



タイトル Title	Berth allocation in a container port : using a continuous location space approach
著者 Author(s)	Imai, Akio / Sun, Xin / Nishimura, Etsuko / Papadimitriou, Stratos
掲載誌・巻号・ページ Citation	Transportation Research Part B: Methodological,39(3):199-221
刊行日 Issue date	2005-03
資源タイプ Resource Type	Journal Article / 学術雑誌論文
版区分 Resource Version	author
権利 Rights	
DOI	10.1016/j.trb.2004.04.004
JaLCDOI	
URL	http://www.lib.kobe-u.ac.jp/handle_kernel/90000056

Berth allocation in a container port: using a continuous location space approach

Akio IMAI

Faculty of Maritime Sciences

Kobe University

Fukae, Higashinada, Kobe 658-0022 Japan

&

World Maritime University, PO Box 500, S-201 24 Malmö, Sweden

Xin SUN, Etsuko NISHIMURA

Faculty of Maritime Sciences

Kobe University

Fukae, Higashinada, Kobe 658-0022 Japan

and

Stratos PAPADIMITRIOU

Department of Maritime Studies, University of Piraeus

80 Karaoli & Dimitriou Str., GR185 32 Piraeus, Greece

ABSTRACT

This paper addresses the berth allocation problem in a multi-user container terminal. There are two typical schemes for berth allocation: one in discrete locations and the other in continuous locations. The former has the advantage of easiness in scheduling but it has a weakness in that terminal usage is not fully efficient. The latter exhibits the complete opposite characteristics. In previous papers, the authors have developed and presented the discrete location version of the berth allocation problem. In view of the steadily growing trend in increasing the container ship size, more flexible berth allocation planning is mandatory, especially in busy hub ports where ships of various sizes are calling. In this paper, a heuristic for the berth allocation problem in continuous locations is presented. A wide variety of experiments were conducted and the results showed that the heuristic works well from a practical point of view.

Keywords: Berth allocation; Terminal management; Container transportation;
Cargo handling; Heuristic; Mathematical programming

1. Introduction

Major container ports in Japan and the US such as Kobe, Yokohama, Los Angeles and Oakland feature terminals leased on a long term basis by shipping operators, which in the literature are referred to as *Dedicated Terminals*. However, container ports in Europe and China furnish the so-called *Multi-User Terminals*, which may be defined as terminals with a long quay where a number of incoming vessels are simultaneously and dynamically allocated to the quay and are not always assigned to specific, same quay locations whenever they call. The Multi-User Terminal is a widespread system in use, especially in busy container ports with heavy container traffic such as Hamburg, Rotterdam and Antwerp in Europe, and Busan, Shanghai and Hong Kong in East Asia. The productivity of these busy container terminals depends highly on the terminal handling system, as well as on the efficient berth allocation of calling vessels. For managerial purposes, the entire quay in a Multi-User Terminal is partitioned into several berths, and the allocation of the ships to the proper quay locations is based on the berth. Throughout this paper, we refer to the problem of allocating ships to quay locations (or berths) as the *Berth Allocation Problem (BAP)*. The problem associated with this allocation scheme based on the berth is, hereafter, referred to as the *Discrete Berth Allocation Problem* or *BAPD*. As there is a broad diversity in ship size - we are mainly concerned with ship length - in real practice ships are sometimes berthed across the berth boundary for enhancing the efficient berth usage.

The authors have previously examined efficient berth scheduling for the Multi-User Terminal in Imai et al. (1997, 2001, 2003) and Nishimura et al. (2001). Although these studies treat the BAP in different settings, they all define it in a discrete quay location space. This paper is motivated by the fact that ships are sometimes moored across the berth boundary even though in most container terminals of Japan the

BAP in practice is treated based on the berth. Thus, as the ship's berthing is actually performed in a continuous location space, the BAP should be solved as such as well, that is without using the berth (hereafter, referred to as the *Continuous Berth Allocation Problem* or *BAPC*). In this study, we address the BAPC with the objective of minimizing the total service time where the ship's handling time depends on the ship's berthing location in the quay.

This paper is organized as follows. The next section carries out an extensive literature review of the existing studies on the BAP and related issues. The problem formulation is defined in Section 3, followed by Section 4 which describes a heuristic algorithm. In Section 5 a large scale of numerical experiments are performed and in Section 6 the paper's conclusions are presented.

2. Literature review

There are few studies dealing with berth allocation. The BAPD is investigated by most of these studies such as Lai and Shih (1992); Brown et al. (1994, 1997); Imai et al. (1997, 2001, 2003), and Nishimura et al. (2001). On the other hand, the BAPC was tackled by Lim (1998), Li et al. (1998), Guan et al. (2002), Park and Kim (2002, 2003), and Kim and Moon (2003).

This study has a close relationship with a study of the BAPD in Imai et al. (2001) since the solution algorithm for this study is based on their BAPD. Their study minimizes the ship's total service time (including the ship's waiting time for berth availability and the ship's handling time) with the assumption that the handling time depends on the quay location where the ship is moored. It introduces two types of BAPs: the static and the dynamic. The static version treats only ships that have already

arrived at the port before the scheduling begins, whilst the dynamic one takes into account ships that have already arrived as well as those that have not arrived at the time of planning and will arrive at some later moment during the planning horizon. It is noted that in some scheduling studies, the term “dynamic” is used to refer to problems treating some events with unknown time of their occurrence. Throughout this paper though, we use the term “dynamic” with the meaning described earlier, in order to preserve the consistency with the existing dynamic BAP studies.

For the BAPD, Lai and Shih (1992) propose a heuristic algorithm for berth allocation, which is motivated by more efficient terminal usage in the HIT (Hong Kong International Terminal) of Hong Kong. Their problem considers a first-come-first-served (FCFS) rule. Brown et al. (1994, 1997) examine ship handling in naval ports. They identify the optimal set of ship-to-berth assignments that maximize the sum of benefits for ships while in port. Imai et al. (1997) develop a heuristic algorithm for the static BAP with two objectives: the minimization of the service time and the minimization of the dissatisfaction of ships with the order of service. They do not take into account the FCFS rule as they aim to maximize the productivity of the terminal. Imai et al. (2001), as mentioned before, introduce both static and dynamic BAPs. They developed a heuristic based on the Lagrangian relaxation. Nishimura et al. (2001) extend the dynamic BAP of Imai et al. to treat some physical restrictions on berthing ships such as water depth and berth length. They employ the genetic algorithm in order to find the approximate solution for the problem. Imai et al. (2003) introduce the service priority of the ship in the dynamic berth allocation circumstances. They first attempt to utilize the Lagrangian relaxation; however, they find that the relaxed problem is reduced to the Quadratic Assignment Problem, which is difficult to solve in the polynomially bounded computation time. Consequently they apply the genetic algorithm for their problem.

Lim (1998) addresses a BAPC with the objective of minimizing the maximum amount of quay space used at any time with the assumption that once a ship is berthed, it will not be moved to any place else along the quay before it departs. He also assumes that every ship is berthed as soon as it arrives at the port. On the other hand, Li et al. (1998) solve a BAPC both with and without the ship's movement restriction. Their objective is to minimize the makespan of the schedule. Guan et al. (2002) developed a heuristic for a BAPC with an objective function that minimizes the total weighted completion time of ship services. Park and Kim (2002) studied a BAPC with an objective that minimizes the costs of delayed departures of ships due to the undesirable service order and those of additional complexity in handling containers when ships are served at non-optimal mooring locations in port. Their work is more practical than the aforementioned BAPC studies in that factors assessed in the objective depend on the quay locations of ships. Kim and Moon (2003) address the same BAPC as the one tackled by Park and Kim (2002), though the former employs the simulated annealing method while the latter applies the subgradient optimization method. We note that all of the aforementioned BAPC studies assume the ship's handling time being independent from the berthing location, which is not the case in our study. Park and Kim (2003) studied a BAPC with a similar objective to those of Park and Kim (2002), and Kim and Moon (2003). The difference in objective is that in addition to the costs considered in previous studies, Park and Kim (2003) take into account the cost resulting from early or late start of ship handling against the estimated times of ship arrival. Their study features an interesting characteristic in that they determine the optimal start times of ship services and associated mooring locations and at the same time they determine the optimal assignment of quay cranes to the ships. In their study, the handling time of a particular ship is a function of the number of quay cranes engaged in the ship; however, the handling time is independent from the mooring location of the ship. In summary, all

the existing BAPC studies assume unchanged handling times regardless of where the ships are handled.

The BAPC has enormous flexibility for the berth allocation, achieving higher efficiency in berth usage and productivity than the discrete one. However, this advantage is offset by the difficulty in solving the problem due to its complexity.

In the BAPC, a scheduling task is geometrically presented as the following problem: given a number of ships under handling being defined as the rectangles and the berth availability being defined as a big box (or a large confined space), how efficiently the rectangles are packed or located in the box. In this representation as shown in Fig. 1, the horizontal edges of the rectangles and the box correspond to the physical length of ships and quay, respectively, while the vertical edge corresponds to the lay time of ships and the planning time required.

