



Berth allocation with service priority

Imai, Akio

Nishimura, Etsuko

Papadimitriou, Stratos

(Citation)

Transportation Research Part B: Methodological, 37(5):437-457

(Issue Date)

2003-06

(Resource Type)

journal article

(Version)

Accepted Manuscript

(URL)

<https://hdl.handle.net/20.500.14094/90000070>



BERTH ALLOCATION WITH SERVICE PRIORITY

Akio IMAI

Department of Transportation and Information Systems Engineering

Kobe University of Mercantile Marine

Fukae, Higashinada, Kobe 658-0022 Japan

&

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

Etsuko NISHIMURA

Department of Transportation and Information Systems Engineering

Kobe University of Mercantile Marine

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

Over the past several years, port related charges in Japanese ports have been substantially higher than those charged in other major international hub ports. All major container ports in Japan feature so-called Dedicated Terminals in which cost-effectiveness is justified by huge container volume to be handled. One of the reasons cited for high port charges is a relative decrease in handling volume compared to the terminal capacity, resulting in inefficient use of the existing capacity. The use of the Multi-User Container Terminal (MUT) concept employed in some of the major container hub ports such as Hong Kong, Pusan, Hamburg and Rotterdam reduces redundant terminal space and results in substantial cost savings in cargo handling costs and therefore is desired for ports in Japan as well. One of the key issues in the MUT operation is the berth allocation to calling vessels. In a recent study, an allocation problem for the MUT was examined, in which each vessel was treated equally. However, as some vessel operators desire high priority services, the goal of this paper is to modify the existing formulation of the berth allocation problem in order to treat calling vessels at various service priorities by developing a genetic algorithm based heuristic for the resulting non-linear problem.

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

1. INTRODUCTION

In major ports in Japan and the US such as Kobe, Yokohama, Los Angeles and Oakland, shipping lines lease the container terminals (referred to as *Dedicated Terminal or DT*) in order for them to be directly involved in the processing and handling of the containers as they aim to achieve higher productivity and economies of scale. Whereas this may be warranted in the case of a firm that handles a large amount of containers with a corresponding number of ship calls, it may not be justified if these quantities are not sufficient, as it will have an adverse effect on costs. Over the past several years port related charges in Japanese ports have been consistently higher than those in other major ports. One of the reasons cited for the increased costs is the over-investment in ports with relatively small cargo volume.

A “Multi-User Container Terminal (MUT)” may be defined as a terminal with a long berth, that is able to handle simultaneously a number of vessels which are dynamically allocated to the berth and are not always assigned to specific berth locations. Some major container ports provide an MUT, while most of them feature DTs. Examples of the MUT are Hong Kong International Terminal (HIT) in Hong Kong, Pusan East Container Terminal (PECT) in Pusan, and Delta Multi-User Terminal (DMU) in Rotterdam. In addition, most container terminals in China are used as MUTs, since the limited terminal space due to a smaller construction budget has to be utilized efficiently in order to meet huge container traffic.

An MUT can reduce the required terminal space while handling containers with the same rate of productivity as a DT, thus resulting in substantial cost savings in cargo handling costs. One of the issues that affect the efficiency of MUT operations is the berth allocation to calling vessels. The berth allocation problem (BAP) has already been addressed and solved through a subgradient optimization with a Lagrangian relaxation technique (Imai et al., 2001), in order to minimize the total service time that comprises the handling time and the waiting time for an idle berth. This problem assumed that each vessel was treated without any differentiated priority. However, according to a survey for a port authority in Japan, it became evident that vessels with a large container handling volume preferred to be given a higher priority over small vessels when the berth was congested. In general the ship priority depends on the total throughput per shipping line; therefore the ship size (actually handling volume of that ship) as an index for the priority must be regarded as an intervening variable in that it is often closely correlated to the power of the shipping company which deploys it. This is the case, for example, in Tanjong Pagar container

terminal of Singapore.

On the other hand, some feeder shipping companies claim that small feeder vessels are very often dominated by large vessels, and they argue that they ought to be given the same priority treatment as larger vessels and, indeed, even higher priority since their handling times are much shorter. In fact, Dalian container terminal of China, which is an MUT, provides higher priorities to small feeder vessels when the terminal is busy, resulting in less waiting time to the following vessels. This advantage is demonstrated as follows: Given the situation that two vessels with different handling volumes have just arrival at the same time, the small vessel is forced to wait for long if the big one is served first, while the latter waits for a short time if it is served second.

From the above discussion, it is clear that service priority is important in terminal operations including berth allocation, especially for a situation which involves various sizes of ships with various cargo handling volumes at a particular port of call.

Xu and Parnas (2000) provide interesting insights into priority scheduling that schedules jobs (or processes) for limited resources (or processors), assuming: (a) the scheduling is performed at run time; (b) processes are assigned fixed priorities and whenever two processes compete for a processor, the process with higher priority wins. In accordance with their paper, the best-known representative of priority scheduling is the *Rate Monotonic Scheduling* (Liu and Layland, 1973). Furthermore, there is another scheduling policy named the *Priority Ceiling Protocol* (Sha et al., 1990). The former assumes that the major characteristics of all processes are known before run-time (that is corresponding to the pre-known handling time in the berth allocation). Fixed priorities are assigned to the processes according to their periods, the shorter the period, the higher the priority. At any time the process with the highest priority among all the processes ready to run is assigned to the processor. The Priority Ceiling Protocol makes the same assumptions as Rate Monotonic Scheduling, except that in addition, processes may have critical sections guarded by semaphores, and a protocol is provided for handling them. Each semaphore is assigned a priority ceiling, which is equal to the priority of the highest priority process that may use this semaphore. The process that has the highest priority among the processes ready to run is assigned to the processors, like the Rate Monotonic Scheduling.

In these scheduling policies the priority is treated explicitly, that is, processes are selected for processing in order of priority value, and a scheduling algorithm arbitrarily determines a solution with such a priority. In fact, terms such as priority rule, heuristics, or scheduling rule are often used synonymously in the scheduling literature.

The goal of this paper is to define a solution to the objective of minimized total

service time, while differentiating priorities to ships by variation of their service times (including the waiting time for an idle berth) in the solution. In this study the priority is evaluated by the resulting service time for each ship; therefore the priority should not be, in a strong sense, defined in the problem formulation like the above scheduling methods. It is not necessarily true that the priorities being given to the formulation explicitly differentiates the resulting service times for ships, because ships have different (estimated or pre-known) handling times which may depend on the berth location they are assigned and they do not always arrive at the port at the same time. Consequently, the above-mentioned scheduling approaches, with the so-called hard priority are restricted to the BAP. By contrast, this study assigns the ships a softly defined-priority, that is, weights related to their handling volumes in a BAP mathematical programming formulation.

The objective of this paper is to modify the existing BAP formulation in order to deal with calling vessels with various service priorities. For solving the BAP with priority we first examine a subgradient method using a Lagrangian relaxation technique like the one used in Imai et al. (2001). However due to its complexity in the solution process, we finally propose a genetic algorithm (GA) based heuristic.

The paper is organized as follows. The next section provides a literature review on the berth allocation planning. A mixed integer programming of the BAP without priority consideration is discussed in Section 3. The fourth section introduces a new formulation of the BAP to take into account the priority concern. Following that, we describe a Lagrangian relaxation approach for the solution, then a GA based solution method for that problem. In the fifth section a number of computational analyses are carried out, while the final section concludes the paper.

2. LITERATURE REVIEW ON THE BAP

Most port studies focus their attention to the strategic and tactical issues facing the port. As many container terminals are privately operated by specific shipping lines, very few studies have been conducted on berth allocation in a multi-user terminal system.

Lai and Shih (1992) propose a heuristic algorithm for berth allocation which is motivated by more efficient terminal (actually berth) usage in the HIT terminal of Hong Kong. Their problem considers a first-come-first-served (FCFS) allocation strategy, which is not the case in our problem. 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. Berth planning in naval ports has important differences from planning in commercial ports. In the former, a berth shift occurs when for proper services, a newly arriving ship must be assigned to a berth where another ship is already being serviced. This treatment is unlikely in commercial ports. Berth shifting as well as other factors less relevant to commercial ports are taken into account in their paper, thus making their study inappropriate for commercial ports.

