

$$P(\max(X) - E(\max(X)) \geq \lambda) \leq P(Z \geq \lambda),$$

where $Z \approx N(0, \max(\sigma^2(X_1), \ldots, \sigma^2(X_n)))$.

Note that the result does not hold for multivariate normal distribution if the mean values are not identical. It suggests that the maximum variance can be used to approximate the spread of $\max(X)$ above the mean. We use the above insight to improve our estimation of the variance for the longest path.

Fixed a buffer length L, define $S_i(L) \subset S_i$ with

$$k \in S_i(L) \text{ if and only if } (E(t_k) + E(p_k)) \geq \max\left(\mu(r_i), \max_{j \in S_i} (E(t_j) + E(p_j))\right) - L.$$

$S_i(L)$ consists of those paths whose expected length is within the buffer L from

$$\max\left(\mu(r_i), \max_{j \in S_i} (E(t_j) + E(p_j))\right),$$

the earliest time vessel i can be berthed at the terminal. It corresponds to those jobs that may potential block the berthing operations of vessel i. We refine our estimate of the variance of the longest path by:

$$\sigma^2(t_i - r_i) = \begin{cases} 0 & \text{if } S_i(L) = \emptyset \\ \max_{j \in S_i(L)} \left(\sigma^2(t_j) + \sigma^2(p_j)\sigma^2(r_i)\right) & \text{otherwise.} \end{cases} \tag{3}$$

Note that the refined estimate for $\sigma^2(t_i - r_i)$ is larger in Eq. 3 than Eq. 2 and grows with the number of elements in $S_i(L)$. This is desirable as it provides a convenient way to penalize against template where there are many other vessels blocking (within buffer of L hours) the berthing operation of another vessel.

For the rest of this paper, we will approximate the distribution of $t_i - r_i$ by a normal distribution $N(\mu(t_i - r_i), \sigma^2(t_i - r_i))$, with the mean and variance given by Eqs. 1 and 3. Hence,

$$\begin{aligned}
E(t_i - r_i) &\approx \int_0^{\infty} (x) \frac{1}{\sigma(t_i - r_i)\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left(\frac{x - \mu(t_i) - \mu(r_i)}{\sigma(t_i - r_i)} \right)^2 \right\} dx \\
&= \int_{\frac{\mu(r_i) - \mu(t_i)}{\sigma(t_i - r_i)}}^{\infty} (y(\sigma(t_i - r_i)) + \mu(t_i) - \mu(r_i)) \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} y^2 \right\} dy \\
&= \sigma(t_i - r_i) L\left(\frac{\mu(r_i) - \mu(t_i)}{\sigma(t_i - r_i)} \right) + (\mu(t_i) - \mu(r_i)) \left(1 - F\left(\frac{\mu(r_i) - \mu(t_i)}{\sigma(t_i - r_i)} \right) \right)
\end{aligned}$$

where $L(\cdot)$ and $F(\cdot)$ are the standard unit normal loss and cumulative density function and $\sigma(t_i - r_i)^2 = \sigma(t_i)^2 + \sigma(r_i)^2$. This value can thus be easily computed to a high degree of accuracy by evaluating a simple integral, provided the value $\mu(t_k)$ and $\sigma(t_k)$ can be determined a priori for $k \in S_i$.

3.2 Estimating the connectivity cost

Given a sequence pair (H, V), let G_S be the directed graph associated with constraints involving the berthing location variables x_i. Let $L_i, U_i, \quad i = 1, \ldots, W$, denote the position of the lower and upper end of wharf i in the terminal. The connectivity-cost problem can be formulated as:

$$\begin{aligned}
(P_C) \quad & \min \sum_{i=1}^{N} \sum_{j=i+1}^{N} c_{ij} d(x_i, x_j) \\
s.t. \quad & x_k + l_k/2 \leq \sum_{i=1}^{W} U_i y_{ik} \quad \forall \quad k = 1, \ldots, N \\
& x_k - l_k/2 \geq \sum_{i=1}^{W} L_i y_{ik} \quad \forall \quad k = 1, \ldots, N \\
& \sum_{i=1}^{W} y_{ik} = 1 \quad \forall \quad k = 1, \ldots, N \\
& x_i + l_i/2 \leq x_j - l_j/2 \ \mathit{if} \ (i, j) \in G_S \\
& y_{ik} \in \{0, 1\}, \quad i = 1, \ldots, W, k = 1, \ldots, N \\
& x_i \leq 0, \quad i = 1, \ldots, N.
\end{aligned}$$

The first three constraints ensure that the vessels are not berthed across different wharfs. By definition, each wharf corresponds to a stretch of linear space along the quay in the terminal. Note that $y_{ik} = 1$ implies that vessel i is berthed in wharf k. The complexity of this problem depends on the structure of $d(\cdot)$. Furthermore, we need an extremely efficient routine to estimate the minimum connectivity cost, because we have to solve this problem repeatedly over a large number of sequence pairs. To this end, it will be convenient if x_i is chosen to be the *smallest* value satisfying the constraints in (P_C), so that a solution to (P_C) can be obtained easily from the space precedence graph G_S , in a recursive manner: For each vertex j, suppose $\max_{i:(i,j)\in G_S}(x_i + l_i/2) - l_j/2 \in [L_k, U_k)$, then

$$x_j = \begin{cases} \max_{i:(i,j)\in G_S}(x_i + l_i/2) + l_j/2 & \text{if } \max_{i:(i,j)\in G_s}(x_i + l_i/2) + l_j < U_k \\ L_{k'} + l_j/2 & k' \text{ is the nearest wharft above } [L_k, U_k), \\ & \text{with } U_{k'} - L_{k'} \geqq l_j. \end{cases} \tag{4}$$