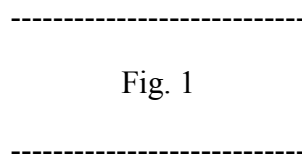


Fig. 1

We can, therefore, imply that there is a similarity between the BAPC and the two dimensional cutting stock problem. The cutting stock problem (CSP) has to do with cutting some form of stock materials to produce smaller pieces of materials in quantities matching those of the orders received, aiming at minimizing trim loss and consequently maximizing profit. Most of the CSPs deal with cutting materials (such as roll paper, wood, glass, etc.) into small *rectangular* pieces. Some consider producing *non-rectangular* pieces. The CSP, hereafter, refers to the rectangular version of the CSP. In one type of the CSP, all pieces have fixed orientation, i.e., a piece of length d and width w is different from a piece of length w and width d . The other type does not impose such an orientation restriction. As the CSP is NP-hard, most studies are concerned with heuristic procedures. Two schemes exist in CSP heuristics: guillotine

cut and non-guillotine cut. The guillotine-cutting pattern, shown in Fig. 2(a), requires that any cut made upon a rectangle must be in a straight line from one edge of the space to the opposite edge. The non-guillotine-cutting pattern is a scheme that permits cuts to be made in a flexible way as shown in Fig. 2(b), thus more space usage is feasible.

Fig. 2

The problem setting of the BAPC is equivalent to the CSP of fixed orientation with the only exception that while all the rectangles (or items) are already available in the CSP, some of them are not yet available in the BAPC. The constraint in the BAPC could be graphically portrayed, in Fig. 1, in that some rectangles cannot be placed below respective predetermined horizontal lines. This restriction simply arises from the fact that ships cannot be berthed before they have arrived at the port. As easily deduced, the static version of the BAPC is, in terms of the problem setting, completely identical to the CSP of fixed orientation.

The BAPCs in Lim (1998), Li et al. (1998) and Guan et al. (2002) that assume the static berth allocation and the handling time independent from the berthing location are equivalent to the CSP with fixed orientation of placing items, whilst studies of Park and Kim (2002, 2003), and Kim and Moon (2003) are based on the dynamic nature of ship arrival with the fixed handling time assumed. However, in a strict sense, the study of Park and Kim (2003) is not a dynamic BAP, since it allows ships to be served earlier than the respective estimated time of arrival, although some penalty is imposed for it in the objective. This treatment policy substantially reduces it to a static problem, because ships can be served any time as long as quay spaces are available. Consequently its constraint is also equivalent to that of the CSP with fixed orientation. On the other hand, the BAPC in this study is quite different from the conventional CSP because of the

varying handling time based on the ship's location as well as the dynamic nature of ship arrival.

3. Problem formulation

3.1. Assumptions

In the sections that follow, we will refer to the dynamic BAPC and dynamic BAPD simply as the BAPC and BAPD. The BAPC assumes that the ships are not moved until their departure time as in Lim (1998), since such ship movements occur very rarely in busy container terminals. In addition to this, the BAPC makes the same assumptions as those made for the BAPD in Imai et al. (2001).

The first assumption is that a ship's handling time depends on the quay location where the particular ship is handled. More precisely, it is assumed that the handling time is defined by the physical relationship between ship's quay location and its container storage location in the yard. The handling time may remain unaffected regardless of where a ship is moored and where its relevant container storage location is, if every ship employs a sufficient number of yard trailers (or straddle carriers) that haul containers between the ship and storage, resulting in no interruption or delay of the quay crane cycle. This is only possible if there is a very large fleet of yard trailers available to cover simultaneously all the ships in the terminal. Such a plan, however, turns out to be very costly as there is a considerable redundant fleet when the terminal is not so busy or when the ships are moored properly nearby their container storage location even when the terminal is at a busy state. Based on this consideration, we assume that due to the limited size of trailer fleet, every ship does not necessarily engage a trailer fleet large enough to keep seamless movements of cranes, especially when the ship is moored far

away from the containers in the yard. This justifies our assumption that the ship's handling time is dependent on its quay location.

In practice, the handling time also depends on the number of quay cranes engaged in handling the ship. Usually the particular number of quay cranes that are assigned to a ship depends on the size of the ship and the container movements to be made. An exception to the rule is made in the occurrence of a late arrival of a ship requiring a quick turnaround, in which case many more cranes are assigned to it. As referred earlier, Park and Kim (2003) determine the quay crane assignment and the berth assignment simultaneously, assuming that the handling time depends only on the number of cranes employed. Their study treats the varying handling time with different number of quay cranes; however it does not explicitly assume that the handling time is dependent on the number of cranes and the quay location.

The second assumption that is made is the dynamic nature of ship arrival as already mentioned. Note that this study does not assume any committed departure time of ship. In reality, if a ship cannot depart before the committed time, some penalty is incurred to the terminal operator. This study focuses on the total efficiency of ship handling, ignoring committed departure time violations. However, as we minimize the delay in ship departure by serving ships properly, the number of cases of violated committed departure time is implicitly minimized.

3.2. Formulation

As illustrated in Fig. 1, we can represent a ship geometrically by a rectangle, such that the length of the rectangle is the length of the ship and the height of the rectangle is the duration of its stay (or handling time). The bottom edge of the rectangle represents the start time of handling the ship. The top edge represents the completion time of handling the ship (or the departure time of the ship). The quay space can be

represented geometrically by an infinitely long rectangle where its length is the quay length and the height defines the time.

The formulation is based on the following notations:

- $i (= 1, \dots, T) \in V$: set of ships
 A_i : estimated time of arrival of ship i
 L_i : length of ship i
 Q : quay length
 M_i : quay location with the minimum handling time of ship i
 C_i : handling time of ship i
 p_i : quay location of ship i
 t_i^B : start time of handling for ship i
 $t_i^F (= t_i^B + C_i)$: completion time of handling for ship i

where p_i s and t_i^B s are variables and integer.

There may be some uncertainty on A_i , because ships are sometimes delayed in arrival due to the delay in cargo handling in previously called ports, etc. However, we treat the BAPC as a deterministic problem with the assumption that no delay takes place in ship arrival.

As assumed in the previous section, the handling time increases proportionally to the distance deviation from the best quay location. We assume a straight-line increase of the handling time with a slope of α_i . We associate a coordinate with the center of the bottom edge of the geometric representation of each ship. The handling time C_i is defined by:

$$C_i = CM_i + |p_i - M_i| \tan \alpha_i \quad (1)$$

where CM_i is the handling time at the best quay location for ship i and $\tan \alpha_i$ is defined as a ratio of increasing handling time to the distance for ship i . We assume that the berthing location is measured with the unit length that makes a significant difference

in the handling time. It is a matter of discussion on whether this linear increase of the handling time is reasonable or not. The correlation between the handling time and quay location (actually as well as the related container storage location in the yard and the number of quay cranes engaged in handling the ship) must be identified by an extensive analysis of the container handling operations in the yard. This is, however, beyond the scope of this research. Even though the correlation is found to be non-linear by such an analysis, it does not have a major effect on the solution procedure of the BAPC as described in Section 4.

Ships are allowed to overlap with each other at least either in time axis or in quay axis. When they are located far apart from each other both in time and quay axes, non-overlapping criteria both in time and quay axes are satisfied; however, we note that the validity of this case is guaranteed by one of the criteria being satisfied. Consequently we have the following non-overlapping constraint for particular ships i and j :

$$|p_i - p_j| \geq \frac{L_i + L_j}{2} \quad \text{or} \quad \left| \frac{t_i^B + t_i^F}{2} - \frac{t_j^B + t_j^F}{2} \right| \geq \frac{C_i + C_j}{2} \quad (2)$$

Note that when two ships are berthed side by side, a certain minimum inter-ship clearance distance must be observed. Each ship has an individual inter-ship clearance distance. Physically speaking, this distance is a function of ship length and width, because some clearance is also needed for mooring lines to tie up a ship against the quay. For convenience, this distance is included in the ship length used in the formulation. Furthermore, a time clearance is also observed when a ship is berthed at the location where the previous ship was served. This clearance is included in the handling time of each ship.

The BAPC is to determine the (p, t) coordinates of boxes in the long rectangle of quay such that all boxes representing ships are non-overlapping and they are not to be

placed below respective predetermined horizontal lines, which represent the arrival times of the ships.

The BAPC may be formulated as follows:

$$[P] \quad \text{Minimize} \quad Z = \sum_{i \in V} (t_i^F - A_i) \quad (3)$$

$$\text{subject to} \quad |p_i - p_j| \delta_{ij}^p \geq \frac{L_i + L_j}{2} \delta_{ij}^p \quad \forall i, j (\neq i) \in V, \quad (4)$$

$$\left| \frac{t_i^B + t_i^F}{2} - \frac{t_j^B + t_j^F}{2} \right| \delta_{ij}^t \geq \frac{C_i + C_j}{2} \delta_{ij}^t \quad \forall i, j (\neq i) \in V, \quad (5)$$

$$\delta_{ij}^p + \delta_{ij}^t = 1 \quad \forall i, j (\neq i) \in V, \quad (6)$$

$$p_i - \frac{L_i}{2} \geq 0 \quad \forall i \in V, \quad (7)$$

$$p_i + \frac{L_i}{2} \leq Q \quad \forall i \in V, \quad (8)$$

$$t_i^B \geq \max(A_i, 0) \quad \forall i \in V, \quad (9)$$

$$p_i, t_i^B \geq 0 \text{ and are integer} \quad \forall i \in V, \quad (10)$$

$$\delta_{ij}^p, \delta_{ij}^t \in \{0, 1\} \quad \forall i, j (\neq i) \in V, \quad (11)$$

where

δ_{ij}^p :=1 if non-overlapping restriction in quay axis is applied for ships i and j ,
=0 otherwise.