Imai et al. (1997) address berth allocation for commercial ports. Most service queues are in general processed on the FCFS basis. They conclude that in order to achieve high port productivity, an optimal set of ship-to-berth assignments should be found without considering the FCFS rule. However, this service principle may result in certain ships being dissatisfied with the order of service. In order to deal with the two criteria to evaluate, i.e., berth performance and dissatisfaction with the order of service, they develop a heuristic to find a set of non-inferior solutions while maximizing the former and minimizing the latter. Their study assumes a static situation where ships to be serviced for a planning horizon have all arrived at a port before one plans the berth allocation. Thus, this study can apply only to tremendously busy ports. As far as container shipping is concerned, such busy ports are neither competitive nor realistic because of the long delay in interchange process at ports. In this context, Imai et al. (2001) extend the static version of the BAP to a dynamic treatment that is similar to the static treatment, but with the difference that some ships arrive while work is in progress. As the first step in this dynamic treatment, only one objective, berth performance, is considered. Due to the difficulty in finding an exact solution, they develop a heuristic by using a subgradient method with Lagrangian relaxation. Their study assumes the same water depth for all the berths, while in practice there are berths with different water depths in certain ports. Nishimura et al. (2001) further extend the dynamic version of the berth allocation problem for the multi-water depth configuration. They employ genetic heuristic algorithms to solve that problem.

There is also another class of the BAP, which is the one with a continuous location index. While in the above mentioned studies the entire terminal space is partitioned into several parts (or berths) and the allocation is planned based on the divided berth space, under this approach ships are allowed to be serviced wherever the empty spaces are available to physically accommodate the ships via a continuous location system. This class of problem resembles more or less the cutting-stock problem where a set of commodities are packed into some boxes in an efficient manner. A ship in service at a berth can be shown by a rectangle in a time-space representation or gantt chart, therefore efficient berth usage is a sort of packing “ship rectangles” into a berth-time availability as a box with some limited

packing scheme such that no rotation of ship rectangles is allowed. As for berth allocation with continuous location, there are very few examples of studies such as Lim (1998) and Li et al. (1998) due to the difficulty in determining a solution.

3. FORMULATION OF THE DYNAMIC BERTH ALLOCATION

3.1 Formulation

This study develops a BAP with priority consideration based on the dynamic version of the BAP (Imai et al., 2001). This section overviews the formulation of the BAP without priority consideration, in order to assess the difficulty in the formulation and solution methodology for the BAP with priority consideration (PBAP).

There are some evaluation criteria to measure berth (or terminal) productivity such as the total handling time and the total service time that includes not only the handling time but also the waiting time of ship for an idle berth. The former is much more related to the port operator than to its users (i.e., calling ships) since no consideration is made for the waiting time. Quick turnaround time at a port is a vital issue for sea-borne transportation, especially for container ship transportation; therefore the latter is used as an objective for the BAP.

The BAP assumes only one long wharf at a multi-user terminal. Considering a variety of ship sizes, especially ship length, a number of ship location alternatives at the wharf are possible. However, for simplicity in the solution procedure, the wharf is virtually divided into several blocks as seen in major container ports, and in the BAP we obtain a set of assignments of ships to those blocks that are hereafter referred to as berths.

We assume that each berth can service one ship at a time regardless of the ship's size and that there are no physical and/or technical restrictions such as the relationship between ship draft and water depth. Furthermore, for generalizing the BAP, the ship handling time is assumed dependent on the berth where it is assigned. The second assumption is justified by the following reasoning: at a private container terminal (or berth), containers to be loaded onto a ship are stored in appropriate yard locations in a terminal alongside the berth where the ship is serviced. For an MUT, ship-to-berth assignments should be, in general, determined in advance of ship arrivals. However, containers for the ships may arrive at the terminal for loading after the berthing decision. Therefore, the distance between a ship and its container location depends on the berth assignment. Although examination of the

terminal handling systems is beyond the scope of this paper, it is obvious that the handling time may also be dependent on the geographical relationship between the ship and the container location in the yard.

In formulating the BAP, we define binary variables x_{ijk} to specify if ship j is serviced as the k th ship at berth i . Other related studies (i.e., berth assignment in naval ports as discussed in the previous section) employ actual unit time for decision variables to index assignment sequences. In the berth allocation planning, it is expected that a ship spends up to 24 h for cargo loading/unloading. Considering the prospective planning horizon of our model (say, at least a few days), we will have an explosively large number of decision variables by describing them with actual time. In addition, our model guarantees consecutive service for all ships without disruptions such as berth shifting.

As the variable x_{ijk} is restricted to 0-1 values, the BAP may be formulated as a mixed integer three-dimensional problem as follows:

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

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

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

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

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

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

Where $i (=1, \dots, I) \in B$ is the set of berths, $j (=1, \dots, T) \in V$ the set of ships, $k (=1, \dots, T) \in U$ the set of service orders, A_j the arrival time of ship j , P_k the subset of U such that $P_k = \{p | p < k \in U\}$, S_i the time when berth location i becomes idle for the planning horizon, W_i the subset of ships with $A_j \geq S_i$, C_{ij} the handling time spent by ship j at berth i , $x_{ijk} = 1$ if ship j is serviced as the k th ship at berth i and $= 0$ otherwise, y_{ijk} the idle time of berth i between the departure of the $(k-1)$ th ship and the arrival of the k th ship when ship j is serviced as the k th ship.

Notice that both sets of ships and service orders have the same number of elements T because a feasible solution may assign all the ships to a particular berth even though a number of berths are provided in the system.

Objective (1) minimizes the sum of waiting and handling times for every ship. Constraint set (2) ensures that every ship must be serviced at some berth in any order of service. Constraints (3) enforce that every berth services up to one ship at any time. Constraints (4) assure that ships are serviced after their arrival.

3.2 Derivation of objective function (1)

The first term of the objective function gives the total service time (including handling and waiting times) when ships are all serviced without any idle condition of berths as shown in Fig. 1. A service time C_{ij} of a specific ship serviced at berth i contributes to the waiting time of the subsequent ships to be serviced at that berth. In other words, the waiting time of any of those ships is represented by a cumulative service time of its predecessors.

The second term of the objective function represents the waiting time incurred with the berth idle time that results from late ship arrival. Fig. 2 shows a berth allocation pattern at berth i with an idle berth. Thin lines represent a ships' wait, while thick lines imply that they are being serviced. Dotted lines portray the berth in idle status. Ship 1 arriving at the port before S_i is the first ship to be serviced while ships 2 and 3 arriving after S_i are the fourth and fifth ships to be serviced, respectively. Berth i is already idle for ships 4 and 5; therefore they get serviced as the second and third ships as soon as they arrive. In general, the handling time C_{ij} of each ship, as mentioned above, contributes to the waiting times for all of its successors. Similarly, the idle time of a berth prior to service for a ship, y_{ijk} , must be summed up for the time its successors spend waiting for service. If a ship arrives before its immediate predecessor's departure, then by definition $y_{ijk}=0$ for that solution. For those ships that arrive before S_i , the time they spend waiting before S_i , i.e., $S_i - A_j$, is added. Conversely, if ship j arrives after S_i (i.e., $S_i - A_j$ is negative), regardless of if $y_{ijk}>0$ or $=0$, then waiting times summed by C_{ij} s and y_{ijk} s of their predecessors are subtracted by the time duration of $A_j - S_i$.

Figs. 1 & 2

Notice that the objective function contains $(T - k + 1)y_{ijk}$, i.e., y_{ijk} is computed for ship j as the k th ship as well as its successors, although it should not be for itself. As stated above, the waiting time of a certain ship given as the sum of C_{ij} and y_{ijk} for its predecessors is subtracted by $A_j - S_i$ if $y_{ijk} > 0$ for it; therefore its y_{ijk} must be added for its own waiting time.