Unfortunately, choosing x_i in the above manner may produce a very bad solution with respect to the objective function $\sum_{i=1}^{N}\sum_{j=i+1}^{N} c_{ij} d(x_i, x_j)$. Furthermore, the periodicity in the vessel schedule introduces additional complexity into this problem. We introduce the notion of virtual wharf mark to address both these issues in the next subsection.

3.3 Rectangle packing on cylinder

We first discuss how the estimation of delays can be modified. Consider Fig. 6, where the template on the left ignores the wrap around effect (periodicity) of the schedule; hence, we can treat $S_4 = S_1 = S_2 = \emptyset$. After computing the values $\mu(t_i)$ and $\sigma(t_i)$ for $i \in \{1, 2, 4\}$, we can proceed to compute $\mu(t_3), \sigma(t_3)$, and $\mu(t_5), \sigma(t_5)$, using $S_3 = \{1\}$, $S_5 = \{1, 2, 4\}$. However, if we factor into the periodicity of the schedule, we find that vessel 5 blocks the berthing operation of vessel 1 and 4 (in the following week).

We address this problem by iteratively computing the first and second moment values using the layout of the template on a plane over a few periods (i.e., few weeks in our case). The overlap and blocking situation in Fig. 6 is thus captured if we look at the layout in the second period (week). We stop the computation after a few iterations so that the wrap -around effect is propagated into the computation. We use this modification as the time cost objective while evaluating the waiting time objective of a particular template.

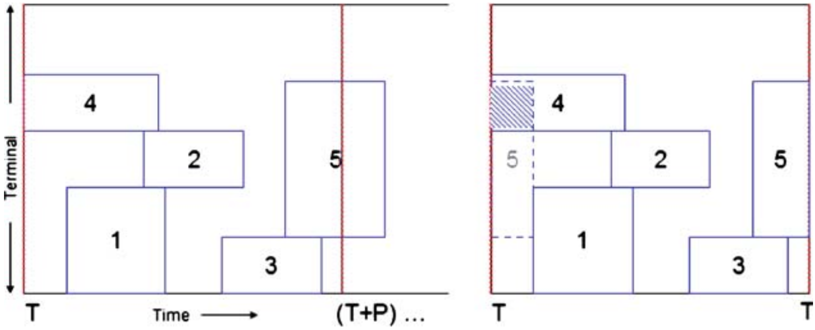

Fig. 6 Template on a plane vs template on a cylinder

We note that, for any nonoverlapping berth template (i.e., packing on a cylinder), it is possible to find a partition such that it can be *unwrapped* to an associated static berth plan (i.e., packing-on-a-plane). Figure 7 shows one such partition obtained by unwrapping. The converse, however ,is not true because we cannot guarantee that the vessels would not overlap when wrapped. Figure 6 shows one such example.

To ensure that vessels can be propped appropriately in the associated static berth plan, we introduce the notion of "virtual wharf mark" (VW). VW's are essentially additional vessels with appropriately chosen r_w, $l_w = 0$ and $p_w = 0$ and with additional lower-bound constraint on the berthing location.

The problem of overlaps due to *wrapping* of the layout in the plane can be avoided by using the virtual wharf mark technique. For example, Fig. 8 shows how the overlap between vessel 2 and 4 (after wrapping around) can be eliminated through the introduction of a virtual wharf mark w, with appropriately chosen lower bound on the berthing location, and arrival time (in this case, T) for w. In Fig. 8, vessel 4 can be kept propped by maintaining $(4, w)$ and $(w, 4)$ in H and V sequences, respectively.

In general, obtaining the appropriate wharf mark set and their associations with the vessels is difficult, as it changes with the sequence pair. Furthermore, a large number of wharf marks may be required to prop all the affected vessels to appropriate height in the layout. Obtaining the set of virtual wharf marks and appropriate lower bounds in the optimal solution is nontrivial, and our strategy is to introduce VW as and when required. However, the dynamically introduced VW aids in enlarging our search space, thereby allowing us to search over a larger neighborhood. More details on implementation involving the virtual wharf mark is presented in "Searching over the sequence pairs."

Because the berth template is tactical in nature, there may be instances where overlaps (in the template) cannot be avoided. These are instances where many vessels need to be serviced at the same time according to the vessel performa. To take care of this issue, we can also allow for vessel overlaps in the packing and try to minimize them by adding a large penalty term to the objective function.