δ_{ij}^t :=1 if non-overlapping restriction in time axis is applied for ships i and j ,
=0 otherwise.

The objective function (3) is the sum of service times for all ships, where the service time is defined as the time spent from arrival to departure including the waiting time for a quay space to become available. Constraint sets (4)-(6) are the non-overlapping restriction. Note that as either quay-non-overlapping or

time-non-overlapping should be satisfied, therefore $\delta_{ij}^p + \delta_{ij}^t = 1$ in constraints (6). Constraint sets (7) and (8) ensure that every ship must be berthed within the quay length. Constraint set (9) assures that the ships are berthed after their arrival.

Note that in practice most scheduling is done on a rolling horizon basis. In each planning horizon, we solve a BAPC assuming the entire quay space is available at time zero. In reality however, most of the times, the quay may still be needed to handle ships left from the previous horizon. That is, the quay may not be available at the beginning of the planning horizon. We can take this into consideration by making a minor change in the solution algorithm.

4. Heuristic algorithm

4.1. BAPD

The BAPC employs, for its solution, the heuristic algorithm of the BAPD; therefore, we briefly describe the latter problem.

The BAPD, which is referred to as the DBAP in Imai et al. (2001), is interpreted as follow: The entire quay space is partitioned into several blocks (or berths) by a specific length (hereafter referred to as a berth length). We solve the BAPD based on the number of partitioned berths. However, the BAPD itself does not explicitly take into account the entire quay length and the berth length, but with an implicit presumption that the berth length is in practice no shorter than the maximum ship length to be served. This nature of the problem implies that the relationship between the BAPD and BAPC corresponds to the one between the guillotine-cutting and non-guillotine-cutting versions of the CSP. Note that despite the fact that the BAPD could be defined without the above presumption, the resulting solution does not make

sense. This is because a big ship moored in a particular berth stretches over the neighboring berths and it may physically collide with neighboring ships. The formulation of BAPD is given in the Appendix A.

The formulation [P] in this study views ships as physical entities since it determines decision variables with respect to ship locations in both quay and time axes. On the other hand, the BAPD determines the ships' berths and order of service with decision variables x_{ijk} of formulation [PD] in the Appendix A. For the consistent discussion with the BAPC, we could assume that ships are moored in the center of the allocated berths in the BAPD. The ship locations in time axis are also determined based on the decision variables x_{ijk} so that any ships do not overlap each other in time axis and are not served before their arrival.

4.2. Upper and lower bounds of the BAPC

This section provides insights on the characteristics of the BAPC and BAPD.

As mentioned in Section 4.1, the BAPD solution could be envisaged as placing ships in the middle of their assigned berths regardless of whether the ships overlap with the neighboring ships or not. Then, we can state the following properties:

Property 1. *The optimal solution to the BAPD, when the berth length is set to the maximum ship length involved in the problem, gives the upper bound of the BAPC.*

Proof. When the berth length is identical to the maximum size of the ships involved, no violations occur in terms of the relationship between ship length and berth length. As shown in the Appendix A, the BAPD is a variation of the Assignment Problem. If the BAPD with this particular berth length follows the continuous location space, it may be formulated as follows:

$$[\text{PU}] \quad \text{Minimize} \quad Z \quad (3)$$

subject to (4)-(11)

$$\sum_m \left(D_m + \frac{L_i}{2} \right) x_{im} \leq p_i \leq \sum_m \left(D_{m+1} - \frac{L_i}{2} \right) x_{im} \quad \forall i \in V, \quad (12)$$

$$\sum_m x_{im} = 1 \quad \forall i \in V, \quad (13)$$

$$\sum_i x_{im} \leq |V| \quad \forall m \in B, \quad (14)$$

$$x_{im} \in \{0, 1\} \quad \forall i \in V, m \in B, \quad (15)$$

where

B : set of berths

$|V|$: the cardinality of V , which is the number of ships

D_m : a coordinate of berth boundary defined as $D_m = (m - 1) \cdot BL$

BL : the berth length

x_{im} : decision variable; and =1 when berth m is allocated to ship i , =0 otherwise.

Constraints (12) guarantee that each ship does not physically exceed the boundaries of its assigned berth. Equalities (13) imply that each ship must be berthed in only one berth. Inequality set (14) defines that each berth has a capacity in terms of time to accommodate all ships. As [P] is a relaxed problem of [PU], the solution to [P] gives a lower bound to [PU]. This in turn means that the solution of [PU] defines the upper bound of [P]. \square

We can set a very small berth length for the BAPD. When we obtain the solution to the BAPD with this setting, it cannot be applied as ships are not confined in the allocated berth in terms of their length and may overlap with other ships in the physical quay space. This occurs because the BAPD formulation does not take into

account physical restrictions. If the berth length is set to a quite small value, e.g., 1 m, which is a much smaller length than the one that makes a difference in the handling time, then we have the following property:

Property 2. *The optimal solution to the BAPD, when the berth length is set to a very small value, gives the lower bound of the BAPC.*

Proof. As the possible minimum length is set as the berth length, ships are placed in locations, in which they have the minimum handling time. As the BAPD imposes no physical restrictions between berth length and ship length, ships assigned to a berth may overlap, both in the time and quay axes, with those assigned to adjacent berths on both sides of its berth. On the other hand, ships do overlap in quay axis and do not in time axis with those ships in the same berth, according to the problem structure of the BAPD (this feature corresponds to a column of the guillotine cutting in Fig. 2). The above-mentioned characteristic of the solution results in non-overlapping constraints in the BAPC only with respect to time axis among ships in the same berth presumed by the BAPD. Also no considerations of the physical restriction allow ships to be moored in berths at both ends of the quay with placing a part of ship beyond the quay end in the BAPC corresponding to the BAPD. The following formulation [PL] is a relaxed version of the BAPC.

$$\begin{aligned}
 \text{[PL]} \quad & \text{Minimize} \quad Z & (3) \\
 & \text{subject to} \quad (5), (6), (9)-(11)
 \end{aligned}$$

The formulation [PL] is not the BAPD, since in the optimality all $\delta_{ij}^t = 0$ in [PL], implying that ships have no overlapping restrictions in terms of time axis. However, bearing in mind that the defined berth length is so small that it does not make any real

difference in the handling time, the solution of [PL] can be composed by reassigning ships at the same berth (those ships are overlapping in terms of quay axis), to neighboring berths in the BAPD formulation without increasing the handling time. As this manipulation results in non-overlapping ships in a specific berth, the solution of [PL] is identical to that of the BAPD. Therefore, the BAPD gives the lower bound of the BAPC. \square

4.3. Outline of the entire solution procedure

Li et al. (1998) study the BAPC only in the context of minimizing the makespan by allowing ships to be delayed for their service. As mentioned before, they addressed two situations: one that allows repositioning ships from one location to another when necessary, while the other does not. The latter case has an analogy to the BAPC, although they allow placing ships in any quay locations during the planning horizon without the consideration of the arrival time and the varying handling time. They employ a simple heuristic based on the First-Fit-Decreasing method, which is one of the well-known efficient heuristics for the Bin Packing Problem. This method sorts items in a non-increasing order of their size and then assigns the items to the bins according to the order they appear in the sorted list. This method is appropriate when the objective is the minimization of the makespan. The BAPC has an objective of minimization of the total service time. If big ships (requiring more handling time) are served before smaller ships, the resulting total service time is greater than the one resulting from the service in the reverse service order. This results from the fact that the service time includes the waiting time for the predecessor to finish its handling. Consequently, the First-Fit-Decreasing method is not suitable. Neither is the reverse selection-scheme since it is likely to split the entire quay-time space into small pieces.

As seen in Section 4.2, the BAPD may serve as a good guide for solving the BAPC. If we can, as a first stage, obtain the solution to the BAPD regardless of the

berth length, then we can, as the second stage, modify the solution by a ship-relocation heuristic (referred to as the BAPC heuristic) to obtain a solution of the BAPC.

The following is the outline of the entire solution procedure:

Step 1. Define the minimum and maximum berth lengths.

Also define a set of different intermediate lengths between the minimum and maximum.

Step 2. Set U = the minimum berth length.

Step 3. Calculate the number of berths based on U and the quay length.

Step 4. Solve the BAPD with the number of berths obtained in Step 3.

Step 5. Modify the solution of the BAPD by the BAPC heuristic.

Step 6. Set U = the next intermediate length.

IF $U >$ the maximum berth length, THEN, STOP;
ELSE, go to Step 3.

In Step 1, berth lengths are defined. For a better solution, we should set very small and large values to the minimum and maximum lengths, respectively, and prepare a number of intermediate lengths. However, calculations with such a parameter setting take a long time. Practitioners may define these berth lengths based on the statistics from the observed ship lengths calling at the terminal concerned. They may, for example, employ a half or less than a half of the smallest feeder ship length for the minimum and the largest mother ship length for the maximum.