3.3 Derivation of constraint set (4)

Constraint set (4) for ship j as the k th ship at berth i yields the following inequality by moving the third term of the left-hand side to the right:

$$\sum_{l \in V'} \sum_{m \in P_k} (C_{il}x_{ilm} + y_{ilm}) + y_{ijk} \geq (A_j - S_i)x_{ijk}$$

The first term in the left-hand side is the time duration between S_i and the time when the last of its predecessors leaves the port. Consequently, the left-hand side, i.e., the time duration between S_i and the start of the service for ship j must be no less than $A_j - S_i$, if $x_{ijk} = 1$.

4. FORMULATION WITH PRIORITY CONSIDERATION

4.1 Formulation of the PBAP

The BAP discussed in the previous section treats ships without distinctive priority. That is, no service priority in terms of ship size, handling volume, etc. is taken into account when determining the berth allocation. However, as previously mentioned there are arguments for differentiating the treatment of vessels. For example, Tanjong Pagar container terminal in Singapore is operated as an MUT, but with a higher priority assigned to ships with large handling volume. On the other hand, Dalian container terminal of China, which is also an MUT, is more likely to serve small feeder vessels when the terminal is congested, resulting in less waiting time to the succeeding vessels.

As mentioned before, the waiting time of each calling ship, which is a part of the service time is the cumulative handling time of all preceding ships at the same berth. That is,

if the service of ship 1 is followed by ships 2 and 3, the handling time of ship 1 is a part of the service time for ships 2 and 3. As will be described later, the priority consideration is taken into account by the service time of a ship weighted by an incident parameter to that ship. Thus, ship 1's handling time contributes to its successor's service times by multiplying it by each successor's parameter. The objective function of the dynamic BAP formulation can therefore be written as:

$$\begin{aligned}
 \text{[PBAP] Minimize } & \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \left\{ C_{ij} + S_i - A_j + \sum_{l \in V} \sum_{m \in P_k} C_{il} x_{ilm} \right\} \alpha_j x_{ijk} + \\
 & \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \left\{ y_{ijk} + \sum_{l \in V} \sum_{m \in P_k} y_{ilm} \right\} \alpha_j
 \end{aligned} \tag{7}$$

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

$$\sum_{j \in V} x_{ijk} \leq 1 \quad \forall i \in B, k \in U, \tag{3}$$

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

$$\begin{aligned}
 x_{ijk} &\in \{0, 1\} & \forall i \in B, j \in V, k \in U, \\
 y_{ijk} &\geq 0 & \forall i \in B, j \in V, k \in U,
 \end{aligned} \tag{5}$$

where α_j is a weight for ship j .

If, for example, we consider priority as a function of the ship's cargo handling volume at a calling port, then a solution may be obtained by using the cargo volume of ship j as a weight α_j . On the other hand, when small feeder ships (with small handling volume) need to be serviced at a higher priority, we can define α_j as the reciprocal of handling volume of ship j , to enhance feeder's evaluation value in the formulation. After all, this formulation has the advantage in that any kind of weight can be attached to individual ships. For instance, when a ship must be handled quickly for a certain reason such as an emergency, high priority may be realized in the resulting solution by adding a high value to it in the formulation.

4.2 Derivation of objective function (7)

As mentioned in Section 4.1, in the BAP formulation the waiting time of a ship for

berth availability is an accumulation of handling time of preceding ships serviced at the berth. Consequently, focusing on a ship in service, the total service time of a berth in objective function (1) is composed by the sum of the ship handling time and the product of the handling time and the potential service queue length after the ship as a part of the successors' waiting time.

In the PBAP, every ship is allowed to have a specific service priority; therefore a weight that represents a priority to a specific ship has to be associated with $\sum_{i \in B} \sum_{k \in U} x_{ijk}$ and its cumulative service time gap of two adjacent ships in service up to that ship, $\sum_{i \in B} \sum_{k \in U} y_{ijk}$.

Keeping this in mind, one focuses on a particular ship's service time which is calculated by the sum of the ship's handling time, $(C_{ij} + S_i - A_j)x_{ijk}$, and the waiting time as the cumulative handling time of preceding ships, $\left(\sum_{l \in V} \sum_{m \in P_k} C_{il} x_{ilm} \right) x_{ijk}$. The first term of function (7) is the total of ship's service time weighted by priority α_j without consideration of the time gap between two adjacent ships in service at a berth. The time gap is taken into account by the second term.

5. SOLUTION PROCEDURE

This section describes a solution procedure of the PBAP. As the BAP that is modified to model the PBAP was solved by the subgradient procedure with a Lagrangian relaxation to the BAP formulation in Imai et al. (2001), we first attempt to develop a Lagrangian relaxation formulation to the PBAP, in order to look into the availability of the subgradient procedure.

5.1 Lagrangian relaxation of the PBAP

Like the BAP, we obtain a Lagrangian relaxation formulation by putting constraints (4) into objective function (7) with corresponding Lagrangian multipliers. The relaxed problem is as follows:

$$[\text{RBAP1}] \text{ Minimize } \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \left\{ C_{ij} + S_i - A_j + \sum_{l \in V} \sum_{m \in P_k} C_{il} x_{ilm} \right\} \alpha_j x_{ijk} +$$

$$\begin{aligned}
& \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \left\{ y_{ijk} + \sum_{l \in V} \sum_{m \in P_k} y_{ilm} \right\} \alpha_j - \\
& \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \lambda_{ijk} \left\{ \sum_{l \in V} \sum_{m \in P_k} (C_{il} x_{ilm} + y_{ilm}) + y_{ijk} - (A_j - S_i) x_{ijk} \right\} \quad (8)
\end{aligned}$$

subject to (2), (3), and (5).

The formulation can be rewritten as follows, because y_{ijk} s are not included in any constraints of [RBAP1] and are redundant.

$$\begin{aligned}
\text{[RBAP2] Minimize } & \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \left\{ C_{ij} + S_i - A_j + \sum_{l \in V} \sum_{m \in P_k} C_{il} x_{ilm} \right\} \alpha_j x_{ijk} + \\
& \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \lambda_{ijk} (A_j - S_i) x_{ijk} - \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \sum_{l \in V} \sum_{m \in P_k} \lambda_{ijk} C_{il} x_{ilm} \quad (9)
\end{aligned}$$

subject to (2), (3), and (5).

Defining $D_{ijk}^* = (C_{ij} + S_i - A_j) \alpha_j + \lambda_{ijk} (A_j - S_i)$, $D_{ijkl}^\# = \lambda_{ijk} C_{il}$ and $E_{ijl} = \alpha_j C_{il}$,

objective function (9) is reformulated as (10).

$$\begin{aligned}
\text{Minimize } & \sum_i \sum_j \sum_k \left\{ D_{ijk}^* + \sum_l \sum_m E_{ijl} x_{ilm} \right\} x_{ijk} - \sum_i \sum_j \sum_k \sum_l \sum_m D_{ijkl}^\# x_{ilm} \\
& = \sum_i \sum_j \sum_k D_{ijk}^* x_{ijk} - \sum_i \sum_j \sum_k \sum_l \sum_m D_{ijkl}^\# x_{ilm} + \sum_i \sum_j \sum_k \sum_l \sum_m E_{ijl} x_{ilm} x_{ijk} \quad (10)
\end{aligned}$$

As shown in Imai et al. (2001), this three dimensional mathematical programming formulation (10), (2), (3), and (5) becomes a two dimensional problem as follows:

$$\text{[RBAP3] Minimize } \sum_{j,n} D'_{jn} x_{jn} + \sum_{j,n} \sum_{p,q} E'_{pqjn} x_{pq} x_{jn} \quad (11)$$

$$\text{subject to } \sum_n x_{jn} = 1 \quad \forall j, \quad (12)$$

$$\sum_j x_{jn} = 1 \quad \forall n, \quad (13)$$

$$x_{jn} \in \{0,1\} \quad \forall j, n, \quad (14)$$

where j and p are ship indices while n and q are resource (i.e., the mixture of berth and service order like Imai et al., (2001)) indices.