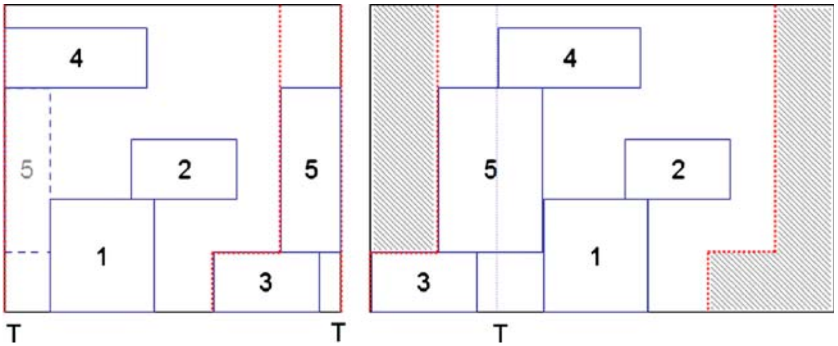

Fig. 7 Unwrapping a feasible template

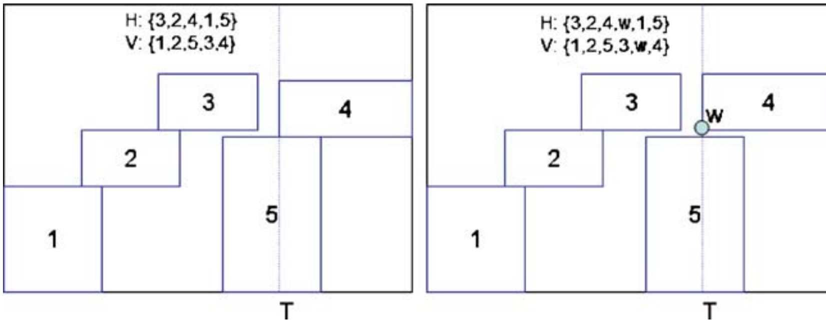

Fig. 8 A virtual wharf mark can be used to add constraints for propping vessels as shown

4 Searching over the sequence pairs

Searching for the optimal sequence pair is nontrivial, and we use a simulated annealing-based local search procedure similar to the one proposed in Dai et al. (unpublished manuscript, 2004). We employ the standard swap and shift neighborhoods and a modification of the greedy neighborhood presented in Dai et al. The greedy neighbor simply represents all the templates that can be obtained by visually manipulating the vessel locations while keeping the berthing time fixed. While manipulating the location, we ensure that the vessels do not overlap even after wrapping the template.

The critical aspect for getting good solutions is to define an appropriate cooling schedule and a sufficiently large neighborhood that can be explored efficiently. With regard to the latter, we use the following standard structures (cf. Dai et al., unpublished manuscript, 2004):

(a) *Single swap* This is obtained by selecting two vessels and swapping them in the sequence by interchanging their positions. Single swap is defined when the swap operation is performed in either H or V sequence.
(b) *Double swap* Double swap neighborhood is obtained by selecting two vessels and swapping them in both H and V sequences.
(c) *Single shift* This neighborhood is obtained by selecting two vessels and sliding one vessel along the sequence until the relative positions are changed; i.e., if $i, j, \ldots, k, l$ is a subsequence, a shift operation involving i and l could transform the subsequence to $j, \ldots, k, l, i$. There are many variants of this operation depending on whether vessel i (or l) is shifted to the left or right of vessel l (or i). We define single shift as a shift operation along one of the sequences.
(d) *Double shift* This defines the neighborhood obtained by shifting along both H and V sequences.

Figure 9 shows examples of the above operations and their impact on the packing obtained. The above neighborhoods described are simple perturbations of the sequence pair, but they result in remarkably different packing when compared visually. Our next neighborhood structure, however, is obtained using visual manipulation of the rectangles in the packing.

(e) *Greedy neighborhood* Given a sequence pair H and V and the associated packing $\mathcal{P}$, we evaluate all possible locations that vessel i can take, with the