4.4. Details of the BAPC heuristic

The BAPC heuristic examines ships one by one like Li et al. (1998). However, we do not process in a non-increasing order of ship length, but in an increasing order of

the start time of ship handling, which is defined in the solution of the BAPD. When a particular ship (called a *target ship* or *TS*) is under examination, we identify a window of the quay space as illustrated in Fig. 3, which is set up between two already examined ships (*ASs*) that are the closest to the TS on each side of the TS. If the TS overlaps with other ships (called *OSs*), we reposition the TS and/or OSs along quay axis in order to resolve the overlapping. Notice that during this process, ASs are never repositioned since they have already been examined and their locations have been fixed. We form a pair of the TS and one of the OSs. We refer to the ship with a bigger increase in handling time (i.e., larger $|p_i - M_i| \tan \alpha_i$) as *the high priority-ship* and the other (i.e., that with a smaller increase) as *the low priority-ship*. Then, we first locate the high priority-ship at its best location within the window in terms of the handling time. We next place the low priority ship at its best within the window so that both ships do not overlap. This process is based on the premise that the two ships are not necessarily located in their best location as this process is based on the solution of the BAPD. In addition, even if they are at the best locations in the BAPD, the locations are not continuous but discrete as defined by the berth length, thus the best location of a ship in the BAPD is not necessarily the best in the BAPC. The order of processing ships is justified in that if the high priority-ship is repositioned after the low priority one and the former cannot be located at its best location, this repositioning most likely results in a higher increase in the handling time than the movement principle we defined.

 Fig. 3

In the above process, the lower-priority ship may not find its best location (as found within the window) without overlapping with the other ship, as illustrated in Fig. 4 where ship j is the low priority. Note that in this figure, ship j 's true best location is

outside the window. Thus, the second best location, which is the best as found within the window, is the one being indicated by M'_j . However, M'_j is not a feasible location to place ship j without overlapping. The next best location seems to be on the left of ship i ; however, it is not feasible either due to the quay space being smaller than the length of ship j . In this case, we try to locate the two ships once again but with the reverse ship order, i.e., the low priority-ship first and the high priority-ship second.

 Fig. 4

After the above two processes, both ships may still not be located without overlapping. As shown in the case of Fig. 5, the best locations of the ships are too close to find an enough space for the second-processed ship no matter which ship is processed first. In this case, considering the two ships as being a collective ship by placing both ships side by side based on the geometric relationship of the ships' best locations, we locate the collective ship so that the total handling time of the two ships is minimized within the window. The handling time of the collective ship as a function of quay location is defined by Eq. (16), assuming ship i is located left of ship j as shown in Fig. 5.

$$C_{ij} = CM_i + \left| p_{ij} - \frac{L_j}{2} - M_i \right| \tan \alpha_i + CM_j + \left| p_{ij} + \frac{L_i}{2} - M_j \right| \tan \alpha_j \quad (16)$$

where C_{ij} is the handling time of the collective ship and p_{ij} is the quay location of the ship as measured in the longitudinal center of the ship.

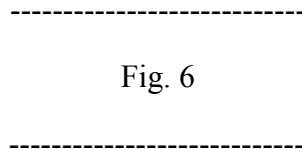
 Fig. 5

After this process is finished, regarding the TS as an AS, we perform the same

process with a pair of the TS and another OS and repeat this until all the OSs are relocated.

Note that if the window length is less than the total of the TS and an OS, the two ships cannot be successfully relocated within the window. In such a case, the service of the OS is delayed by moving backwards (or upwards) in time axis. Also note that if no vessels are served immediately before the TS, i.e., there is an empty space in the quay location where the TS is to be served, the service of the TS will start as early as possible after the relevant process, ensuring that the TS no longer overlaps with any OSs.

The solution of the BAPD is not exact but approximate, which means that the present locations of the OSs are not necessarily the best. Consequently the OSs may not move their locations next to the TS. Such movements cause a window split as shown in Fig. 6, if the new locations of the OSs are far from the TS due to shorter handling times. All of the split windows are used in the subsequent process to treat the overlap of the remaining OSs and the TS. This implies that when there are more than one window in the subsequent process for the next OS of the TS, the OS and TS under process may be located at their best locations in any different windows. We, hereafter, refer to those windows (either single or multiple) available for processing an OS (and the associated TS), as its *available windows*.



A more formal procedure of relocating ships is as follows, where j is a ship under examination and i is a ship that overlaps with ship j :

[Main-routine]

Step 1. Number all ships in ascending order of their handling start time. Set $j = 1$.

- Step 2.* Identify the available window to ship j .
- Step 3.* Identify ships overlapping with ship j and number them in ascending order of the start time of their handling. Set $i = 1$.
- Step 4.* Relocate ships i and j by sub-routine RELOC.
- Step 5.* IF ship i is the last overlapping ship,
 THEN, go to the next step;
 ELSE, perform $i = i + 1$ and go to Step 4.
- Step 6.* IF ship j is the last ship involved in the system,
 THEN, STOP;
 ELSE, perform $j = j + 1$ and go to Step 2.

[Sub-routine RELOC]

- Step 1.* Define the priorities for ships i and j based on the increase in handling time associated with the change of their mooring location at the quay.
 Examine all the available windows to ship j one by one from the closest window to ship j .
 IF no windows are large enough to relocate the two ships,
 THEN, reposition ship i upwards in time axis in order to resolve the overlapping and RETURN.
 ELSE, i.e., a window is found, go to Step 2.
- Step 2.* Define the high and low priority ships.
 Reposition the high-priority ship to the best quay location in any of the available windows.
 Next reposition the low-priority ship to the best quay location in any of the available windows.
- Step 3.* IF the low-priority ship does not find a space to locate itself without overlapping with the high-priority ship,
 THEN, go to the next step;
 ELSE, go to Step 6.
- Step 4.* Perform Step 2 with the reverse order (i.e., try to relocate the low-priority ship

first and the high-priority ship second).

Step 5. IF both are successfully relocated,

THEN, go to Step 6;

ELSE, define a collective ship that represents both ships located side by side as done in Step 2. Put the ship in its best quay location in any of the available windows. Go to Step 6.

Step 6. IF the processed windows are split by the relocation of the two ships,

THEN, register the split windows as additional available windows to ship j .

RETURN.

As shown in Section 3.2, the rate of increase is defined as a linear function. However, if this is not the case, no major problems arise. In the process of relocating overlapping ships, we should identify the minimum handling time by searching within available windows with the given function.

5. Numerical experiments

The solution procedure is coded in “C” language on a Sun SPARC-64G workstation. Problems used in the experiments were generated randomly, but systematically. We developed nine basic problems with different quay lengths and numbers of ships served as shown in Table 1. For each basic problem, we set up two different handling time data sets and five different ship arrival data sets with different seeds for pseudo random numbers to generate ten different problem samples.

Tables 1

Each ship involved in the problem has both a handling time at its best quay location and a rate of increase in handling time as the actual mooring location deviates. The handling times are defined by ship size and the number of handled containers, both of which follow the uniform distribution. We provide two ship data sets with the different distribution of the handling time. The first data set is generated by the uniform distribution with an average ship size of 200 m and an average handling volume of 1,100 containers, while the second data set is generated with an average ship size of 180 m and an average handling volume of 950 containers. The average handling time per ship at the best quay location for the first data set is approximately 10 h, whilst the one for the second data set is roughly 9 h. In addition, the distribution of the former data set is more widely spread than the latter. The ship arrival pattern follows the exponential distribution with an average interval of 3 h.

In the solution procedure, a problem is first solved by the BAPD heuristic and consequently processed by the BAPC heuristic based on the BAPD solution. In the BAPD process, the entire quay space is partitioned into several berths. The number of berths may influence the quality of the final solutions. In the algorithm described in Section 4.3, the numbers of berths are calculated with given berth lengths. However, for convenience, we directly vary the number of berths in the experiment as shown in Table 2, in order to examine the relationship between the solution quality and the number of berths. In principle, we provide the number of berths ranging from value A to value B with an increase by 5, where value A is the maximum number of berths as long as the BAPD yields a feasible solution to the BAPC and value B is the number of berths with a small berth length. At the same time it is worth to split the quay into fewer berths by a long berth length, so that each berth accommodates the respective assigned ships physically within the berth length, in order to examine the relationship in the solution quality between the BAPD and BAPC.

Table 2

We first look into the relationship between the solution quality of the BAPC and the number of berths. Fig. 7 illustrates the total service time of all involved ships, which is calculated as the average over ten different problem samples for each problem scenario. For each quay length, the total service time decreases in accordance with the number of berths up to a certain figure depending on the quay length, i.e., 2, 3 and 4 berths for the quay length of 800, 1200 and 1600 m, respectively. These numbers are the maximum numbers of berths as long as the BAPD solutions define feasible solutions to the BAPC. The service time, then, increases as the number of berths increases. In some cases, we observe that once again they decline. The fluctuation of the service time over the varying number of berths is significant in cases with a large number of ships, as we observe that in a 1600-m quay length, the maximum of the service time is three times greater than the minimum for the cases of 60 ships, while the maximum is only twice larger than the minimum for the 20 ship cases. The best of the average service time per ship with the smallest number of ships in each problem of a different quay length is figured out to be around 20 h, while the best service time with the largest number of ships is approximately 24 h. As every ship data set has the same average of arrival interval, the objective function value of the problem (i.e., the total service time) is supposed to be the same. However, there is such a variance in the objective function value with the different number of ships. The reason may be the solution quality of the BAPD, which is better for a problem with fewer ships and berths. The service time per ship at the best number of berths is twice as much as the average handling time, resulting in a wait of 10 h. Such a solution is caused by that the entire quay is in an extremely congested situation with an average interval of ship arrival of 3 h and a

handling time of 10 h.