[RBAP3] is a quadratic assignment problem (QAP) for which it is difficult to find an optimal solution because it is an NP-hard problem. Due to the NP-hardness, the QAP has been solved not only exactly but also approximately. As there is a huge literature on the QAP, we focus on relatively recent works as follows. Some studies such as Resende et al. (1996), Pardalos et al. (1997a), Angel and Zissimopoulos (1998), Ishii and Sato(1998), Lim et al.(2000), Ahuja et al.(2000), and Arkin et al. (2001), propose heuristic algorithms for the QAP, while Clausen and Perregaard (1997), Pardalos et al. (1997b), Hahn et al. (1998), Sergeev (1999a,b), Karisch et al. (1999), Anstreicher (2000), Anstreicher and Brixius (2001), develop exact algorithms or new and strong lower bounds for existing optimal solution methods. Interestingly Ball et al. (1998) attempt to formulate the QAP as an integer linear programming in two different forms; however it remains difficult to solve in a sense of polynomially- bounded time because of the integer programming.

When the subgradient method incorporating the Lagrangian relaxation is utilized, the optimal solution to the relaxed problem (i.e., the QAP) must be identified in the first half stage at each iteration of the method. Consequently, this forces us to use an exact solution method to the QAP. This approach, however, is not attractive because of the heavy computational burden resulting from the NP-hardness. As a result, we decided to employ a heuristic to entirely solve the PBAP. As seen in the QAP literature, the so-called modern heuristic based on artificial intelligence techniques such as neural network and GA are widely adapted for a number of NP-hard combinatorial problems. An existing study regarding the BAP (Nishimura et al., 2001) attempted the GA and found that for the BAP, the GA worked as well as the subgradient method using the Lagrangian relaxation. Therefore, we employ the GA to approximately solve the PBAP.

5.2 Solution procedure using the genetic algorithm

5.2.1. Outline of solution procedure

As shown in Section 4.1, the problem [PBAP] is a non-linear formulation that is difficult to solve. To facilitate the solution procedure we employ a GA-based heuristic which is widely used in solving difficult problems and has a practical, short computational time. We aim to carry out scheduling for a planning horizon that is as long as accurate ship arrival information is available. To solve [PBAP], we partition the problem into N

sub-problems (SUBs) of the berth allocation in terms of a temporal factor as presented in Fig. 3. Given the resulting time for [PBAP] during the previous planning horizon when every berth becomes idle with the departure of the last ship assigned, the first SUB is solved by a GA. Given the solution of the first SUB that gives the times when the berths become idle for the next SUB, the next SUB is solved. This process is iterated until the final SUB (i.e., the N th SUB) is solved while a solution to a particular SUB (*intermediate solution*) is used as the start time of an idle berth for the next SUB. As the GA produces the best solution (it is not necessarily the best in a strong sense because the GA identifies approximate solutions), the second best, ..., K th best solutions each of which can be handed over to the next SUB, the final solution may be affected by the choice of the intermediate solutions; in other words, the best intermediate solutions being handed over through a series of SUBs does not necessarily lead to the best solution to the entire problem and the second and third best intermediate solutions may result in a better overall solution. However, our preliminary experiments using worse intermediate solutions showed no significant improvement in solution quality of the overall problem. Thus, the best intermediate solutions are inherited over a series of SUBs.

5.2.2. Heuristic using genetic algorithm

GA is like a heuristic method in that the optimality of the answers cannot be determined. It works on the principle of evolving a population of trial solutions, over many iterations, to adapt them to the fitness landscape expressed in the objective function. The procedure of GA is outlined in Fig. 4. In this figure, the objective function value and solution alternatives of the BAP correspond to the fitness value and individuals, respectively. The number of individuals in a generation is set to 30 for our heuristic.

5.2.3. Representation

In the GA's application that was developed, we have chosen to work with scheduling order rather than directly with berth schedules (and berthing times). Furthermore, instead of using the classical binary bit string representation, the chromosomes are represented as character strings. Fig. 5 shows a typical chromosome representation for a two-berth scheduling problem. The length of the string of digits is the number of ships plus the number of berths minus one. It consists of two parts separated with zero, each of which represents a service queue for one of the two berths. The example shows a schedule with ships 2, 8, 5, and 9 serviced in that order at berth 1, and ships 4, 7, 3, 1, and 6 at berth 2.

Figs. 3, 4 & 5

5.2.4. Fitness

The problem [PBAP] is a minimization problem; thus, the smaller the objective function value is, the higher the fitness value must be. For this, the fitness function could be defined by the reciprocal of the objective function (Kim and Kim, 1996). However, by this definition the best solution likely has an extremely good fitness value among solutions obtained where there is no significant difference between them in the objective function value. As this chromosome is always selected as a parent, it is difficult to maintain the variety of chromosome by crossover. Other alternatives of the fitness function are the exponential and sigmoid functions. As a result of some tests we conducted with these functions, the sigmoid function as defined in (15) was found to be better where $y(x)$ denotes the objective function value:

$$f(x) = 1 / (1 + \exp(y(x) / 10000)), \quad (15)$$

Note that $f(x)$ has a value ranging from 0 to 0.5.

6. 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 four basic problems in which 25, 50, 75, and 150 calling ships are served with various handling volumes at an MUT with five berths. Two data sets of ship sizes (actually container handling volume at the terminal being considered) were prepared for each of the four arriving ship sets. In one data set (Ship data A), the ship sizes range from 50 to 1000 TEUs based on a uniform distribution pattern. The other set (Ship data B) contains two groups of ships: one contains small feeder ships with the volume around 100 TEUs and the other large mother vessels of around 800 TEUs. The total handling volumes of ships in both data sets are given in Table 1. Those ship handling times are given as follows. As mentioned in Section 3, the handling time is dependent on the berth; therefore each ship has its own minimum handling time at a particular berth and longer times at the others. The minimum handling time is calculated by the handling volume multiplied by the average handling time per one container observed in the Port of Kobe. The handling times at the

other berths are generated by random numbers. The association of the minimum time with the berth is also made randomly. Four different ship arrival patterns are generated by an exponential distribution with random numbers. After all, 32 cases are calculated for the experiments.

Table 1

The planning horizon that services 25 ships is six days and each of the above-mentioned problem samples is computed with various sizes of SUBs with time spans of one, two, three and six days. The weight α_j is set to the following five different but systematic values: 1 (U), the container volume (CV), squared value of CV (CVS), the reciprocal of the container volume (RCV), and the reciprocal of the container volume squared (RCVS). CVS and RCVS are expected to give high priorities to bigger and smaller ships in solutions, respectively.

First we roughly show how α_j differentiates the ship priority. The objective function of the problem is the total service time weighted by α_j . Thus, different service times for each ship are expected with different values of α_j . As mentioned before, the service time comprises the handling and waiting times and represents the priority that is implicitly given by the terminal operator. However, though the handling time may differ based on the berth in which the ship is serviced, the differential magnitude among berths should not be significant. Therefore, in addition to the service time, the waiting time a ship spends and how often a ship is overtaken or passed by others (representing a relatively low level of treatment in an allocation scheduling) could help evaluate the priority.

Figs. 6 and 7 illustrate the average value of the service time, waiting time, and the number of passings of a ship over problems with four ship arrival patterns. The computational result is so huge that Fig. 6 shows the results only for the 50 and 150 ship problems using ship data A, whilst Fig. 7 represents those for ship data B. A graph for a computation with a specific SUB time span shows average values per ship with various α_j for two different ship groups (i.e., one for ships of under 500 TEUs and the other for those of over 500 TEUs) and for all the ships involved.

Figs. 6 & 7

As expected, for both ship data sets A and B ships with bigger volume (hereafter referred to as big ships) have shorter service times or are treated with higher priority with CVS and CV parameters than with RCV and RCVS. In contrast, small ships have longer service times with CVS and CV. However these trends are not significant with short time spans. This result occurs because in these cases the berths are not so congested (in other words, the ships do not come very often) and every ship is serviced without long delay. This insight is confirmed by the considerably small waiting times in these cases.