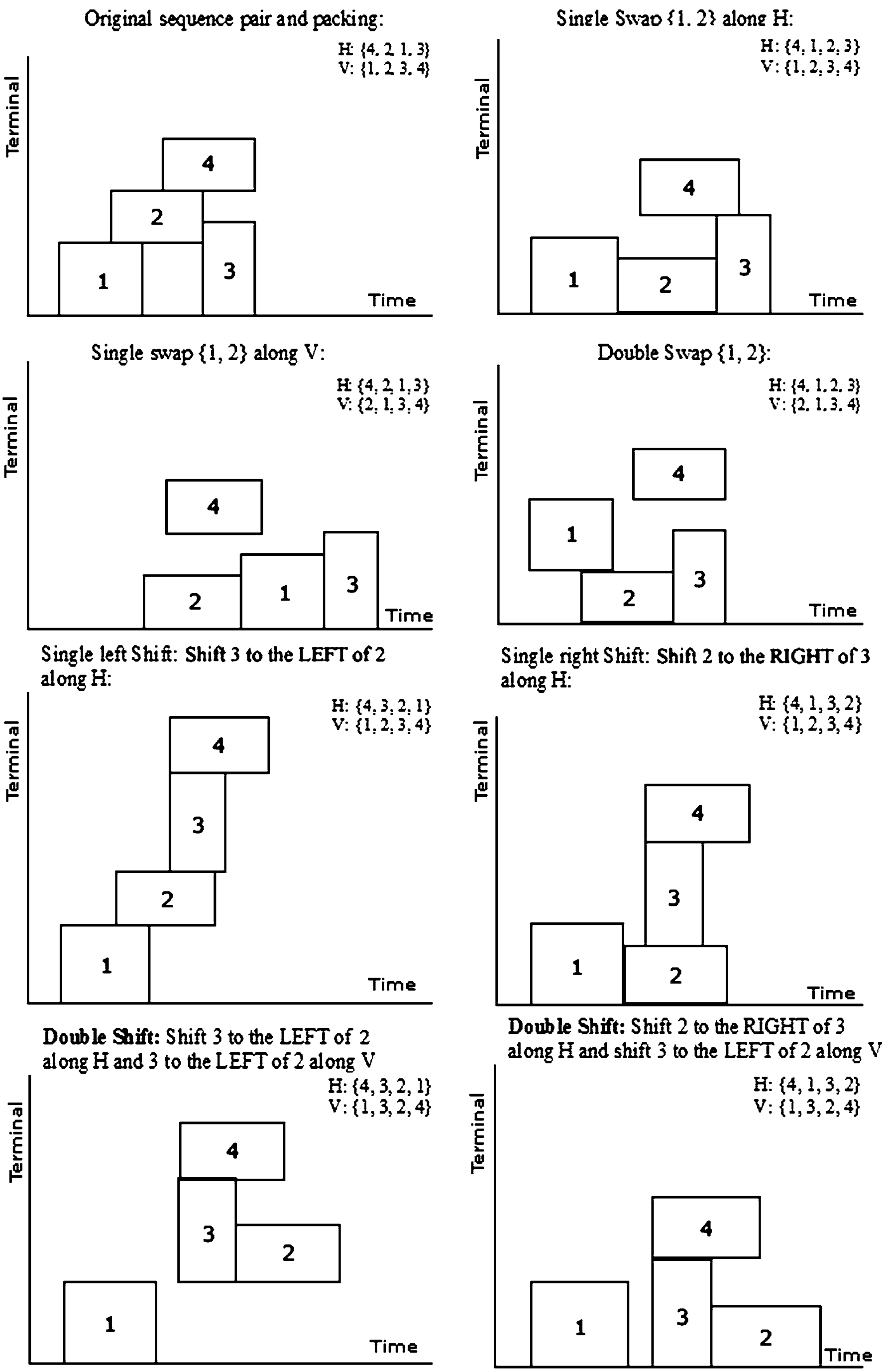

Fig. 9 Examples of swap and shift neighborhoods

rest of the vessels fixed in their respective positions. If there is a better location for the vessel, then we set the berth location of i to its new location. Note that because the time is kept constant, it is easy to check whether there is an overlap along the space dimension. Once the vessel is placed in the new location, we repeat the procedure for the rest of the vessels, until no improvement is possible.

Figure 10 shows the packing and the corresponding sequence pair obtained from a simple greedy neighborhood. The greedy neighborhood artificially modifies the position of the vessel along the space dimension, while keeping the berthing time decision fixed. Once we find the best berthing location for a vessel and obtain the corresponding sequence pair, we can then proceed to the next iteration of the simulated annealing algorithm. However, because we confine ourselves to packing where the berthing location decisions are obtained via a greedy manner (cf. Eq. 4), we need to ensure that the packing obtained from the greedy neighborhood exploration is suitably propped in the packing with additional VWs. New VWs are thus dynamically added and dropped from the search procedure as we explore the various neighborhoods in the simulated annealing algorithm.

Implementation incorporating virtual wharf mark The problem in adding virtual wharf marks as additional vessels in the search space is that it increases the problem size and, hence, the computation time. Here, we propose a cost-effective way of implementing the approach by employing dynamic lower bounds.

Note that the vessels need to be propped after we employ the greedy neighborhood. Instead of adding virtual wharf marks, the idea is to (dynamically) set lower bounds for the berthing location for those vessels that need to be propped by a virtual wharf mark in the packing.

We retain these lower bounds while exploring the neighborhood using operators (a)–(d) and change the lower bounds only when operator (e) changes the packing and introduces new virtual wharf marks. Searching the greedy neighborhood is computationally more expensive than simple sequence pair manipulation; hence, for the experiments, the operators (a)–(d) are employed successively, while the operator (e) is used whenever the other operators get stuck in a local optima.