Fig. 7

The initial solution by the BAPD algorithm spreads ships sparsely with fewer berths, while with more berths it locates them densely, where they are likely to overlap. The BAPC was expected to yield a good solution no matter what number of berths was given in the BAPD. However, according to the computational results, the BAPC algorithm performs better with fewer berths. In other words, the algorithm performs better in filling empty spaces with ships, which are sparsely located in the quay-time space of a solution of the BAPD with fewer berths, than in resolving the ships' overlapping in a solution of the BAPD with more berths. It is interesting to note that the best solutions are found with the maximum number of berths for which the BAPD yields feasible solutions to the BAPC. We note, however, that in the cases of quay length of 1600 m, the solutions with five berths are also acceptable.

We provided two different seed sets for random numbers to generate ships' handling times. The data sets with seed 1 (called data set 1) have larger average values than those with seed 2 (called data set 2). Also, the former has larger variances in the handling time; therefore, it is expected that as the idle space is likely split into small pieces in the quay-time space of the BAPC, which is converted from the solution of the BAPD, it is not easy for the BAPC algorithm to locate ships orderly in order to make the total service time as small as possible. Figs. 8 and 9 graphically represent the average values of the total service time over five different computation samples by data sets 1 and 2, respectively. Both graphs have the same trend; however as assumed, data set 1 created more difficult computation samples than data set 2.

Figs. 8 & 9

With fewer berths, the BAPD calculation produces the feasible solution to the BAPC. In the next step, we explore how the BAPC algorithm improves the feasible solution being given by the BAPD. Fig. 10 demonstrates the degree of improvement for three different quay lengths, where the improvement is defined as follows:

Improvement = (solution value of the BAPD - solution value of the BAPC) / solution value of the BAPC

There is a reduced degree of improvement as the number of berths increases for every quay length; the declining curve is steeper as the berth length increases (or the number of berths decreases). The reason for this trend taking place can be explained by the fact that as we have found the best BAPD solution is obtained at the largest number of berths in Fig. 10, thus resulting in a small possible improvement by the BAPC heuristic. However, this implies that the BAPD heuristic improves the total service time by at least 10%.

Fig. 10

Even though the improvement is the minimum with the maximum number of berths as long as the BAPD solution is feasible in terms of the BAPC, the objective function value of the resulting BAPC solution is the best among the others. According to the conducted experiments, a suggested principle could be to find the initial solution by the BAPD with the berth length that is given by dividing the quay length by the maximum ship length and improve it with the BAPC procedure.

6. Conclusions

Major container ports in Europe and China furnish the so-called Multi-User Terminals, where the effective berth allocation is one of the crucial issues for high productivity. For managerial purposes, the entire quay in a Multi-User Terminal is partitioned into several berths, and the ship allocation to the proper quay location is based on the berth. In previous papers, the authors have examined efficient berth scheduling for the Multi-User Terminal (called the berth allocation in the discrete quay location), where ships are all located within any of the partitioned berths. Meanwhile, as there is a large diversity in ship size (especially where ship length is concerned), ships are sometimes, in reality, berthed across the berth boundary for efficiency in berth usage. Therefore, more flexible berth allocation is desired, which locates the ships by ignoring the berth boundary. The problem relevant to this berth allocation circumstance is called the berth allocation in the continuous quay location (BAPC). For this problem, a heuristic algorithm incorporated with the existing berth allocation in the discrete quay location (BAPD) was developed. This algorithm solves the problem in two stages: in the first stage the algorithm of the BAPD identifies a solution given the number of partitioned berths, and in the second stage the other procedure relocates the ships that may overlap or be located sparsely in a scheduling space, which is defined by the BAPD algorithm.

According to the numerical experiments conducted, we arrived at the following findings: (a) as a general trend, the solution to the BAPC is worsening with increasing number of berths provided in the BAPD calculation; (b) the average service time per ship in the solution of the BAPC is also worse with an increased number of ships in the

BAPD; and (c) the best solution with a specific number of ships is identified when the BAPD algorithm yields a feasible solution to the BAPC with as many berths as possible. While to the authors' knowledge, no terminals employ such scientific methods, including the existing studies, for the berth scheduling, terminal planners and operators may use them to improve the terminal operation.

Acknowledgements

The authors would like to thank the three anonymous referees for their time, support and insightful suggestions that have improved this paper to a great extent.

Appendix A. Formulation of the dynamic BAPD

$$[\text{PD}] \quad \text{Minimize} \quad \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \{(T - k + 1)C_{ij} + S_i - A_j\} x_{ijk} + \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} (T - k + 1)y_{ijk} \quad (\text{A.1})$$

$$\text{subject to} \quad \sum_{i \in B} \sum_{k \in U} x_{ijk} = 1 \quad \forall j \in V, \quad (\text{A.2})$$

$$\sum_{j \in V} x_{ijk} \leq 1 \quad \forall i \in B, k \in U, \quad (\text{A.3})$$

$$\sum_{l \in V} \sum_{m \in P_k} (C_{il} x_{ilm} + y_{ilm}) + y_{ijk} - (A_j - S_i) x_{ijk} \geq 0 \quad \forall i \in B, j \in W_i, k \in U, \quad (\text{A.4})$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in B, j \in V, k \in U, \quad (\text{A.5})$$

$$y_{ijk} \geq 0 \quad \forall i \in B, j \in V, k \in U, \quad (\text{A.6})$$

where

$i (= 1, \dots, I) \in B$: set of berths

$j (= 1, \dots, T) \in V$: set of ships

$k (= 1, \dots, T) \in U$: set of service orders

A_j : arrival time of ship j

P_k	: subset of U such that $P_k = \{p p < k \in U\}$
S_i	: time when berth i becomes idle for the planning horizon
W_i	: subset of ships with $A_j \geq S_i$
C_{ij}	: handling time spent by ship j at berth i
x_{ijk}	: 1 if ship j is served as the $(T - k + 1)$ th last ship at berth i , and 0 otherwise
y_{ijk}	: idle time of berth i between the departure of the $(T - k + 2)$ th last ship and the arrival of the $(T - k + 1)$ th last ship when ship j is served as the $(T - k + 1)$ th last ship

The decision variables are x_{ijk} s and y_{ijk} s. Objective (A.1) minimizes the total service time. Constraint set (A.2) ensures that every ship must be served at some berth in any order of service. Constraints (A.3) enforce that every berth serves up to one ship at any time. Constraints (A.4) assure that ships are served after their arrival. For the detailed derivation of the formulation, see Imai et al. (2001, in press).

References

- Brown, G.G., Lawphongpanich, S., Thurman, K.P., 1994. Optimizing ship berthing. *Naval Research Logistics* 41, 1-15.
- Brown, G.G., Cormican, K.J., Lawphongpanich, S., Widdis, D.B., 1997. Optimizing submarine berthing with a persistence incentive. *Naval Research Logistics* 44, 301-318.
- Guan, Y., Xiao, W.-Q., Cheung, R.K., Li, C.-L., 2002. A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis, *Operations Research Letters* 30, 343-350.
- Imai, A., Nagaiwa, K., Chan, W.T., 1997. Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation* 31, 75-94.
- Imai, A., Nishimura E., Papadimitriou, S., 2001. The dynamic berth allocation for a container port. *Transportation Research Part B* 35, 401-417.
- Imai, A., Nishimura E., Papadimitriou, S., 2003. Berth allocation with service priority. *Transportation Research Part B* 37, 437-457.

- Imai, A., Nishimura E., Papadimitriou, S., in press. Corrigendum to “The dynamic berth allocation for a container port” [Transportation Research Part B 35 (2001) 401-417]. Transportation Research Part B, in press, doi:10.1016/j.trb.2004.03.004.
- Kim, K.H., Moon, K.C., 2003. Berth scheduling by simulated annealing. Transportation Research Part B 37, 541-560.
- Lai, K.K., Shih, K., 1992. A study of container berth allocation. Journal of Advanced Transportation 26, 45-60.
- Li, C.-L., Cai, X., Lee, C.-Y., 1998. Scheduling with multiple-job-on-one-processor pattern. IIE Transactions 30, 433-445.
- Lim, A., 1998. The berth planning problem. Operations Research Letters 22, 105-110.
- Nishimura, E., Imai, A., Papadimitriou, S., 2001. Berth allocation planning in the public berth system by genetic algorithms. European Journal of Operational Research 131, 282-292.
- Park, K.T., Kim, K.H., 2002. Berth scheduling for container terminals by using a sub-gradient optimization technique. Journal of the Operational Research Society 53, 1054-1062.
- Park, Y.-M., Kim, K.H., 2003. A scheduling method for berth and quay cranes. OR Spectrum 25, 1-23.