As small ships have smaller volume than bigger ships, the average service time is expected to be longer for the big ships than for the small ones. However, except for cases of one through three SUB time spans in both ship data sets, the big ships have shorter service times than the small ships with the parameter CVS. This implies that CVS works well in providing very high priorities to the big ships. The fluctuation of the service time by varying α_j is typical in the cases of big SUB time span. There is no significant fluctuation of the average service time of all the ships over cases with different time spans.

The trends of the waiting time and the number of passings resemble that of the service time. Nevertheless, the fluctuations of the average values of the waiting time and the number of passings among all the ships over the computations with different time spans are more significant than the service time, and these values increase with larger time spans. It is notable that the average service time of all the ships is minimized with $\alpha_j=1$, that is the case without any restriction in service priority. This is normal since the results are evaluated based on the total service time (actually average service time) but are not weighted. It is also interesting that in most cases, the waiting time and the number of passings are also minimized with $\alpha_j=1$.

The computational results of the two ship data sets are not significantly different. The service time of the data A is, however, a bit higher than that of data B because the total volume of data A is bigger than B. This observation is significant for the cases of six day time span.

As far as the average service time over the four different ship arrival patterns is concerned, the priority is closely related to α_j . However, this is not always the case. For instance, Fig. 8 shows an insignificant relationship between the total service time and α_j for a computational result of one arrival pattern out of the four using ship data B. Bigger problems, i.e., the 75- and 150-ship problems reveal significance in differentiated service times, while smaller ones represent less diversity, especially for ships of less than 500 TEUs.

This trend is seen in the analysis with the average value in Fig. 7. However, as seen in Fig. 8 individual computational cases have some irregularity with respect to the weight such as the 25-ship problem with the span of four days and the 75-ship problem with the three day span.

Fig. 8

7. CONCLUSIONS

In this study, we examined how the service priority can be incorporated into the BAP. The existing dynamic berth allocation formulation was extended in order to take the service priority into account in the objective function. The resulting formulation became non-linear and was difficult to solve. First we discussed how the problem was reduced to a Lagrangian relaxation problem in order to look into the availability of the subgradient optimization. Although the subgradient method was adaptable to this problem, enormous computational effort was expected because the relaxed problem was a quadratic assignment problem which was NP-hard. Therefore, we eventually employed a GA based heuristic algorithm, which is widely utilized for complicated combinatorial problems. A number of numerical experiments were conducted, showing that the service priority varied in accordance with different values of weight α_j . Finally it became apparent that α_j performed as an index that can be used to differentiate ship priority of handling in a decision making process for the berth allocation. Berth allocation that takes into account priority consideration is of high importance to port operators who function in a setting of intense competition as it allows them be more flexible in their decision making and it provides them with a range of alternatives to consider for servicing their clients.

REFERENCES

- Ahuja, R.K., Orlin, J.B., Tiwri, A., 2000. A greedy genetic algorithm for the quadratic assignment problem. *Computers and Operations Research* 27, 917-934.
- Angel, E., Zissimopoulos, V. 1998. On the quality of local search for the quadratic assignment problem. *Discrete Applied Mathematics* 82, 15-25.
- Anstreicher, K.M., 2000. Eigenvalue bounds versus semidefinite relaxations for the

- quadratic assignment problem. *SIAM Journal on Optimization* 11, 254-265.
- Anstreicher, K.M., Brixius, N.W., 2001. A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming, Series A* 89, 341-357.
- Arkin, E.M., Hassin, R., Sviridenko, M., 2001. Approximating the maximum quadratic assignment problem. *Information Processing Letters* 77, 13-16.
- Ball, M.O., Kaku, B.K., Vakhutinsky, A., 1998. Network-based formulations of the quadratic assignment problem. *European Journal of Operational Research* 104, 241-249.
- 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., 1997. Optimizing submarine berthing with a persistence incentive. *Naval Research Logistics* 44, 301-318.
- Clausen, J., Perregaard, M., 1997. Solving large quadratic assignment problems in parallel. *Computational Optimization and Applications* 8, 111-127.
- Hahn, P., Grant, T., Hall, N. 1998. A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method. *European Journal of Operational Research* 108, 629-640.
- 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.
- Ishii, S., Sato, M., 1998. Constrained neural approaches to quadratic assignment problems. *Neural Networks* 11, 1073-1082.
- Karisch, S.E., Cela, E., Clausen, J., Espersen, T., 1999. A dual framework for lower bounds of the quadratic assignment problem based on linearization. *Computing* 63, 351-403.
- Kim, J.U., Kim, Y.D., 1996. Simulated annealing and genetic algorithms for scheduling products with multi-level product structure. *Computers and Operations Research* 23, 857-868.
- 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.
- Liu, C.L., Layland, J.W., 1973. Scheduling algorithms for multiprogramming in a hard–real-time environment. *Journal of ACM* 20, 46-61.

- Lim, A., 1998. The berth scheduling problem. *Operations Research Letters* 22, 105-110.
- Lim, M.H., Yuan, Y., Omatu, S., 2000. Efficient genetic algorithms using simple genes exchange local search policy for the quadratic assignment problem. *Computational Optimization and Applications* 15, 249-268.
- 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.
- Pardalos, P.M., Pitsoulis, L.S., Resende, M.G.C., 1997a. Algorithm 769: Fortran subroutines for approximate solution of sparse quadratic assignment problems using GRASP. *ACM Transactions on Mathematical Software* 23, 196-208.
- Pardalos, P.M., Ramakrishnan, K.G., Resende, M.G.C., Li, Y., 1997b. Implementation of a variance reduction-based lower bound in a branch-and-bound algorithm for the quadratic assignment problem. *SIAM Journal on Optimization* 7, 280-294.
- Resende, M.G.C., Pardalos, P.M., Li, Y., 1996. Algorithm 754: Fortran subroutines for approximate solution of dense quadratic assignment problems using GRASP. *ACM Transactions on Mathematical Software* 22, 104-118.
- Sergeev, S.I., 1999a. The quadratic assignment problem: I. New lower bounds in paired assignment. *Automation and Remote Control* 60, 1162-1178.
- Sergeev, S.I., 1999b. The quadratic assignment problem: II. Refined Gilmore-Lawler algorithm. *Automation and Remote Control* 60, 1326-1331.
- Sha, L., Rajkumar, R., Lehoczky, J.P., 1990. Priority inheritance protocol: An approach to real-time synchronization. *IEEE Transactions on Computers* 39, 1175-1185.
- Xu, J., Parnas, D.L., 2000. Priority scheduling versus pre-run-time scheduling. *global Journal of Time-Critical Computing Systems* 18, 7-23.