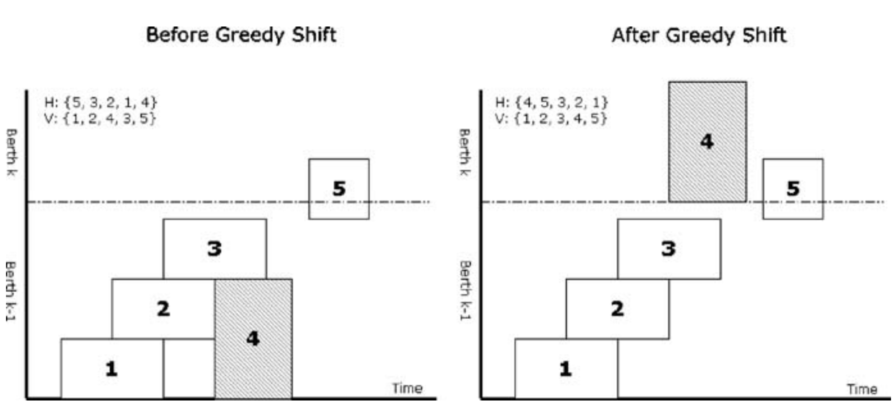

Fig. 10 An example for greedy neighbor

The dynamic lower bounding technique described above is equivalent to adding a virtual wharf mark w_k to i (with w_k coming immediately after i in the H sequence, and w_k immediately before i in the V sequence) and performing all neighborhood searches treating iw_k in H and w_ki in V, as *virtual* vessel. Note that swapping or shifting iw_k with j in H or swapping or shifting w_ki with j in V has the same effect of swapping or shifting i with j in the original H sequence, but maintaining a lower bound (determined by virtual wharf mark w_k) on vessel i in the neighborhood.

5 Computational and simulation results

In the previous section, we described a procedure where good home berth template can be constructed, using the sequence pair manipulation. In this section, we evaluate the usefulness of this approach, by simulating the performance of the template using a dynamic berth allocation package developed recently by Dai et al. (unpublished manuscript, 2004). The simulation allows us to track the berthing performance of the vessels over several months. We use the BOA statistics (percentage of vessels berthed on arrival) and the connectivity cost (based on actual berthing location of the vessels in the simulation) to construct the efficient frontier for the template.

Throughout our simulation, we use the following observation to create realistic input parameters for our model: In a typical port, the connectivity profile normally exhibits the following feature. On the average, each vessel exchanges containers with about 30% of the rest of the vessels calling at the port. Fifteen percent of these container exchanges are in the range $[0, 100]$, and another 15% in the range $[100, 1000]$. We use $d(x_i, x_j) = |x_i - x_j|$ in our experiment.

To simplify our simulation, we also make the simplifying assumption that port stay time is deterministic, and we focus on the importance of capturing the variability of the arrival time of the vessels at the port. The variability of the port stay time is within the control of the terminal management and can be controlled or influenced by deploying appropriate amount of resources. The vessels' arrival time, however, is not within the control of the terminal; hence, it is deemed to play a more important role in the berth management process.

The general simulation environment for the experiments is as follows:

- We use a data set that represents a vessel arrival pattern at a typical port. The expected arrival time of vessels is the same as used in the template. It is assumed that the berth planner does not get any updates on the vessel arrival time between the time the template is drawn and when the actual deployment is done.
- Based on the vessel location in the template, a stepwise constant space cost is generated for the problem. We call the berth that the vessel is expected to be moored in (according to template) as the *preferred berth*. The space cost is considered to be constant within each berth in the terminal. The cost of allocating a vessel in any berth other than the preferred berth is set to be proportional to the distance between the two berths.
- During dynamic deployment, the berth planner plans the berth allocation to a set of vessels arriving within a scheduling window. Of course, as time rolls by, the

scheduling window rolls forward too, hence, including newer vessel arrivals. For the experiments, we set the scheduling window to *48 h*.

– To prevent last-minute reshuffling of resources, the berth plan is frozen for two shifts (i.e., 16 h) from the current time. Any changes in the vessel's arrival time will be accommodated within this window, but the vessel's berth location will remain fixed. This rule is of practical importance for the port operator as it eases the resource bottlenecks.
– We assume that the vessels update their exact arrival time around *8 h* from the current time. For sensible comparison, we generate the actual arrival information (from a normal distribution with mean r_i , and a small variance) beforehand for all the experiments.
– We compute the container movement cost between vessels once the simulation is over. In this step, we assume that only the vessels within the same period are *connected*. Specifically, for a vessel berthed at time t_i' , we assume it to be *connected* only o the vessels berthed between $t_i' - \frac{P}{2}$ and $t_i' + \frac{P}{2}$.
– In dynamic deployment, we consider a vessel to be berthed-on-arrival if it is berthed earlier than $(\hat{r}_i + 2)$ hours. Here, $\hat{r}_i$ is the actual arrival time. BOA directly corresponds to the service levels that a port guarantees to the vessels, and maximizing BOA is a prime concern during deployment.
– The statistics on the percentage of vessels allocated to the preferred berth is collected. This is a measure of the amount of replanning the berth planner has to do when using a template.