Table 1. Number of ships to be served

Quay length (m)	Composition of # of ships involved in the problem		
800	10	20	30
1200	15	30	45
1600	20	40	60

Table 2. The number of berths for the BAPD procedure

Quay length (m)	Composition of # of berths									
800	1	2	5	10	15					
1200	1	2	3	5	10	15	20	25		
1600	1	2	3	4	5	10	15	20	25	30

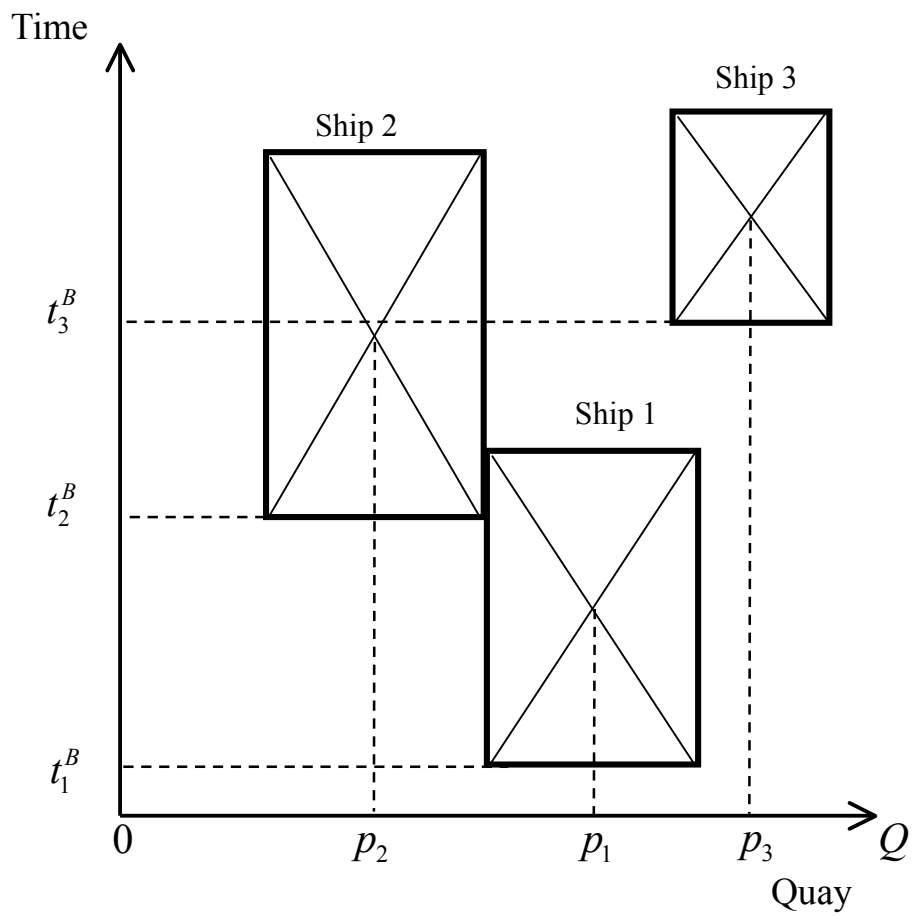
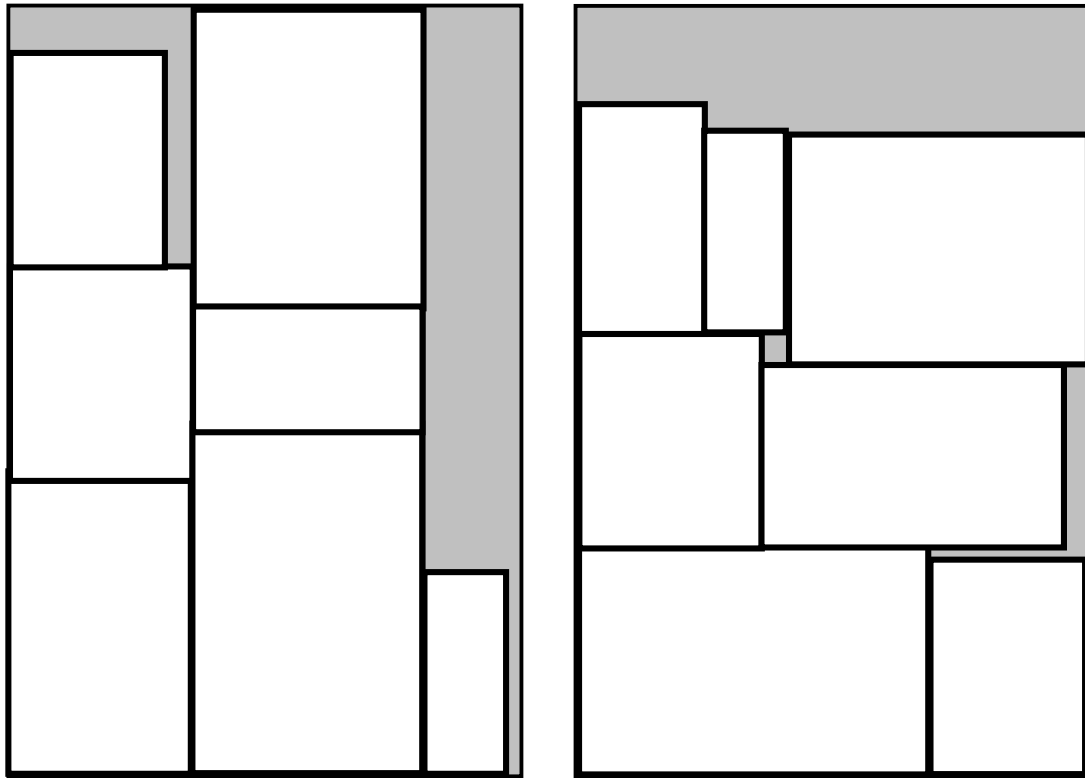


Fig. 1. Graphical representation of berth allocation



(a) Guillotine cut

(b) Non-guillotine cut

Fig. 2. Cutting stock schemes

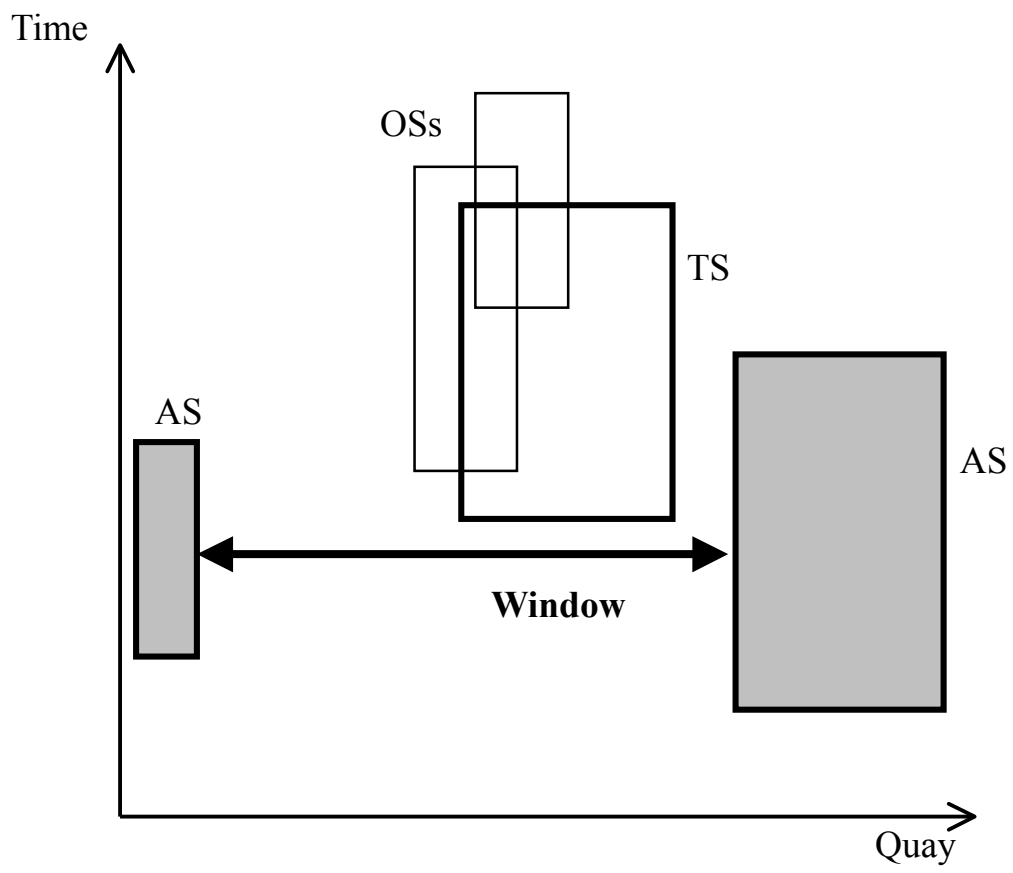
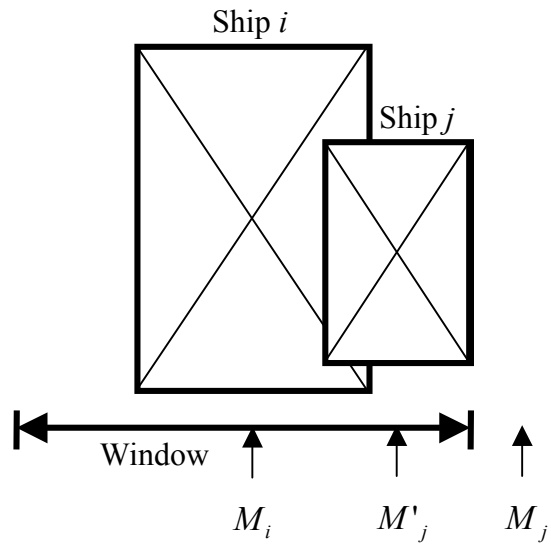