Table 1. Total handling volume of ships

# of ships	25	50	75	150
Ship data A	12593	27422	42509	83985
Ship data B	13101	25225	37588	73521

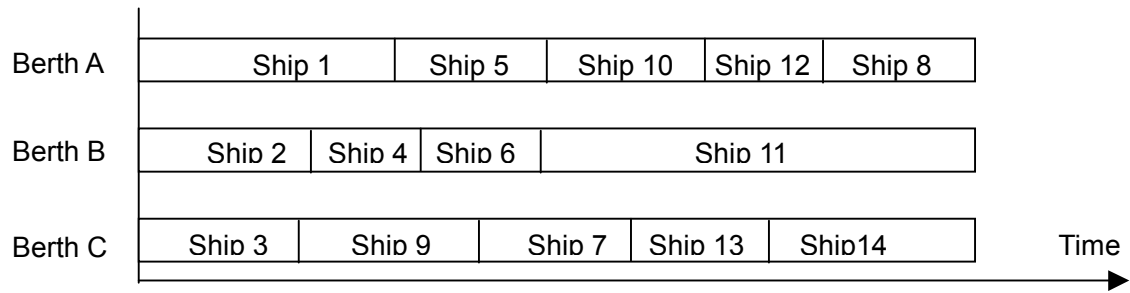


Fig. 1. Berth allocation

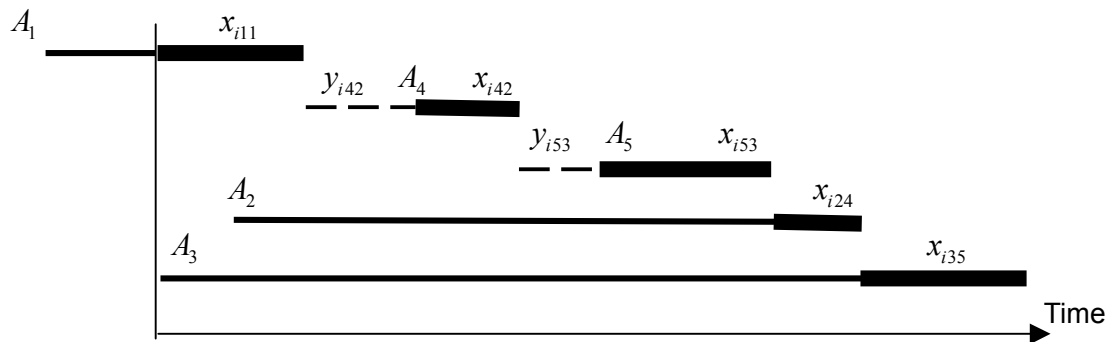


Fig. 2. Dynamic allocation to berth i

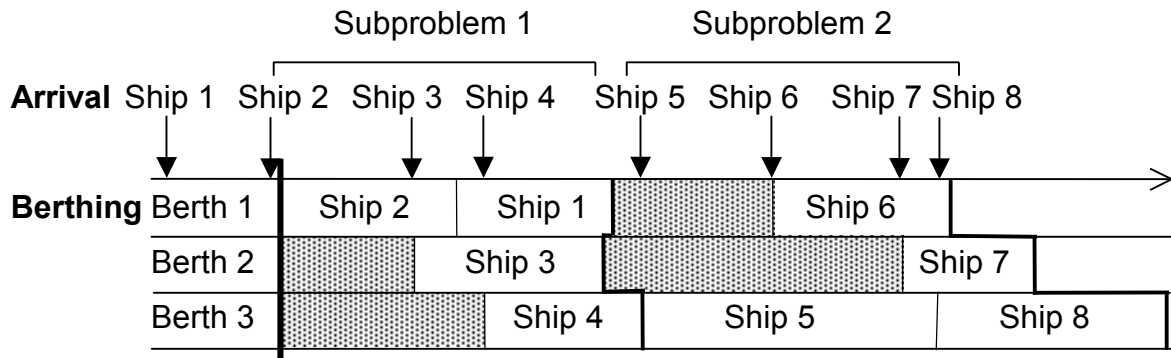


Fig. 3. Berth Schedule

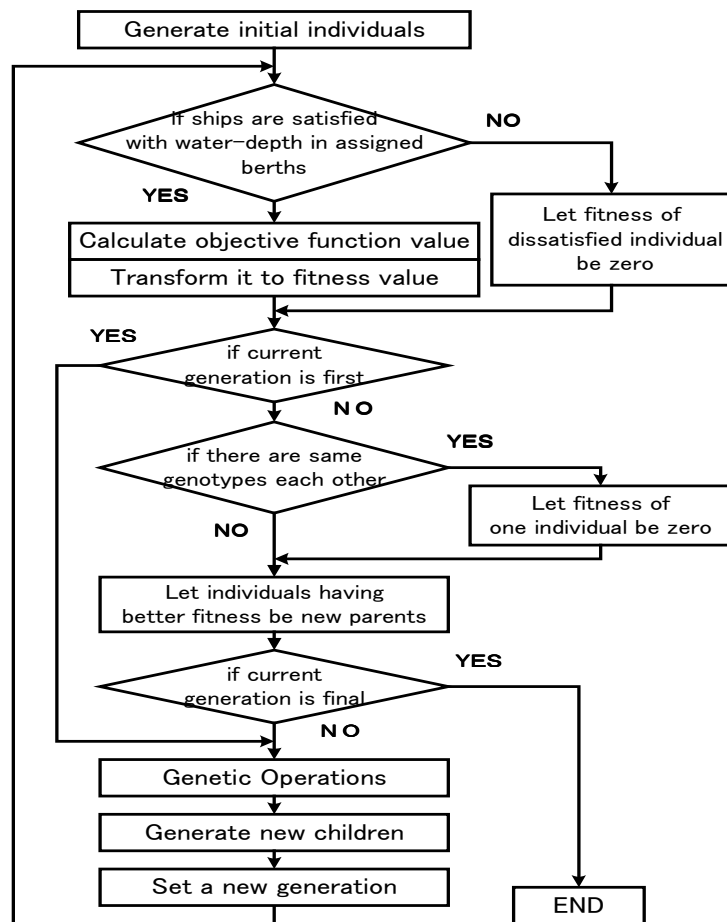


Fig. 4. GA procedure

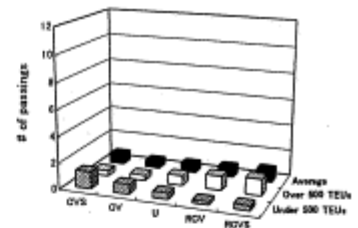
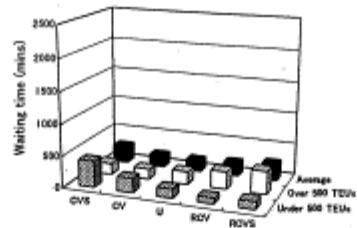
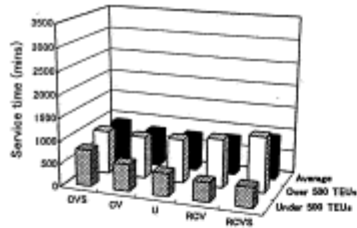
GENE: SHIP # 1-9, 0(ZERO)

(CELL #)	1	2	3	4	5	6	7	8	9	10
CHROMOSOME	2	8	5	9	0	4	7	3	1	6
BERTH #	1	1	1	1		2	2	2	2	2
ORDER OF SERVICE	1	2	3	4		1	2	3	4	5

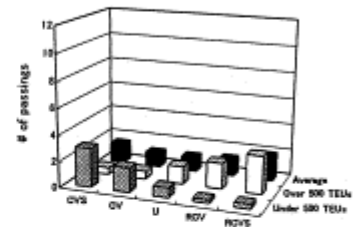
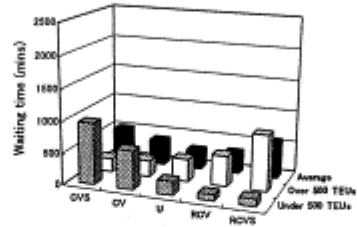
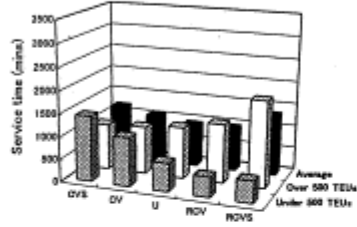
↑

BOUNDARY OF BERTHS

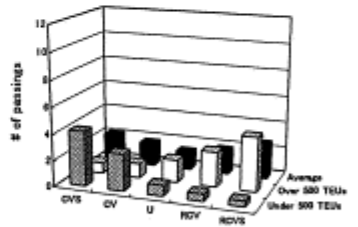
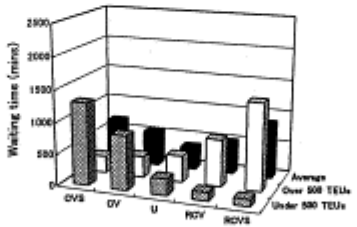
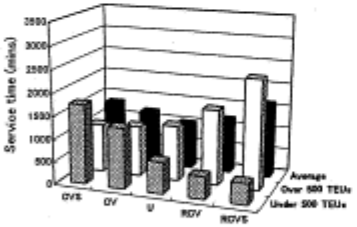
Fig. 5. Chromosome representation