5.1 The impact of variability on berthing performance

In practice, the home berth template is usually designed assuming that the vessels' time-of-call is precise and does not fluctuate from the scheduled time of arrival. Furthermore, the planners normally try to design a layout with as few overlaps as

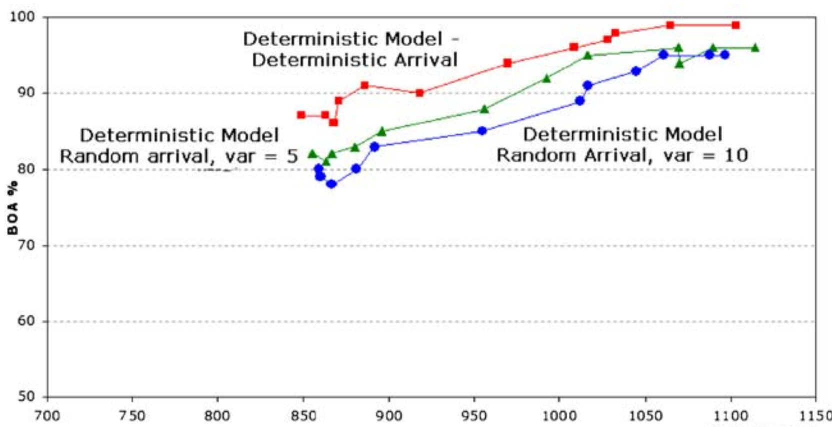

Fig. 11 Plot shows a steady drop in performance as variance in arrival data is increased

possible. We can incorporate these considerations in our home berth model, using $\sigma^2(r_i) = 0$ and a huge, lumpy cost to penalize for overlaps in our template.

Using the template obtained this way, Figure 11 shows the efficient frontiers obtained under three different scenarios: (1) $\sigma^2(r_i) = 0$, (2) $\sigma^2(r_i) = 5$, and (3) $\sigma^2(r_i) = 10$. As expected, the efficient frontier drops steadily from case 1 to 3, as the variability of the arrival data slowly erodes the performance of the home berth template.

5.2 Evaluating the template obtained from the robust model

We design another template, using the robust model outlined in the previous sections. Instead of using lumpy cost to minimize the number of overlapping vessels in the layout, we use the connectivity and delay cost estimation methods proposed in this paper to distinguish between templates. By varying the weights on the connectivity and waiting time cost components, we actually obtain different home berth templates. The planners can then choose the right template to balance the two objective functions in berthing operations.

Sensitivity to parameters Figure 12 shows the variation of connectivity and waiting time cost components (obtained from our model) when the weights of delays (time penalty), with respect to connectivity cost, are increased steadily. We benchmarked against the template obtained using a deterministic approach, i.e., ignoring the information on the variance of arrival time data.

The templates obtained from the stochastic model performs consistently well in terms of waiting time cost component for all time penalty weighing factors. By increasing the weights for time penalty, the model is able to find new templates with slightly better waiting time performance, but at the expense of a large increase in connectivity cost.

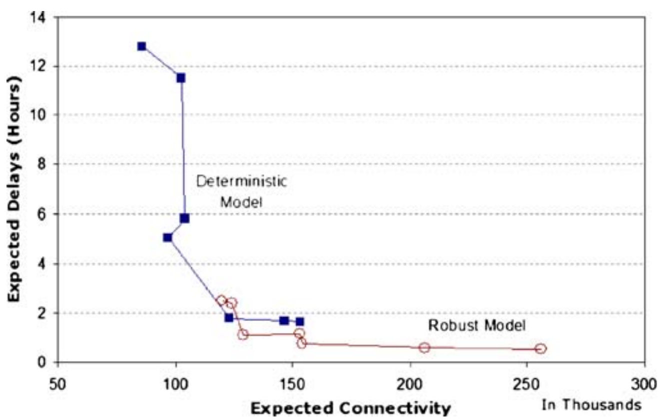

Fig. 12 Comparing robust templates with varying penalties on time. For similar connectivity cost, one can expect to have smaller expected delays while using a robust measure for the waiting time objective

The templates obtained from the deterministic model, however, is very sensitive to the weights used for the time penalty cost function. This suggests that it is very important that the parameters should be set correctly to ensure desirable results from the proposed model. This behavior arises, we believe, because the deterministic model is not able to capture the effect of delays propagation on the rest of the vessels.

Resource conflicts Ideally, the home berth template should have a minimal number of overlap between vessels. This is important because too many overlaps (though quantified in the form of delays) would eventually make the real time vessel deployment problem tough. Figure 13 shows the overlaps obtained from templates produced from the robust model, while we vary the weights on the time penalty cost function.

The robust model, as it is seen, is successful in keeping the number of overlaps down. This is an inherent quality of the model rather than an exception since, while evaluating the waiting time objective using a robust measure, the concerns posed by overlaps are captured implicitly in the waiting time estimation in the model.

Performance of the robust model in simulation Figure 14 shows an efficient frontier comparison for *good* templates. For the sake of fair comparison, we choose the home berth plan that balances the connectivity cost and expected delay, one from each model. The results indicate that the robust model is the better choice. One could choose a minimum service level expected and could pick a template that can achieve the service levels with smaller connectivity cost. For example, the minimum connectivity cost that can be achieved with a service level of 90% is 836,446 and is possible using the robust template.