Fig. 3. Definition of window



M_i : Ship i 's best location

M_j : Ship j 's best location

M'_j : Ship j 's best location within the window

Fig. 4. Repositioning two ships

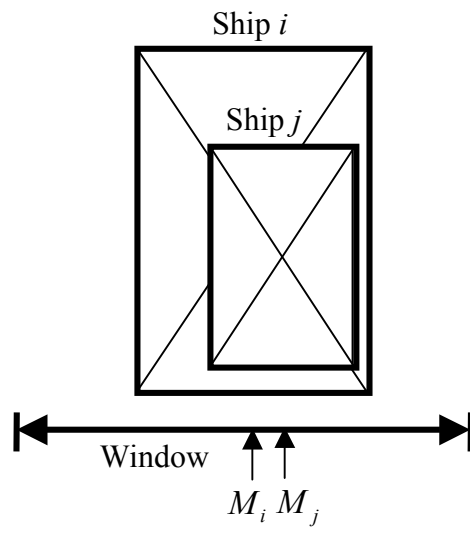


Fig. 5. Unsuccessful case of locating two ships separately

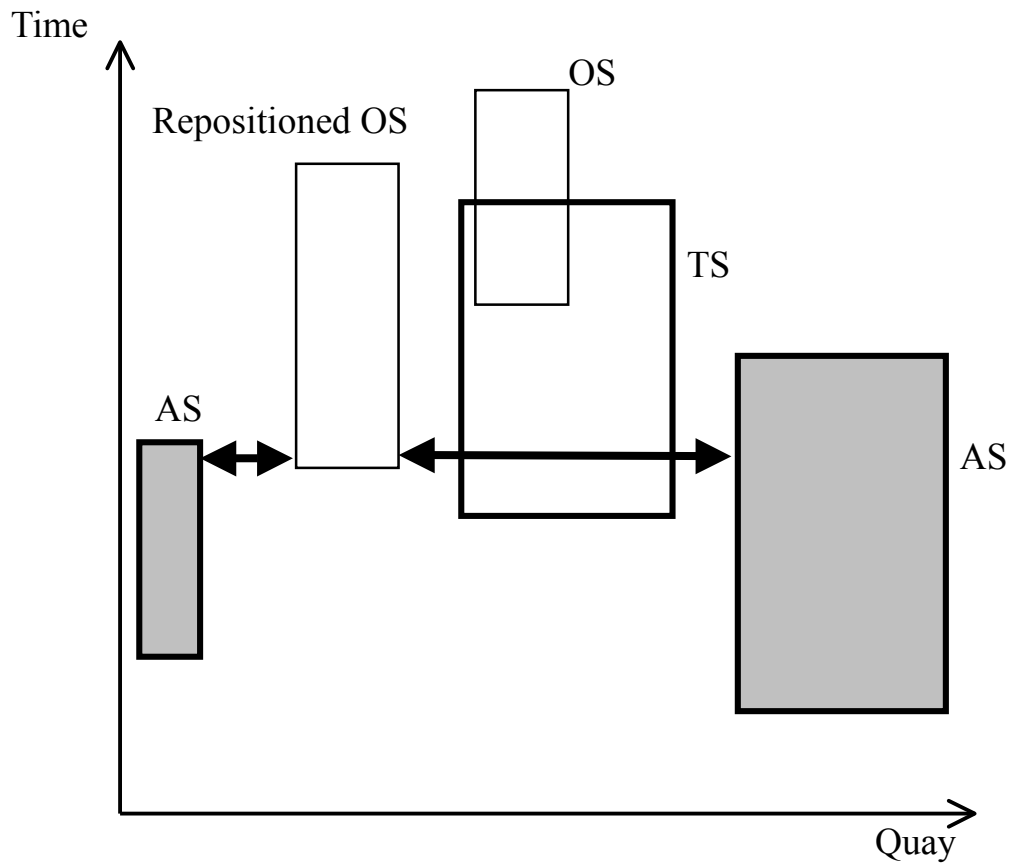
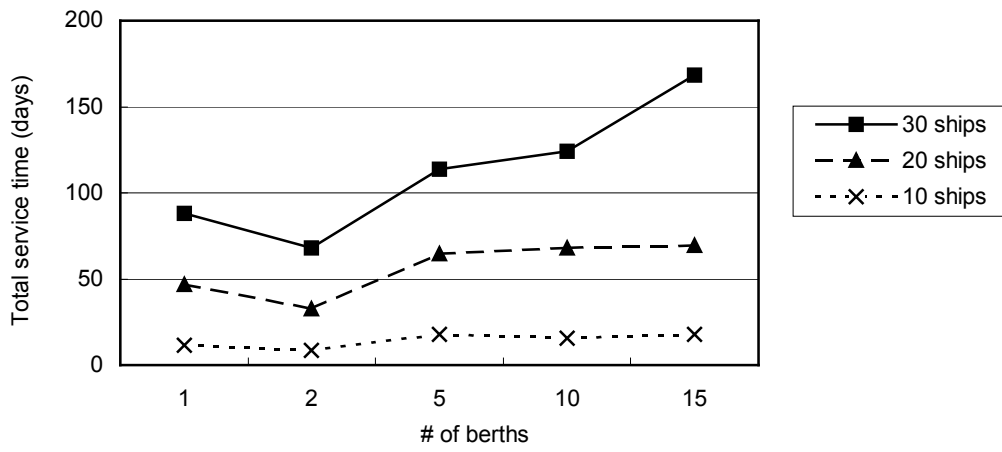
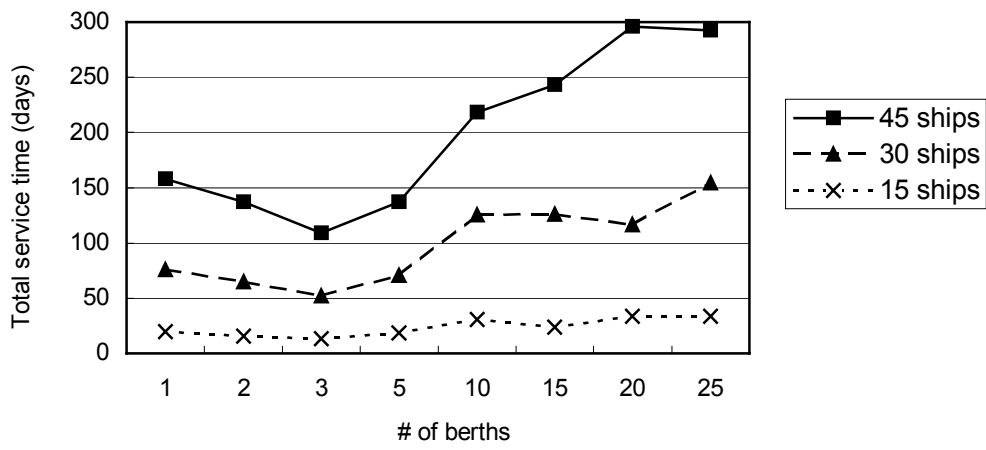