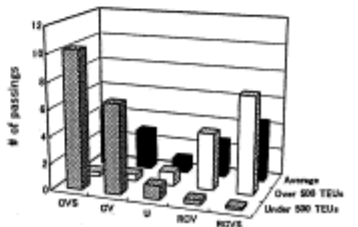
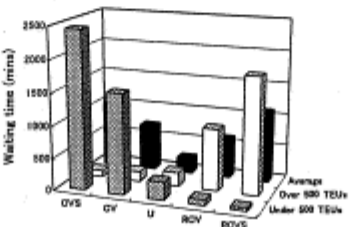
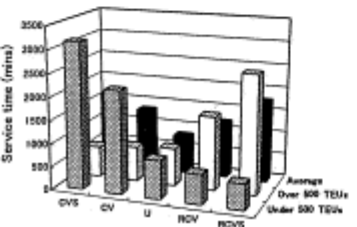
(a) time span for SUB =one day



(b) time span for SUB =two days

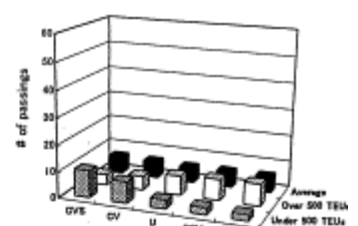
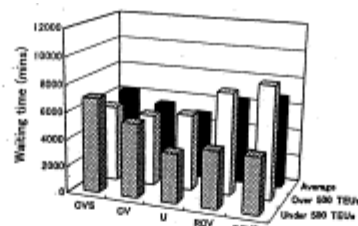
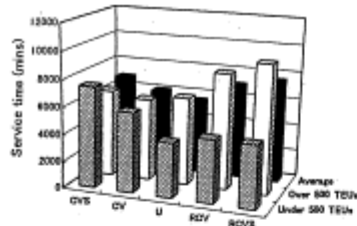


(c) time span for SUB =three days

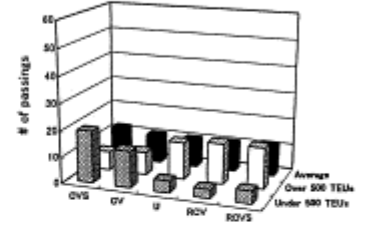
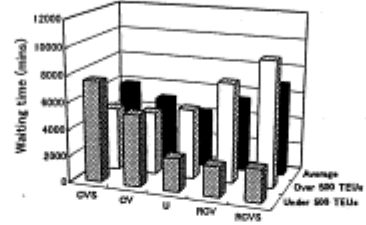
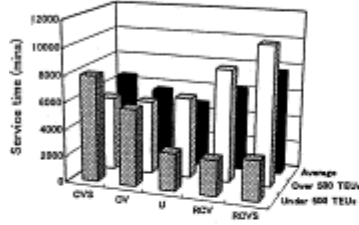


(d) time span for SUB =six days

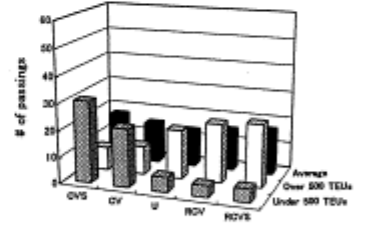
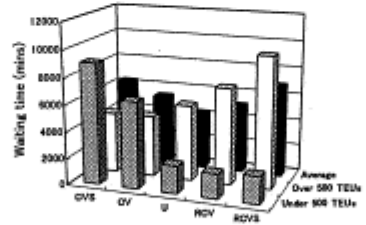
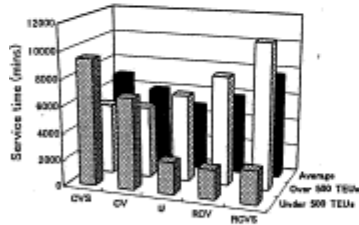
Fig. 6.(i) Results for 50-ship problem with ship data A



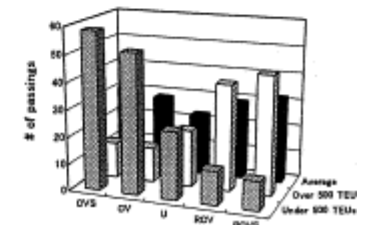
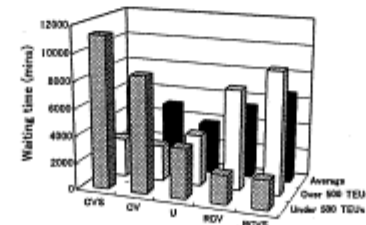
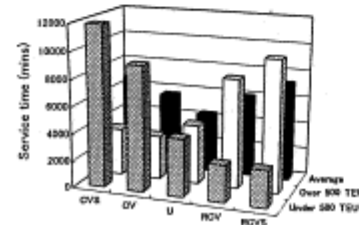
(a) time span for SUB =one day



(b) time span for SUB =two days

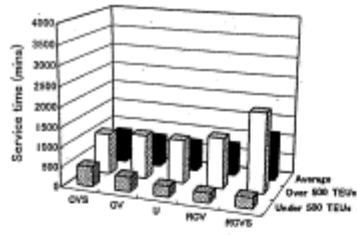


(c) time span for SUB =three days

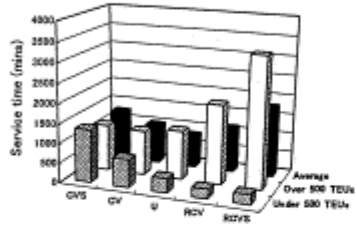
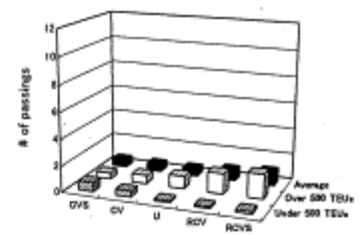
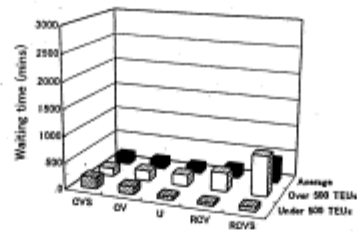


(d) time span for SUB =six days

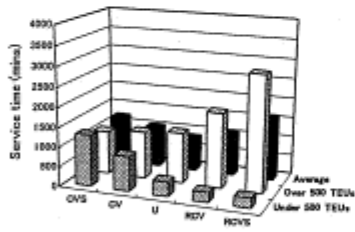
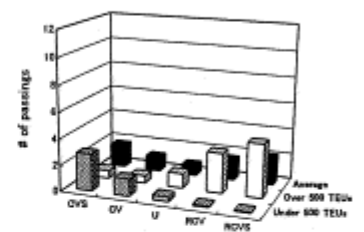
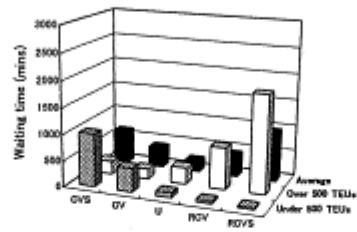
Fig. 6.(ii) Results for 150-ship problem with ship data A



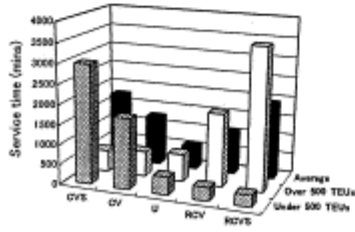
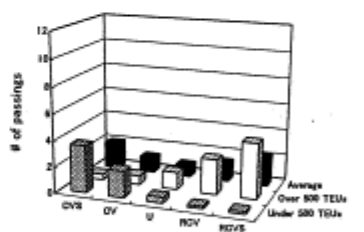
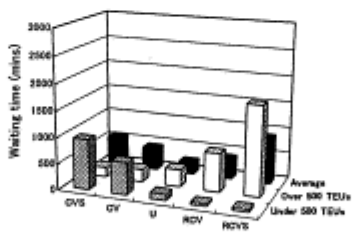
(a) time span for SUB =one day



(b) time span for SUB =two days



(c) time span for SUB =three days



(d) time span for SUB =six days

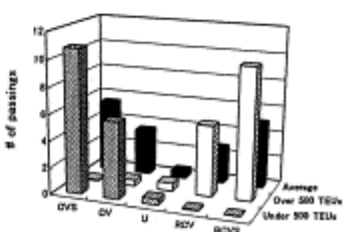
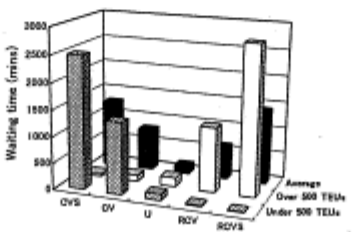