If one compares the delays experienced by an individual vessel, another inherent property of the robust model is revealed. In the presence of traffic congestion, the robust model would ensure that vessels are distributed so as to minimize the delay propagation. The deterministic model fails to employ the information from the immediate past. This means that specific vessels may suffer recurring high delays when the deterministic home berth plan is used. This observation is corroborated by Fig. 15. The delay experienced by individual vessels fluctuates widely while using the deterministic measure for waiting cost objective, whereas the fluctuation is more controlled while employing the robust measure in evaluating a home berth template. The average delays in the

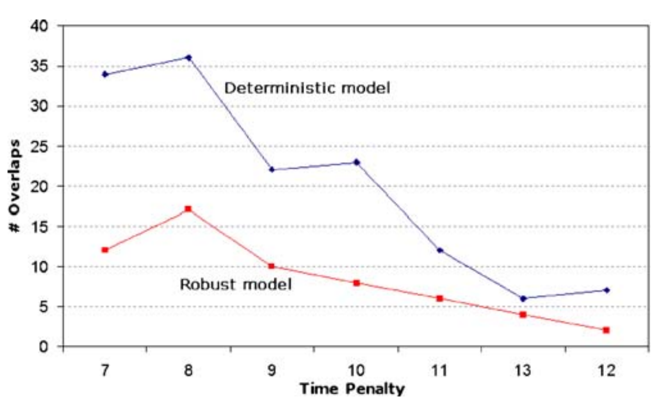

Fig. 13 Variation of overlaps vs weights in time penalty function

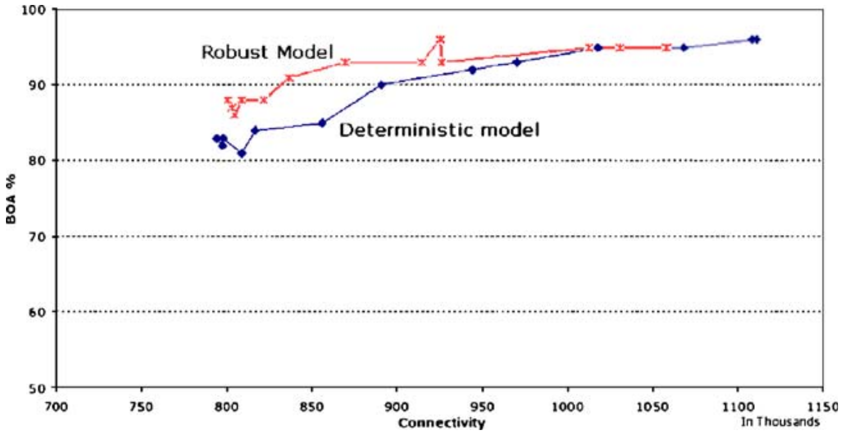

Fig. 14 Efficient frontiers for "good" templates

deterministic model is 1.65 h with a variance of 2.75, whereas the average delay in the robust model is 0.75 h with a variance of 1.13. Furthermore, 27 vessels in the robust model's template have an expected delay of 0 h, as opposed to 13 vessels in the deterministic model.

5.3 Comparison to "optimal" template

It is desirable to obtain an absolute measure comparing the solution produced by the robust model to the optimal template or with respect to a tight bound, as it is important to understand how good the proposed robust model performs. To address this concern, we create an artificial problem instance wherein a real-life berth template is chosen and the connectivity data are modified so that the given template is near optimal (i.e., vessels in close proximity will have higher number of container exchanges). Vessels in the same berth have high connectivity (1,000), vessels in adjacent berths have low connectivity (300), and the connectivity is 0 for the rest.

We did not modify the arrival time information or adjust the layout of the template. As the template has been deployed in practice, the planners must have already visually inspected the layout and are satisfied that all potential bottlenecks have been resolved from the template. Using these data, we create another home berth template using the robust model.

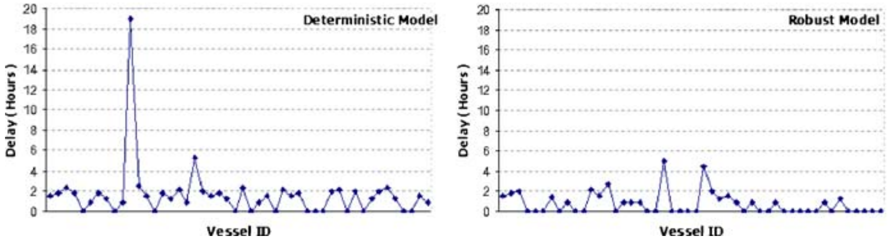

Fig. 15 Expected delays for each vessel based on the templates

The connectivity optimal template and the robust template are as shown in Fig. 16. The respective efficient frontiers are as shown in Fig. 17. We observe that even in this scenario, the efficient frontier obtained from the robust model slightly outperforms the optimal template. The results are promising and one can expect to obtain better dynamic performance in the actual deployment of the robust model.

6 Conclusion and extensions

In this paper, we have addressed an important tactical planning problem in container terminal management, the home berth allocation. The problem is modelled as a bicriteria optimization problem. We study and provide a framework to deal with the trade-off between the operational cost and service levels demanded by customers.

The combinatorial nature of the home berth problem poses a primary challenge and we address it by modelling the problem as one of packing rectangles on a cylinder. We motivate the use of sequence pair for defining the search space and provide a series of extensions to adapt the approach for the problem presented herein. The sequence pair approach naturally decomposes the home berth problem into a space and time subproblem, and we use this insight to address the temporal problem with random inputs. Because the home berth problem is tactical in scope, the temporal data, specifically the vessel arrival time, is stochastic. To address this inherent randomness, we borrow techniques from stochastic project scheduling and explicitly derive an expression for expected delays.