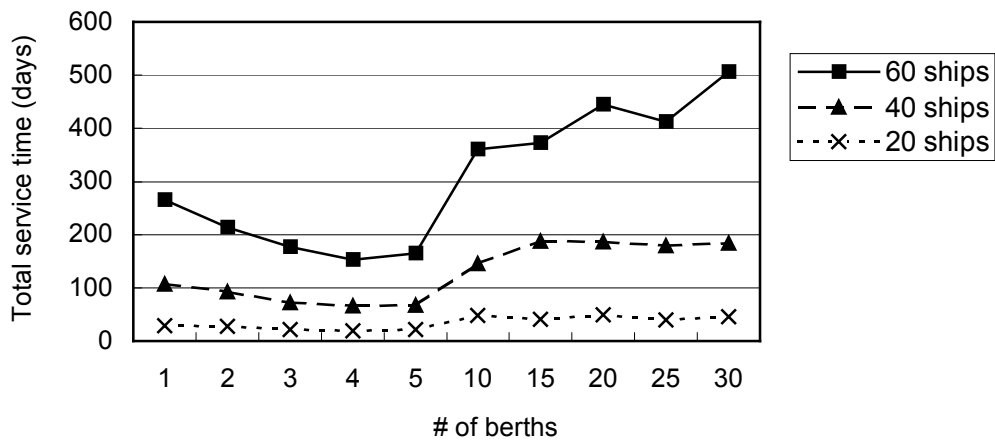
Fig. 6. Split window



(i) Quay length=800m

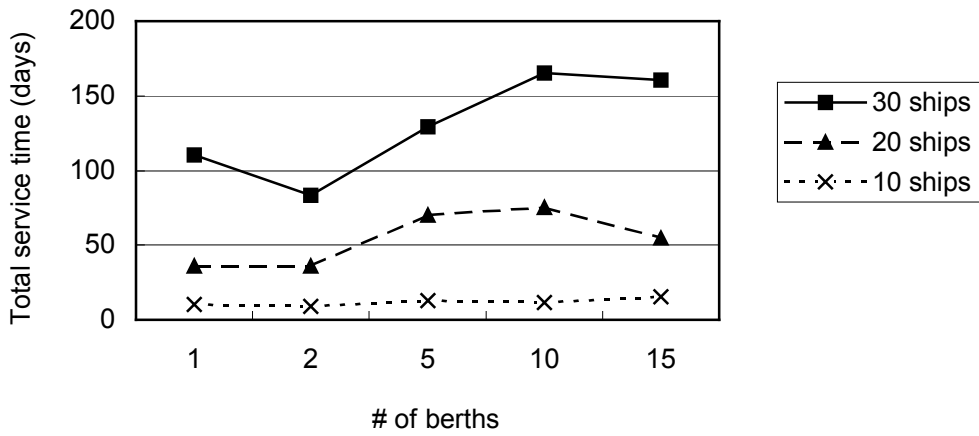


(ii) Quay length=1,200m

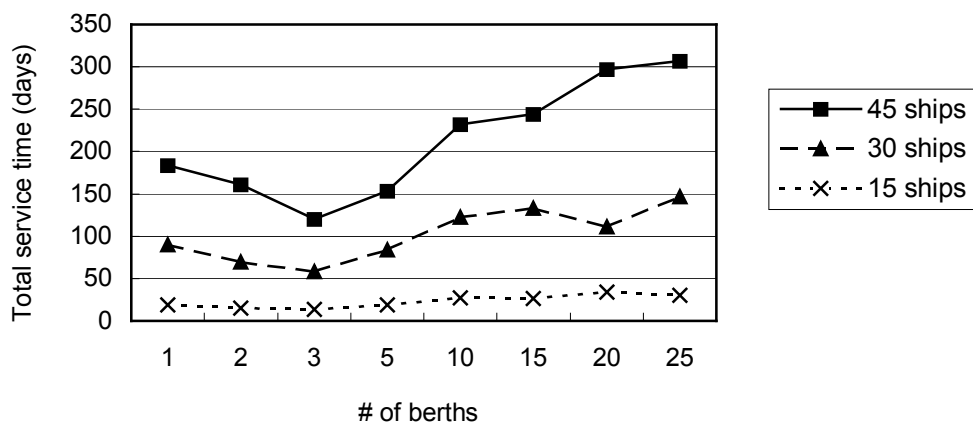


(iii) Quay length=1,600m

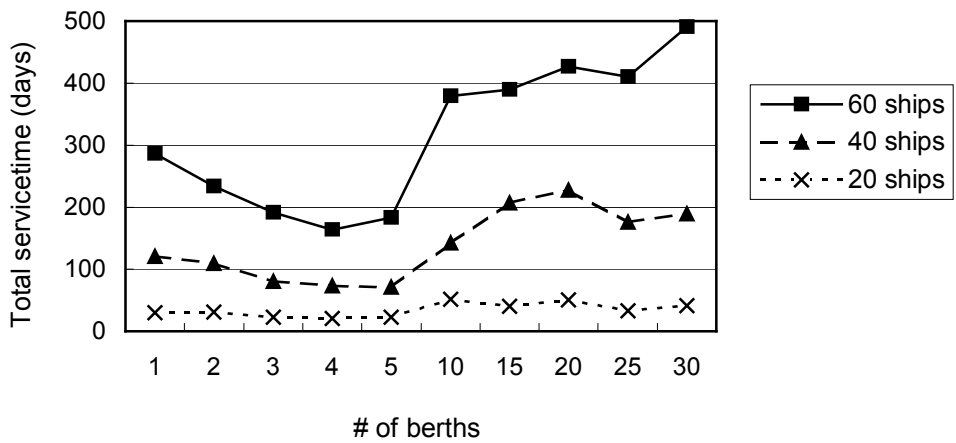
Fig. 7. Solution quality versus the number of berths



(i) Quay length=800m

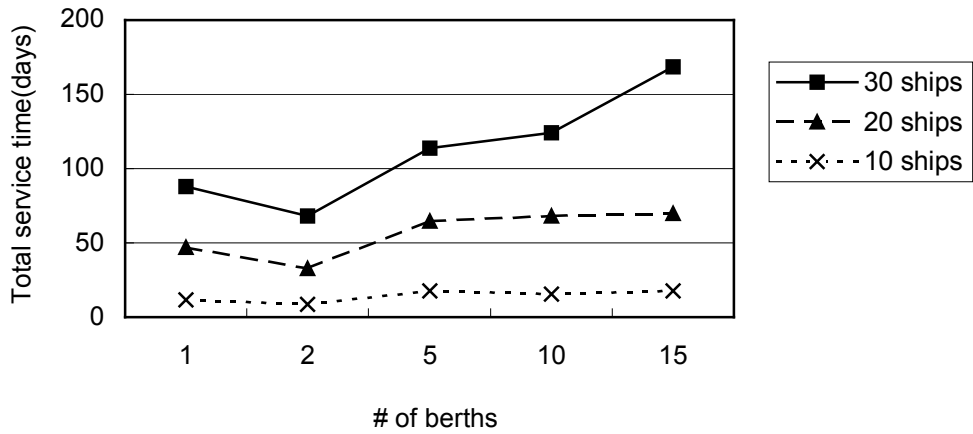


(ii) Quay length=1,200m

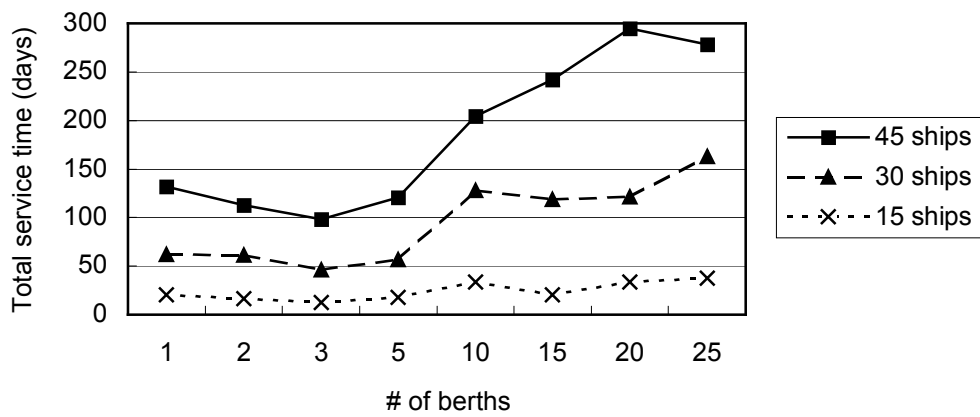


(iii) Quay length=1,600m

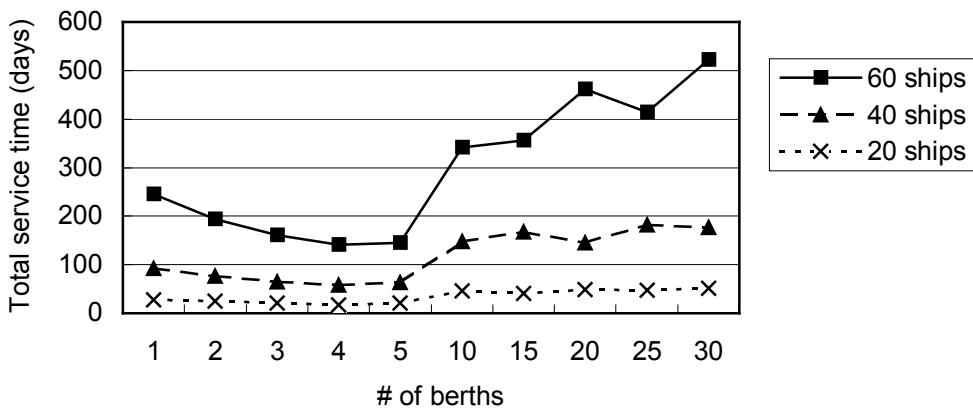
Fig. 8. Solution quality versus the number of berths (ship data set 1)



(i) Quay length=800m



(ii) Quay length=1,200m



(iii) Quay length=1,600m

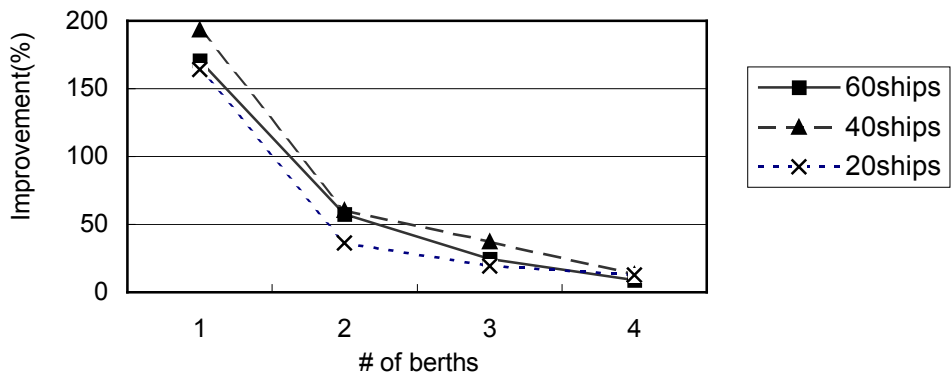
Fig. 9. Solution quality versus the number of berths (ship data set 2)



(i) Quay length=800m



(ii) Quay length=1,200m



(iii) Quay length=1,600m

Fig. 10. Solution qualities between the BAPD and BAPC