Fig. 7.(i) Results for 50-ship problem with ship data B

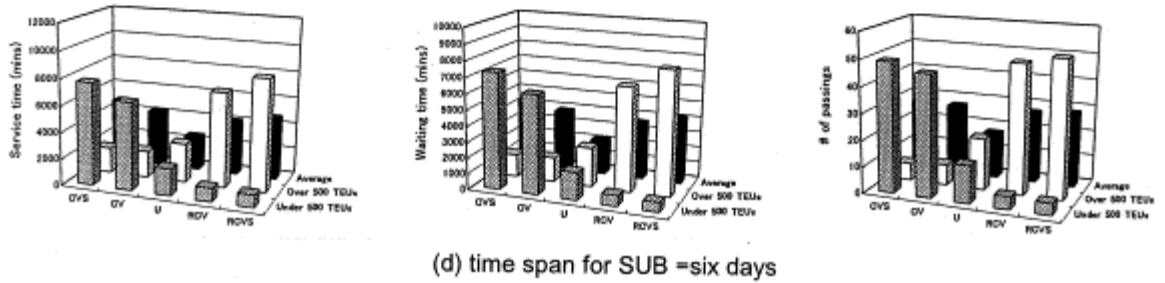
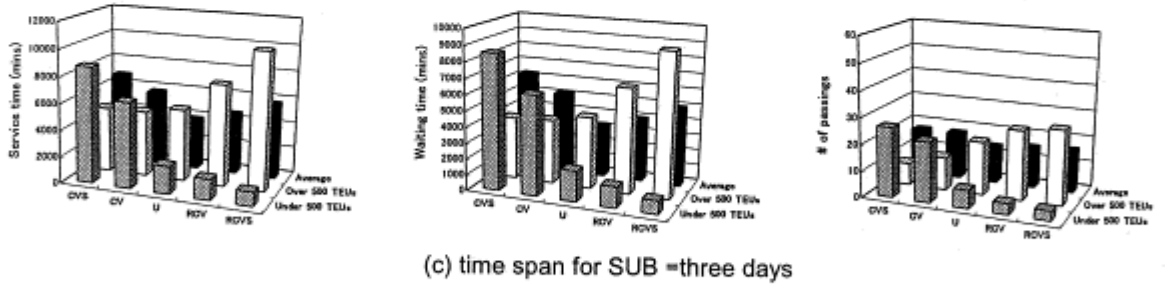
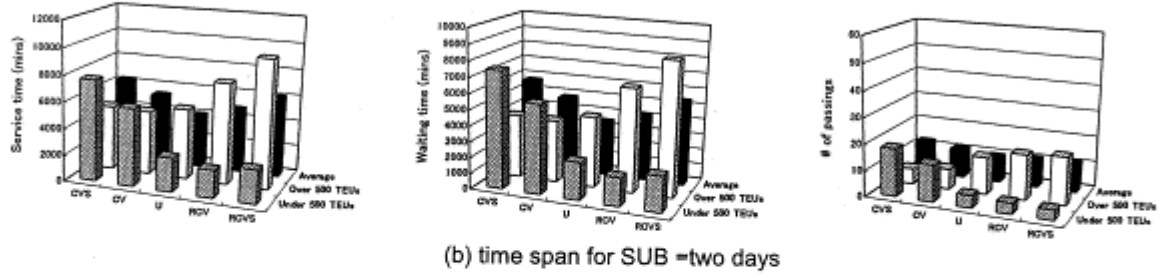
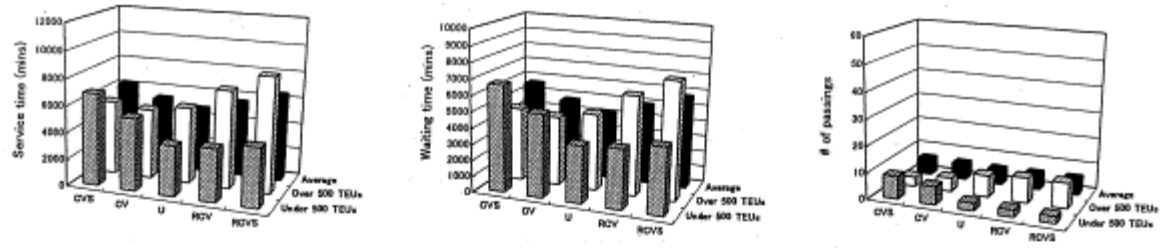


Figure 7-(ii). Results for 150-ship problem with ship data B

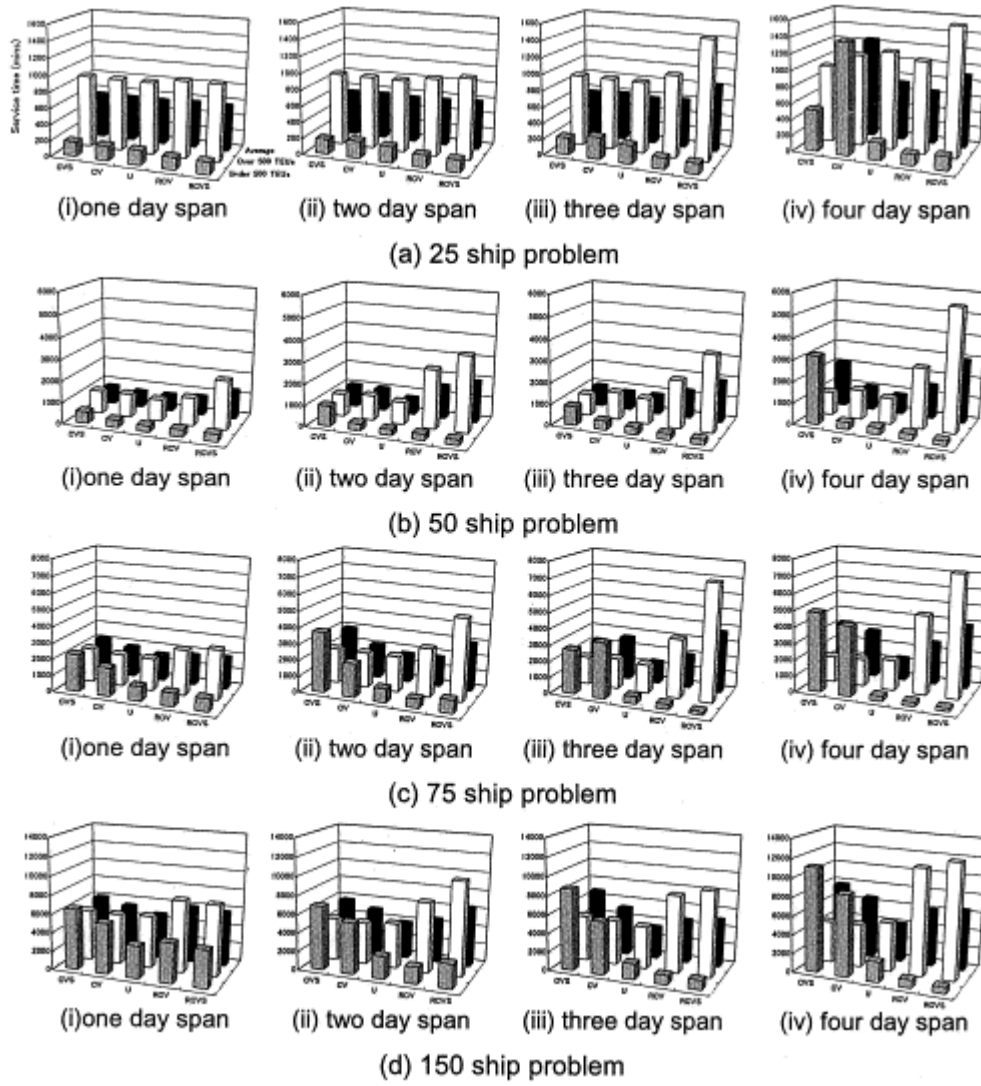


Fig. 8. Service time with ship data B