In brief, we address both the combinatorial and stochastic nature of the problem in our proposed framework and create robust home berth allocations, which translates to better service levels and better resource management during actual operations.

In addition, the paper provides extensive computational experiments and simulations to analyze the effect of solving the problem in the robust home berth allocation framework. The results show that the model cannot only effectively

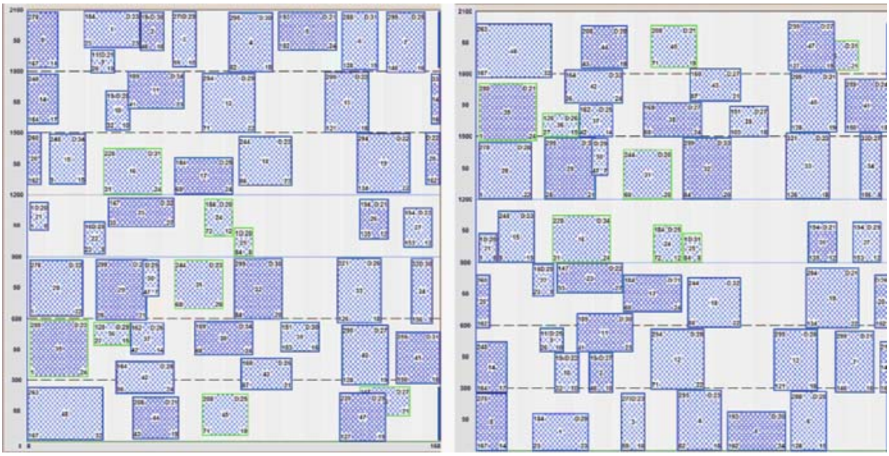

Fig. 16 A tale of two templates

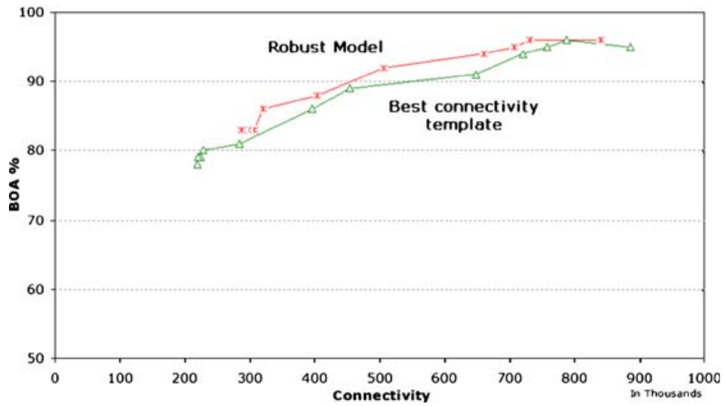

Fig. 17 Efficient frontier analysis shows that the robust model outperforms the "optimal" template

measure the dynamic impact of the randomness but also has other advantages like minimizing overlaps in the template.

To the best of our knowledge, this is the first paper that solves a berth template problem and analyzes the impact of the template on the real time berth allocation. However, our approach is limited by the fact that the template is relevant only when a substantial number of vessels arrive periodically and within the same period. An extension of the problem to create a template for vessels with different periods would be interesting and we leave it for future research. We do not address the problem of crane allocation in this paper. We leave this problem as an extension and note that the crane allocation decision affects the port stay of vessels, and it requires a more sophisticated model and careful study to analyze its impact on the template and during dynamic deployment.

Recent trends in port management has been toward making the operations flexible. This means that megaports would have to frequently redraw customer contracts and appropriately change internal operations viz, yard plans. In such a situation, flexibility in drawing reliable home berth allocation becomes ever more important. We hope that with the model described in the paper, a berth operator can create berth templates that can then be follow with minor modifications during actual deployment.

References

Brown GG, Lawphongpanich S, Thurman KP (1994) Optimizing ship berthing. Nav Res Logist 41:1–15

Chen CY, Hsieh TW (1999) A time–space network model for the berth allocation problem. In: 19th IFIP TC7 conference on system modeling and optimization, Cambridge, England, 12–16 July 1999

Imahori S, Yagiura M, Ibaraki T (2003) Local search algorithms for the rectangle packing problem with general spatial costs. Math Program Ser B 97:543–569

Lim A (1998) On the ship berthing problem. Oper Res Lett 22(2–3):105–110

Murata H, Fujiyoshi K, Nakatake S, Kajitani Y (1996) VLSI module placement based on rectangle packing by the sequence pair. IEEE Trans Comput-Aided Des Integr Circuits Syst 15(12):1518–1524

Murata H, Fujiyoshi K, Kaneko M (1998). VLSI/PCB placement with obstacles based on sequence pair. IEEE Trans Comput-Aided Des Integr Circuits Syst 17:60–68
Murty KG, Liu JY, Wan YW, Linn R (2005) A decision support system for operations in a container terminal. Decis Support Syst 39(3):309–332
Tang X, Tian R, Wong DF (2000) Fast evaluation of sequence pair in block placement by longest common subsequence computation. In: Proceedings of the conference on design,, automation and test in Europe, pp 106–111